

THESIS TITLE

SEBASTIAN SALASSI

SUPERVISOR: DR. GIULIA ROSSI

ADVISOR: PROF. ANNALISA RELINI



## LIST OF ACRONYMS

<b>FF</b>	Force Field
<b>CG</b>	Coarse-Grained
<b>COM</b>	Center of Mass
<b>DOF</b>	Degrees of Freedom
<b>PW</b>	Polarizable Water
<b>ESM</b>	Ewald Summation Method
<b>PME</b>	Particle Mesh Ewald
<b>FFT</b>	Fast Fourier Transform
<b>PEF</b>	Potential Energy Function
<b>PBC</b>	Periodic Boundary Conditions
<b>MD</b>	Molecular Dynamics
<b>DNA</b>	Deoxyribonucleic Acid
<b>DMPC</b>	1,2-dimyristoyl- <i>sn</i> -glycero-3-phosphocholine
<b>RMS</b>	Root-Mean-Square



## INTRODUCTION



# CONTENTS

LIST OF ACRONYMS	iii
INTRODUCTION	v
1 INTRODUCTION TO MOLECULAR DYNAMICS	1
1.1 Review of classical mechanics . . . . .	1
1.2 Review of statistical mechanics . . . . .	4
1.2.1 Microcanonical ensemble . . . . .	7
1.2.2 Isothermal–isobaric ensemble . . . . .	7
1.3 Molecular Dynamics simulation . . . . .	8
1.3.1 Initial configuration . . . . .	9
1.3.2 Periodic boundary conditions . . . . .	10
1.3.3 Numerical integrators . . . . .	11
1.3.4 Neighbor list . . . . .	13
1.3.5 Thermostat algorithms . . . . .	15
1.3.6 Barostats algorithms . . . . .	17
1.4 Empirical Force–Field model . . . . .	18
1.4.1 Inter–particles interactions . . . . .	20
1.4.2 Non–bonded interactions . . . . .	21
1.4.3 Van der Waals interactions . . . . .	22
1.4.4 Electrostatic interactions . . . . .	24
1.4.5 Charge representation . . . . .	31
1.4.6 Polarization . . . . .	32
1.4.7 Coarse–Grained model . . . . .	33
1.5 MARTINI: a Coarse–Grained Force–Field . . . . .	34
1.5.1 Mapping . . . . .	35
1.5.2 Interactions potential . . . . .	36
1.5.3 Simulation parameters . . . . .	37
1.5.4 Parametrization . . . . .	38
1.5.5 Polarizable Water model . . . . .	39
1.5.6 Limitations of MARTINI FF . . . . .	41
1.6 Advanced sampling methods . . . . .	43
1.6.1 Metadynamics . . . . .	43
1.6.2 Umbrella sampling . . . . .	43
2 DUE	45

3 TRE

47

48



# 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 will introduce the computational Molecular Dynamics (MD) method and analyze the main aspects of this technique. Following some details about how the system under study is modeled are given: the *empirical Force-Field* (FF). This includes the way to consider and treat the inter-particle interactions at the MD level with some methods to increase the computational efficiency such as the *cut-off* methods for the so called *short range* interactions and the Particle Mesh Ewald (PME) technique for treatment of the *long range* interactions. Then MARTINI Force Field (FF), the main FF used in this thesis work, are summarized. Lastly some advanced sampling techniques are introduced: the *Umbrella Sampling* and *Metadynamics*. Some parts of this introductory Chapter are based on the books of Tuckerman [13], Leach [7], Frenkel and Smit [6] and Allen and Tildesley [1] to which the reader is addressed for a more complete discussion.

### 1.1 REVIEW OF CLASSICAL MECHANICS

The classical behavior of a system of  $N$  particles with mass  $m_i$  and coordinates  $\vec{r}_1, \dots, \vec{r}_N$  is described by the Newton's second law. Each particle in

the system will experience a total force  $\vec{F}_i$  and the so that second law for particle  $i$  reads

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

The total force is defined as

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

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

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

$$\frac{d\vec{p}_i}{dt} = \vec{F}_i(\vec{r}_1, \dots, \vec{r}_N) \quad (1.1.2)$$

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

$$\vec{x}_t = (\vec{r}_1(t), \dots, \vec{r}_N(t), \vec{p}_1(t), \dots, \vec{p}_N(t)) \quad (1.1.3)$$

called *phase space vector* or the *microstate* of the system at a time  $t$ . All the possible microstates of a system generate a  $6N$ -dimensional space called *phase space* of the system, indicated with  $\Omega$ . Thus  $\vec{x}_t$  describes a particular trajectory in the phase space, i.e. the system evolution is the motion of a phase space point.

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

$$\vec{F}_i(\vec{r}_1, \dots, \vec{r}_N) = -\partial_{r_{i\alpha}} U(\vec{r}_1, \dots, \vec{r}_N) \hat{e}_\alpha = -\vec{\nabla}_i U(\vec{r}_1, \dots, \vec{r}_N) \quad (1.1.4)$$

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

The kinetics energy of the system is defined as

$$K(\dot{\vec{r}}_1, \dots, \dot{\vec{r}}_N) = \sum_{i=0}^N \frac{1}{2} m_i \dot{\vec{r}}_i \cdot \dot{\vec{r}}_i \quad (1.1.5)$$

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

$$\mathcal{L}(\vec{r}_1, \dots, \vec{r}_N, \dot{\vec{r}}_1, \dots, \dot{\vec{r}}_N) = K(\dot{\vec{r}}_1, \dots, \dot{\vec{r}}_N) - U(\vec{r}_1, \dots, \vec{r}_N) \quad (1.1.6)$$

such that

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{r}_{i\alpha}} \right) - \frac{\partial \mathcal{L}}{\partial r_{i\alpha}} \quad (1.1.7)$$

This is a set of  $3N$  equations where for each  $i, \alpha = 1, 2, 3$ . These 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 are a sort of generator of the equations of motion.

Using the definition of  $\mathcal{L}$  [\(1.1.6\)](#) we have

$$p_{i\alpha} = \frac{\partial \mathcal{L}}{\partial \dot{r}_{i\alpha}} = m_i \dot{r}_{i\alpha} \quad (1.1.8)$$

thus we can express particles velocities as a function of particles momenta. Equations [\(1.1.5\)](#) and [\(1.1.8\)](#) let us to express the kinetics energy in the form

$$K = \sum_{i=0}^N \frac{\vec{p}_i \cdot \vec{p}_i}{2m_i} \quad (1.1.9)$$

To describe the system we can define another scalar function, called *Hamiltonian* of the system

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

Substituting [\(1.1.6\)](#) and using [\(1.1.9\)](#) the Hamiltonian of the system is nothing that

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

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

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

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

## 1.2 REVIEW OF STATISTICAL MECHANICS

Using the picture of the classical mechanics described above we have a good and sophisticated machinery that allows us, knowing some information about the system in exame, i.e. initial positions and velocities of all particles and their interactions, to completely solve the equations of motion in order to get the dynamics of the system at every time. So classical mechanics encodes all the information about the *microscopic* view of a system and, in principle, we can extract all the information we want about the *macroscopic* proprieties of such system. The main task of this process is to obtain the thermodynamics proprieties of a system (temperature, pressure and 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 ( $N_A = 6.022 \cdot 10^{23} \text{ mol}^{-1}$ ); we can not think of solving such a number of equations analytically even if we consider to solve it numerically: that is almost impossible. Thus the problem is to extract the macroscopic information from classical mechanics and to establish a well computable link between microscopic and macroscopic to obtain "easily" the thermodynamics information required.

The solution of that problem comes from the *statistical mechanics* developed, principally, by Boltzmann and Gibbs. Statistical mechanics involves all the rules and methods through which the microscopic and macroscopic worlds are related each other; this provides also a rigorous derivation of thermodynamics from the microscopic proprieties: without that thermodynamics would be only a phenomenological theory. The first step to the solution of the problem is to recognize that *a macroscopic observable of a system does not strongly depend on the complete dynamics of each particle in the system, but rather on an average that cancels 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 generates a specific macroscopic state; certainly we can do the contrary and for sure, if the system is real, we will 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 exists a number of microscopic state that yield to the same properties.

The most important idea, that makes this idea practicable, is the concept of *statistical ensemble*. Based on the previous considerations a general definition of an ensemble is *a collection of systems all subject to a set of common interactions and all sharing the same macroscopic proprieties*. This concept lays the foundations of thermodynamics and suggests a procedure for computing many macroscopic observable. In more details a N-particle system in a specific microscopic state is described by its microstate:  $\vec{x} = (\vec{r}_1, \dots, \vec{r}_N, \vec{p}_1, \dots, \vec{p}_N)$  and hence each systems is described as a point in the phase space, then *an ensemble is a set of points in the phase space that are subject to the constraint to be part of the ensemble itself*. Each system evolves in time with the equations of motion, so the time evolution of an ensemble is described by the flow of a set of points in the phase space according to the classical mechanics. Once an ensemble is defined we are able to compute, at every time, the macroscopic observables simply averaging over all the systems in the ensemble. To do this we have to know, at every time, which 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  $dx = dr_1 \dots dr_{3N}, dp_1, \dots, dp_{3N}$  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 microstates 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

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

Giving the ensemble distribution function, the ensemble average of an 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

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

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

$$\mathcal{Z} = \int_{\Omega} f(\mathcal{H}(\vec{x})) \, d\vec{x}$$

In order to compute the partition function we need to specified the thermodynamic observables, called *control variables*, that characterize the ensemble itself. By definition of an ensemble at thermodynamic equilibrium those 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– $\mu p T$ .

The averages computed in 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 ensemble averages and how to compute them but we need also a link between statistical averages and the experimental values. When we measure a macroscopic observable  $A$  we prepare an experiment with *only one* 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 fluctuates over time due to 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, in principle, the time for average is infinity then we have the “real” mean value of the observable

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

In order for a comparison to be made, an identity between ensemble and time averages must be established. This link is provided by the *ergodic theorem* and the *ergodic hypothesis*. A system is said to be ergodic if, over a long period of time, all the microstates in the phase space with the same

energy are accessible with the same probability. Then the ergodic theorem says that the time and ensemble averages are equal *almost everywhere* in the phase space. So we can write the follow identity

$$\bar{A} = \lim_{\tau \rightarrow +\infty} \frac{1}{\tau} \int_{t_0}^{\tau} A(\vec{x}_t) dt = \int_{\Omega} A(\vec{x}) \rho(\vec{x}, t) d\mathbf{x} = \langle A \rangle(t)$$

The most important ensembles are the microcanonical and the isothermal–isobaric. In the following we will describe them briefly with particular attention to the isothermal–isobaric ensemble, the most relevant for this thesis work.

### 1.2.1 Microcanonical ensemble

The microcanonical ensemble is composed of the systems whose number of particles (N), volume (V) and energy (E) are constant. Due to the constant energy it describes a Hamiltonian system for which

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

this let us define the partition function as follow

$$\mathcal{Z}_{NVE} = \frac{1}{N! h^{3N}} \int_{\Omega} \delta(\mathcal{H}(\vec{x}) - E) d\mathbf{x} \quad (1.2.1)$$

where the normalization factor  $N!$  takes into account the particles indistinguishability and  $h^{3N}$  is for compatibility of statistical mechanics with quantum mechanics: It comes from Heisenberg's uncertainty principle and it is the smallest phase space volume element.

### 1.2.2 Isothermal–isobaric ensemble

The isothermal–isobaric ensemble contains those systems with constant particles number (N), pressure (p) and temperature (T). This is useful in many chemical, biological and physical systems since their proprieties are reported in conditions of standard temperature and pressure. To 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 a high thermal capacity. The second can be idealized like a piston connected to the system changing the volume will adjust the pressure. The instantaneous work done by the system against the external piston is defined by  $pV$ , where  $V$  is

the instantaneous system volume. Then we have to correct the Hamiltonian of the system:  $\mathcal{H}(\vec{x}) \rightarrow \mathcal{H}(\vec{x}) + pV$ . The partition function is then defined considering the Boltzmann ensemble distribution function

$$\mathcal{Z}_{NPT} = \frac{1}{N!h^{3N}} \int_0^{+\infty} dV \int_{\Omega} e^{-\beta(\mathcal{H}(\vec{x}) + pV)} d\mathbf{x}$$

where  $\beta^{-1} \equiv k_B T$ ,  $k_B = 1.3806505(24) \cdot 10^{-23} \text{ JK}^{-1}$  is the Boltzmann constant and the normalization factor is the same as equation (1.2.1). The right thermodynamic potential to obtain all the macroscopic observables is the Gibbs free energy  $G$  defined by

$$G = -k_B T \ln \mathcal{Z}_{NPT}$$

it describes the maximum reversible work that may be performed by the system. The other thermodynamic quantities can be obtained deriving  $G$

$$\mu = \left( \frac{\partial G}{\partial N} \right)_{pT} \quad \langle V \rangle = \left( \frac{\partial G}{\partial p} \right)_{NT} \quad S = \left( \frac{\partial G}{\partial T} \right)_{Np}$$

where  $S$  is the system entropy.

### 1.3 MOLECULAR DYNAMICS SIMULATION

Molecular Dynamics ([MD](#)) is a set of techniques that allow us to prepare a “computer experiments” in which we have a virtual system set with some initial conditions, i.e. some microstates of all system particles. Then, solving numerically the classical equations of motion we will be 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 conditions and/or with different control parameters, such as temperature or pressure, in a straightforward and safer manner. The model system model contains all the information to obtain an approximation of the interactions among all system particles or, most commonly, to compute the [PEF](#) from which the forces are derived by equation (1.1.4). Solving the equations of motion With a numerical integrator, an [MD](#) simulation generates a set of phase space vectors, a *trajectory*, at discrete times that are multiples of the fundamental time discretization parameter, called *MD time step*:  $\delta t$ . Starting from an initial phase space vector  $\vec{x}_0$ , at each step, the forces are computed from the [PEF](#). Then the equations



of motion are integrated and a new phase space vector  $\vec{x}_{\delta t}$  is generated, thus a new set of forces is computed and so on. In order to compute time averages we need to discretize equation (1.2) so the time integration is substituted with a summation over the collected data at certain time step  $\Delta\tau = i\delta t$ ,  $i = 1, 2, 3, \dots$ . If  $i > 1$  only a subset of the collected data is used to compute time averages. The formula become

$$\langle A \rangle = \frac{1}{M} \sum_{n=1}^M A(\vec{x}_{n\Delta\tau}) \quad (1.3.1)$$

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

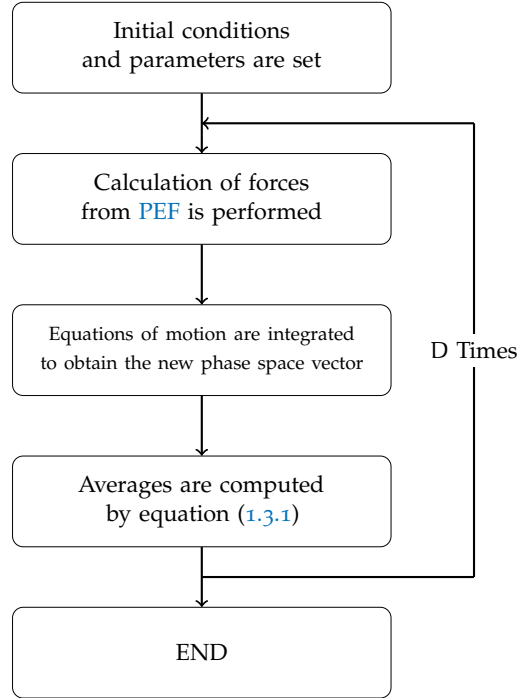


Figure 1: Schematic representation of an MD simulation.

### 1.3.1 Initial configuration

Before an MD simulation can be performed it is necessary to select an *initial configuration*. Its choice can be nontrivial and it depends on the complexity of the system. Moreover, all successive simulations will depend on the initial configuration, so if it is wrong, maybe even the simulations will produce

wrong results. Then, careful attention must be paid in setting up the initial configuration.

Setting up an initial configuration means to prepare an  $N$ -particle system and give all the particle 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 have to 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. This is done because, in general, the total force acting on the system  $\vec{F} = \sum_{k=1}^N \vec{F}_i$  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 taken into account because it reduces the system Degrees of Freedom (DOF) by 3.

### 1.3.2 Periodic boundary conditions

In all experiments our system is necessarily confined into a box with its initial size. Even in an 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 gives 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 surface effects and to consider only an infinite bulk system, Periodic Boundary Conditions (PBC) are imposed to the simulation box. This gives us also the possibility to simulate system’s bulk proprieties without considering a large number of particles. To give a better idea, in figure (2) an example of a two-dimensional box with PBC is shown. The central red contoured box is the simulation box. The idea is to replicate that box in space along each side 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 leaves 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 simulation box particles which have

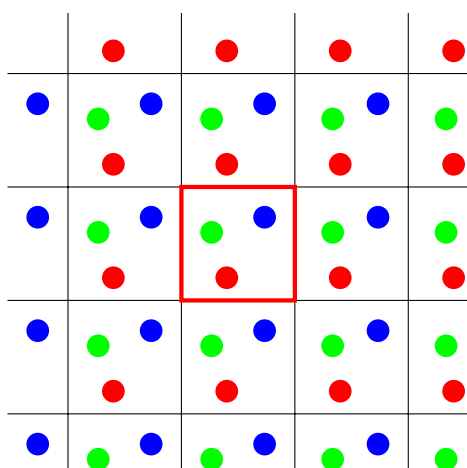


Figure 2: Schematic view of a two-dimensional box with [PBC](#) imposed. The central, red contoured, box is the simulation box and it is replicated along each side.

left the main box. For example, if 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 it leaves 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 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 cell length. This cause, obviously, the impossibility to sample those vibrating modes. Another 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 it is a difficult and time consuming task to do the same with the electrostatic interaction that it treated with a more sophisticated methods (this will be better explained in section [1.4.4](#)).

### 1.3.3 Numerical integrators

As we have seen above we need to solve numerically the equations of motion. Since the [PEF](#) is a continuous function of the phase space 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

$$\vec{r}_i(t + \delta t) = \vec{r}_i(t) + \vec{v}_i(t) \delta t + \frac{1}{2m_i} \vec{f}_i(t) (\delta t)^2 + o((\delta t)^3) \quad (1.3.2)$$

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 its implementation the *leap-frog algorithm*, which is the default used in our MD tools for this thesis work.

### Verlet algorithm

The Verlet algorithm requires the positions and the forces at a time  $t$  and the positions at a time  $t - \delta t$  to calculate the positions at a time  $t + \delta t$ . Starting from equation (1.3.2) we can write

$$\vec{r}_i(t + \delta t) \simeq \vec{r}_i(t) + \vec{v}_i(t) \delta t + \frac{1}{2m_i} \vec{f}_i(t) (\delta t)^2 \quad (1.3.3)$$

$$\vec{r}_i(t - \delta t) \simeq \vec{r}_i(t) - \vec{v}_i(t) \delta t + \frac{1}{2m_i} \vec{f}_i(t) (\delta t)^2 \quad (1.3.4)$$

from their sum we obtain the new positions at a time  $t + \delta t$

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

The velocities do not appear in the equation above and can be obtained taking the difference of equation (1.3.3) and (1.3.4)

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

Since positions are computed as differences this is a fourth order algorithm and the precision is up to  $o(\delta t)^4$ , but they also contain a small term ( $o(\delta t)^2$ ) which is summed to a difference of 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 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 (1.3.4) truncated at the first order:  $\vec{r}_i(-\delta t) \simeq \vec{r}_i(0) - \vec{v}_i(0) \delta t$ .

*Leap-Frog algorithm*

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

First it calculates 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

$$\vec{v}_i(t) \simeq \frac{\vec{v}_i(t + 1/2\delta t) + \vec{v}_i(t - 1/2\delta t)}{2} \quad (1.3.5)$$

Another advantage is that the velocities are part of the algorithm itself and moreover it does not require the calculation of the difference between two large numbers, with an increasing of precision. The obvious disadvantage is that the positions and velocities are not synchronized so the equation (1.3.5) is necessary to calculate the velocities at a time  $t$ . The need to have velocities at the same time of positions, as for the Verlet algorithm, derives from the calculation of the kinetics energy contribution to the total energy: It must be computed with positions and velocities at the same time.

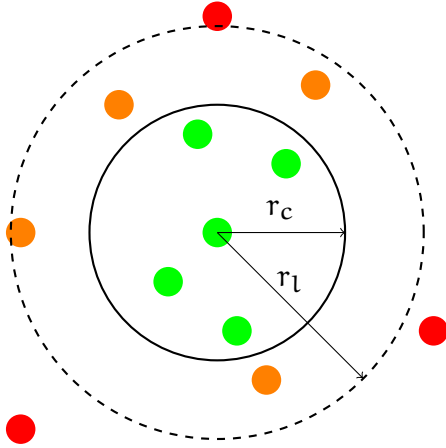
## 1.3.4 Neighbor list

In order to solve the classical equations of motion, it is necessary to know the forces, and so the PEF, acting on the system's particles. As we shall see in detail in the next section, this is one of the most time consuming part of an MD simulation. Even for a simple pairwise potential to know the forces acting on one particle, in principle, it is necessary to calculate the contribution from all particles in the simulation box and all other images. The most popular way to speed up the simulation is the use of a truncation of the interaction potentials within a cutoff. The general idea is to compute the interactions between those particles that are closer to the selected one by a certain cutoff distance  $r_c$ . By itself, the use of a truncation of the potential, may not dramatically reduce the time consuming in computing the

inter-particles interactions. This is because, in order to decide for what particles we have to compute the interactions, we have to compute the distances between every pair of particles in the system. A marked increase of the performance is achieved by the use of a *neighbor pair-list*. The simplest way is to consider, for each particle, a list of its neighbor particles that lie within a sphere of radius  $r_c$  surrounding the selected particle. Then, for each particle, the interactions are computed between the selected particle and those that are in its pair-list. There is a gain in performance if the pair-list is updated at least every  $M > 1$  MD steps. This is possible because of the assumption that a particle neighbors do not change significantly for  $M$  time steps, typically  $M = 10$  or  $20$ .

Anyway particles move during the non-updating time so that some of them may cross the pair-list causing an over- or under-estimation of the inter-particles contribution. To partially solve this problem, as suggested by Verlet [14], one can consider a *buffered pair-list* or a *Verlet cut-off scheme* in which the pair-list is constructed considering those particles that are close to the selected one by a distance of  $r_l > r_c$ , called *list radius*. That pair-list is updated every some time steps, but every time steps the pairwise contribution is computed only between those particles in the pair-list for which the distance is less than  $r_c$ . Thus, at the cost of slightly decrease the performance compared to the un-buffered pair-list, almost none of the interacting particles within the cutoff is neglected. Nevertheless, since a too big list radius and a too small list update frequency leads to a loss of performance, particle-pairs could still move enough during the non-update time and have a chance to cross the boundary of the buffered pair-list  $r_l$ . This chance leads to a small energy drift proportional to the system temperature. In simulations with a constant temperature coupling i.e. in a canonical or isothermal-isobaric ensemble, the extra radius of the buffered pair-list can be dynamically determined, during the simulation, by fixing an upper limit to the energy drift. That tolerance is often called *Verlet buffer tolerance* and commonly is set to  $0.005 \text{ kJ}/(\text{mol}\cdot\text{ps})$ . In figure (3) a schematic representation of the buffered pair-list construction is shown.

A better performance can be achieved adding a dynamic update algorithm of the pair-list refresh rate during the simulation. A nice way is considering the maximum distances traveled by a particles in the pair-list: if this distance is greater than  $r_b = r_l - r_c$  then the pair-list is certainly updated. From this one can fix the refresh rate to a higher value increasing the performance.



**Figure 3:** Schematic representation of the buffered pair-list construction respect to the central particle. Green particles are in the pair-list below the cutoff radius  $r_c$ , therefore included in the calculation of the interactions. Orange particles are in the pair-list for which every step is checked if their distances became lower than  $r_c$ . Red particles are not in the pair-list thus are no more considered in the distance check, until the next list update.

### 1.3.5 Thermostat algorithms

A thermostat is an external tool that allows us to maintain a system at a constant temperature. In MD simulations if one wants to use an isothermal ensemble, so with constant temperature, some algorithms have to be developed in order to mimic the thermostat effect on the system and which generates the correct ensemble. One way to impose a constant temperature is acting on the particles' velocities. A common practice, as suggested by Bussi *et al.* [2], is related to a *velocity rescale* algorithm in which the velocities of all particles are scaled by some factor. The simplest way is to consider the total kinetic energy  $K$  of the system as in equation (1.1.5) and the average kinetic energy  $\langle K \rangle$ . This is obtained from the *equipartition theorem* as follows

$$\langle K \rangle = \frac{1}{2} N_f k_B T$$

where  $N_f$  is the *total DOF* of the system and  $T$  is the target temperature. Thus the scaling factor is defined by

$$\alpha_T = \sqrt{\frac{\langle K \rangle}{K}}$$

The scaling operation is usually performed at a fixed rate during the simulation, or when the kinetic energy exceeds the limits of an interval centered around the target value. However, the sampled ensemble is not explicitly known but, since in the thermodynamic limit the average properties do not depend on the ensemble chosen, even this very simple algorithm can be used to produce useful results. Despite this, for small systems or when the observables of interest are dependent on the fluctuations rather than on the

averages or when other methods assume a canonical sampling, this method cannot be safely used.

In order to obtain the correct canonical sampling Bussi *et al.* modify the way to calculate  $\alpha_T$  so as to enforce a canonical distribution for the kinetic energy. The new scaling factor is obtained from

$$\alpha_T = \sqrt{\frac{K_T}{K}}$$

where  $K_T$  is extracted with a stochastic procedure from the equilibrium canonical distribution of the kinetic energy, given by

$$P(K_T)dK_T \propto K_T^{N_f/2-1} e^{-\beta K_T} dK_T$$

where  $P(K_T)dK_T$  is probability that the system have a kinetic energy between  $K_T$  and  $K_T + dK_T$ .

Since velocities are scaled only after some MD steps this can cause a discontinuity in the particles velocities just before and after the scaling step. To avoid this problem, the authors suggest that the choice of  $K_T$  can be based on the previous value of  $K$  so as to obtain a smoother evolution. The procedure proposed by Bussi *et al.* consist by the following steps

- Evolve the system for a single time step solving the equations of motion, so as to be in an NVE ensemble;
- Calculate the total kinetic energy and evolve it for a single time step using an auxiliary continuous stochastic dynamics;
- Rescale the velocities by  $\alpha_T$  so as to enforce this new value of the kinetic energy.

The authors have shown that an auxiliary dynamics for the kinetic energy of the form

$$dK = \frac{\langle K \rangle - K}{\tau} dt + 2\sqrt{\frac{K\langle K \rangle}{N_f\tau}} dW$$

can do the job. Where  $dW$  is a Wiener stochastic noise and  $\tau$  is an arbitrary parameter which is related to the response time of the thermostat. In fact it reduces to the simple velocity rescale if  $\tau \rightarrow 0$  and to an Hamiltonian dynamics (as an NVE ensemble) if  $\tau \rightarrow +\infty$ . Until the system is at non-equilibrium the first determinist part is dominant and drives the system to equilibrium with a characteristic time  $\tau$ . Then the stochastic contribution sample the canonical distribution.



#### 1.3.6 Barostats algorithms

*Berendsen algorithm*

*Parrinello–Rahman algorithm*

## 1.4 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  $N$ -particles 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 as set of atoms, molecules or more complex system such as protein and macromolecules systems. In each of these 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 is played by 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. Further 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  $N \sim 100$ ), too 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, are mediated by the electrons interactions. Thus, to 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 additive 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 into 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 expression of them and they are chosen as a compromise between accuracy and computational efficiency. Further 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 thus changing some parameters without retesting whole **FF** is not allowed because, maybe, one can destroy the whole **FF**.

For biomolecular applications two main classes of **FFs** exist: The *atomistic* **FFs** in which basic particles are atoms, and the *coarse-grained* **FFs** in which the basic particles represent atom groups or small chemical moieties. In this case, even the way to do the coarse-graining of the atoms in the molecules, called *mapping*, is part of the **FF** itself. Different Coarse-Grained (**CG**) **FFs** can use different mapping methods even with the same functional forms. In the following we will add some other information about **FFs** and describe the principal functional forms for modeling the inter-particles interactions and how to treat them in a **MD** simulation. While in the next section we will focus on the main **CG FF** used in this thesis work: The MARTINI **CG FF** developed by Marrink *et al.* [9].

**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 to reproduce a small set of target properties on 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.

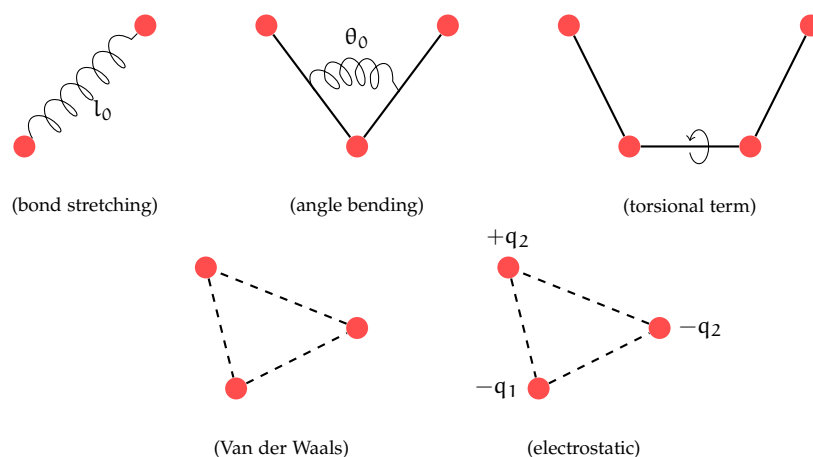
**TRANSFERABILITY** As described above the parameterization of a **FF** involves a small set of test systems for which some set of target proprieties are reproduced. The main characteristic of a **FF** is the *transferability* that means the ability of the model to describe different situations that 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.

## 1.4.1 Inter-particles interactions

For biomolecular applications the inter-particles interaction potentials are divided into two main classes: The *bonded interactions* involving particles within the same molecules and the *non-bonded interactions* engaging all particles in the system and which usually represent the Van der Waals and the electrostatic interactions. The most common and general functional form for the PEF is the following one

$$\begin{aligned}
 U(\vec{r}_1, \dots, \vec{r}_N) = & \frac{1}{2} \sum_{\text{bonds}} \frac{1}{2} k_i^b (l_i - l_{i0})^2 + \frac{1}{2} \sum_{\text{angles}} k_i^a (\theta_i - \theta_{i0})^2 + \\
 & + \frac{1}{2} \sum_{\text{torsions}} V_n (1 + \cos(n\omega - \gamma)) + \\
 & + \sum_{i=1}^N \sum_{j>i} \left( 4\epsilon_{ij} \left( \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right) + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right)
 \end{aligned} \tag{1.4.1}$$

The first two terms in equation (1.4.1) are harmonic potentials which model respectively the energy contribution due to deviation from reference bond length  $l_{i0}$  and bond angle  $\theta_{i0}$ . 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 involves a set of three particles in the same molecule. The middle line of equation (1.4.1) concerns the energy contribution due to the bond torsional change where  $\omega$  is the torsional angle. It involves four particles in the same molecule and mimic the energy barrier needed to rotate the bond angle along the bond axis.  $\gamma$  is a phase factor,  $V_n$  qualitatively describes 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 (1.4.1) 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  $\sigma_{ij}$  and  $\epsilon_{ij}$  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 figure (4).



**Figure 4:** Schematic representation of the common inter-atoms interactions for biomolecular applications: bond stretching, angle bending, torsional term, Van der Waals and electrostatic interactions.

#### 1.4.2 Non-bonded interactions

The bonded interactions, as we can see in equation (1.4.1), are at *fixed range*, meaning that they depend, for example, on the equilibrium bond length that is fixed. The same does not hold for the non-bonded interactions because they depend 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 dimensionality  $s$  of the system they are split into *short range* if  $d > s$  and *long range* interactions if  $1 \leq d < s$ . For example, as we shall see later, the Lennard-Jones 12-6 potential decays to zero as  $r^{-6}$  then it is a short range interaction, while the electrostatic is a long range interaction since it decays to zero as  $r$ .

##### *Cut-off, shift and switch methods*

As we have mentioned in 1.3.4 the calculations of the non-bonded interactions energy contributions is one of the most time consuming part of an MD simulations. Since they are pairwise interactions their calculation scale as  $\sim N^2$ . Especially for the short range interactions, various methods were developed in order to speed-up the simulations. The *cut-off* method is the most used to treat the short range interactions and, in some cases, even the long range one. Obviously this is strictly connected to the way neighbor list are computed as we have seen a cut-off method alone is not enough to speed up the simulation. Taking one particle into account, the general idea is to evaluate the non-bonded interactions with all other particles that are closer

to the first for a distance  $r_c$ , called *cut-off* radius, otherwise the interactions is set to 0. This means that the new potential is of the form

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

This generates a discontinuity in the potential and in its 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 also a *shift* of the potential value at  $r_c$ , since it is a constant it does not affect the forces. We have

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

Moreover to solve 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

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

Although, the shift methods make the potential quite different from the “true” one and this makes difficult to retrieve the correct thermodynamics proprieties. Thus, even if it can solve some instabilities, it must be carefully used.

Another powerful method is the *switch* method. The general idea is to consider two cut-off radii  $r_{c1}$  and  $r_{c2}$ . If  $r \leq r_{c1}$  the “true” forms are used; while for  $r > r_{c2}$  the potential is set to zero. For  $r_{c1} < r \leq r_{c2}$  a *switching function* is considered in order to *smoothly* switch the potential to zero.

It is important to stress out that even the method used to treat the interactions, 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 parameterizations, and should never be changed without retesting some target properties.

#### 1.4.3 Van der Waals interactions

Van der Waals forces are a set of interactions that are divided into two main contributions: an attracting interaction and a repulsive one. The main contribution to both is due to quantum dynamics effect of the electron cloud

interactions through the Pauli exclusion principle and to the instantaneous electrostatic interactions, even if both atoms are neutral, such as dipole-dipole, induced dipole-dipole and induced dipole-induced dipole interactions which, in a more rigorous description should be treated quantum mechanically. Both London dispersion forces, involving polar and non-polar atoms and related to instantaneous multipoles interactions, and hydrogen bonding, due to quantum effects, instantaneous electrostatics interactions and entropy effects, contribute to the attractive part of the potential.

The usual model to treat Van der Waals interactions is a Lennard-Jones potential. The most common exponents for the attractive and repulsive contributions to the potential are 6 and 12, respectively, although 6 and 9 can also be found depending on the system. The general form for a 12 – 6 Lennard-Jones potential is the following

$$v(r) = 4\epsilon \left( \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right) = \frac{C_{12}}{r^{12}} - \frac{C_6}{r^6} \quad (1.4.2)$$

where  $C_{12} = 4\epsilon\sigma^{12}$ ,  $C_6 = 4\epsilon\sigma^6$  and  $r$  is the pairwise particles distance.  $\epsilon$  is related to the absolute value of minimum while  $\sigma$  is related to the position of the minimum of the potential:  $r_{\min} = 2^{1/6}\sigma$ , often referred to by Van der Waals radius. These constants are proper for each particle pair. 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 (5) there is a example plot of the function (1.4.2) with  $\epsilon = \sigma = 1$ .

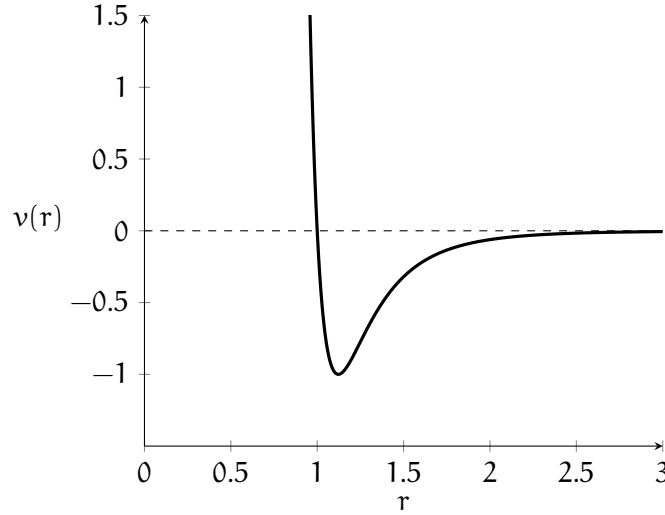


Figure 5: Example plot of a Lennard-Jones function with  $\epsilon = \sigma = 1$ .

The simplest and computationally most 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 potential and/or a continuous forces. As we can see from figure (5) 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 choice for the cut–off is then of the order of  $r_c \sim 2\sigma \div 3\sigma$ .

#### 1.4.4 Electrostatic interactions

One of the most important long range interaction is the electrostatic one. 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 interaction by a cut–off method<sup>1</sup>. Of course this is an approximation and can lead to a serious issues in those proprieties or systems that strongly depend on the electrostatic interaction. Most issues due to a bad treatment of the Coulomb interactions are related to the development of a good polar solvent model (a good treatment of the electrostatic proprieties of water is really important for biological applications), as well as to the study of the interactions of charged particles with polar solvents, transport processes of charged moieties, calculations of the electrostatic potential inside 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 PBC: even all the infinity images of all particles need to be taken into account.

If we consider only the simulation box the energy contribution is

$$U = \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i}^N \frac{1}{4\pi\epsilon_0} \frac{q_i q_j}{r_{ij}} \quad (1.4.3)$$

where  $q_i$  and  $q_j$  are the charge of particles  $i$  and  $j$ , respectively, and  $r_{ij}$  is the distance between  $i$  and  $j$ . But we need also all image boxes. Supposing, for simplicity, that the box is a cube of size  $L$ , then we can define a tern of integer numbers  $(n_x, n_y, n_z)$ ,  $n_i = 0, 1, 2, \dots$  so that the position of all other image

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



boxes, with respect to the central simulation box, is  $\vec{n} = L(n_x, n_y, n_z)$ . Then the energy contribution becomes

$$U = \frac{1}{2} \sum'_{n_x, n_y, n_z} \sum_{i=1}^N \sum_{j=1}^N \frac{1}{4\pi\epsilon_0} \frac{q_i q_j}{\|\vec{r}_i - \vec{r}_j + \vec{n}\|} \quad (1.4.4)$$

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

As described above, a cut-off method is a good easy way to solve equation (1.4.3) and sometimes it produces good results. However, the increasing of computer power can lead to develop more rigorous methods to solve equation (1.4.4), even for very large systems. The main problem is that the summation in equation (1.4.4) is *conditionally convergent*<sup>2</sup> and converges extremely slowly so that it would need so many terms to converge that its computational cost would be too high, especially for large systems (of the order of  $N \sim 3 \cdot 10^4$ ). The most important methods developed to solve this problem are based on the *Ewald Summation Method* (ESM). We shall describe those used in this thesis work: the Ewald Summation Method (ESM) 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.* [3].

#### *Ewald summation method*

The Ewald Summation Method (ESM) is the first method introduced by Ewald *et al.* or a correct treatment of the electrostatic energy contribution in an ionic crystal that can be modeled as the electrostatic interactions of a periodic charge density. The basic idea is to split the summation in equation (1.4.4) in two series both rapidly convergent. The method is based on the following identity

$$\frac{1}{r} = \frac{f(r)}{r} + \frac{1-f(r)}{r} \quad (1.4.5)$$

the trick is to choose a function  $f(r)$  that will deal with 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.

The ESM for electrostatic interactions works, as illustrated in figure (6), considering each point-like charge in the system surrounded by a neutral-

<sup>2</sup> 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.

izing charge distribution of equal magnitude but opposite sign that decays rapidly to zero. Simplifying 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

$$\rho_i(r) = \frac{q_i \alpha^3}{\pi^{3/2}} e^{-\alpha^2(r-r_i)^2} \quad (1.4.6)$$

that obeys the relation

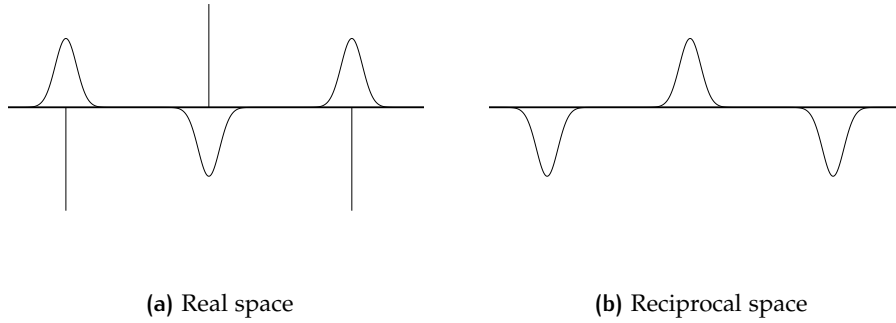
$$\frac{q_i \alpha^3}{\pi^{3/2}} \int_{r_i-\epsilon}^{r_i+\epsilon} e^{-\alpha^2(r-r_i)^2} dr \simeq q_i$$

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

$$U_r = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \sum'_{n_x, n_y, n_z} \frac{q_i q_j}{4\pi\epsilon_0} \frac{\text{erfc}(\alpha \|\vec{r}_i - \vec{r}_j + \vec{n}\|)}{\|\vec{r}_i - \vec{r}_j + \vec{n}\|} \quad (1.4.7)$$

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

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{+\infty} e^{-t^2} dt, \quad \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (1.4.8)$$



**Figure 6:** 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 point is that the summation involving the complementary error function in equation (1.4.7) is rapidly convergent and it needs very few terms so that a cut-off method can be safely used. The rate of convergence depends on the  $\alpha$  parameter, the bigger is  $\alpha$  the more rapidly converge and the shorter can be the cut-off radius. Thus the ESM use the  $\text{erfc}(r)$  as  $f(r)$  function in equation (1.4.5). Of course since we added a non physical neutralizing charge in the system, in order to restore the real charge distribution, we must consider another distribution, called counteracts charge distribution, of equal magnitude but opposite sign. Considering the identity in equation (1.4.5) this lead to an energy contribution of the form  $(1 - f(r))/r$ , so, using equation (1.4.8), it is of the form  $\text{erf}(r)/r$ . Another trick is to compute the former in the *real space*, the latter in the *reciprocal space*, thus considering its Fourier transform. This energy contribution is given by

$$U_f = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \sum_{k_x, k_y, k_z} \frac{1}{4\pi\epsilon_0} \frac{4\pi}{L^3 k^2} e^{-k^2/(4\alpha^2)} e^{i\vec{k} \cdot (\vec{r}_i - \vec{r}_j)} \quad (1.4.9)$$

where  $\vec{k} = 2\pi\vec{n}/L$  are the reciprocal lattice vectors. Even  $U_f$  converges rapidly as  $U_r$  in equation (1.4.7); then a cut-off method can be safely used. Nevertheless, as opposite to  $U_r$ , the smaller is  $\alpha$  the shorter can be the cut-off. Clearly a proper *balance* between the real and reciprocal space summation is needed.

Since in equation (1.4.7) even the self interaction with each Gaussian is included we need to add another item for cancel it out; this is done by the self-term

$$U_{\text{self}} = -\frac{\alpha}{\sqrt{\pi}} \sum_{i=1}^N \frac{q_i}{4\pi\epsilon_0} \quad (1.4.10)$$

Summarizing, the energy contribution of the electrostatic interactions by the ESM, is computed summing equations (1.4.7), (1.4.9) and (1.4.10) to obtain

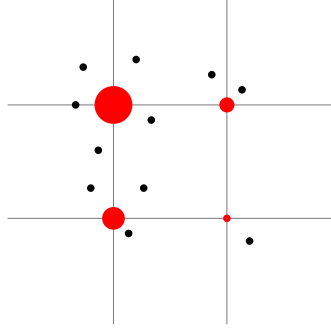
$$\begin{aligned} U = & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \sum'_{n_x, n_y, n_z} \frac{q_i q_j}{4\pi\epsilon_0} \frac{\text{erfc}(\alpha \|\vec{r}_i - \vec{r}_j + \vec{n}\|)}{\|\vec{r}_i - \vec{r}_j + \vec{n}\|} + \\ & + \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \sum_{k_x, k_y, k_z} \frac{1}{4\pi\epsilon_0} \frac{4\pi}{L^3 k^2} e^{-k^2/(4\alpha^2)} e^{i\vec{k} \cdot (\vec{r}_i - \vec{r}_j)} + \\ & - \frac{\alpha}{\sqrt{\pi}} \sum_{i=1}^N \frac{q_i}{4\pi\epsilon_0} \end{aligned} \quad (1.4.11)$$

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 computation. The [ESM](#) offers a well defined method to properly treat electrostatic interactions, nevertheless it is quite expensive in term of computational resources. If  $\alpha$  and the cut-off are constant, then the computation scales as  $\sim N^2$ ; while if  $\alpha$  and the cut-off are dynamically updated it scales as  $\sim N^{3/2}$ ; this can lead to an incompatibility between the Van der Waals interactions cut-off and the electrostatic one thus compromising the efficiency gain. Despite we have described [ESM](#) applying it to the electrostatic interactions, it can be used, with some changes, with 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 contribution.

For biomolecular applications most [MD](#) tools set an equal cut-off radius for both Van der Waals interaction and the real part of the Ewald summation ([1.4.11](#)) 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 ([1.4.11](#)) will be very inefficient as it scales as  $\sim N^2$ . In order to increase the efficiency of the calculation of the Fourier transform various of advanced methods can be used. They are all based on the use of the Fast Fourier Transform ([FFT](#)) method. In this way the reciprocal part can scale as  $\sim N \ln N$ . Since [FFT](#) requires discretized quantities, the idea of such methods, 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 the [FFT](#) method; this can be done, for example, exploiting the [PBC](#) in order to discretize and make periodic the Poisson's equation. Such algorithms 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 depends strongly on the way in which the charges are attributed to mesh points, this makes the methods different. In the following we will describe the one used in this thesis work, the Particle Mesh Ewald ([PME](#)) method.

#### *Particle mesh Ewald method*

Particle Mesh Ewald ([PME](#)) method developed by Darden *et al.* [[4](#)] is based on the [ESM](#) so that the starting point is equation ([1.4.11](#)) in which, as described above, the first part of the Ewald summation is computed in the real space together with the Van der Waals contribution using the same cut-off radius while the reciprocal part is computed using [FFT](#) method, in order to have a gain of performance. To do this, first, we need to consider a grid mesh onto which the Gaussian counteracts charge distribution is spread. The basic



**Figure 7:** A schematic representation of the charge assignment. The black filled circles are a unit particle charge, while the red ones, are the charges assigned to grid points. The bigger is the circle, the more is the charge.

idea, then, is to calculate the electrostatic energy solving Poisson's equation through FFT methods. The efficiency and accuracy depend on the way the charges are distributed onto the grid. To do 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

$$q_M(r_p) = \int_0^L W(r_p - r) \rho(r) dr \quad (1.4.12)$$

where  $L$  is the box length and, if  $h$  is the grid spacing,  $M = L/h$  is the number of mesh point. In figure (7) the charges assignment is schematically represented. The assignment function should have the following properties: 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 as possible the discretization of the charge density. However the computational cost increases as the number of grid points! So a balance between efficiency and accuracy is clearly needed.

A nice way to solve the problem of charge assignment is to shift the problem to the discretization of the Fourier transform. This can be viewed as an interpolation problem. Consider the  $e^{-i\vec{k} \cdot \vec{r}_j}$  term in the Fourier transform of equation (1.4.9). 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

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

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

where  $W_p$  denotes the interpolation coefficient. A  $p$ -order interpolation means that only the  $p$  mesh points nearest to  $r_j$  contribute to the sum. Assuming a point-like charge distribution the Fourier transform of the charge density is therefore

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

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 (1.4.12), 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 former is to use, as interpolating function, a polynomial function of degree  $\leq (n - 1)$  where  $n$  is the number of points to interpolate, that passes through all the  $n$  points, and which 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 sets of points to interpolate. The main disadvantage of this method is that, even if  $P(x)$  is continuous everywhere, its derivative is not, thus it can lead to some instability in MD simulations. The latter 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 through each point, the SPLINE method uses a *piecewise polynomial function*, called SPLINE, in which each piece is 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  $\lambda$  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 Euler SPLINE method is referred to as *smooth* Particle Mesh Ewald and the reader is addressed to the article by Essmann *et al.* [5] for more technical details about the interpolation procedure.

Summarizing, the [PME](#) method is implemented with the following scheme

- By the interpolation of the complex exponential in the Fourier transform of the Ewald summation, the Gaussian counteracts charge distribution are spread onto the mesh grid and a discretized Fourier transform is obtained;
- Poisson's equation for the discretized charges are solved through the [FFT](#) methods;
- The reciprocal energy contribution is obtained considering the inverse Fourier transform;
- Electrostatic forces are computed and assigned to the charged system particles.

The main advantages of the [PME](#) algorithm are that the potential energy and forces are smooth functions of the particles positions, offers a good energy 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 as the Lennard-Jones potential. Nevertheless it does not conserve very well the particles momentum due to Root-Mean-Square ([RMS](#)) errors in the forces calculations that have to be cancelled out by removing the forces averages.

#### 1.4.5 Charge representation

Even if some methods are developed to speed up the electrostatic energy contribution such as 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 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 as 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.

#### 1.4.6 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 another 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 as 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 dielectric constant  $\epsilon_r$ . In this way the polarization effect is taken into account considering a mean field theory and solving the Poisson's equation, for determine the electrostatic potential due to system charges, by the substitution  $\epsilon_0 \rightarrow \epsilon_0 \epsilon_r$ . If that method give an incomparable gain in performance, it must be carefully used. The main disadvantage is that the mean polarization effect is added to all system particles. This 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 dielectric constant can not be the same, this can lead to a serious incorrect results of the properties of the organic matters and maybe of the simulation.

The way to correct the above behavior is to use an explicit methods. As the name suggest, the polarization effect is taken 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.

#### 1.4.7 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 properties 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.

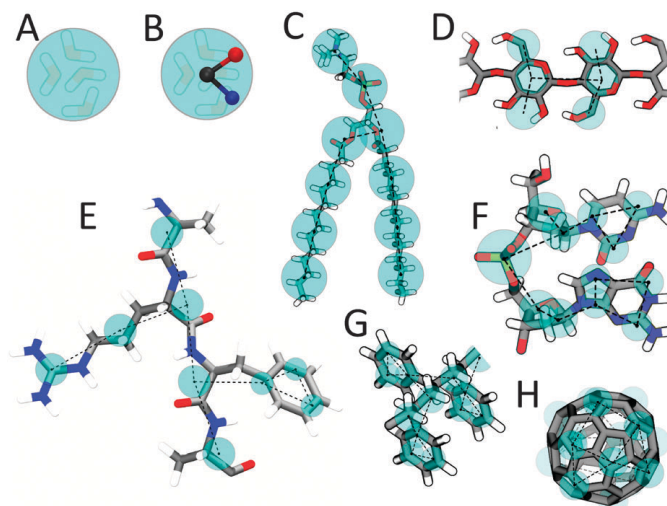
## 1.5 MARTINI: A COARSE-GRAINED FORCE-FIELD

MARTINI is a CG FF originally developed by Marrink *et al.* [9] for organic solvents and lipids and then extended to proteins [11], carbohydrates [8] and a broad class of polymers [12]. 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 philosophy 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 whole the FF. This is possible, because, instead of focusing on the accurate reproducing of structural details of a particular system, it is based on the reproducing accurately the interactions between polar and non-polar chemical compounds. These are the main target properties: The *partitioning free energy* between water and a large number of organic systems, 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, this is the guide for the mapping procedure.

### 1.5.1 Mapping

The mapping of 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 modeled with one MARTINI bead. In figure (8) examples and comparisons of some molecules between atomistic and MARTINI FF is shown. There are



**Figure 8:** MARTINI mapping and atomistic structures compares of some molecules: (A) Standard water bead, (B) polarizable water bead, (C) DMPC lipid, (D) Polysaccharide fragment, (E) Peptide, (F) DNA fragment, (G) Polystyrene fragment and (H) Fullerene molecule. In all cases MARTINI CG beads are shown as cyan transparent beads overlaying the atomistic structure. Taken from [10].

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 subtypes 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), ..., highest polarity (5).

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

Level	$\epsilon$ [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

### 1.5.2 Interactions potential

**VAN DER WAALS INTERACTIONS** The functional form describing the pairwise Van der Waals interaction is a Lennard–Jones 12 – 6 potential as in equation (1.4.2). For the most beads the  $\sigma$  parameter is set equal to 0.47 nm except for the Q–C<sub>1</sub> and Q–C<sub>2</sub> interactions that are set to  $\sigma = 0.62$  nm. This is consistent for reproducing the hydration shell when a charged beads (Q) is dragged into an apolar medium. The strength of the interactions is instead divided into ten levels, reported in table (1) The association of the interactions strength with the MARTINI beads is shown in figure (9).

	sub	Q				P					N				C				
		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	I	III	I	IV	V	VI	VII	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	I	IV	V	VI	VI	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
N	1	I	I	I	III	O	II	II	II	II	II	II	II	III	IV	IV	IV	V	VI
	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	II	III	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
C	0	IV	IV	IV	IV	IV	IV	IV	III	III	IV	IV	IV	IV	IV	IV	IV	V	VI
	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	V	V	V	IV	IV	IV

**Figure 9:** Interactions strength associations matrix for the MARTINI beads type and subtype. Taken from [9].

**ELECTROSTATIC INTERACTIONS** The electrostatic charges are assigned using the atom–centered approximation, as described in 1.4.5. But, in this

case, the charges are no more fractional and they are empirically assigned at the center of the beads trying to following as much as possible the net charge of the associated chemical moieties. A special case is the water, modeled as a P<sub>4</sub> bead, since it interact only with the Van der Waals interaction the polarizability effect is not very well described. For repair this lack it is used an implicit medium with a dielectric constant  $\epsilon_r = 15$ . However, as we will see later in 1.5.5, for avoid the problems of the implicit medium described in 1.4.6, especially for the main aim of the MARTINI FF, the lipids membranes for that the dielectric constant in the hydroponic region is much smaller, Yesylevskyy *et al.* [15] has developed a more sophisticated CG water model, called *polarizable water* (PW), to take into account a better water's behavior.

**BONDED INTERACTIONS** They includes only a bond length and an angle harmonic contributions. The former use a harmonic potential as the first term in equation (1.4.1), with the same bond constant for all beads type:  $k^b = 1250 \text{ kJ}/(\text{mol nm}^2)$  and an equilibrium distance of  $l_0 = 0.47 \text{ nm}$ . The later is modeled as a cosine-type harmonic potential

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

and the parameters are:  $k^a = 25 \text{ kJ}/\text{mol}$  and  $\theta_0 = 180^\circ$  for aliphatic chains;  $k^a = 45 \text{ kJ}/\text{mol}$  and  $\theta_0 = 120^\circ$  for *cis* double bonds and  $k^a = 45 \text{ kJ}/\text{mol}$  and  $\theta_0 = 180^\circ$  for *trans* unsaturated bonds. Moreover, especially 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  $\theta_{ijkl}$  denotes the angle between the planes constituted between atoms  $i, j, k$  and  $j, k, l$ ;  $k_{id}$  and  $\theta_0$  are, as usual, the force constant and equilibrium angle.

### 1.5.3 Simulation parameters

MARTINI FF was originally developed using a shifted cut-off scheme for both Lennard-Jones and electrostatic potentials with a cut-off radius of  $r_c = 1.2 \text{ nm}$ . The Lennard-Jones was shifted from  $r_s = 0.9 \text{ nm}$  to  $r_c$  while from  $r_s = 0.0 \text{ nm}$  to  $r_c$  for the electrostatic potential. The neighbor list is constructed as described in the first part of 1.3.4 with a refresh rate of 10 MD steps. Recently the more efficient Verlet cut-off scheme was tested by

Marrink *et al.* [10] and used with the MARTINI FF with a cut-off radius of  $r_c = 1.1$  nm, a Verlet buffer tolerance of 0.005 kJ/(mol·ps) and a minimum refresh rate of 10 MD steps (often, depending on the hardware it can be dynamically increased to 30 or 40 MD steps getting a performance gain). Moreover, the treatment of the electrostatic interaction can be safely updated to the PME method together with the Verlet cut-off scheme. These were largely tested by Yesylevskyy *et al.* [15]. In this case the cut-off radius was set to  $r_c = 1.2$  nm with the same Verlet buffer tolerance; the PME grid spacing was set to have a 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), as we shall see, the dielectric constant should be reduced to  $\epsilon_r = 2.5$ . 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 these simulation parameters must be followed by a retest of the main properties of the MARTINI FF.

#### 1.5.4 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. Similarly 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

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

These simulations were 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 particle for that the partitioning free energy one wants to compute, is placed in the simulation box. From the equilibrium densities of the particles in water  $\rho_{\text{wat}}$  and in organic solvent  $\rho_{\text{oil}}$ , the free energy of transfer can be computed from

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

where S indicate the organic solvent.

In figure (10) a summary of the results is reported. 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 vapor-liquid system, but as long as one does not study that systems, the partitioning free energy is much important then the other.

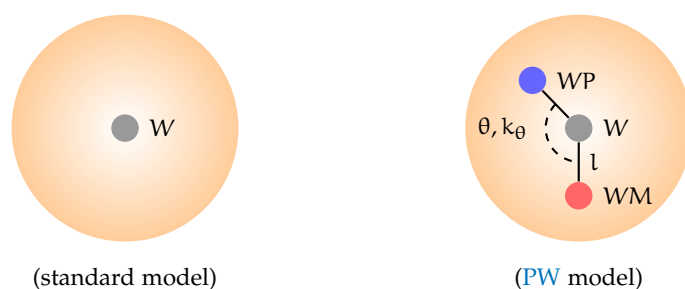
type	building block	examples	$\Delta G^{\text{vap}}$		$\Delta G^{\text{hyd}}$		$\Delta G^{\text{part}}_{\text{HW}}$		$\Delta G^{\text{part}}_{\text{CW}}$		$\Delta G^{\text{part}}_{\text{EW}}$		$\Delta G^{\text{part}}_{\text{OW}}$	
			exp	CG	exp	CG	exp	CG	exp	CG	exp	CG	exp	CG
Q <sub>da</sub>	H <sub>3</sub> N <sup>+</sup> -C <sub>2</sub> -OH	ethanolamine (protonated)			-25		< -30		-18		-13		-18	
Q <sub>d</sub>	H <sub>3</sub> N <sup>+</sup> -C <sub>3</sub>	1-propylamine (protonated)			-25		< -30		-18		-13		-18	
	NA <sup>+</sup> OH	sodium (hydrated)			-25		< -30		-18		-13		-18	
Q <sub>a</sub>	PO <sub>4</sub> <sup>-</sup>	phosphate			-25		< -30		-18		-13		-18	
	CL <sup>-</sup> HO	chloride (hydrated)			-25		< -30		-18		-13		-18	
Q <sub>0</sub>	C <sub>3</sub> N <sup>+</sup>	choline			-25		< -30		-18		-13		-18	
P <sub>5</sub>	H <sub>2</sub> N-C <sub>2</sub> =O	acetamide	sol	sol	-40	-25	-27	-28	(-20)	-18	-15	-13	-8	-10
P <sub>4</sub>	HOH (× 4)	water	-27	-18	-27	-18	-25	-23		-14	-10	-7	-8	-9
	HO-C <sub>2</sub> -OH	ethanediol	-35	-18	-33	-18	-21	-23		-14		-7	-8	-9
P <sub>3</sub>	HO-C <sub>2</sub> =O	acetic acid	-31	-18	-29	-18	-19	-21	-9	-10	-2	-6	-1	-7
	C-NH-C=O	methylformamide	-35	-18		-18		-21		-10		-6	-5	-7
P <sub>2</sub>	C <sub>2</sub> -OH	ethanol	-22	-16	-21	-14	-13	-17	-5	-2	-3	1	-2	-2
P <sub>1</sub>	C <sub>3</sub> -OH	1-propanol	-23	-16	-21	-14	-9	-11	-2	-2	0	1	1	-1
		2-propanol	-22	-16	-20	-14	-10	-11	-2	-2	-1	1	0	-1
N <sub>da</sub>	C <sub>4</sub> -OH	1-butanol	-25	-16	-20	-9	-5	-7	2	0	4	2	4	3
N <sub>d</sub>	H <sub>2</sub> N-C <sub>3</sub>	1-propylamine	-17	-13	-18	-9	(-6)	-7	(1)	0	(-3)	2	(3)	3
N <sub>a</sub>	C <sub>3</sub> =O	2-propanone	-17	-13	-16	-9	-6	-7	1	0	-1	2	-1	3
	C-NO <sub>2</sub>	nitromethane	-23	-13	-17	-9	-6	-7		0		2	-2	3
	C <sub>3</sub> =N	propionitrile	-22	-13	-17	-9	-5	-7		0		2	1	3
	C-O-C=O	methylformate	-16	-13	-12	-9	(-6)	-7	(4)	0	(-1)	2	(0)	3
	C <sub>2</sub> HC=O	propanal	-13	-15	-9	-4	-7		0	2	2	3	3	
N <sub>0</sub>	C-O-C <sub>2</sub>	methoxyethane	-13	-10	(-8)	-2	(1)	-2		6	(3)	6	(3)	5
C <sub>5</sub>	C <sub>3</sub> -SH	1-propanethiol	-17	-10		1		5		10		10		6
	C-S-C <sub>2</sub>	methyl ethyl sulfide	-17	-10	-6	1	(7)	5		10		10	(9)	6
C <sub>4</sub>	C <sub>2</sub> =C <sub>2</sub>	2-butyne	-15	-10	-1	5		9		13		13	9	9
	C=C-C=C	1,3-butadiene		-10	2	5	11	9		13		13	11	9
	C-X <sub>4</sub>	chloroform	-18	-10	-4	5	(7)	9	14	13		13	11	9
C <sub>3</sub>	C <sub>2</sub> =C <sub>2</sub>	2-butene		-10		5		13		13		13	13	14
	C <sub>3</sub> -X	1-chloropropane	-16	-10	-1	5	12	13		13		13	12	14
		2-bromopropane	-16	-10	-2	5		13		13		13	12	14
C <sub>2</sub>	C <sub>3</sub>	propane	gas	-10	8	10		16		15		14	14	16
C <sub>1</sub>	C <sub>4</sub>	butane	-11 <sup>b</sup>	-10	9	14	18	18		18		14	16	17
		isopropane	gas	-10	10	14		18		18		14	16	17

**Figure 10:** Results summary: free energies of vaporization  $\Delta G^{\text{vap}}$ , hydration  $\Delta G^{\text{hyd}}$  and partitioning  $\Delta G^{\text{part}}$  between water (W) and organic solvents (hexadecane (H), chloroform (C), ether (E) and octanol(O)) compared to experimental values. Experimental properties between parentheses are estimates obtained from comparison to similar compounds. The statistical accuracy of the free energies obtained from the simulations is  $\pm 1$  kJ/mol. <sup>b</sup> The temperature for the experimental data is 273 K. Taken from [9].

### 1.5.5 Polarizable Water model

Water play a crucial role in any biomolecular systems then it is important to describe correctly its behavior. Since the MARTINI water model does not take into account directly the electrostatic interactions between water and the other molecules because it does not have any charge and it interact only via Van der Waals interactions, for take into account the main effects

of water, screening and polarizable effects, an simple implicit medium is used. However any biomolecular process involve charged species moving between region of different dielectric constant. Due to the change in electrostatic screening between those environments, the strength of the interactions between the moving charges and the surrounding molecules also changes, but this effect can not be consider in an implicit medium model. Thus can have important consequences for the way biological activity is controlled. In order to capture the inhomogeneous nature of the dielectric response an explicit medium have to be used.



**Figure 11:** Schematic representation of the **PW** bead. Shaded orange spheres correspond to the Van der Waals radii of the central neutral particle  $W$ . The blue particle is the positively charged while the red is the negatively charged.

In same the fashion of the **MARTINI** philosophy, Yesylevskyy *et al.* [15] has developed a Polarizable Water (**PW**) model that better describe the real behavior of water. As before four water molecules is associated to one **PW** bead. The new water bead actually consists of three particles instead of one in the standard **MARTINI** model. In figure (11) the topology of the **PW** and a comparison between the old model is shown. The central particle  $W$  is neutral and interacts with other particles in the system with only Lennard-Jones potential, just like the standard water bead thus as a  $P_4$  **MARTINI** bead (see figure (9) for the interactions matrix). There are two additional particles, namely  $WP$  and  $WM$ , that are bound to the central particle and carry a positive and negative charge  $|q| = 0.46e$  respectively, where  $e = 1.60217653(14) \cdot 10^{-19}$  C is the unit electron charge. They interact with other particles in the system by the Coulomb interaction only. The bonds  $W - WP$  and  $W - WM$  are constrained to have a fixed distance  $l = 0.14$  nm. The electrostatic interaction between  $WP$  and  $WM$  inside the same bead are exclude thus they are transparent toward each other and they can rotate around the  $W$  particle. As a consequence the dipole momentum of the water bead depends on the relative angular position  $\theta$  of  $WP$  and  $WM$ : It can vary from zero ( $\theta = 0$ ), to  $2ql$  ( $\theta = \pi$ ). A harmonic angle potential with equilibrium angle fixed to  $\theta_0 = 0$  and a force constant  $k_\theta = 4.2$  kJ/(mol $\cdot$ rad $^2$ )



is added to control the rotation of WP and WM particles around the W particle so to adjust the distribution of the dipole momentum. The value of the equilibrium angle is consistent with the fact that in an apolar medium the total dipole momentum of a water molecule is zero. Since in this model the screening and polarization effects are treated explicitly the global dielectric constant is then reduced from  $\epsilon_r = 15$ , used in the standard MARTINI, to  $\epsilon_r = 2.5$ . Moreover, since the PW beads attract each other stronger than the standard water beads because of additional electrostatic interactions the strength  $\epsilon_{WW}$  of the Lennard-Jones interaction between W particles must be reduced. They found that change the Lennard-Jones strength from an I level to an III level can do the trick (see table (1) for the various interaction levels). While  $\sigma$  remains set to 0.47 nm.

The parametrization of  $q$ ,  $k_\theta$  and  $\epsilon_{WW}$  are obtained, in addition to the basic target properties of the MARTINI FF, also trying to reproduce the dielectric constant, density and dipole momentum of a pure water phase. For a more details about the parameterizations and testing methods the reader is addressed to the article by Yesylevsky *et al.*[15]. An important question is the use of the PME method to treat the long range electrostatic interaction. The authors found that, despite the loss of performance, in addition to the PW model, the use of the PME method contributes to a more realistic description of the processes involved in a biomolecular environments. In particular, some interesting results of utility for this thesis work, concern a better description of some properties of lipids membranes. One is the translocation of charged ions through a lipid bilayer that is described in a more realist detail: The authors found that the simulations with PW and PME are approaching the atomistic results better than the standard MARTINI FF. Another important phenomenon about lipids membranes consist in the electroporation of membrane by PW beads due to an electric field across the membrane (for example, created by a cross membrane ions imbalance) and, related to this, the translocation of ions across the membrane helped by a so called “water finger” a water defect inside the membrane that seems to be the trigger of the translation. Such phenomenons are clearly evident in atomistic simulation however never occurred using the standard MARTINI FF thus the importance to use the PW model together with PME method.

#### 1.5.6 Limitations of MARTINI FF

As we have seen in 1.4.7 the CG FFs are computationally advantages, however a price has to be paid. Although the MARTINI FF is still a finer CG FF, some limitations are shared with other CG models at a fundamental level, such as the chemical and spatial resolution, which are both limited

compared to atomistic models. One of the most important question is the **DOF** reduction: This affects the entropy of the system which is underestimated and, as consequence of the **MARTINI** parameterizations, also the balance between enthalpy is not correct described. Since the **MARTINI** model is built with the constraint that the partitioning free energy  $\Delta G = \Delta H - T\Delta S$  must be conserved with the atomistic case, together with an intrinsic loss of entropy, the enthalpic contribution must decrease with respect to the atomistic model<sup>3</sup>. Moreover this mean that the temperature dependence of a **CG** model is a priori not correct, anyway this is not our case since we use only a **NVT** or a **NPT** ensemble with an external thermostat coupling.

Another consequence of **CG FFs** is related to the **PEF** that becomes smoother respect to the atomistic case. This effectively results in more sampling of the energy landscape in a given time period, speeding up the kinetics of the system and allowing the use of an higher time steps with longer simulation times. However the speed-up is not easily predictable and is not likely to be the same for different systems and maybe it is dependent on the type of molecule. Nevertheless for the **MARTINI FF** an average scaling factor of 4, based on lateral diffusion coefficients of lipids in membranes, is commonly used, of course with some care. An other smaller source of errors is due to the choice of masses: Since ensemble properties is not affected by particles mass, in order to increase the efficiency, all the **MARTINI** beads have the same mass of 72 amu. Thus leads in some uncertainty in the dynamics of the system making this time scaling for different beads non-trivial.

A problem involving the Lennard–Jones potential as a model of the Van der Waals interaction in **MARTINI**, is that the steep repulsion leads to an over-structuring of fluids compared to atomistic models. The direct and most evident implication is the melting point of the water that is  $290 \pm 5$  K. A practical partial solution is the use of the so called “anti-freeze” particles named **BP<sub>4</sub>** type. The Lennard–Jones interaction between these particles and water is modified with a slightly larger Van der Waals radius parameter,  $\sigma = 0.57$  nm and a stronger interaction to be a level O (see table (1) for the interaction level). Marrink *et al.* suggest that a mole fraction of  $n_{af} = 0.1$  is sufficient to prevent freezing without affecting the other properties of water. Some other properties of water are not accurate such us the surface tension of air/water interface that leads in problem to water/oil interfaces formation. Using the **PW** model these properties improve slightly. For a more comprehensive discussion about the limitations of the **MARTINI FF** the reader is addressed to review by Marrink *et al.* [10].

<sup>3</sup> If a **NVT** ensemble is used the correct potential is  $\Delta A = \Delta U - T\Delta S$ , thus the incorrect balance is between the internal energy and the entropy

## 1.6 ADVANCED SAMPLING METHODS

### 1.6.1 Metadynamics

### 1.6.2 Umbrella sampling



2 | DUE



# 3 | TRE





## BIBLIOGRAPHY

- [1] M.P. Allen and D.J. Tildesley. *Computer Simulation of Liquids*. Oxford Science Publ. Clarendon Press, 1989. ISBN: 9780198556459. URL: <https://books.google.it/books?id=032VXB9e5P4C>.
- [2] Giovanni Bussi, Davide Donadio, and Michele Parrinello. "Canonical sampling through velocity rescaling". In: *The Journal of Chemical Physics* 126.1, 014101 (2007). DOI: <http://dx.doi.org/10.1063/1.2408420>. URL: <http://scitation.aip.org/content/aip/journal/jcp/126/1/10.1063/1.2408420>.
- [3] 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>.
- [4] Tom Darden, Darrin York, and Lee Pedersen. "Particle mesh Ewald: An  $N \cdot \ln(N)$  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>.
- [5] 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>.
- [6] 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=5qTzldS9R0IC>.
- [7] A. R. Leach. *Molecular Modelling: Principles and Applications*. Pearson Education. Prentice Hall, 2001. ISBN: 9780582382107. URL: <https://books.google.it/books?id=kB7jsbV-uhkC>.
- [8] 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](http://dx.doi.org/10.1021/ct900313w). URL: <http://dx.doi.org/10.1021/ct900313w>.

- [9] 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>.
- [10] Siewert J. Marrink and D. Peter Tieleman. "Perspective on the Martini model". In: *Chem. Soc. Rev.* 42 (16 2013), pp. 6801–6822. DOI: [10.1039/C3CS60093A](http://dx.doi.org/10.1039/C3CS60093A). URL: <http://dx.doi.org/10.1039/C3CS60093A>.
- [11] 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>.
- [12] 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](http://dx.doi.org/10.1039/C0SM00481B). URL: <http://dx.doi.org/10.1039/C0SM00481B>.
- [13] M. Tuckerman. *Statistical Mechanics: Theory and Molecular Simulation*. Oxford Graduate Texts. OUP Oxford, 2010. ISBN: 9780191523465. URL: <https://books.google.it/books?id=Lo3Jqc0pgrcC>.
- [14] L. Verlet. "Computer "Experiments" on Classical Fluids. II. Equilibrium Correlation Functions". In: *Phys. Rev.* 165 (1 Jan. 1968), pp. 201–214. DOI: [10.1103/PhysRev.165.201](http://link.aps.org/doi/10.1103/PhysRev.165.201). URL: <http://link.aps.org/doi/10.1103/PhysRev.165.201>.
- [15] 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](http://dx.doi.org/10.1371/journal.pcbi.1000810). URL: <http://dx.doi.org/10.1371/journal.pcbi.1000810>.