

KINETICS AND THERMODYNAMICS OF INTERACTION BETWEEN  
MONOLAYER-PROTECTED, CHARGED METAL NANOPARTICLES  
AND MODEL BIOLOGICAL MEMBRANES

SEBASTIAN SALASSI

SUPERVISOR: DR. GIULIA ROSSI

ADVISOR: PROF. ANNALISA RELINI

*What do you get if you multiply six by nine?*

*Six by nine. Forty two.*

*That's it. That's all there is.*

*I always thought something was fundamentally wrong with the universe.*

DOUGLAS ADAMS – The Restaurant at the End of the Universe

## LIST OF ACRONYMS

<b>ALU</b>	arithmetic logic unit
<b>AuNP</b>	gold nanoparticle
<b>CG</b>	coarse-grained
<b>CH<sub>2</sub></b>	methylene
<b>CH<sub>3</sub></b>	methyl
<b>COM</b>	center of mass
<b>CV</b>	collective variable
<b>DMPC</b>	1,2-dimyristoyl- <i>sn</i> -glycero-3-phosphocholine
<b>DOF</b>	degrees of freedom
<b>ESM</b>	ewald summation method
<b>FES</b>	free energy surface
<b>FF</b>	force field
<b>FFT</b>	fast fourier transform
<b>FPU</b>	floating point unit
<b>MD</b>	molecular dynamics
<b>MUS</b>	11-mercaptoundecane sulphonate
<b>NP</b>	nanoparticle
<b>OT</b>	octanethiol
<b>PBC</b>	periodic boundary conditions
<b>PEF</b>	potential energy function
<b>PME</b>	particle mesh ewald
<b>POPC</b>	1-palmitoyl-2-oleoyl- <i>sn</i> -glycero-3-phosphocholine
<b>PP</b>	particle-particle
<b>PW</b>	polarizable water
<b>SIMD</b>	single instruction – multiple data

$\text{SO}_4^-$  sulphonate

**WHAM** weighted histogram analysis method

# CONTENTS

LIST OF ACRONYMS	iii
1 INTRODUCTION TO MOLECULAR DYNAMICS	1
1.1 Review of classical mechanics	1
1.2 Review of statistical mechanics	3
1.2.1 Microcanonical ensemble	6
1.2.2 Isothermal-isobaric ensemble	7
1.3 Molecular Dynamics simulation	9
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 Barostat algorithms	16
1.4 Empirical Force-Field model	19
1.4.1 Inter-particle interactions	20
1.4.2 Non-bonded interactions: general aspects	22
1.4.3 Van der Waals interactions	23
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	34
1.5.2 Interactions potential	35
1.5.3 Simulation parameters	37
1.5.4 Parametrization	37
1.5.5 Polarizable Water model	38
1.5.6 Limitations of MARTINI FF	41
1.6 Advanced sampling methods	42
1.6.1 Umbrella sampling	44
1.6.2 Metadynamics	47
1.6.3 Umbrella sampling and metadynamics remarks	51
2 MD TOOLS, PARALLELIZATION AND HARDWARE ACCELERATION	53
2.1 GROMACS package	53
2.1.1 Parallelization	53
2.1.2 PP and PME nodes	54
2.1.3 Hardware acceleration	56
2.2 PLUMED package	57

2.3 Simulation parameters . . . . .	59
3 MODEL OF CELL MEMBRANE AND MONOLAYER-PROTECTED NP	61
3.1 NanoParticle model . . . . .	61
3.1.1 Passivated gold core . . . . .	62
3.1.2 Functionalizing ligands . . . . .	63
3.2 Cell membranes . . . . .	65
3.2.1 Real cell membranes . . . . .	65
3.2.2 Model cell membrane . . . . .	68
3.3 NP-Membrane interaction . . . . .	70
3.3.1 Three-stage process . . . . .	70
3.3.2 Preliminary metadynamics results . . . . .	71
4 RESULTS FROM METADYNAMICS SIMULATIONS	73
4.1 Metadynamics set-up . . . . .	73
4.2 Models compare . . . . .	73
4.3 Convergence problems . . . . .	73
4.4 Head lipids dragging . . . . .	73
4.5 Membrane leaflet engulfment . . . . .	73
5 RESULTS FROM UNBIASED SIMULATIONS	75
6 SUMMARY AND CONCLUSIONS	77
	78

# 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 Newton's laws, 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, 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 with some details about the *empirical* force field (FF): the container of system model and parameters under study. This includes the way to consider and treat the inter-particle interactions at MD level. Then, coarse-graining procedure are introduced, with particular attention to the MARTINI FF, the main FF used in this thesis work. Lastly advanced sampling techniques are explained: the *Umbrella Sampling* and *Metadynamics*. This introductory Chapter are based on the books of Tuckerman [29], Leach [16], Frenkel and Smit [8] and Allen and Tildesley [2] to which the reader is addressed for a more complete discussion.

### 1.1 REVIEW OF CLASSICAL MECHANICS

Let us consider a system of  $N$  particles with mass  $m_i$  and coordinates  $\vec{r}_1, \dots, \vec{r}_N$ . According to the Newton's second law each particle will experience a total force  $\vec{F}_i$  such that

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

The total force on each particle is defined as

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

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 referred 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 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. For simplicity of notation let us introduce also the ordered vectors  $\vec{r} = (\vec{r}_1, \dots, \vec{r}_N)$ ,  $\dot{\vec{r}} = (\dot{\vec{r}}_1, \dots, \dot{\vec{r}}_N)$  and  $\vec{p} = (\vec{p}_1, \dots, \vec{p}_N)$  which are the coordinates, velocities and momenta off all particles.

Let us suppose that all the forces acting on the  $N$ -particle system are conservative; this means that it must exist a scalar function  $U = U(\vec{r})$  called potential energy function (PEF), such that

$$\vec{F}(\vec{r}) = -\partial_{r_i} U(\vec{r}) \hat{e}_i = -\vec{\nabla}_{\vec{r}} U(\vec{r}) \quad (1.1.4)$$

where  $\vec{F} = (\vec{F}_1, \dots, \vec{F}_N)$ . Thus we have only to know the PEF of the system at any time and the initial conditions to solve Newton's laws.

The kinetic energy of the system is defined as

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

If the system is conservative we can define a scalar function, called *Lagrangian* of the system

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

such that

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{r}_i} \right) - \frac{\partial \mathcal{L}}{\partial r_i} = 0 \quad (1.1.7)$$

These set of  $3N$  equations are called *Euler–Lagrange equations of motion*. It is easy to show that substituting the definition of  $\mathcal{L}$  (1.1.6) 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

$$p_i = \frac{\partial \mathcal{L}}{\partial \dot{r}_i} = m_i \dot{r}_i \quad (1.1.8)$$

thus we can express particle velocities as a function of particle momenta. Equations (1.1.5) and (1.1.8) let us to express the kinetic energy in the form

$$K(\vec{p}) = \sum_{i=0}^N \frac{1}{2m_i} \vec{p}_i \cdot \vec{p}_i \quad (1.1.9)$$

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

$$\mathcal{H}(\vec{r}, \vec{p}) = \sum_{i=0}^N \vec{p}_i \cdot \dot{\vec{r}}_i - \mathcal{L}(\vec{r}, \dot{\vec{r}}(\vec{p}))$$

*Hamiltonian formalism*

Substituting (1.1.6) and using (1.1.9), the Hamiltonian of the system is nothing that

$$\mathcal{H}(\vec{r}, \vec{p}) = K(\vec{p}) + U(\vec{r}) \quad (1.1.10)$$

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

$$\begin{aligned} \dot{r}_i &= \frac{\partial \mathcal{H}}{\partial p_i} \\ \dot{p}_i &= -\frac{\partial \mathcal{H}}{\partial r_i} \end{aligned} \quad (1.1.11)$$

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 exam, 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* properties of such system. The main task of this process is to obtain the thermodynamics properties 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.

The solution of that problem comes from *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 to each other. This provides also a rigorous derivation of thermodynamics from the microscopic properties: 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.* This it is intuitively true. We can think to set up an experiment with a system in a specific microscopic state that corresponds to a macroscopic state. Certainly we can do the contrary and for sure we will not find the same microscopic state. Then we can iterate the experiment and we will find that for a specific macroscopic state of a system there exists a number of microscopic states that yield to the same properties.

*statistical ensemble concept*

The most important concept that makes this idea practicable is the *statistical ensemble*. Based on the previous considerations, a general definition of an ensemble is *a collection of systems subject to a set of common interactions and sharing the same macroscopic properties.* This concept lays the foundations of thermodynamic and suggests a procedure to compute many macroscopic observables. In more details a  $N$ -particle system in a specific microscopic state is described by its microstate  $\vec{x}$ , 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 systems in the ensemble. To do this we have to know, at every time, which microstates of the phase space are part of the ensemble. For this purpose we define the *ensemble distribution function*  $\tilde{\rho} = \tilde{\rho}(\vec{x}, t)$ . Let  $dx = dr_1 \cdots dr_{3N} dp_1 \cdots dp_{3N}$  the infinitesimal phase space volume, then

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

where  $N$  is the total number of microstates in the ensemble,  $\rho(\vec{x}, t)dx$  can be interpreted as the probability to find a system in the ensemble with microstate  $\vec{x}$  at a time  $t$  and  $\rho(\vec{x}, t)$  is the more convenient normalized ensemble 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. Thus, it can

be shown [29] that exist a scalar function of the Hamiltonian  $f(\mathcal{H}(\vec{x}))$  such that the time-independent ensemble average of the equilibrium observable  $A$  can be expressed as

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

where  $Z$ , known as *partition function*, is specific for the ensemble in exam and it is defined as follow

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

*ensemble averages*

In order to compute the partition function we need to specify 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–μpT

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 particle 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, if the system is ergodic, the time and ensemble averages are equal *almost everywhere* in the phase space. So we can write the follow identity

$$\bar{A} = \lim_{\tau \rightarrow +\infty} \frac{1}{\tau} \int_{t_0}^{\tau} A(\vec{x}(t)) \, dt = \int_{\Omega} A(\vec{x}) \rho(\vec{x}, t) \, dx = \langle A \rangle(t) \quad (1.2.1)$$

*ergodic theorem and ergodic hypothesis*

For biomolecular applications 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 to define the partition function as follow

$$\mathcal{Z}_{NVE} = \frac{1}{N!h^{3N}} \int_{\Omega} \delta(\mathcal{H}(\vec{x}) - E) dx \quad (1.2.2)$$

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. The right thermodynamic potential, i.e. the constant of motion, to obtain all the macroscopic observables is the entropy  $S$  given by

$$S = k_B \ln \mathcal{Z}_{NVE}$$

where  $k_B = 1.3806505(24) \cdot 10^{-23}$  J/K is the Boltzmann constant. The other thermodynamic quantities can be obtained by

$$\frac{1}{T} = \left( \frac{\partial S}{\partial E} \right)_{NV} \quad \frac{p}{T} = \left( \frac{\partial S}{\partial V} \right)_{NE} \quad \frac{\mu}{T} = \left( \frac{\partial S}{\partial N} \right)_{VE}$$

The link between microscopic functions of phase space points and macroscopic observables, like kinetic energy or pressure, is provided by the *classical virial theorem* which states that

$$\left\langle x_i \frac{\partial \mathcal{H}}{\partial x_k} \right\rangle = k_B T \delta_{ik} \quad (1.2.3)$$

where  $x_i$  is some phase space variable.

**KINETIC ENERGY** Since a system in a microcanonical ensemble is defined to be Hamiltonian, from equations (1.1.10) and (1.1.9) according to the virial theorem with the choice  $x_i = p_i$  we obtain

$$\left\langle \frac{p_i^2}{m_i} \right\rangle = k_B T$$

then summing both side over all particles and over the three coordinates we obtain the average kinetic energy at equilibrium

$$\langle K \rangle = \left\langle \sum_{i=1}^N \sum_{\alpha=1}^3 \frac{p_{i+\alpha}^2}{2m_i} \right\rangle = \left\langle \sum_{i=1}^N \frac{\vec{p}_i \cdot \vec{p}_i}{2m_i} \right\rangle = \frac{3}{2} N k_B T \quad (1.2.4)$$

this is like the well known equipartition theorem in which  $3N$  is the number of degrees of freedom (DOF).

**PRESSURE** Choosing  $x_i = r_i$  in equation (1.2.3) and summing both side over all particles and substituting equations (1.1.11) and (1.2.4), we obtain

$$W = \left\langle \sum_{i=1}^{3N} r_i \dot{p}_i \right\rangle = -3N k_B T = -2 \langle K \rangle$$

the quantity  $W$  is often called *virial*. For a  $N$ -particle non-interacting system it is well known that  $pV = Nk_B T$  thus the virial is  $W = -3pV$ . For a real interacting system we need to include in the virial the contribution due to the inter-particles interactions  $U(\vec{r}_1, \dots, \vec{r}_N)$ , thus the virial becomes<sup>1</sup>

$$W = -3pV + \left\langle \sum_{i=1}^N \sum_{j=i+1}^N r_{ij} f_{ij} \right\rangle = -2 \langle K \rangle$$

where  $r_{ij}$  and  $f_{ij}$  are the distance and force between particles  $i$  and  $j$ . Thus the pressure of the system is given by

$$p = \frac{1}{3V} \left( 2 \langle K \rangle + \left\langle \sum_{i=1}^N \sum_{j=i+1}^N r_{ij} f_{ij} \right\rangle \right)$$

The instantaneous pressure in terms of the phase space points  $\vec{x}(t)$  is obtained substituting equation (1.2.4) and getting the quantity in the average bracket, so

$$p(\vec{x}_t) = \frac{1}{3V} \sum_{i=1}^N \left( \frac{\vec{p}_i \cdot \vec{p}_i}{m_i} + \sum_{j=i+1}^N r_{ij} f_{ij} \right) \quad (1.2.5)$$

### 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 properties 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*

---

<sup>1</sup> We assume that the inter-particles interactions are pairwise additive which they depends only on distances between particles  $i$  and  $j$ .

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: it change the volume in order to 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

$$\mathcal{Z}_{NPT} = \frac{1}{N!h^{3N}} \int_0^{+\infty} dV \int_{\Omega} e^{-\beta(\mathcal{H}(\vec{x})+pV)} dx \quad (1.2.6)$$

where  $\beta^{-1} \equiv k_B T$  and the normalization factor is the same as in equation (1.2.2). The right thermodynamic potential, i.e. the constant of motion, to obtain the other thermodynamic quantities is the Gibbs free energy  $G = H - TS$  (where  $H$  is the enthalpy) 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 by

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

For anisotropic systems, volume can undergo different changes in the three Cartesian directories even if external pressure is applied isotropically. In these cases it is necessary to take this into account in the partition function. The way is to define a matrix formed by the three basis vectors of the system box  $\vec{a}$ ,  $\vec{b}$  and  $\vec{c}$

$$\mathbf{H} = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \quad (1.2.7)$$

so that its volume is  $V = \vec{a} \cdot \vec{b} \times \vec{c} = |\det \mathbf{H}|$ . The partition function become

$$\mathcal{Z}_{NpT} = \frac{1}{N!h^{3N}} \int d\mathbf{H} \int_{\Omega} \frac{1}{(\det \mathbf{H})^2} e^{-\beta(\mathcal{H}(\vec{x})+p|\det \mathbf{H}|)} dx$$

where  $\int d\mathbf{H}$  is an integral over all nine components of  $\mathbf{H}$ . Hence, the instantaneous pressure can no longer be described by a single quantity. Instead a  $3 \times 3$  pressure matrix  $\mathbf{P}$  is needed. What one find, if the system is isotropically coupled, is that on average, this pressure matrix reduces to a diagonal pressure matrix such that

$$\text{Tr} \langle \mathbf{P} \rangle = 3p$$

### 1.3 MOLECULAR DYNAMICS SIMULATION

molecular dynamics (MD) is a set of techniques that allow us to prepare a “computer experiments” in which solving numerically the classical equations of motion of a virtual system we are able to know its time evolution. Such virtual experiment approach has the advantage that many experiments can be set up with different initial conditions and/or with different control parameters, such as temperature or pressure. Obviously that experiment is carried out using a model that approximates the real system. The main parts of that model are the information required to obtain an approximation of the interactions among all system particles, i.e. 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 (1.2.1) 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 becomes

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

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).

#### 1.3.1 Initial configuration

In order to perform an MD simulation it is necessary to select an *initial configuration*. Its choice can be nontrivial and it depends on the complexity of the system. 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 assign all particle positions and velocities, i.e. all the  $6N$  coordinates of the initial phase space vector  $\vec{x}(0)$ . A common choice to assign the initial velocities is to extract them 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 has to rescale all the velocities in such a way that the total system’s momentum  $\vec{P} = \sum_{k=1}^N m_k \vec{v}_k$  is zero, this is

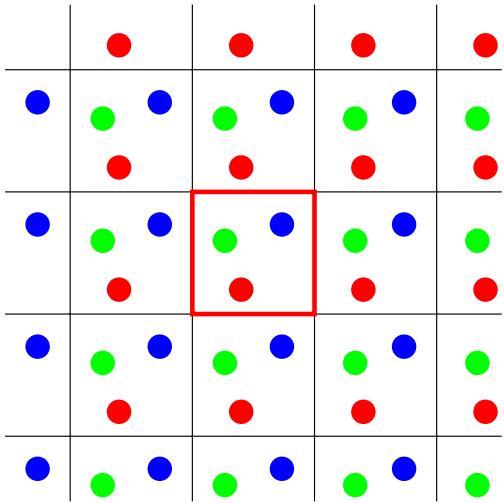


Figure 1: Schematic representation of an MD simulation.

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 DOF by three.

### 1.3.2 Periodic boundary conditions

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 coordinates. 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 too 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



**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.

particles is achieved adjusting the positions of the simulation box particles which have 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. Only box geometries compatible with translational symmetry can be used. For example, neither a spherical nor a icosahedral box could do the job. However 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 than 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 than 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 interactions that are treated with a more sophisticated methods, as 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  $t + \delta t$

$$\vec{r}_i(t + \delta t) \simeq 2\vec{r}_i(t) - \vec{r}_i(t - \delta t) + \frac{1}{m_i} \vec{f}_i(t) (\delta t)^2 \quad (1.3.5)$$

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 in equation (1.3.5) 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 of order  $o(\delta t)^2$ ,  $(\vec{f}_i(t)/(2m_i))$  which is summed to a difference of larger terms  $(2\vec{r}_i(t) - \vec{r}_i(t - \delta t))$ . This may cause a loss of precision due to computer numerical representation.

The main disadvantage is that 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  $t - \delta t$ . So at  $t = 0$  we need a trick to obtain the previous nonexistent positions. The trick is to use equation (1.3.4) truncated at the first order:  $\vec{r}_i(-\delta t) \simeq \vec{r}(0) - \vec{v}_i(0)\delta t$ .

### *Leap-Frog algorithm*

The leap-frog algorithm is a variant of the Verlet one and it is commonly implemented in many MD tools, as it is in our case. It computes the positions at a time

$t$  and the velocities at a time  $t + 1/2\delta t$  from the forces at a time  $t$  and the velocities at a time  $t - 1/2\delta t$ . 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  $t + 1/2\delta t$  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

$$\vec{v}_i(t) \simeq \frac{\vec{v}_i(t + 1/2\delta t) + \vec{v}_i(t - 1/2\delta t)}{2} \quad (1.3.6)$$

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 a gain of precision. The obvious disadvantage is that the positions and velocities are not synchronized so the equation (1.3.6) 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 kinetic energy contribution to the total energy: it must be computed with positions and velocities at the same time.

#### 1.3.4 Neighbor list

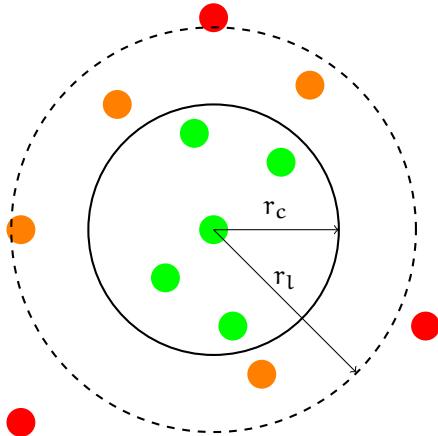
In order to solve the classical equations of motion, it is necessary to know the forces, and so the PEF, acting on the system's particles. As we shall see in detail in the next section, this is one of the most time consuming part of an MD simulation. To know the forces acting on one particle, in principle, it is necessary to calculate the contribution from all particles in the simulation box and all other periodic images. The most popular way to speed up the simulation is to use a truncation of the interaction potentials within a cutoff. The general idea is to compute the energy contribution of particle  $i$  considering the interaction only with particles  $j$  that are closer to a certain cutoff distance  $r_c$ , thus such that  $\|\vec{r}_i - \vec{r}_j\| \leq r_c$ . This is summarized in the following expressions

$$U_i(\vec{r}) = \sum_{j \neq i}^N U_{ij}^*(\vec{r}), \quad U_{ij}^*(\vec{r}) = \begin{cases} U_{ij}(\vec{r}) & \|\vec{r}_i - \vec{r}_j\| \leq r_c \\ 0 & \text{otherwise} \end{cases}$$

where  $U_{ij}(\vec{r})$  is the interaction potential between particles  $i$  and  $j$ .

By itself, the use of a truncation of the potential may not dramatically reduce the time spent in computing the inter-particle 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

**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 at every step is checked if their distances became smaller then  $r_c$ . Red particles are not in the pair-list and they are completely neglected until the next list update.



increase of 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 taking into account that, typically, in MD simulations of liquids system at ambient temperature and pressure, a particle neighbors do not change significantly for  $M < 20$  time steps.

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-particle energy contribution. To partially solve this problem, as suggested by Verlet [33], 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 then  $r_c$ . 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-updating 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*. 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 to consider the maximum distance traveled by a particle in the pair-list: if this distance is greater than  $r_b = r_l - r_c$  then the pair-list is certainly updated. Thus one can fix the refresh rate to a higher value increasing the performance.

### 1.3.5 Thermostat algorithms

A thermostat is an external tool that allows to maintain a system at a constant temperature. Several algorithms are available. Some are based acting on particle velocities (Anderson, Berendsen, Bussi) other introduce some more DOF in the system that take into account a real temperature bath coupling (Nosé–Hoover). We will describe in more detail the one used in this thesis work.

As suggested by Bussi *et al.* [4], a common practice to implement a thermostat acting on particle velocities 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  $\langle K \rangle$  obtained from equation (1.2.4) with the substitution  $3N \rightarrow N_f$

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

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

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

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, often called *weak coupling thermostat*, 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.* [4] 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

$$\alpha_T \equiv \sqrt{\frac{K_T}{\langle K \rangle}}$$

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 has 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 particle 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.* consists in 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

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

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

### 1.3.6 Barostat algorithms

As the thermostat maintains the system at a constant temperature  $T$ , a barostat is needed to maintain the system at a constant pressure  $p$ . To do this the system is coupled to a sort of piston so that changing the volume of the simulation box will adjust the pressure.

#### *Berendsen algorithm*

A common way, as proposed by Berendsen, is to scale both volume and particle coordinates. The rate of change of the pressure is given by

$$\frac{dp}{dt} = \frac{p_0 - p}{\tau_p}$$

where  $p_0$  is the target pressure,  $p$  is the instantaneous system pressure given by equation (1.2.5) and  $\tau_p$  is a coupling constant related to the response time of the barostat. Thus if we consider a time step  $\delta t$  then the volume scaling factor  $\lambda$  is given by

$$\lambda = 1 - k_T(p_0 - p) \frac{\delta t}{\tau_p}$$

where  $k_t$  is the isothermal compressibility defined as

$$k_T = -\frac{1}{V} \left( \frac{\partial V}{\partial p} \right)_T$$

while the factor  $\delta t/\tau_p$  gives a scaling factor for the isothermal compressibility that allow us to take into account the finite response time of the barostat. If  $\tau_p \rightarrow 0$  the system has an infinity isothermal compressibility so it is necessary a really small change in volume to achieve the correct pressure; on the contrary if  $\tau_p \rightarrow +\infty$  the system reduces to the Hamiltonian dynamics<sup>2</sup>. The particle coordinates is scaled by the factor  $\lambda^{1/3}$ .

In the case of anisotropic system the pressure matrix  $P$  and the volume matrix  $H$  (1.2.7) have to be considered and the Berendsen algorithm can be generalized such that even  $\lambda$  becomes a  $3 \times 3$  matrix. However the main problem, as for the weak coupling thermostat, is that the sampled ensemble is not known. Thus a new approach, in order to correct sample the isobaric ensemble, has been derived by Parrinello and Rahman.

#### *Parrinello–Rahman algorithm*

The Parrinello–Rahman barostat [22][23] is based on genuinely treat the coupling of the system to an external piston of “mass”  $M_h$  through a Lagrangian method in which the volume becomes a Lagrangian variable of the system and its time evolution can be computed solving the Euler–Lagrange equations. Both the size and the shape of the simulation box are allowed to fluctuate. So it is perfectly compatible with anisotropic system. The shape and size of the simulation box are described by the volume matrix  $H$  in equation (1.2.7) such that  $V = |\det H|$ .  $H$  is the Lagrangian coordinate of the external piston. If the external pressure  $p_0$  is applied to the piston then its potential energy is given by

$$U_p = p_0 |\det H|$$

instead, if  $H$  varies in time then a “kinetic energy” is associated to the piston as follow

$$K_p = \frac{1}{2} M_h \text{Tr}(\dot{H}^T \dot{H})$$

where  ${}^T(\cdot)$  denote the transpose operation.

In order to write the Lagrangian of the system the particle coordinates must be expressed in terms of  $H$ . This can be done defining the vector  $\vec{s}_i$  so that  $\vec{r}_i = H \vec{s}_i$ . The square displacement can be obtained as  $\vec{r}_i \cdot \vec{r}_i = {}^T(H \vec{s}_i) H \vec{s}_i = {}^T \vec{s}_i {}^T H H \vec{s}_i$ , often  $G = {}^T H H$ . Parrinello *et al.* write the Lagrangian as follow

$$\mathcal{L} = \frac{1}{2} \sum_{i=1}^N m_i \dot{\vec{s}}_i {}^T G \dot{\vec{s}}_i - \sum_{i=1}^N \sum_{j=i+1}^N U_{ij}(\vec{r}) + \frac{1}{2} M_h \text{Tr}({}^T \dot{H} \dot{H}) - p_0 |\det H|$$

---

<sup>2</sup> If the system is coupled to a thermostat, then the canonical ensemble is sampled.

thus the equations of motion for  $\mathbf{H}$  and  $\vec{s}_i$  can be computed solving the Euler–Lagrange equations. It can be shown that the equations of motion, derived in [22] and [23], correctly sample an isobaric ensemble. A practical way to show it, is to consider the Hamiltonian of the system. Using equation (1.1.10) the Hamiltonian is

$$\mathcal{H} = \frac{1}{2} \sum_{i=1}^N m_i \dot{\vec{s}}_i^t \mathbf{G} \dot{\vec{s}}_i + \sum_{i=1}^N \sum_{j=i+1}^N U_{ij}(\vec{r}) + \frac{1}{2} M_h \text{Tr}({}^t \dot{\mathbf{H}} \dot{\mathbf{H}}) + p_0 |\det \mathbf{H}|$$

if the system is in equilibrium at temperature  $T$ , using the equipartition theorem, the kinetic term of the system particles contributes to the energy by  $3Nk_B T/2$  while the kinetic term of the piston by  $9k_B T/2$ . Since  $3N \gg 9$  the Hamiltonian can be approximated by

$$\mathcal{H} \simeq \langle K \rangle + U + p_0 V = H$$

that is the enthalpy. Since the Hamiltonian is a constant of motion the correct isobaric ensemble is sampled. If the system is also coupled to a thermostat, the  $-TS$  term must be added, so the constant of motion becomes the Gibbs free energy  $G = H - TS$  and the isobaric–isotherm ensemble is correctly sampled.

In order to understand the meaning of the  $M_h$  parameter (the piston “mass”) we report the dynamic equation of  $\mathbf{H}$

$$M_h \ddot{\mathbf{H}} = V(p - p_0)({}^t \mathbf{H})^{-1}$$

The Parrinello–Rahman barostat is like a second order system. When there is an imbalance between the instantaneous internal pressure  $p$  (see equation (1.2.5)) and the target pressure  $p_0$ , the system recovers this imbalance in a characteristic time governed by the parameter  $M_h$ . Since at equilibrium the properties of a system are independent of the masses of its constituent parts,  $M_h$  can be arbitrarily chosen if one is interested only in static averages, otherwise a more appropriate choice must be made to obtain accurate dynamical properties. In this case, the authors suggest a choice of  $M_h$  such that the relaxation time is of the order of  $L/c$  where  $L$  is simulation box size and  $c$  is the sound velocity.

Interesting is the fact that this algorithm can be generalized to an anisotropic pressure coupling making use of the theory of elasticity. Differently from the Berendsen algorithm, Parrinello–Rahman method is much slower in changing the volume until the equilibrium value is reached and it is less stable: if the system is not well equilibrated can lead to a large volume fluctuation which can compromise the simulation success. On the other hand the Parrinello–Rahman samples correctly the isobaric ensemble. A common strategy is to use the Berendsen barostat to reach equilibrium then switch to the Parrinello–Rahman algorithm to correctly sample the phase space associated to the isobaric ensemble.

## 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  $N$ -particle system. 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. Considering soft or condensed matter, the most relevant for this thesis, the main processes of interest ranging from protein folding to glass transitions, from surface diffusion to ligand–receptor binding take place on longer time scales (several ns) and involve a larger number of atoms ( $N \gg 10^4$ ).

Hence with soft or condensed matter systems, in order to remove some DOF due to the necessity to increase the time step and speed up the simulation, 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. This is done in order to integrate out the high frequency electrons' motions. If, for some particular system, we want to know precisely the dynamics of the electrons we have to introduce quantum mechanical methods that are, even for a small number of particles ( $N \sim 10^2$ ), very computationally time consuming. 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 atom or molecule interactions, such as bond formation, are mediated by the electron 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–atom interactions* that mimics the interactions behavior. Since forces are derived from the PEF, a typical choice, when it is possible, for example with organic systems, is to model the inter–particle interactions by a set of pairwise additive potentials that avoid any many–body calculations. Nevertheless for some systems this is impracticable, for example, for metal cores we need to consider the many–body calculations or an effective many–body potential that take into account the behavior of all metal atoms in the core. A typical solution is to use the many–body potential developed by Gupta (see [16] for more details about metals treatment at MD level). The set of functional forms of the inter–particle interaction potentials and its parameterization are collected into the so called empirical force field (FF). The meaning of *empirical* is that most of the functional forms of the interactions has no “first principle” justification: they are chosen as a compromise between accuracy and computational efficiency.

For biomolecular applications two main classes of FFs exist: the *atomistic* FFs in which basic particles are atoms, and the *coarse–grained* (CG) FFs in which the basic particles represent atom groups or small chemical moieties. In this case, even the

way to realize the grouping, called *mapping*, is part of the FF itself. Different 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-particle interactions and how to treat them in an MD simulation. While in the next section we will focus on the main CG FF used in this thesis work: the MARTINI FF developed by Marrink *et al.* [19].

**PARAMETERIZATION** In general the functional forms for potential interactions are common to all particles in the system. 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 transition temperatures.

#### 1.4.1 Inter-particle interactions

For biomolecular applications the inter-particle 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. In the following the most common and general functional form for the PEF is described. In general the PEF can be always splitted in a bonded and non-bonded contributions:

$$U(\vec{r}) = U_b(\vec{r}) + U_{nb}(\vec{r})$$

The bonded contribution is generally described by the following terms

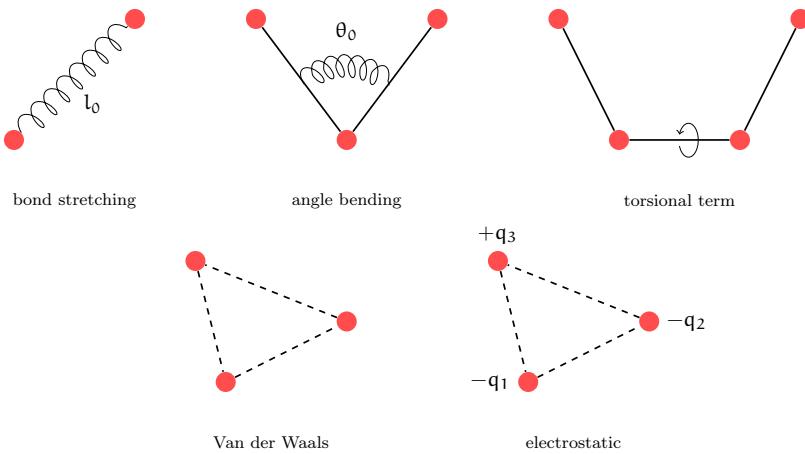
$$\begin{aligned} U_b(\vec{r}) = & \frac{1}{2} \sum_{\text{bonds}} \frac{1}{2} k_i^b (l_i - l_{i0})^2 + \frac{1}{2} \sum_{\text{angles}} k_i^a (\theta_i - \theta_{i0})^2 + \\ & + \frac{1}{2} \sum_{\text{torsions}} V_n (1 + \cos(n\omega - \gamma)) \end{aligned} \quad (1.4.1)$$

the first two terms are harmonic potentials which model respectively the energy contribution due to the deviation from reference bond length  $l_{i0}$  and a bond angle  $\theta_{i0}$ .  $k_{bi}$  and  $k_{ia}$  are the bond and angle elastic constants. The bond contribution involve a set of two particles in the same molecule while the angle contribution a set of three particles in the same molecule. The last term in equation (1.4.1) concerns the energy contribution due to the bond torsional change. It involves four particles in the same molecule and mimic the energy barrier needed to rotate the bond angle along the bond axis.  $\omega$  is the torsional angle,  $\gamma$  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 non-bonded contribution is described by the following terms

$$U_{nb}(\vec{r}) = \sum_{i=1}^N \sum_{j>i} \left( 4\epsilon_{ij} \left( \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right) + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right) \quad (1.4.2)$$

The first term is the energy contribution due to the Van der Waals interactions modeled as a Lennard-Jones 12 – 6 potential, fully characterized by the constants  $\sigma_{ij}$  and  $\epsilon_{ij}$  proper for each particles pair. The last term is the electrostatic energy contribution described by the particle 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-atom interactions for biomolecular applications: bond stretching, angle bending, torsional term, Van der Waals and electrostatic interactions.

### 1.4.2 Non-bonded interactions: general aspects

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. Moreover, usually in standard MD simulations the bond break is not taken into account. The non-bonded interactions, instead, are at non-fixed range because they depend on the inter-particles distance  $r_{ij}$  and 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 section 1.3.4 the calculations of non-bonded interaction energy contributions is one of the most time consuming part of an MD simulation. Even if we use a simple pairwise additive potential their calculation scale as  $\sim N^2$ . Thus, especially for short range interactions, various methods were developed in order to speed-up the simulation. The *cut-off* method is the most used to treat the short range interactions and, in some cases, even the long range ones. 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 derivatives, 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  $r_c$  so that  $v^*(r_c) = 0$ . Adding a constant will not affect the force calculations. The potential becomes

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

Moreover, to solve the discontinuity of the forces, that can cause some instability, a simple possibility is to consider a linear term proportional to the first derivative of the potential, such as

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

The shift methods can make the potential quite different from the original one. So this have to be properly taken into account in order to retrieve the correct thermodynamics proprieties.

Another powerful method to solve the discontinuity problem of the potential and forces 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 original form is used; while if  $r > r_{c2}$  the potential is set to zero. If  $r_{c1} < r \leq r_{c2}$  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, in turn, part of the FF. So they are interdependent with the model parameterization, and should never be changed without retesting the target properties of the parameterization.

#### 1.4.3 Van der Waals interactions

Van der Waals forces are a set of interactions that can be ascribed to quantum dynamic effects. In general they are described by a sum of a repulsive and an attracting term. The former effectually takes into account the Pauli exclusion principle between electron clouds; the latter is related to the dipole-dipole interactions (Keesom forces), dipole-induce dipole interactions (Debye forces), induced dipole-induced dipole interactions, London dispersion forces, hydrogen bonding and entropy effects, that involves both polar and non-polar atoms.

The usual way to treat the Van der Waals interactions is to use a pairwise Lennard-Jones potential. The most common exponents for the attractive and repulsive contributions are 6 and 12, respectively. The former has physical meaning: calculating the repulsive part of the Van der Waals interaction for a simple system in a semi-classical approximation gives a potential energy contribution that vanishes as  $r^{-6}$ . While the latter is only for computational reason: giving the  $r^{-6}$  term it is efficiency to calculate  $r^{-12}$  as the square of  $r^{-6}$ . Thus it is just a common choice and other exponents can do the job, depending on the system in study. The general form for a 12 – 6 Lennard-Jones potential is the following

$$v(r) = 4\epsilon \left( \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right) = \frac{C_{12}}{r^{12}} - \frac{C_6}{r^6} \quad (1.4.3)$$

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) a plot of the function (1.4.3) with  $\epsilon = \sigma = 1$  is shown.

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

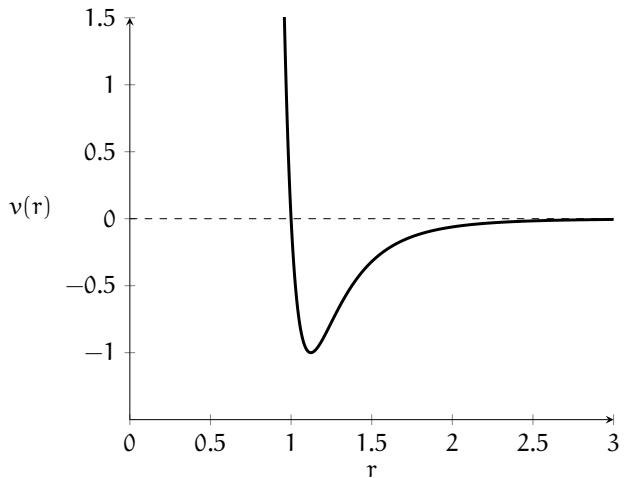


Figure 5: Plot of a Lennard–Jones interaction potential with  $\epsilon = \sigma = 1$ .

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 vanishes rapidly with distance: at  $r \sim 2\sigma$  its value is less than 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

Despite the long range characteristic of the electrostatic interaction, for 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>3</sup>. Of course this is an approximation and can lead to serious issues in those proprieties or systems that strongly depend on the electrostatic interactions. As an example we report some situations in which the use of the short–range electrostatics can lead to artifacts: they 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 calculation of the electrostatic energy contribution requires to take into account *all particles* in a system, even all the periodic images due to the PBC. This leads to a loss of computational efficiency.

If we consider only the simulation box the electrostatic energy contribution is

$$U = \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i}^N \frac{1}{4\pi\epsilon_0} \frac{q_i q_j}{r_{ij}} \quad (1.4.4)$$

<sup>3</sup> In general, for computational reason, a common choice is to consider the same cut–off for Van der Waals and electrostatic interactions.

where  $q_i$ ,  $q_j$  and  $r_{ij}$  are the charges and the distance, respectively, between particles  $i$  and  $j$ . But we need also all image boxes. Let us suppose, for simplicity, a cubic box of size  $L$ . 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 boxes, with respect to the central simulation box, is  $\vec{n} = L(n_x, n_y, n_z)$ . Then the energy contribution becomes

$$U = \frac{1}{2} \sum_{n_x, n_y, n_z}^{+\infty} \sum'_{i=1}^N \sum_{j=1}^N \frac{1}{4\pi\epsilon_0} \frac{q_i q_j}{\|\vec{r}_i - \vec{r}_j + \vec{n}\|} \quad (1.4.5)$$

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: in the inner sum it must be  $j \neq i$ .

As described above, a cut-off method is a good easy way to solve equation (1.4.5) and sometimes it produces sufficiently good results. However, the increasing of computer power can lead to develop more rigorous methods to solve equation (1.4.5), even for very large systems. The main problem is that the summation in equation (1.4.5) is *conditionally convergent*<sup>4</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  $N \gg 10^4$ ). 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 ESM itself and the particle mesh ewald (PME) method. For a more complete discussion about the advanced methods developed to treat the electrostatic interaction, for biological applications, the reader is addressed to the review of Cisneros *et al.* [5].

### Ewald summation method

The ewald summation method (ESM) is the first method introduced by Ewald for a correct treatment of the electrostatic energy contribution in an ionic crystal. The basic idea is to split the summation in equation (1.4.5) in two series both rapidly convergent. The method is based on the following identity

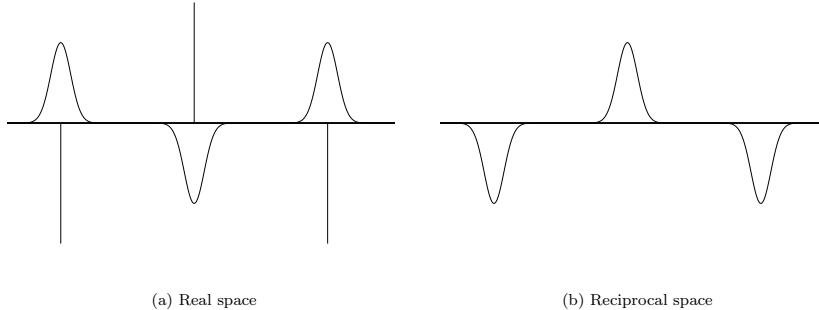
$$\frac{1}{r} = \frac{f(r)}{r} + \frac{1-f(r)}{r} \quad (1.4.6)$$

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 interaction works, as illustrated in figure (6), considering each point-like charge in the system surrounded by a neutralizing 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

---

<sup>4</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.



**Figure 6:** Schematic illustration of the ESM charge distribution: in (a) point charges (represented by vertical lines) and the neutralizing Gaussian charge distribution; in (b) the counteracting Gaussian distribution.

is a Gaussian distribution centered in the position  $r_i$  of the point-like charge  $q_i$ , of the form

$$\rho_i(r) = \frac{q_i \alpha^3}{\pi^{3/2}} e^{-\alpha^2(r-r_i)^2} \quad (1.4.7)$$

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

$$U_r = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N' \sum_{n_x, n_y, n_z} \frac{q_i q_j}{4\pi \epsilon_0} \frac{\operatorname{erfc}(\alpha \|\vec{r}_i - \vec{r}_j + \vec{n}\|)}{\|\vec{r}_i - \vec{r}_j + \vec{n}\|} \quad (1.4.8)$$

where the prime indicate that for  $\vec{n} = 0$  must be  $i \neq j$ ,  $\operatorname{erfc}(x) = 1 - \operatorname{erf}(x)$  is the complementary error function and  $\operatorname{erf}(x)$  is the error function. They are given by

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{+\infty} e^{-t^2} dt, \quad \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (1.4.9)$$

The point is that the summation involving the complementary error function in equation (1.4.8) 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 the series converges and the shorter the cut-off radius can be. Thus the ESM use the  $\operatorname{erfc}(r)$  as  $f(r)$  function in equation (1.4.6). 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 counteracting charge distribution, of equal magnitude but opposite sign. Considering the identity in equation (1.4.6) this lead to an energy contribution  $U_f$  of the form  $(1 - f(r))/r$ . But using the relation between  $\operatorname{erfc}$  and  $\operatorname{erf}$  functions, it

becomes of the form  $\text{erf}(r)/r$ . Another important trick is to compute  $U_r$  in the *real space* while  $U_f$  in the *reciprocal space*, thus considering its Fourier transform. This energy contribution is given by

$$U_f = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \sum_{k_x, k_y, k_z} \frac{1}{4\pi\epsilon_0 L^3 k^2} \frac{4\pi}{L^3 k^2} e^{-k^2/(4\alpha^2)} e^{ik \cdot (\vec{r}_i - \vec{r}_j)} \quad (1.4.10)$$

where  $\vec{k} = 2\pi\vec{n}/L$  are the reciprocal lattice vectors. Even  $U_f$  converges rapidly as  $U_r$  in equation (1.4.8); then a cut-off method can be safely used. Nevertheless, opposite to  $U_r$ , the smaller is  $\alpha$ , the shorter the cut-off can be. Clearly a proper *balance* between the real and reciprocal space summation is needed.

Since in equation (1.4.8) 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

$$U_{\text{self}} = -\frac{\alpha}{\sqrt{\pi}} \sum_{i=1}^N \frac{q_i}{4\pi\epsilon_0} \quad (1.4.11)$$

Summarizing, the energy contribution of the electrostatic interactions by the ESM is computed summing equations (1.4.8), (1.4.10) and (1.4.11) to obtain

$$\begin{aligned} U = & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \sum'_{n_x, n_y, n_z} \frac{q_i q_j}{4\pi\epsilon_0} \frac{\text{erfc}(\alpha \|\vec{r}_i - \vec{r}_j + \vec{n}\|)}{\|\vec{r}_i - \vec{r}_j + \vec{n}\|} + \\ & + \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \sum_{k_x, k_y, k_z} \frac{1}{4\pi\epsilon_0 L^3 k^2} \frac{4\pi}{L^3 k^2} e^{-k^2/(4\alpha^2)} e^{ik \cdot (\vec{r}_i - \vec{r}_j)} + \\ & - \frac{\alpha}{\sqrt{\pi}} \sum_{i=1}^N \frac{q_i}{4\pi\epsilon_0} \end{aligned} \quad (1.4.12)$$

The first line is the real space contribution while the second is the Fourier energy contribution. Since the 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  $\sim N^2$ .

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 (1.4.12) 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 (1.4.12) 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 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<sup>5</sup> 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 the 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*

The particle mesh ewald (PME) method developed by Darden *et al.* [6] is based on the ESM so the starting point is equation (1.4.12). 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. The reciprocal part instead, is computed using FFT method, in order to have a gain of performance. First, we need to consider a grid mesh onto which the Gaussian counteracting charge distribution is spread. The basic 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

$$q_M(r_p) = \int_0^L W(r_p - r)\rho(r) dr \quad (1.4.13)$$

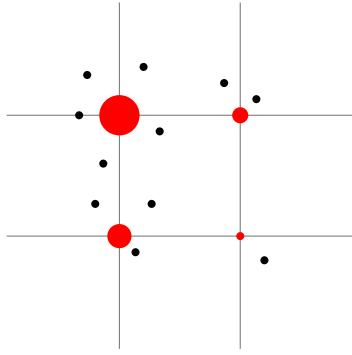
where  $L$  is the box length and, if  $h$  is the grid spacing,  $M = L/h$  is the number of mesh point. In figure (7) the charges assignment is schematically represented.

The assignment function should have the following proprieties: it should be an even function and it 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: 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  $e^{-ik_j \cdot \vec{r}_j}$  term in the Fourier transform of equation (1.4.10). In general  $\vec{r}_j$  does not correspond to a mesh grid point, so this term is not part of a

---

<sup>5</sup> Given a charge distribution  $\rho(\vec{r})$  then the associated electrostatic potential  $\phi(\vec{r})$  can be calculated solving the Poisson's equation  $\nabla^2\phi(\vec{r}) = -\frac{1}{\epsilon_0}\rho(\vec{r})$ . If a charge  $q$  is at position  $\vec{r}$  its electrostatic potential energy is given by  $U = q\phi(\vec{r})$ .



**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.

discrete Fourier transform. The idea, thus, is to interpolate it in terms of the 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  $[ \cdot ]$  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 (1.4.13), 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

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

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

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

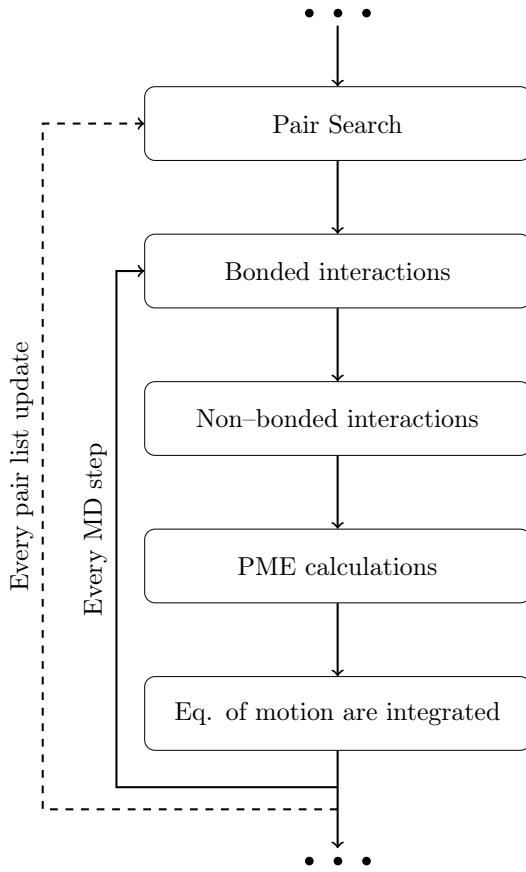
where  $\lambda$  is a complex parameter and  $z$  is a complex variable. The main properties of such SPLINE is that, it is  $n - 1$  times analytic, continuously differentiable and then can solve the instability problem of the Lagrange interpolation method. In literature the Euler SPLINE method is referred to as *smooth* PME and the reader is addressed to the article by Essmann *et al.* [7] 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 counteracting charge distribution are spread onto the mesh grid;
- Poisson's equation for the discretized charges are solved through the FFT methods;
- The  $U_f$  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. The method offers a good energy conservation and offers a good balance between accuracy and computational efficiency since it scales as  $\sim N \ln N$ . Moreover, despite we have described ESM and PME applied to the electrostatic interaction, they 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.

The general workflow shown in figure (1) can be update adding the calculation of the bonded and the non-bonded interactions and the PME contribution. In figure (8) a summary of the new workflow is shown.



**Figure 8:** Schematic workflow of the particle interaction calculations inside an MD simulation.

#### 1.4.5 Charge representation

Even if some methods, such as the PME one, have been developed to speed up the computation of electrostatic energy contribution one of the main problems of FFs for biomolecular applications remains the *charge representation*: the way in which the charges of atoms or molecules are assigned to the system particles. The problem arises from the necessity to represent the electron clouds and the interactions that generate. Nevertheless this is crucial for a better description of most electrostatic phenomena such as polarizability of molecules and polar solvent, solvation shell of charged ions, protein-ligand interactions, ion transport through polar and non-polar medium, self assembly processes and so on.

The most common solution is the *atom-centered “partial charge” approximation* in which the full charge density of the molecule is replaced by fractional point-like charges assigned to each atom of the molecule. Traditionally most non-polarizable FFs (as those we will use in this thesis work) assign to each atom of a molecule a fixed partial-charge. The most used procedure for extracting partial-charges from molecular wave functions is based on fitting atomic charges with the molecular electrostatic potential, computed with *ab initio* calculation such as *density func-*

*tional theory.* The fitting procedure consists in minimizing the deviation between the electrostatic potential produced by the assigned charges and the molecular electrostatic potential. Such representation is believed to be an important source of error in the electrostatics treatment. Moreover with fixed charge assignment it is impossible to take into account those phenomena that involve a transfer of charge inside the molecule, as polarization effect. The use of off-centered charges and/or higher order atomic multipoles can significantly improve the treatment of electrostatics but of course it is necessary a good balance between accuracy and performances since the electrostatic problem can rapidly drive to a loss of computational efficiency.

#### 1.4.6 Polarization

Polarization refers to the redistribution of the electron charge density of a molecule in presence of an external electric field, generated, for example, by charged ions or by another molecule. Polarization is responsible to non-additive attractive inter- or intra-molecular interactions which have many-body characteristics. These effects have been recognized to have an important role in many biological interactions in which different compounds are present. An increasing number of studies show that the lack of these effects can lead to serious limitations, particularly, for systems that involve different environments such us water and proteins or water and lipids. In MD simulations polarization effects are included using either *implicit* or *explicit* methods.

The implicit method completely avoids the many-body calculations by including a mean polarization effect in the functional form of the interaction potential. The general idea is to surround all the simulation box by a transparent medium with a relative 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 to determine the electrostatic potential due to system charges by the substitution  $\epsilon_0 \rightarrow \epsilon_0 \epsilon_r$ . Since it avoids many-body calculations, this method gives an incomparable gain in performances but it must be carefully used. The main disadvantage is that the mean polarization effect is added to all system particles and this wash out all the details about a possible polarization effect in a molecule. Moreover the electrostatic interaction between charged particles is affected by the mean field effect. This produce a correct electrostatic interaction for particles inside the same solvent for which the implicit medium is parameterized. Otherwise, if the simulation box is composed by different chemical environments such as a polar and non-polar compounds, using the same dielectric constant would lead to badly calculate the electrostatic interactions for particles in the non-polar environment. Thus, this method can be safely used when our system is composed principally by one kind of solvent, for example water.

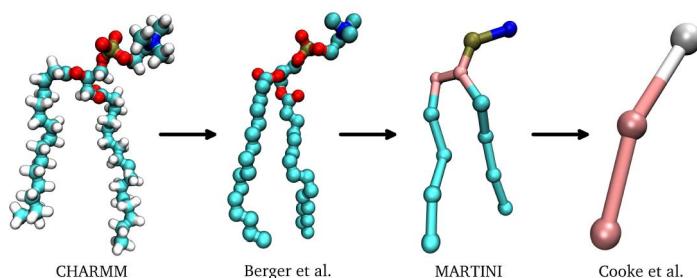
The way to correct the above behavior is to use an explicit method. As the name suggests, the polarization effect is taken into account for every molecule in the system by an its proper model included in the FF. The general idea is to add some more internal DOF to a molecule or atom to take into account 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. This can be done for every molecule or atom in the system and thus it is the optimum to better describe systems with different chemical environments. Obviously this method is more time consuming compared to the first.

#### 1.4.7 Coarse-Grained model

As we have introduced at the beginning of this section, for biomolecular applications two main classes of classical FFs exist: atomistic FFs and CG FFs. Since the atomistic model takes into account all the atoms in a molecule it is obviously the most realist and accurate FF. Nevertheless the large number of DOF of atomistic FFs leads to a loss of computational performance. Moreover, basically, the atomistic FFs are efficient until the physical proprieties can be properly sampled on a time scale of hundred of nanoseconds over a length scale of a few nanometers. As the time and length scales increase more and more time is needed to carry out a complete simulation. Unfortunately many biological processes involving lipid membranes and other organic molecules, including synthetic compounds, take place on much longer time (of the order of microseconds nor milliseconds) and length scales (of the order of tens nanometers).

One possible solution is to *integrate out* some DOF, preserving those that are relevant for the problem in exam: this procedure is called *coarse-graining*. The basic units of CG FFs are called *beads*, each representing a group of atoms or a well defined chemical moiety. 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: functional forms need to be chosen and their parameterization need to be adjusted so as to reproduce the desired target properties. In figure (9) different levels of coarse-graining are showed.



**Figure 9:** Different levels of the coarse-graining of a phospholipid. From left to right increasing levels of coarse-graining. Taken from [11].

The first step in the development of a CG model is the *mapping* procedure. This establishes a link between the atomistic model and the CG one. There is not a unique or correct procedure to perform the mapping because it depends on the desired coarse-graining level, on the time and length scales that one wants to correctly sample and on the properties one wants to reproduce. For biological ap-

plications CG FFs are often designed to reproduce specific thermodynamic properties such as surface tension, free energy of partitioning, free energy of hydration and so on. CG FFs employed in the material sciences (e.g. polymer science) often use as a target the material structural properties.

In general a CG FF is more computationally efficient than an atomistic one for the following reasons

- the DOF of the system are reduced due to the CG procedure with the consequence that a smaller number of interactions and forces has to be computed;
- bead-bead interactions, which result from the removal of finer structural details, are softer than atom-atom interactions;
- 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 which leads to faster diffusion.

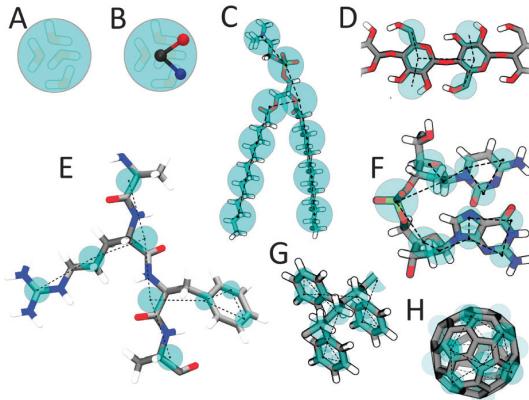
## 1.5 MARTINI: A COARSE-GRAINED FORCE-FIELD

**MARTINI** is a CG FF developed by Marrink *et al.* [19] for organic solvents and lipids and then extended to proteins [21], carbohydrates [17] and a broad class of polymers [24]. **MARTINI** is based on a chemical building block approach. Such **MARTINI** blocks or beads represent small chemical moieties and they are extensively calibrated in order to construct a large variety of organic molecules. As we shall see in section 1.5.4, the FF is based on accurately reproducing the interaction between polar and non-polar chemical compounds. The main target property is intact the *partitioning free energy* between water and a large number of organic solvents, i.e. the free energy of transfer of chemical moieties from polar and non-polar solvents.

### 1.5.1 Mapping

The mapping of the **MARTINI** beads is based on a four-to-one scheme that groups four heavy atoms like C, S, O and so on, plus their associated hydrogen atoms, into a single interaction site. Consistently four water molecules are modeled with one **MARTINI** bead. An example of the mapping procedure including both atomistic and CG descriptions is shown in figure (10).

There are four main bead types: polar (P), non-polar (N), apolar (A) and charged (Q). Each bead type has a number of subtypes to take into account a more accurate representation of the chemical nature of the moieties due to the underlying specific atomistic structure. These subtypes are distinguished by the hydrogen bonding capabilities: donor (d), acceptor (a), both donor and acceptor (da) and none (0) and/or by their degree of polarity: lowest polarity (1), ..., highest polarity (5).



**Figure 10:** 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 [20].

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

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

### 1.5.2 Interactions potential

**VAN DER WAALS INTERACTIONS** The functional form describing pairwise Van der Waals interaction is a Lennard–Jones 12 – 6 potential as in equation (1.4.3). For 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 for which  $\sigma = 0.62$  nm. This is consistent with reproducing the hydration shell when a charged bead (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 interaction strength with each MARTINI beads is shown in figure (11).

**ELECTROSTATIC INTERACTIONS** Electrostatic charges are assigned using the atom-centered approximation, as described in 1.4.5. The charges of the MARTINI beads are empirically assigned at the center of the beads and correspond to the net

		Q				P				N				C					
	sub	da	d	a	0	5	4	3	2	1	da	d	a	0	5	4	3	2	1
Q	da	O	O	O	II	O	O	O	I	I	I	I	I	IV	V	VI	VII	IX	IX
	d	O	I	O	II	O	O	O	I	I	I	III	I	IV	V	VI	VII	IX	IX
	a	O	O	I	II	O	O	O	I	I	I	I	III	IV	V	VI	VII	IX	IX
	0	II	II	II	IV	I	O	I	II	III	III	III	IV	V	VI	VII	IX	IX	
	5	O	O	O	I	O	O	O	O	O	I	I	I	I	V	VI	VI	VII	VIII
P	4	O	O	O	O	I	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	V	V	VI	VII	VII
	2	I	I	I	II	O	II	II	II	II	II	II	II	IV	IV	V	VI	VII	VII
	1	I	I	I	III	O	II	II	II	II	II	II	II	IV	IV	V	V	VI	VI
	0	IV	IV	IV	IV	IV	IV	IV	IV	IV	IV	IV	IV	IV	IV	IV	V	V	V
N	5	V	V	V	V	V	IV	IV	IV	IV	IV	IV	IV	IV	IV	IV	IV	V	V
	4	VI	VI	VI	VI	VI	V	IV	IV	IV	V	V	V	IV	IV	IV	V	V	V
	3	VII	VII	VII	VII	VI	VI	V	IV	VI	VI	VI	IV	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 11:** Interaction strength association matrix for the MARTINI bead types and subtypes. Taken from [19].

charge of the chemical moiety they represent. Water, for example, is described by a neutral P<sub>4</sub> beads. Van der Waals interactions are responsible for take into account, effectively, the effects of polarizability together with the use of an implicit medium with a dielectric constant  $\epsilon_r = 15$ . However, as we shall see in section 1.5.5, to avoid the problems with the implicit medium described in 1.4.6, especially for lipid membranes for which the dielectric constant in the hydrophobic region is much smaller, Yesylevskyy *et al.* [35] have developed a more sophisticated CG water model, called polarizable water (PW), to take into account a better water behavior.

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

$$U = \frac{1}{2}k^a(\cos(\theta) - \cos(\theta_0))^2 \quad (1.5.1)$$

whose parameters are:  $k^a = 25 \text{ kJ/mol}$  and  $\theta_0 = 180^\circ$  for aliphatic chains;  $k^a = 45 \text{ kJ/mol}$  and  $\theta_0 = 120^\circ$  for cis double bonds and  $k^a = 45 \text{ kJ/mol}$  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 described by 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

The MARTINI FF was originally developed using a shifted cut-off scheme for both Lennard-Jones and electrostatic potentials with a cut-off radius  $r_c = 1.2$  nm. The Lennard-Jones potential was shifted from  $r_s = 0.9$  nm to  $r_c$  while from  $r_s = 0.0$  nm to  $r_c$  for the electrostatic potential. The neighbor list is constructed as 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.* [20] 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 better performances). Moreover, the treatment of the electrostatic interaction can be safely updated to the PME method together with the Verlet cut-off scheme. This new set-up was largely tested by Yesylevskyy *et al.* [35]. 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 a fourth-order. Moreover, with the use of PW, as we shall see, the dielectric constant should be reduced to  $\epsilon_r = 2.5$ . In all cases a time step  $20 \leq \delta t \leq 40$  is suitable for a great number of applications. 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 against 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 CG beads between bulk water in equilibrium with its vapor. Similarly the free energy of vaporization was obtained considering a simulation box with the selected CG bead in a liquid phase in equilibrium with its vapor. From the equilibrium densities of the particles in both phases the related free energies can be computed from

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

All the simulations were performed in a canonical NVT ensemble.

The partitioning free energy between water and an organic solvent was obtained in a NPT ensemble, considering a simulation box half filled of water and half of the organic solvent. Then a small fraction of the CG particle for which the partitioning free energy is to be computed, was 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

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

where S indicate the organic solvent.

In figure (12) a summary of the results is reported. As one can see the model has bad performances for what concerns the free energies of vaporization and hydration, which are too high with respect to experimental data. Instead, the partitioning free energies match very well. Thus the model is not very accurate for vapor–liquid systems but it is reliable at describing the behavior of liquid phases with different degrees of hydrophobicity.

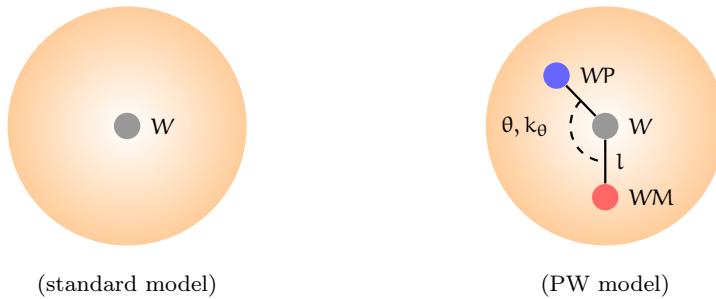
type	building block	examples	$\Delta G^{\text{vap}}$		$\Delta G^{\text{hydr}}$		$\Delta G_{HW}^{\text{part}}$		$\Delta G_{CW}^{\text{part}}$		$\Delta G_{EW}^{\text{part}}$		$\Delta G_{OW}^{\text{part}}$	
			exp	CG	exp	CG	exp	CG	exp	CG	exp	CG	exp	CG
Q <sub>da</sub>	H <sub>3</sub> N <sup>+</sup> —C <sub>2</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>3-</sup>	phosphate			-25		< -30		-18		-13		-18	
	CL—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	-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 12:** Results summary: free energies of vaporization  $\Delta G^{\text{vap}}$ , hydration  $\Delta G^{\text{hydr}}$  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 in 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 [19].

### 1.5.5 Polarizable Water model

Water plays a crucial role in any biomolecular system. It is important thus to correctly describe its behavior. Since the MARTINI water model does not directly take into account the electrostatic interaction between water and water and other

environment because it does not have any charge and it interact only via Van der Waals interaction, a simple implicit medium is used to take into account the main electrostatic effects of water, screening and polarizability. In reality, however, any biomolecular process involves charged species moving between regions of different dielectric constant. Due to the change in electrostatic screening between those environments, the strength of the interaction between the moving charges and the surrounding molecules also changes. This effect can not be taken into account by implicit medium models. In order to capture the inhomogeneous nature of the dielectric response an explicit medium has to be used.



**Figure 13:** Schematic representation of the PW bead. Shaded orange spheres correspond to the Van der Walls radii of the central neutral particle  $W$ . The blue particle is the positively charged while the red is the negatively charged.

In the same fashion of the **MARTINI** philosophy, Yesylevskyy *et al.* [35] have developed a polarizable water (PW) model that better describes the real behavior of water. As before one PW bead is associated to four water molecules. The new water bead consists of three particles instead of one in the standard **MARTINI** model. In figure (13) the topology of the PW and a comparison with the old model is shown. The central particle  $W$  is neutral and interacts with other particles in the system with the only Lennard-Jones potential. 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  $e = 1.60217653(14) \cdot 10^{-19} \text{ C}$  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  $l = 0.14 \text{ nm}$ . The electrostatic interaction between  $WP$  and  $WM$  inside the same bead are exclude, thus they are invisible to 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  $k_\theta = 4.2 \text{ kJ}/(\text{mol rad}^2)$  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

**Table 2:** Summary of the parameters used in the PW model. See [35] for details.

$ q  = 0.46e$
$l = 0.14 \text{ nm}$
$\theta_0 = 0$
$k_\theta = 4.2 \text{ kJ}/(\text{mol rad}^2)$
$\epsilon_{WW} = 4 \text{ kJ/mol}$
$\sigma = 0.47 \text{ nm}$

`MARTINI`, to  $\varepsilon_r = 2.5$ . Moreover, since the PW beads attract each other stronger than the standard water beads ( $P_4$  type), because of the additional electrostatic interactions, the strength  $\epsilon_{WW}$  of the Lennard–Jones interaction between  $W$  particles must be reduced. Water properties, especially the hydration free energy compared to the standard water model, and the lipid membrane behavior are reproduced satisfactory if the Lennard–Jones strength is changed from a I level (see table (1)) to a 95% of it. Since even in this case the hydration free energy was too high, the same apply to the cross interaction terms between  $W$  particle and the other `MARTINI` beads. Instead,  $\sigma$  remains unchanged to 0.47 nm. In addition to this, in order to recover the correct partitioning behavior, the self terms between  $Q$  type beads are generally decreased; while the cross terms are generally increased. In figure (14) a summary of the new interaction terms between  $Q$  type beads and the other beads is shown.

POL	$Q_{da}$	$Q_d$	$Q_a$	$Q_o$	$P_5$	$P_4$	$P_3$	$P_2$	$P_1$	$N_{da}$	$N_d$	$N_a$	$N_o$	$C_5$	$C_4$	$C_3$	$C_2$	$C_1$
$Q_{da}$	<b>O</b>	<b>O</b>	I	I	<b>IV</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>III</b>	<b>IV</b>	<b>V</b>	<b>VI</b>	<b>VII</b>
	O	O	O	O	II	O	O	O	I	I	I	I	I	IV	V	VI	VII	IX
$Q_d$	I	I	<b>IV</b>	<b>III</b>	<b>VII</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>II</b>	<b>O</b>	<b>III</b>	<b>IV</b>	<b>V</b>	<b>VI</b>	<b>VII</b>	<b>VII</b>
	O	O	I	O	II	O	O	O	I	I	I	III	I	IV	V	VI	VII	IX
$Q_a$	I	I	<b>III</b>	<b>IV</b>	<b>VII</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>II</b>	<b>III</b>	<b>IV</b>	<b>V</b>	<b>VI</b>	<b>VII</b>	<b>VII</b>
	O	O	O	I	II	O	O	O	I	I	I	III	IV	V	VI	VII	IX	IX
$Q_o$	<b>II</b>	<b>IV</b>	<b>VII</b>	<b>VII</b>	<b>IV</b>	<b>O</b>	<b>O</b>	<b>O</b>	I	<b>II</b>	<b>II</b>	<b>II</b>	<b>III</b>	<b>IV</b>	<b>V</b>	<b>VI</b>	<b>VII</b>	<b>VII</b>
	O	II	II	II	IV	I	O	I	II	III	III	III	III	IV	V	VI	VII	IX

**Figure 14:** New interaction strength between  $Q$  type beads and the other beads. New values in bold font, old values in normal font. See table (1) for the interaction levels. POL is the name of the `MARTINI` bead type associated to particle  $W$  inside a PW bead. Taken from [35].

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  $\epsilon_W$ , density  $\rho$  and dipole momentum of a pure water phase. The above parameters of the PW model are summarized in table (2) while a comparison of the results obtained with the PW, the standard `MARTINI` water and the experimental data are summarized in table (3). For more details about the parameterizations and testing methods the reader is addressed to the article by Yesylevskyy *et al.*[35].

Moreover, the authors found that, in addition to the PW model, the use of PME method contributes to a more realistic description of the processes involved in biomolecular environments. In particular, some interesting results of utility for

	PW	MARTINI water	Experimental
$\rho$ [kg/m <sup>3</sup> ]	1043	1013	997
$\bar{p}$ [D]	4.9	—	4.4 <sup>a</sup>
$\epsilon_W$	75.6	—	78.4
T <sub>melt</sub> [K]	282	290	273
$\Delta G^{\text{hyd}}$ [kJ/mol]	-18.7	-18	-27
D <sub>WW</sub> [10 <sup>-5</sup> cm <sup>2</sup> /s]	2.5	2.0	2.3

**Table 3:** Summary of the results obtained with the PW, standard MARTINI water and the experimental data at T = 300 K.  $\bar{p}$  is the average dipole momentum of a pure water box and D<sub>WW</sub> is the self-diffusion coefficient. <sup>a</sup> In order for a comparison to be made, this data is obtained from an atomistic simulation of 1600 SPC/E water molecules and the average dipole momentum  $\bar{p}$  is obtained averaging over a groups of four molecules, randomly chosen; see [35] for more details.

this thesis work concern a better description of the properties of lipid membranes, as we shall see in Chapter 3.

### 1.5.6 Limitations of MARTINI FF

As we have seen in 1.4.7 the CG FFs are computationally advantageous, still a price has to be paid. Although the MARTINI FF is still a fine 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. An important issue is the underestimation of the entropy which respect to the atomistic case, that is a consequence of the DOF reduction process. Since in the MARTINI FF the partitioning free energy  $\Delta G = \Delta H - T\Delta S$  must be consistent with the experimental data, the intrinsic loss of entropy imply a reduction of the enthalpy contribution, generating an imbalance between them. If a NVT ensemble is used the correct potential is the Helmholtz free energy  $\Delta A = \Delta U - T\Delta S$ , thus the imbalance is between the internal energy and the entropy. This means that the use of a CG model with a NVE ensemble leads to a problem of energy conservation. Anyway this is not our case since we only use a NVT or a NPT ensemble with an external thermostat coupling.

Another consequence of the use of a CG FFs is related to the free energy surface (FES) 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 dynamics of the system. Moreover even the PEF becomes smoother, in particular the bonded contributions, 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 four, based on lateral diffusion coefficients of lipids in membranes, is commonly used, of course with some care. Another smaller source of errors is due to the choice of

masses: since ensemble properties are not affected by particle masses, in order to increase the efficiency, all the MARTINI beads have the same mass of 72 amu. This leads in some uncertainty in the dynamics of the system making the time scaling for different beads non-trivial.

A problem involving the Lennard–Jones potential as a model of Van der Waals interactions in MARTINI , is that the steep repulsion leads to an over-structuring of fluids compared to atomistic models. As we can see from table (3) the direct and most evident implication is the melting point of the standard MARTINI 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. Using the PW model these properties improve slightly, see table (3). For a more comprehensive discussion about the limitations of the MARTINI FF the reader is addressed to the review by Marrink *et al.* [20].

## 1.6 ADVANCED SAMPLING METHODS

The free energy is a quantity of particular importance for the equilibrium statistical mechanics. The Gibbs free energy G is specific for the NPT isobaric–isothermal ensemble and the Helmholtz free energy A for the NVE canonical ensemble. The time evolution of a biomolecular system and its equilibrium properties are determined by the system’s FES. This because free energy differences tell us, for example, if a chemical reaction occurs spontaneously, whether a given solute is hydrophobic or hydrophilic, if a protein conformational change can take place or whether some molecules in water solution are able to self–assemble into a more complex system and so forth. Being related to the partition function of an ensemble, the free energy is the generator through which other thermodynamic quantities are obtained via differentiation. However often we are particularly interested in the free energy difference between two thermodynamic states, instead of the absolute value.

**COLLECTIVE VARIABLES** Often, we are interested in the FES in function of some generalized coordinates, called collective variables (CVs) of the system. These small set of variables can describe, in a simple and useful manner, some chemical, thermodynamic or mechanical processes that take place in the system. For example, the free energy in function of the distance between the COM of two molecules give us information about their attraction or repulsion and if they form a bound state. Thus the FES, in the CVs space, provides a map of the stable conformations, the relative stability of these conformations and the barrier heights that must be crossed for the transition to take place. Moreover, the energy landscape, even for a small molecule, can be extremely wrinkled and the large number of minima can

not be sampled in a typical MD simulation. The principal reason for the use of the CVs, is thus to limit the sampling to those regions of phase space that are most important for the process under study, hoping that the limited sampled regions are sufficient to correctly describe the process.

**FREE ENERGY SURFACE** The calculation of free energy difference between two thermodynamic states (that requires an *a priori* knowledge of the two stable states) and the calculation of the FES are one of the main challenges in MD simulations for biomolecular applications. Let us suppose that we are interested in the FES in function of the CV  $s(\vec{r})$  and we are working in a NPT ensemble<sup>6</sup> with an isotropic system. Following section 1.2, the Gibbs free energy along the CV is obtained as

$$G(s) = -k_B T \ln Q(s) \quad (1.6.1)$$

where  $Q(s)$  is the partition function that integrate out all the DOF of the system expect for  $s(\vec{r})$ :

$$Q(s) = \frac{1}{Z_{NpT}} \int_0^{+\infty} dV \int_{\Omega} e^{-\beta(\mathcal{H}(\vec{x}) + pV)} \delta(s(\vec{r}) - s) dx$$

since  $s(\vec{r})$  does not depend on particle momenta, from equations (1.1.10) and (1.2.6),  $Q(s)$  can be rewritten as

$$Q(s) = \frac{\int_{\Omega} e^{-\beta U(\vec{r})} \delta(s(\vec{r}) - s) dr}{\int_{\Omega} e^{-\beta U(\vec{r})} dr} \quad (1.6.2)$$

where  $U(\vec{r})$  is the PEF.  $Q(s)ds$  can be interpreted as the probability to find the system with  $s(\vec{x})$  between  $s$  and  $s + ds$ . Since this equation contains a direct phase space integration it can be rewritten in a more useful manner using the ensemble averages and then, using the ergodic theorem in equation (1.2.1), as a time average:

$$Q(s) = \langle \delta(s(\vec{r}) - s) \rangle = \lim_{t \rightarrow +\infty} \frac{1}{\tau} \int_0^{\tau} \delta(s(\vec{r}(t)) - s) dt \quad (1.6.3)$$

The time sampling of equation (1.6.3) can be derived, in principle, via MD. Unfortunately, since the time can not be infinite, the main problem related to MD simulations is whether we are able to correctly sample *all* the phase space of an ensemble in order to compute the ensemble average. Clearly the answer depends on the system in exam, if it is really simple maybe we can do that, otherwise probably not or indeed it take too much time and/or we are not able to collect sufficient data. This sampling problem can be summarized as follows: regions in phase space around a local minimum of the FES are typically sampled well, whereas regions of higher energy are sampled rarely. These high energy regions provide with a small contribution to the partition function, due to their unfavorable Boltzmann factor. However they must be overcome in order to sample other

---

<sup>6</sup> The following is still valid even for a NVE ensemble with the substitution  $G \rightarrow A$  and the use of the correct partition function  $Z_{NVE}$ .

minima that can give, instead, a important contribution to the ensemble averages. These transitions are called *rare events*. When the system is moving in the energy landscape the only way to escape from a local minimum is to exploit thermal fluctuations and so energy barriers that are higher then  $\sim k_B T$  have a small probability of being overcome.

**BIAS BASED ADVANCED SAMPLING** Several methods are developed and are still in development in order to solve the just described sampling problem. These methods are all based on advanced sampling techniques that allow us to

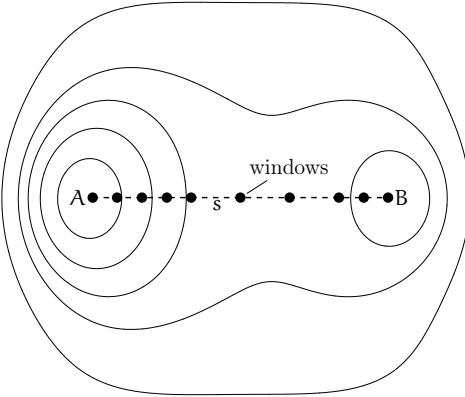
- Escape from a local energy minimum, in order to explore other regions of the phase space;
- Calculate the free energy difference between two thermodynamic states;
- Compute the FES along one or a small set of CVs.

The common basic idea is to introduce a additional *bias potential* able to confine the sampling to a limited region of the CV space and/or drive the transition between two metastable configuration. In particular we describe only those methods used in this thesis work: *umbrella sampling method* and *metadynamics*. For a more comprehensive discussion the reader is addressed to the review by Kästner [12] for the former, and to the review by Laio and Gervasio [13] for the latter. Moreover in the books by Tuckerman [29] other enhanced sampling methods are described.

### 1.6.1 Umbrella sampling

The umbrella sampling method was developed by Torrie and Valleau and today is one of the most mature and broadly accepted method for calculating free energy differences. The basic idea is to drive the system from a known state A to an other known state B through a deterministic path defined on the CV  $s(\vec{r})$  chosen to describe the process. This method is well suited for one CV, otherwise the computational performance degrades rapidly with the number of CVs. The idea is to identify a path connecting A to B, and divide it into a discrete number of windows  $N_w$  and take a subset  $\{s_i\}$ ,  $i = 1, \dots, N_w$  of the values assumed by the continuos CV from state A to B along the path. For each of these target values  $s_i$  a bias potential  $w_i(s)$ , depending only on  $s$ , is added to PEF in order to restrain the system to the  $s_i$  target value. Then, MD simulations are performed for each window. All the data collected within the  $N_w$  windows are combined together to compute the biased FES along the chosen CV. Eventually, the unbiased FES is recovered. In figure (15) an example of the windows selection is shown.

Suppose  $w_i(s)$  to be the bias potential applied on window  $i$  in function of the CV  $s(\vec{r})$ . Then the biased PEF of window  $i$  become  $U_i^b(\vec{r}) = U(\vec{r}) + w_i(s)$ ; where the superscript b denotes biased quantities. With the substitution  $U(\vec{r}) \rightarrow U_i^b(\vec{r})$



**Figure 15:** Example of an umbrella sampling windows selection. The contour plot represent a 2-minima free energy profile through which the system is evolved. The dashed line is the path along the CV  $s(\vec{r})$  that connect state A to B. The black filled circles represent the selected windows for different values of  $s$ .

in equation (1.6.2) the corresponding biased partition function integrated over all DOF but  $s$  yields to

$$Q_i^b(s) = e^{\beta w_i(s)} \frac{\int_{\Omega} e^{-\beta U(\vec{r})} \delta(s(\vec{r}) - s) d\vec{r}}{\int_{\Omega} e^{-\beta(U(\vec{r}) + w_i(s(\vec{r})))} d\vec{r}}$$

In order to use equation (1.6.1) to obtain the FES, we need to recover the unbiased partition fraction  $Q_i(s)$  like in equation (1.6.2). This can be done (see [12]), obtaining the following expression

$$\begin{aligned} Q_i(s) &= Q_i^b(s) e^{\beta w_i(s)} \frac{\int_{\Omega} e^{-\beta U(\vec{r})} e^{-\beta w_i(s(\vec{r}))} d\vec{r}}{\int_{\Omega} e^{-\beta U(\vec{r})} d\vec{r}} = \\ &= Q_i^b(s) e^{\beta w_i(s)} \langle e^{-\beta w_i(s)} \rangle \end{aligned} \quad (1.6.4)$$

then the FES for window  $i$  is obtained simply by equation (1.6.1)

$$G_i(s) = -k_B T \ln Q_i^b(s) - w_i(s) + C_i$$

where  $Q_i^b(s)$  is obtained as an ensemble average via the biased MD simulation like equation (1.6.3) and  $w_i(s)$  is a known function.  $C_i$  is an additive constant independent of  $s$  that connects the free energy curves  $G_i(s)$  of different windows. As we shall see, in order to combine more windows to calculate a global FES, all  $\{C_i\}$  must be computed.

**WEIGHTED HISTOGRAM ANALYSIS METHOD** Different analysis methods have been developed in order combine informations collected from the  $N_w$  windows. The aim of those methods is to compute the global unbiased partition function  $Q(s)$  from the set of biased partition functions  $\{Q_i^n(s)\}$  in order to extract the global FES  $G(s)$ . The most commonly used method is the weighted histogram analysis method (WHAM) [25], [10]. The philosophy behind it is to minimize the statistical error in the calculation of  $Q(s)$ . For each of the  $N_w$  biased windows a set of his-

tograms  $\{h_i(s)\}$  with  $n_i$  bins, representing the set  $\{Q_i^b(s)\}$ , is recorded. The global distribution function is computed as follow

$$Q(s) = \sum_{i=0}^{N_w} p_i h_i(s), \quad \sum_{i=0}^{N_w} p_i = 1 \quad (1.6.5)$$

where  $p_i$  are weights chosen to minimize statistical error on  $Q(s)$ . Those leads to

$$p_i = \frac{n_i e^{-\beta(w_i(s) - C_i)}}{\sum_{j=0}^{N_w} n_j e^{-\beta(w_j(s) - C_j)}} \quad (1.6.6)$$

where  $n_i$  is the total number of independent bins in the  $i$ -th histogram. The problem now is to compute the set of constants  $\{C_i\}$ . We can not use the integrals in equation (1.6.4) for the problems already described at the beginning of this section, but the second line give us an idea. The ensemble average can be computed using the global distribution function  $Q(s)$  as follow

$$e^{-\beta C_i} = \langle e^{-\beta w_i(s)} \rangle = \int Q(s) e^{-\beta w_i(s)} ds \quad (1.6.7)$$

Because  $Q(s)$  in equation (1.6.5) depends on the set of constants  $\{C_i\}$  and *vice versa*, both equations must be solved in a iterative self-consistent manner. A first guess of set  $\{C_i\}$  is used to compute the weights from equation (1.6.6) then from equation (1.6.5)  $Q(s)$  is computed and it is used to obtain a new set of constants  $\{C_i\}$  from equation (1.6.7) and so on until both equations are satisfied. When the iteration procedure is completed the global FES  $G(s)$  is obtained from equation (1.6.1). One important consideration about WHAM is that the histograms in adjacent windows must be sufficiently overlapped otherwise the statistical error due to the combining procedure can be too high or the iterative procedure itself can lead to convergence problems.

**BIAS POTENTIAL** The bias potential is chosen such that sampling along the CV is uniform. Together with WHAM a simple harmonic bias potential is most commonly used for its simplicity. In order to restrain the system to the target value  $s_i$  of the CV  $s(\vec{r})$  along the path chosen to connect state A and B, each window is biased with a harmonic potential of the form

$$w_i(s) = \frac{1}{2} K(s - s_i)^2$$

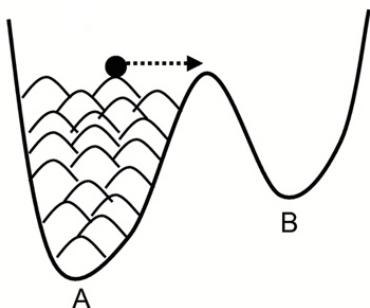
The choice of  $K$ , the strength of the bias potential, is a critical point.  $K$  has to be large enough to appropriately sample the corresponding modes. However too large  $K$  cause too narrow distributions that can cause overlap problems, hence the necessity for  $K$  to be as small as possible to allow for some overlap between windows.

The implementation of the umbrella sampling method can be summarized in the following procedure

- A CV that well describes the transition process from state A to B and a connecting path are chosen;
- A subset of the values assumed by the CV along the path are taken and for each a biased MD simulation is performed<sup>7</sup>;
- Using WHAM the set of biased partition functions  $\{Q_i^b(s)\}$  are combined in order to compute the global unbiased partition function  $Q(s)$ . Then  $G(s)$  is calculated.

### 1.6.2 Metadynamics

The metadynamics method was originally developed by Parrinello and Laio [14] with the first aim to accelerate the escaping from a free energy minimum in MD simulations. The success of the method soon led to the creation of a unified framework for accelerating rare events and computing free energies in function of a small set of CVs. The main advantages with respect to umbrella sampling method is that several CVs, instead of a single one, can be simultaneously used without affecting the simulation performance. The basic idea of metadynamics is to enhance the dynamics of a system along some CVs simply by filling the corresponding energy minimum with an history-dependent bias potential, in order to sample larger and larger portions of the phase space. Supposing a two state process from state A to B, in the CV space, if the deposited energy is sufficient to fill the energy well of state A, the system is favorably disposed to overcome the barrier and go to the state B. In figure (16) a simple scheme of that cartoon is shown.



**Figure 16:** Schematic cartoon of the energy well filling by the history-dependent bias potential and the consequently transition from state A to state B.

This novel idea is supported by the assumption of Parrinello and Laio, based on experimental and heuristic results, that iteratively summing the deposited potential during the biased MD simulation leads to an estimator of the FES along the chosen CVs in the region explored. If  $\vec{s}(\vec{r}) = (s_1(\vec{r}), \dots, s_n(\vec{r}))$  is the CVs vector, where  $n$  is a small number, and  $w(\vec{s}, t)$  is the bias potential deposited every  $\tau$  MD

---

<sup>7</sup> Since each biased simulation is independent they can be performed in parallel with the other.

time steps the “metadynamics” history-dependent potential acting on the system at a time  $t$  is given by

$$w_M(\vec{s}, t) = \sum_{\substack{i=0 \\ i\tau < t}} w(\vec{s}, i\tau)$$

where  $t$  is the time in unit of MD time step. The time dependence in the bias potential  $w(\vec{s}, t)$  is needed since it has to depend on the values assumed by the CVs at some previous MD steps. The Parrinello and Laio assumption yields to the following expression

$$\lim_{t \rightarrow +\infty} w_M(\vec{s}, t) \simeq -G(\vec{s}) + C \quad (1.6.8)$$

where  $C$  is an additive constant. Since the history-dependent potential iteratively compensates the underlying FES, the system evolved with metadynamics *tends to escape from any energy minima via the lowest saddle point*. Thus, to the contrary of umbrella sampling metadynamics is suitable, not only to compute efficiently FES, but also to explore new reaction path and accelerate the observation of rare events. If the CVs are chosen sensibly the system will quickly find its way over the lowest free energy saddle point and evolve over the next minimum as it would do in a *very long* MD simulation.

A possible bias potential can be deduced considering equation (1.6.3). The Dirac delta function can be expressed by its approximant:

$$\delta(x - a) = \lim_{\sigma \rightarrow 0} \frac{1}{\sqrt{2\pi}\sigma^2} e^{-(x-a)^2/(2\sigma^2)}$$

then substituting in equation (1.6.3) we have

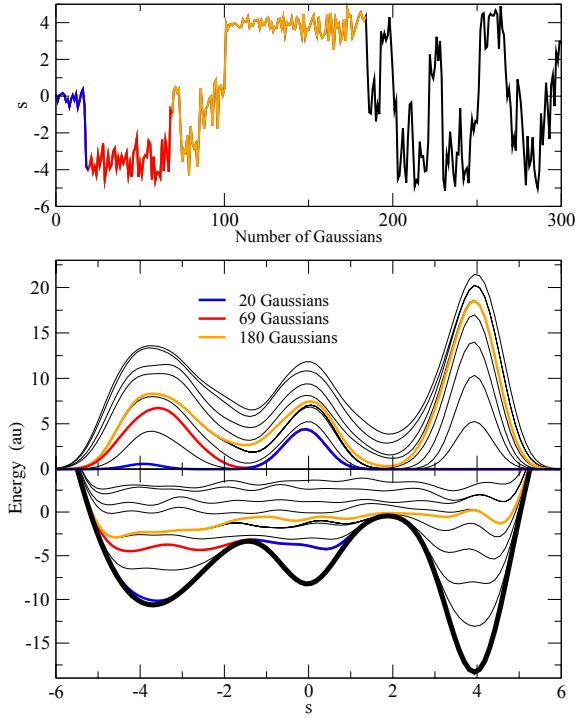
$$Q(s) = \frac{1}{\sqrt{2\pi}} \lim_{t \rightarrow +\infty} \lim_{\sigma \rightarrow 0} \frac{1}{\sigma t} \int_0^t \exp\left(-\frac{(s(\vec{r}(\tau)) - s)^2}{2\sigma^2}\right) d\tau$$

the equation above suggests the use of a Gaussian function centered around the values assumed by the CV at a time  $t$ . Then the bias potential component, extended to multiple CVs, is chosen as follow

$$w(\vec{s}, t) = w \exp\left(-\sum_{i=0}^n \frac{(s_i - s_i(\vec{r}(t)))^2}{2\delta s_i^2}\right)$$

where  $w$  is the height and  $\delta s_i$  is the width of the deposited Gaussians.

For setting up a metadynamics simulation, there are three parameters to choose carefully: the height  $w$  and the width  $\delta s_i$  of the deposited Gaussians and the stride of deposition  $\tau$ . All these parameters affect the accuracy and the efficiency of the free energy profile reconstruction. Clearly if the Gaussians are big and large or placed too quickly the FES will be explored at a fast pace but the reconstructed profile will be affected by large errors. Instead if they are small or placed infrequently the reconstruction will be accurate but will take a longer time. Moreover the time required to escape from a local minimum is determined by the number of Gaussians necessary to fill the well. This number is proportional to  $(1/\delta s)^n$  [13], hence



**Figure 17:** Example of a time evolution of a metadynamics simulation. Top: time evolution of the CV of a system evolved on the 3-minima FES represented by the black thick line in the lower panel. Middle: time evolution of  $w_M$ , the history-dependent bias potential. Blue line:  $w_M$  when the first minimum is filled and the system escapes to the second minimum; red line:  $w_M$  when also the second minimum is filled; yellow line: when the entire profile is filled and the dynamics becomes diffusive on the whole energy landscape. Lower panel: time evolution of the sum of  $w_M$  (same color scheme). Taken from [13].

the necessity to maintain the number  $n$  of CVs as small as possible or increase the Gaussian width. On the other hand the history-dependent potential can only reproduce features of the FES on a scale larger than  $\sim \delta s$ . Empirical criteria can be used to choose the  $\delta s$  and  $\tau$  parameters; the former by monitoring the standard deviation of the CVs in an unbiased MD simulation and the latter by considering the relaxation time of the system after a Gaussian is deposited: clearly the bigger is the Gaussian the more time is needed by the system to relax. However it does not exist an universal and general recipe to choose these parameters, only some knowledge of the system and process under study can give useful hints. They can certainly be fine tuned in successive steps.

In figure (17) an example of the time evolution of a metadynamics simulation is shown. We can see that as the number of deposited Gaussians increases the system is able to visit more regions of the phase space. Moreover, as the simulation time increases, when the energy landscape (lower panel) becomes flat the system becomes diffusive (upper panel) in the CVs space. Thus we can stop the metadynamics simulation obtaining the unbiased FES as in equation (1.6.8).

Keeping in mind figure (17) a summary of the behavior of metadynamics and the validity of equation (1.6.8) can be qualitatively understood in the limit of slow deposition. This means that  $w_M(\vec{s}, t)$  varies slowly and the probability to observe  $\vec{s}$  is approximately proportional to the Boltzmann factor  $e^{-\beta(G(\vec{s}) - w_M(\vec{s}, t))}$ . If the function in the exponential has some local minimum due to  $G(\vec{s})$ ,  $\vec{s}$  will preferentially be localized in the neighborhood of this minimum and an increasing number of Gaussians have to be added until it is completely filled. When the minimum is filled the system reaches the condition  $G(\vec{s}) \sim w_M(\vec{s}, t)$ , the probability dis-

tribution would be approximately flat and we can say that the system becomes diffusive in the region explored by the simulation. Then, in this region, the placement of new Gaussians is no more affected by the difference between  $G(\vec{s})$  and  $w_M(\vec{s}, t)$  and they are deposited randomly over the flat free energy profile. The FESs reconstructed after that point, as visible in figure (17), are affected by some corrugations, due to newly deposited Gaussians, of the order of  $w$ . From that point the metadynamics has reached the *convergence* in the sampled region.

**CONVERGENCE** As just explained before, the convergence of the metadynamics in a specific region of the CVs space is reached when the system become diffusive in this region, i.e. when the CVs can assume all possible values compatible to the sampled region. This is a crucial point for the metadynamics method in order to obtain the best possible estimator of the FES, in the sampled region. Clearly, in order to save computational time, if one want only to give the possibility to the system to escape from an energy minimum is not necessary to reach the convergence otherwise it is an important point. Unfortunately in most systems it is not trivial to identify precisely when the diffusive regime has been reached. In most cases a practical method to assess the convergence of metadynamics simulations is based on monitoring the free energy difference between two reference point: when the difference is approximately flat then the convergence should be reached.

**ERROR ESTIMATION** After convergence is reached the error on the FES estimator is clearly dependent on the chosen parameters. The best way to estimate the statistical error introduced by the metadynamics is to perform several statistically independent metadynamics runs. The arithmetic average of all the history-dependent potentials taken at the same time, after the convergence is reached, is equal to the best estimate of the FES. The statistical error of the FES of a run can be considered as the standard deviation between the history-dependent potential  $w_M(s, t)$  of that run and the averaged FES  $G(\vec{s}) \sim -\overline{w_M(\vec{s}, t)}$  and average it over the whole CVs space, as follow

$$\bar{\epsilon}^2 = \frac{1}{\Omega_s} \int_{\Omega_s} (\overline{w_M(\vec{s}, t)} - \overline{w_M(\vec{s}, t)})^2 ds$$

where  $\Omega_s$  is the whole CVs space. Laio *et al.* in [15], by performing extensive numerical simulations of a Langevin stochastic system, have derived an approximate expression for the error estimator in function of the system and metadynamics parameters, as follow

$$\bar{\epsilon}^2 \propto \frac{LT}{D} \frac{w}{\tau} \delta s$$

where  $L$  is the size of the simulation box,  $D$  is the system diffusion coefficient and  $T$  is the system temperature. Since  $\delta s$  is approximately fixed by the fluctuation of the CVs in a unbiased MD run and/or by the granularity to be achieved in the FES estimator, the error is dominated by the ratio  $w/\tau$ . Thus a fine tuning of the Gaussians height and deposition pace is often needed in order to minimize the

statistical error. Despite this, it is commonly accepted to follow a procedure for the FES and error estimators:

- Several statistically independent metadynamics simulations are performed;
- After the convergence is reached for each run, the estimator of the FES is the arithmetic average of the FES obtained from equation (1.6.1) for each simulations (each free energy profile appropriately normalized);
- The statistical error on the averaged FES is performed considering its standard deviation.

Alternatively one can perform a very long metadynamics run in which, for example, the system is diffusive for half simulation. Then one can chose as statistically independent history-dependent potentials some of them computed at different times in the diffusive region and follow the above procedure from the second point. Clearly one has to be careful about the decorrelation time between the chosen profiles, otherwise they are not statistically independent due the continuos nature of the metadynamics algorithm.

**PERFORMANCE OPTIMIZATION** In general the computational overhead of adding metadynamics to an MD simulation is usually not excessive, even if it depends on the implemented algorithms. However as the number of deposited Gaussians increases, in each MD step a larger and larger number of exponential terms have to be computed and summed in order to calculate the derivative of the history-dependent potential, i.e. the forces due to the metadynamics. Thus if the Gaussians are big or frequently deposited or the system takes a lot of time to reach the convergence, the computational overhead scales as the number of the deposited Gaussians. This can clearly lead to a loss of performance as the simulation time increase. A simple solution is to implement a discrete mesh grid on the CVs space in which the history-dependent potential is spread and stored. When a new Gaussian is added the potential is updated in the whole grid. While, at each MD step, in order to compute the derivative, the potential in a non-grid point is only estimated from an interpolation of some neighbor grid points, depending on the interpolation order. By this trick the computational overhead remains approximately constant as the simulation time increase.

### 1.6.3 Umbrella sampling and metadynamics remarks

The use of metadynamics is growing up in the computational community. The most relevant reasons are its simplicity of implementation and the direct way to control the efficiency and performance by changing the parameters of the Gaussians entering in the history-dependent potential. This gives to the metadynamics the possibility, with only one framework, to overcome different situations such as passing continuously from a fast and coarse exploration of the energy landscape to an accurate evaluation of the free energy profile, predict new stable configu-

rations and structures, new reaction pathways, calculate free energy profiles and free energy differences.

The main difference of metadynamics respect to umbrella sampling is the process of reconstruction of the free energy profile. In the latter it follows a predefined scheme designed for covering only the CV space along the chosen path. In contrast a well implemented metadynamics reconstruct efficiently the free energy profile: starting from the current minimum it explores a larger and larger region of the CVs space in order to recursively reconstruct the FES. However a careful choice of the CVs is needed, otherwise the history-dependent potential and the FES can evolve in an unphysical manner.

In a recent work by Davide Bochicchio *et al.* [3] both methods for the FES estimation included the error estimation, performance and accuracy are extensively compared with MD simulations involving the transfer of hydrophobic oligomers from the water phase to the hydrophobic core of a lipid membrane. The authors consider the system both at atomistic and CG levels (MARTINI FF). They found that, if the CVs are properly chosen and the parameters for the metadynamics are carefully tuned, the reconstructed free energy profiles are identical between umbrella sampling and metadynamics, but the latter yields the same accuracy in less simulated time.

# 2

## MD TOOLS, PARALLELIZATION AND HARDWARE ACCELERATION

Different tools for setting up an MD simulation, a metadynamics run and to analyze the collected data exist. In this Chapter we present the main tools used in this thesis: the **GROMACS** and the **PLUMED** packages. Moreover, with the increasing of the systems complexity, different *parallelization* techniques are developed in order to exploit all the computational resources of the modern multi–core CPUs and multi–node cluster. We shall describe the main used in this thesis work. Then we give the main input parameters used to setup the MD and metadynamics runs.

### 2.1 GROMACS PACKAGE

The MD tool and the main analysis tool used in this thesis work is the **GROMACS** [1] package. It is compatible with both atomistic and CG FFs and support a variety of options and parameters such as different thermostat and barostat algorithms, different methods to treat the electrostatic interaction such as the ESM and the PME method, different integrators, cut–off schemes and so on. For what concern the trajectory analysis it holds a comprehensive framework for obtaining different quantities. Moreover it supports different kind of parallelization techniques that is, it has the compatibility with both multi–core single processor (home PC) and multi–core multi–processor (cluster) architectures with the support of both MPI and OpenMP libraries for the inter–nodes and inter–cores information exchange and the support of different hardware accelerated frameworks such as NVIDIA CUDA<sup>®</sup>, OpenCL<sup>TM</sup> and OpenACC.

#### 2.1.1 Parallelization

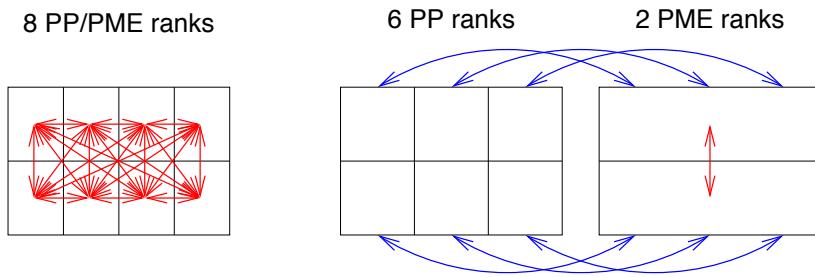
Parallelization refers to the possibility of running a program, in this case an MD simulation, in parallel with the use of more than one CPUs or more than one cores per CPU or both. The efficiency of a parallelization scheme is mainly determined by how much information each core and/or CPU has to exchange with the others, the less the better. The reason is that the libraries (MPI and OpenMP) needed for the exchange of the information are not fast as the cores are. Hence when the waiting time for the information exchange becomes higher than the total time spent in executing the parallel code, the program has reached its scaling limit, that is, no more cores have to be used.

For what concern an MD simulation there exists one main kind of parallelization scheme: the domain decomposition. It exploits the local character of most of the interactions used in common FFs. The simulation box is divided into cells, each

of which is assigned to a cores such that it store only a portion of the whole system: the particles which lie in that region and forces between them and the atom positions and forces from neighboring regions owned by other cores. In order to minimize the communication between different cores each cells has to minimize its surface respect to the volume, so it is important to use cells that are as cubic as possible. Indeed if the cells are not too small, the particle update can be carried out every time the neighbor list is updated and not every MD step. Then, this also contribute to reduce the number of communications. A common choice is to use cell greater than two times the cut-off radius. The scaling limit with the domain decomposition is reached when the cells are so small that a few number of particles lied in each cell; for **GROMACS** it means  $\sim 100 \div 200$  atoms/core.

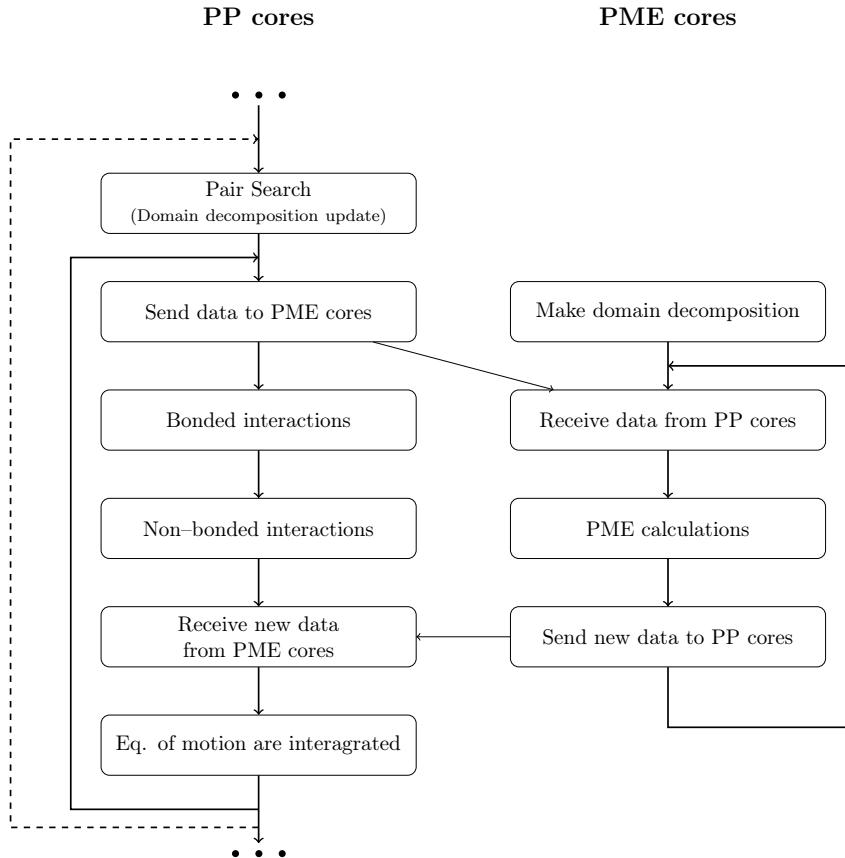
### 2.1.2 PP and PME nodes

When using the PME method for the electrostatic interaction a special parallelization scheme is needed in order to speed up the simulation. The reason is related to the fact that the PME method intrinsically need to know all the particle positions in the whole simulation box. In a domain decomposition scheme this results in a massive use of the so called *all-to-all* communications, i.e. all cores have to communicate to each other. Hence this leads to a loss of computational performance. In order to minimize the all-to-all communications, **GROMACS** implement the possibility to assign a subset of the cores exclusively to the calculation of the energy contribution of the electrostatic interaction using the PME method, while the rest of the work is assigned to the remaining cores, called the particle–particle (PP) cores. In figure (18) a schematic view of the all-to-all communications needed



**Figure 18:** Left: example of the use of 8 cores for both PP and PME. Right: same but 2 cores are used only for PME and the other for the PP contribution. Red arrows represent the all-to-all communications while the blue arrows are the communications between one PP core and one PME core. Taken from [1].

in a comparison between PP/PME in the same cores and the use of the separate PME cores is shown. As one can see the all-to-all communications (red arrows) are smaller for the separate PME scheme than that for PP/PME scheme. With this parallelization scheme the workflow of figure (8) is updated to one shown in figure (19).



**Figure 19:** Schematic workflow of the load balancing between PP and PME cores.

**PME DOMAIN DECOMPOSITION** To optimize the load in the PME cores a domain decomposition is still used but with a 2D or a “pencil” scheme. This because the FFT method is very inefficient if parallelized with too many cores due to the all-to-all communications, thus the domain decomposition acts on the  $xy$ -plane while the  $z$  axis is assigned to one core; pencil refers to the shape of the domain at high parallelization. Moreover the number of domains along the  $x$  axis have to be equal to the number of domains in the PP decomposition and eventually a 1D scheme can be used along the  $y$  axis only if the PP decomposition has one domain along  $x$  axis. To avoid superfluous communication of coordinates and forces between the PP and PME cores, the number of cells along the  $x$  axis should ideally be the same or a multiple of the number of the PME cores. Most of this issues are taken into account approximately automatically by **GROMACS** at the beginning of the simulation.

**PP – PME LOAD BALANCING** In order to achieve the better performance a well balancing of the load between PP and PME cores is needed, i.e. no core should expect to each other. In **GROMACS** this is done balancing the real part and the PME part of the electrostatic interaction. Since the real part is computed on PP cores as non-bonded interaction, changing the Coulomb cut-off between the real and the

PME part, will adjust the load balancing between PP and PME cores. Off course this has to be done keeping unchanged the desired accuracy of the electrostatic energy contribution. In **GROMACS** the ratio between the starting cut-off radius and the PME grid spacing gives the accuracy of the electrostatic energy contribution. Then changing both the Coulomb cut-off and PME grid spacing, leaving the ratio unchanged, will adjust the PP – PME load without affecting the accuracy. We have to stress out that at the beginning of the simulation the Coulomb, the Van der Waals and the pair list cut-off radii must be the same. Then **GROMACS** only adjust the ratio between the Coulomb cut-off and the PME grid spacing, without affecting the cut-off of the pair list or of the Van der Waals interaction.

### 2.1.3 Hardware acceleration

Excluding the PME, an other most time consuming part of an MD simulation, is the computation of the non-bonded interactions. The way to speed up the simulation without recur to more cores, because, for example, the system has reached its scaling limit or there are no longer cores, is to use an hardware accelerator. By offloading the computation of the non-bonded interactions to the hardware accelerator, the CPU has the possibility to do *in concurrency* other calculations such as the PME calculations. **GROMACS** support completely the NVIDIA GPU with the NVIDIA CUDA® framework acceleration together with the use of the Verlet cut-off scheme. Other GPUs or other cut-off scheme are partially supported with OpenCL or OpenACC libraries.

The speed up of the simulation by offloading the non-bonded interactions to a GPU using NVIDIA CUDA® is of the order of two or three times faster than that of a non accelerated simulation. Moreover, to increase the performance, avoiding the waiting time of the CPU for the GPU to finish the non-bonded calculations, the same load balancing between PP and PME part can be used. Adjusting both the Coulomb cut-off and PME grid spacing, leaving the ratio unchanged, will adjust the load between the real and the reciprocal part of the electrostatic energy contribution. Since the real part is computed in the GPU as non-bonded interaction, it will adjust the load balancing between the GPU and the CPU. Using an hardware accelerator the schematic workflow of figure (8) is updated to one shown in figure (22).

Basically and briefly the main difference, as showed in figure (20), between a CPU and a GPU is the total number of *computing modules* assigned to do vectorized calculations i.e. calculation that involve many vectors of different length. Now days the total number of hardware cores in a high level CPU is eight cores. Instead, a middle level GPU has a number of computing modules of the order of  $> 10^3$ . In the GPU case, the computing module is essentially a single instruction – multiple data (SIMD) computing unit which is composed by several cores which each of it is equipped by several arithmetic logic units (ALUs) and floating point units (FPUs). A SIMD unit is a specific module for vector calculation: it give several instruction set to perform the same instruction on a multiple data stream i.e. multiple data arranged in a vector fashion. In figure (21) a schematic represen-



Figure 20: Schematic comparison between a CPU and a GPU.

tation of a SIMD that execute in parallel four two number sums i.e. the sum of

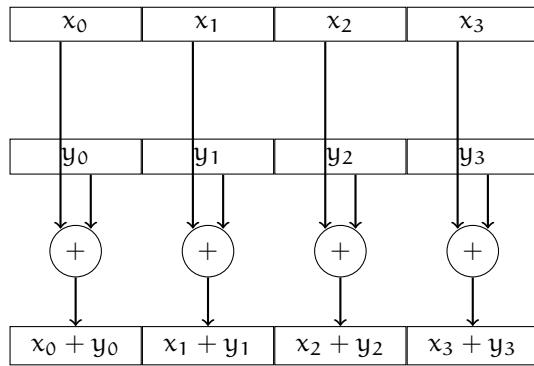


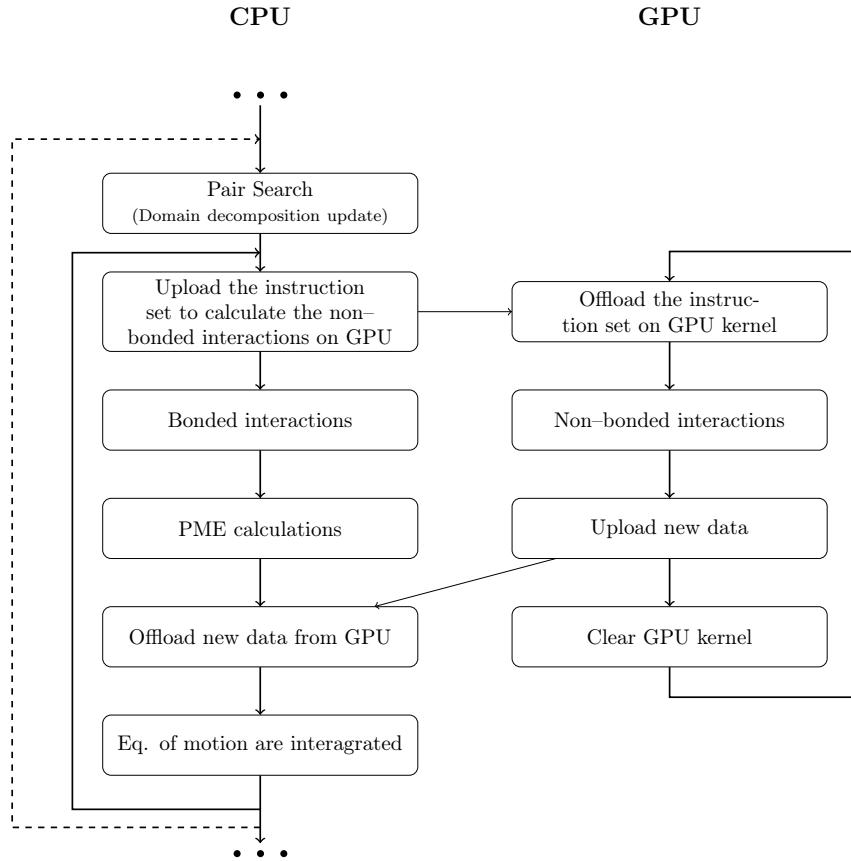
Figure 21: Block diagram of a SIMD computation on a four element vectors.

each cell of the two vectors, is shown. Essentially this is possible thinking a 32-bit register as vector of four 8-bit or two 16-bit numbers, then the same instruction is applied to each cell pair of the vectors. Even a modern CPU has the possibility of SIMD computation but the goal of the GPU implementation is the extremely high number of modules arranged in a efficiency pipeline, respect to CPU one. Hence the *concurrency* calculations on a streamed data are much faster if executed on a GPU then on a CPU. On the contrary the CPU has the ability to perform several task in parallel (or approximately in parallel).

## 2.2 PLUMED PACKAGE

The tool used for setting up and analyze a metadynamics simulation is the **PLUMED** package. It is an open source project composed by a set of libraries used for analyze the data obtained with a metadynamics simulation (first all, for obtaining the FES from the history-dependent bias potential) or other advanced sampling technique, such as umbrella sampling. Moreover it works together with the most popular MD tools, **GROMACS** included. In that sense it is a sort of plug-in that insert the advanced sampling code and algorithms inside the main MD loop of the used MD tool.

For setting up a metadynamics simulation **PLUMED** require an input file in which we have to specify what are the CVs and the parameters to be used in the



**Figure 22:** Schematic workflow of the calculation of the interactions with the use of a GPU as hardware accelerator.

metadynamics run. Moreover there is the possibility to add more feature such as, online analysis or more output files and options.

For what concern the parallelization we have noted that it is badly parallelized between more than one node, that is, we observe good computational performance if we use the parallelization between cores inside the same CPU with the OpenMP library and the use of the GPU as hardware accelerator. Otherwise, for example with more than one node of a cluster, we obtain worse computational performance. Moreover, depending on the system and the parameters of the metadynamics, we observe a general decrease of computational performance compared to an unbiased MD simulation.

Below we show a typical input file used in this thesis work for a metadynamics simulation.

```

# global variable: z distance from COM of particles 385-7040 and paticle 308
memb: COM ATOMS=385-7040
mus: COM ATOMS=308
dist: DISTANCE ATOMS=memb,mus COMPONENTS

# Activate metadynamics in dist.z
METAD ...
LABEL=metad
ARG=dist.z
  
```

```

PACE=50000
HEIGHT=2.479
SIGMA=0.06
FILE=HILLS
GRID_MIN=-9
GRID_MAX=9
GRID_SPACING=0.005
... METAD

# monitor the two variables and the metadynamics bias potential
PRINT STRIDE=10000 ARG=dist.z,metad.bias FILE=COLVAR

```

## 2.3 SIMULATION PARAMETERS

Since **GROMACS** is compatible both with atomistic and CG FFs and with different kind of MD simulations, such as with a NVE or NVT or NpT ensemble, different MD integrators, different algorithms to treat the long- and short-range interactions and so forth, few input files must be given at **GROMACS** input. Briefly these input files are the initial configuration file in which the position and eventually the velocity of all particles and the three box dimensions are saved; the input parameters file in which, following the FF receipt, all the MD parameters are given; the topology file in which all the particles of a system are described: what kind of forces between them, the assigned charge, the mass and all the constant refer to the particles and the interactions between them, even this file, generally, is created under the receipt of the FF used.

In table (4) we report the most important MD parameters used in this thesis work. In table (4) “VbT” means Verlet–buffer–tolerance; “VdW” means Van der Waals; T and p coupling are the thermostat and barostat algorithms used, respectively;  $\tau_T$  and  $\tau_p$  are the related time constants; p type, is the pressure coupling type: semi-isotropic means two independent pressure coupling for xy axes and for z axis; K is the related compressibility (in this case equal for both pressure coupling). In general when starting an MD simulation, an equilibration run (Eq. in table (4)) of few nanoseconds, that also generate the random velocities from the Maxwell–Boltzmann distribution at fixed temperature, is needed for equilibrating the volume of the simulation box and the whole system. Then one have to switch to the production run (Run in table (4)) in which the correct ensemble is sampled.

For what concern the metadynamics parameters, in accordance with [27], in table (5) we report the main input parameters used to set up a metadynamics run.

Input parameter	Eq. PME	Run PME	Eq. PME&PW	Run PME&PW
Time step [fs]	10	20	5	20
cut-off scheme	Verlet	Verlet	Verlet	Verlet
VbT [J/(mol ps)]	5	5	5	5
Coulomb Type	PME	PME	PME	PME
r Coulomb [nm]	1.2	1.2	1.2	1.2
$\epsilon_r$	15	15	2.5	2.5
PME grid [nm]	0.12	0.12	0.12	0.12
PME order	4	4	4	4
VdW type	cut-off	cut-off	cut-off	cut-off
r VdW [nm]	1.2	1.2	1.2	1.2
T coupling	v-rescale	v-rescale	v-rescale	v-rescale
T [K]	310	310	310	310
$\tau_T$ [ps]	2.0	2.0	2.0	2.0
p coupling	Berendsen	Parrinello–Rahman	Berendsen	Parrinello–Rahman
p type	semiisotropic	semiisotropic	semiisotropic	semiisotropic
p [bar]	1.0	1.0	1.0	1.0
$\tau_p$ [ps]	4.0	12.0	4.0	12.0
K [ $\text{bar}^{-1}$ ]	$4.5 \cdot 10^{-5}$	$3.0 \cdot 10^{-4}$	$4.5 \cdot 10^{-5}$	$3.0 \cdot 10^{-4}$
v random	yes (310 K)	no	yes (310 K)	no

**Table 4:** Summary of the main input parameters used in my MD runs.

Deposition pace ( $\tau$ )	Gaussian height ( $w$ )	Gaussian width ( $\delta s$ )	Grid spacing
50000 MD steps <sup>a</sup>	2.479 kJ <sup>b</sup>	0.06 nm	0.005 nm

**Table 5:** Main metadynamics parameters used in this thesis work. <sup>a</sup> With a 20 fs time step it means a deposition of a Gaussian every nanoseconds. <sup>b</sup> It is approximately equal to  $1 k_B T$ .

# 3

## MODEL OF CELL MEMBRANE AND MONOLAYER-PROTECTED NP

In the first part of this chapter we will present and describe the model of the charged monolayer-protected gold nanoparticle (AuNP) developed by Federica Simonelli and co-workers in [26] and [27]. The gold core is treated by an all atoms representation, while the ligands are modeled at CG level by the MARTINI FF. In the second part of the chapter we will describe the most important physical and chemical features of the cell membrane. Then we will summarize the characteristics of the CG model used to treat it. Finally, we will describe the interaction mechanism and some preliminary thermodynamic results about the interaction of the AuNP and the model cell membrane, as outlined in [27]. For a more precise discussion about the nanoparticle (NP)-membrane interaction and the models parameterization the reader is addressed to the work of Federica Simonelli *et al.* [27] and her thesis work [26]. For more details about the gold core used, its properties, equilibrium structure and so forth, the reader is addressed to the work of Lopez-Acevedo *et al.* [18] while for a general discussion about thiolated AuNPs to the work of Häkkinen [9]. For what concerns cell membranes and biological lipids the reader is addressed to the book of Yeagle [34].

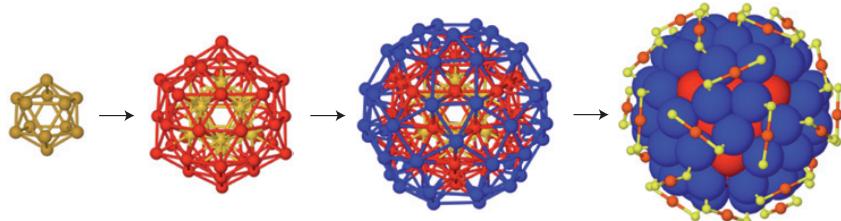
### 3.1 NANOPARTICLE MODEL

Monolayer-protected AuNPs have the advantage of having a well defined molecular structure. That is, mono-dispersed solutions can be synthesized and structurally determined. In particular AuNPs are biocompatible, have a facile surface chemistry and efficiently convert light into heat. Monolayer-protected AuNPs have a definite mass and molecular composition, and their structure is stabilized by the covalently bound ligands shell. Most commonly they are thiolated compounds because they covalently bind to the gold NP by Au-S surface bonds, that is, a robust but modifiable interaction, crucial in stabilizing the NP and transmitting electronic interactions between gold and sulfur-containing organic molecules [9]. Moreover monodisperse AuNPs can be synthesized in the 1 – 4 nm range. Subtle changes of size, structure and ligand compositions and arrangements, can affect NP properties such as their optical properties, important for biological sensing and therapeutics. Several stable thiolated AuNPs are identified, differing in size of the gold core and number of ligands [9]. In this thesis work we will consider the  $\text{Au}_{144}(\text{SR})_{60}$  thiolated AuNP, where R are the aliphatic chain of the thiol compounds. The equilibrium structure of the gold core is described by *ab-initio* calculation in [18].

Changing the composition of the aliphatic chains bonded to the thiol group, different properties of the thiolated AuNP can be achieved, such as different net charge, different level of hydrophobicity, different size and so on. In particular, as we shall see, in the model we will use we consider only octanethiol (OT) and 11-mercaptoundecane sulphonate (MUS) ligands that cover the NP gold core with a monolayer with different compositions and surface arrangements.

### 3.1.1 Passivated gold core

The gold core is composed of 144 atoms, it has icosahedral symmetry and it is made of three bulk shell with 12, 42 and 60 atoms, respectively. A surface shell of 30 atoms completes the gold cluster structure. Then 60 sulphur atoms, which bind the aliphatic chains ( $R$ ) to the gold core, are bounded to the gold atoms on the surface through the typical bond structure  $RS-Au-SR$ . The shell construction is shown in figure (23).



**Figure 23:** First three frame: the concentric 12-(yellow), 42-(red) and 60-(blue) atom gold internal shell, surrounded (last frame) by 30 gold (red small) and 60 sulphur (yellow small) surface atoms. The R chains are not shown. Taken from [9].

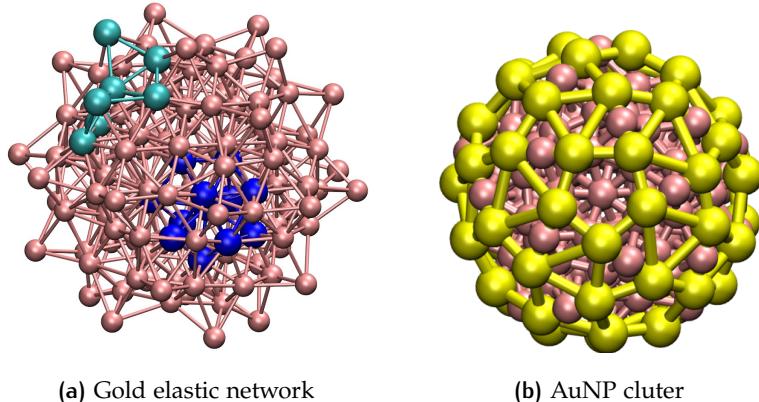
The resulting diameter of the gold core is about 2 nm. When passivated by thiols, its overall size depends on the length of the aliphatic chains bound to the sulphur atoms. The monolayer-protected AuNPs we will consider have a total diameter of about 4 nm.

Despite the computational cost associated to atomistically describe the NP core, all gold and sulfur atoms are taken into account. Bonds between gold atoms and sulfur atoms are allowed to vibrate. As we have seen in a previous section, a many-body potential should be used. Instead, as we are interested in the vibrational modes of the core atoms, a more efficient way, as shown in figure (24), is to use an elastic network associate the potential energy

$$U = \frac{1}{2} \sum_i \sum_{j \neq i}' k_{ij} (r_{ij} - r_{ij}^0)^2$$

where  $r_{ij}$  is the distance,  $k_{ij}$  is the bond constant for  $i-j$  atom pair and the prime indicates that only neighbor atoms within a certain cutoff are considered.

The bond constant is assigned to  $k = 32500 \text{ kJ}/(\text{mol nm}^2)$  for Au–Au surface atoms,  $k = 11000 \text{ kJ}/(\text{mol nm}^2)$  for Au–Au bulk atoms,  $k = 1250 \text{ kJ}/(\text{mol nm}^2)$  for S–S atoms and  $k = 32500 \text{ kJ}/(\text{mol nm}^2)$  for Au–S bonds, as summarized in



**Figure 24:** Left: gold elastic network. In cyan a surface atom and its neighbors; in blue bulk atom and its neighbors. Right: AuNP cluster. The elastic network for both gold and sulfur atoms are represented by sticks. Taken from [26] and [27].

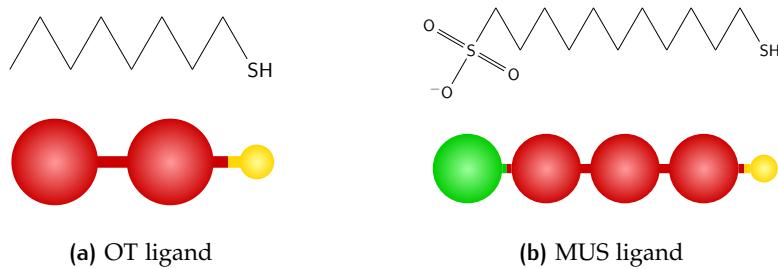
table (6). Instead, the equilibrium distances are derived from *ab-initio* data in [18]. Moreover, to prevent the penetration of other particles inside the NP core, a purely repulsive interaction of the form  $C/r^{-12}$  where  $C = 0.92953 \cdot 10^{-6}$  (kJ nm<sup>12</sup>)/mol, is added between gold and sulfur atoms, gold and all other particles and sulfur and all other particles. A gold atom is considered as bulk atom if it has at least nine gold atom neighbors otherwise as a surface atom. Two gold atoms  $i$  and  $j$  are neighbors if their distance is  $r_{ij} \leq 0.35$  nm. Instead, two sulfur atoms are considered neighbors if they lie in a sphere shell of radius 0.55 nm, i.e. each sulfur atom have at least five neighbors.

Bond	$k$ [kJ/(mol nm <sup>2</sup> )]
Au–Au (bulk)	11000
Au–Au (surface)	32500
Au–S	32500
S–S	1250

**Table 6:** Summary of the bond constants for the elastic network of the NP core.

### 3.1.2 Functionalizing ligands

Our AuNP core is functionalized with MUS and OT ligands as shown in figure 25. MUS is a charged compound made of an alkyl chain ( $\text{CH}_2$ )<sub>11</sub> and a charged terminal  $\text{SO}_4^-$  group. The charged terminal group makes MUS partially hydrophilic. OT, instead, is completely hydrophobic and it is made by an alkyl chain ( $\text{CH}_2$ )<sub>7</sub> and one  $\text{CH}_3$  terminal group. Using both hydrophilic and hydrophobic groups guarantees that NPs are stable, that is, they do not aggregate in aqueous environments.



**Figure 25:** Top: chemical structure. Bottom: CG MARTINI model (red: C<sub>1</sub> bead, green: Qda negatively charged bead and yellow: sulfur atom).

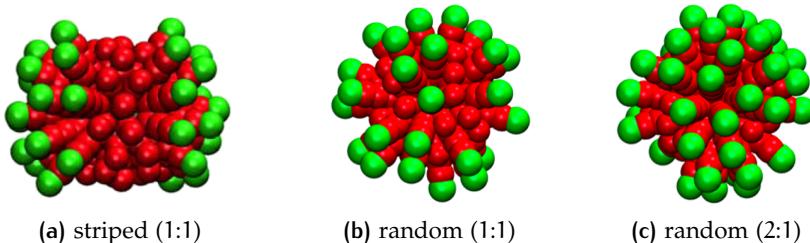
**OT MODEL** Two MARTINI beads of type C<sub>1</sub> model the eight carbon atoms of the OT backbone and their hydrogen atoms. The chemical structure and the resulting CG MARTINI model is shown in figure (25a). The first bead of each OT ligand is bound to a sulphur atom via a harmonic potential with a bond constant of 1250 kJ/(mol nm<sup>2</sup>) and equilibrium length of 0.47 nm. The second bead is connected to the first by the same bond potential. An angle potential as in equation (1.5.1) is used among the three particles. Parameters are fixed in accordance with the standard MARTINI parameters for alkanes.

**MUS MODEL** Three MARTINI beads of type C<sub>1</sub> model the hydrophobic chain of the MUS ligand. The charged group is modeled as a Qda bead with a charge of -e. The chemical structure and the resulting CG MARTINI model is shown in figure (25b). Even in this case the first bead of a MUS ligand is bound to the sulphur atom through a harmonic potential with the same parameter: bond constant of 1250 kJ/(mol nm<sup>2</sup>) and equilibrium length of 0.47 nm. The same potential is used to bind all other beads to the previous one. An angle potential as in equation (1.5.1) is used among the sulfur atom, the first C<sub>1</sub> and second C<sub>1</sub>, among the first, the second and the third C<sub>1</sub> beads and so on for all four beads. Parameters are fixed in accordance with the standard MARTINI parameters for alkanes.

**LEVEL OF HYDROPHOBICITY** The AuNP core can be functionalized with both ligands at different composition. In particular varying the ratio between the OT and MUS ligands different levels of hydrophobicity can be reached. Two surface compositions will be considered in this thesis work: (MUS:OT 1:1) and (2:1), the former is the main used in this thesis work. This choice is due to the possibility to compare to previous experimental and simulation data [28], [31], [32] and [30].

**SURFACE ARRANGEMENTS** The ligands on the AuNP surface can be arranged in two possible ways: randomly or with a predetermined scheme. We will consider both NPs with a random ligand arrangement and NPs with a striped ligand arrangement. The striped scheme is obtained dividing the NP surface in three stripes: the external two stripes are covered with MUS ligands while the central with OT ligands. For this thesis work we consider three type of NPs: striped

(MUS:OT 1:1), random (MUS:OT 1:1) and random (MUS:OT 2:1). In figure (26) the different coatings for the NP core is shown.



**Figure 26:** AuNP with different ligands surface arrangements and composition. From left to right: striped (MUS:OT 1:1), random (MUS:OT 1:1) and random (MUS:OT 2:1). Hydrophobic beads are shown in red while the negatively charged beads are green.

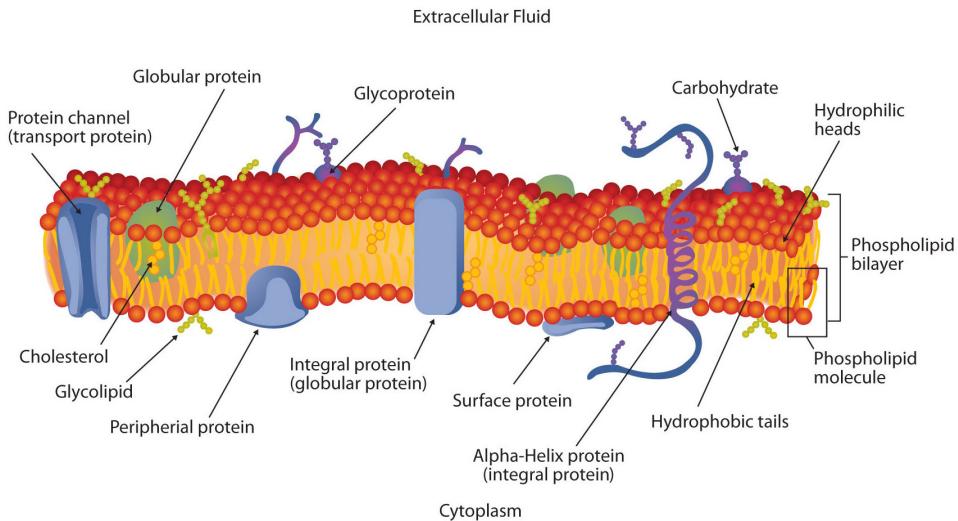
## 3.2 CELL MEMBRANES

The cell membrane or cytoplasmic membrane is a biological membrane that separate the interior and the external environments of a living cells and controls communications and nutrient flow into and out the cell. This is essential to enable the function and stabilize the structure of cells. The membrane main function is the compartmentalization of a biological environment into two well defined not independent subsection: the cell in which the “life reactions” take place and the external cell fluid in which all the necessary biological compounds are dissolved. The compartmentalization in such environments enhance the probability of chemical reactions to take place but also allow to develop different chemical reactions inside and outside the cell. From an evolutionary point of view it is believed that a simple membrane that defines an inside and an outside environments and that protect and concentrates molecules in a closed space, it was necessary for the development of life.

### 3.2.1 Real cell membranes

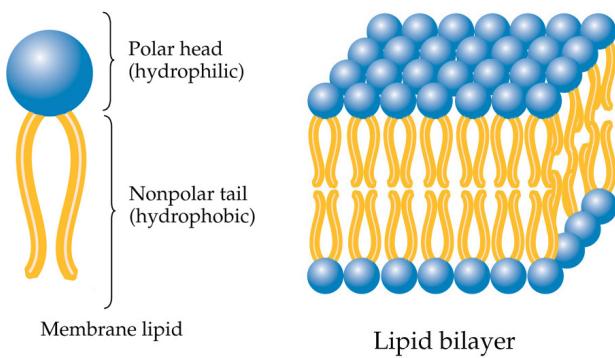
Since the compartmentalization, the cell membrane have to allow the exchange of biological molecules from and to the outside world. This role is, in a small part, fulfilled by the membrane itself through the passive translocation of small molecules or ions. But the main part is fulfilled through proteins and other compounds that are solved inside the membrane. As we can see from a cartoon of a real cell membrane in figure (27), the biological membrane is a crowded environment consisting of phospholipids, glycolipids, carbohydrates, proteins and other organic molecules.

The plasma membrane by mass is essentially composed by half lipids and half proteins. The lipids that constitute the membrane are a particular type of lipids, called *phospholipids*. They are made of a neutral (polar) or charged head, which is



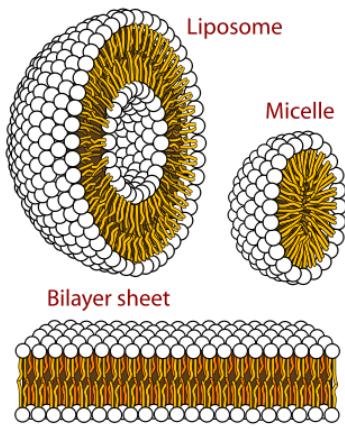
**Figure 27:** Schematic representation of a biological cell membrane.

hydrophilic and one or two fatty acid hydrocarbon chains, often called lipid tails, which are instead hydrophobic, as shown schematically to the left of figure (28). Due to have both properties a phospholipid is an *amphiphilic* molecule. This amphiphilic nature, through the *hydrophobic effect*, play a key role in the membrane formation. In fact when we put a relevant concentration of phospholipids in a water solution at a certain temperature, the lipid tails tend to move away from water molecules due to the hydrophobic effect, so they tend to get closer between them in order to minimize the hydrophobic surface in contact with water. Instead, the polar heads tend to make favorable bonds with water molecules. The resultant bilayer configuration, proper of the biological membrane, is shown schematically to the right of figure (28). Hence, the phospholipids in a water solution tend to *spon-*



**Figure 28:** Left: schematic representation of a phospholipid. Right: self-assembled bilayer configuration assumed by an aqueous solution of phospholipids.

*taneously* self-organize in a *ordered* state that minimize the Gibbs free energy and that depends on the lipids concentration and the temperature of the solution. As schematically shown in figure (29) the final configurations of such self-assembly process are essentially three: the bilayer sheet, a liposome and a micelle.



**Figure 29:** Cross-sectional view of the structures that can be formed by a self-assembly process of a phospholipid aqueous solution.

A real lipid bilayer often contains hundred of different lipid species. They differ in the length of the hydrocarbon chains, in the degree of unsaturation, i.e. in the number of double bond in the hydrocarbon chains, and in different composition of the head that can be neutral (polar) or charged. There are two main classes of phospholipids that make a cell membrane of animals: glycerophospholipid (phosphatidyl-choline, phosphatidyl-ethanolamine, phosphatidyl-serine, phosphatidyl-serine) and phosphosphingolipids (sphingomyelin). In the former group the lipid tails are bound to a glycerol group while the latter do not have glycerol and the lipid tails have a backbone of sphingoid bases, absent in the former. These five types take into account for more than half of the lipids in most membranes.

The cell membrane has quasi-liquid properties at physiological temperature. This is in part due to some disorder in the alignment of the lipid tails produced by the presence of unsaturated chains. Another contribution arises from the area occupied by the lipid heads which determines the distance between the hydrocarbon chains. This fluid character makes the lipid bilayer like a solvent in which the other molecules (lipids and proteins) are dissolved and are free to diffuse. Moreover the lipids themselves can move in different ways. The main movements and the associated time scales are summarized as follows

- lipids conformational change (few nanoseconds);
- lipids protrusions out-of-plane (tens of picoseconds);
- diffusion within a leaflet (order of tens of nanoseconds);
- bilayer undulation and thickness involve collective motion of many lipids.

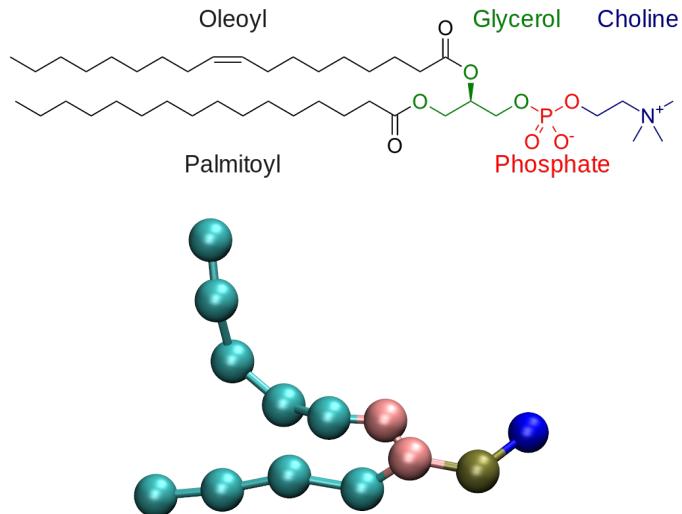
There are also many rare events that take place on the order of hours or even days, such as lipid flip-flop, in which a lipid flips from one leaflet to the opposite one; ion translocation; *electroporation* by water, for example due to a cross membrane ion imbalance, in which water translocates across the bilayer; water assisted ion permeation via formation of a *water-finger* and so forth.

For what concerns the length scales, the bilayer thickness is determined by the length of the lipid tails and their degree of unsaturation. Typically the hydrophobic region is  $\sim 3$  nm thick while each hydrophilic regions is  $\sim 1$  nm thick. Hence the typical bilayer thickness is around  $\sim 4 \div 5$  nm.

### 3.2.2 Model cell membrane

As we have seen above, the cell membrane is an extremely complex environment due to the large number of different biological molecules (lipids, proteins and so on) which composes and resides in the membrane. The model membranes we will consider in this thesis will be composed of lipids only. This choice is dictated by three main reasons. First, current models and computational power could badly aim at reproducing the complexity of a real plasma membrane. Second, the use of a model system allows to tackle fundamental questions concerning the physical and molecular ?? interaction between NPs and membranes, last but not least, the model membrane we will consider resembles closely the model membranes used in a number of experimental and simulations results.

In the bilayer model we will use, we consider model biological membrane consisting of 1-palmitoyl-2-oleoyl-*sn*-glycero-3-phosphocholine (POPC) lipids; the chemical structure is shown in the top of figure (30). It is a zwitterionic glycerophospholipid of type phosphatidyl-choline whose head is made of a phosphate ( $\text{PO}_4^-$ ) and a choline ( $\text{C}_5\text{H}_{14}\text{NO}^+$ ) groups. It has two hydrocarbon chains: one is a saturated chain (palmitoyl) and the other is an unsaturated chain (oleoyl). The head groups and tails are both bounded to the glycerol group ( $\text{C}_3\text{H}_8\text{O}_3$ ).



**Figure 30:** Top: chemical structure of a POPC lipid. Bottom: MARTINI CG model. The tan bead is the phosphate group, choline is in blue, the two pink beads represents the glycerol group and the hydrophobic chains in cyan.

**CG MODEL** It is clear that the number of lipids that constitute a real cell membrane is enormous and it is impossible to take into account an entire cell membrane in a MD simulation. A first approximation is to consider only a small area of the model bilayer. Given the medium area per lipid of about  $0.65 \text{ nm}^2$  and a portion of bilayer of  $\sim 160 \text{ nm}^2$ , which corresponds to about 250 lipids per leaflet, the total number of particles to be included in a atomistic simulation (excluding hydrogen atoms) are about  $26 \cdot 10^3$  plus the water molecules ( $\sim 7 \cdot 10^4$ ). This has a very expensive computational cost and the range of phenomena which can be studied on these time and length scales are very limited, calling for the adoption of a CG approach.

**MARTINI MODEL** As described in section 1.5, we will use the CG MARTINI FF for lipids [19]. The MARTINI model for the POPC lipid maps the choline and the phosphate groups into two beads of type Q<sub>0</sub> and Q<sub>a</sub> negatively and positively charged, respectively. The saturated tail is modeled with four beads of type C<sub>1</sub> while the unsaturated tail is built up of four C<sub>1</sub> beads and one C<sub>3</sub> bead which corresponds to the unsaturated group of atoms. The glycerol group is modeled with two beads of type N<sub>a</sub>. A comparison between the chemical structure and CG model is shown in figure (30).

**MODEL ACCURACY** The standard MARTINI FF is able to capture the main physical properties of a lipid bilayer. These properties include the area per lipid, the distribution of groups across the membrane, the trend of the bending and the area compression moduli in function of the lipids composition and the unsaturation degree of the lipids, the stress profile across the membrane, the process of lipid desorption and flip-flopping, and many other as better described in [19] and [20]. Nevertheless many other properties, prevalently mediated by the electrostatic interaction, are not well described. As we have seen in the previous chapter this is because the MARTINI FF does not take into account the long range treatment of the electrostatic interaction and because the standard MARTINI water is prevalently insensible to the electrostatic interaction<sup>1</sup>. To overcome to this problem the use of the PW model and the PME method, as outlined in [20] and [35], are crucial to better describe the process that involve lipid bilayer, water and charged ions. These are ions translocation; electroporation of the membrane by water, due to a cross membrane ions imbalance; water-helped ions permeation and many other water defects inside the membrane as better described in the works of Marrink and Yesylevskyy.

---

<sup>1</sup> Moreover we can not forget that the MARTINI water bead takes into account four real water molecules. Hence the probability for a MARTINI water bead to permeate the hydrophobic region of the membrane is much less than for a water molecule in an atomistic FF.

### 3.3 NP-MEMBRANE INTERACTION

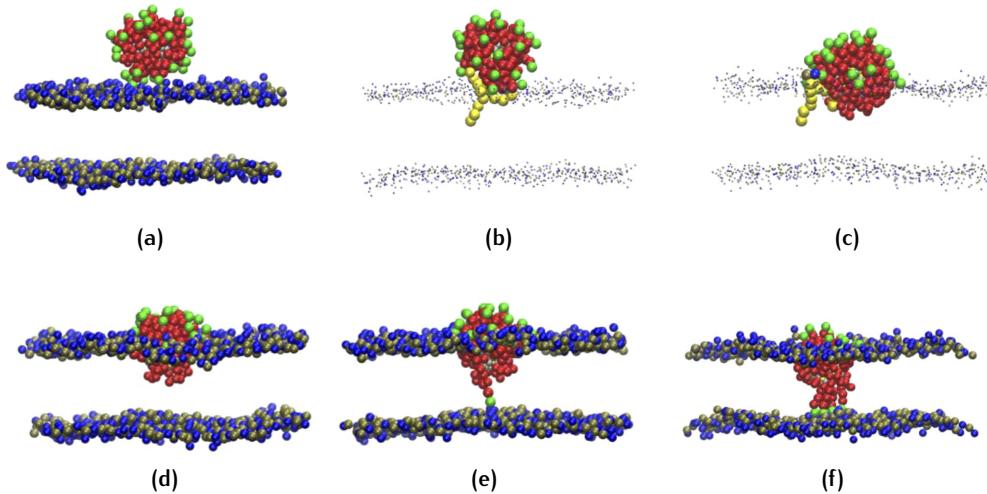
Recently the literature for what concern the computational modeling about the interaction of such anionic monolayer-protected AuNP and model lipid membranes has expanded contributing to sketch a possible mechanism of such interaction. The beginning of the path way mechanism is the electrostatic attraction between the charged ligands and the polar head of the zwitterionic phospholipids in a fluid water phase: it is recognized to be the driving force for the adhesion of the AuNP to the membrane surface, see figure (31a). To the other end of the path way, it is known, on thermodynamic basis, that the most stable state for the AuNP corresponds to the so-called *snorkeling* configuration in which the NP is embedded in the hydrophobic region of the membrane, while the charged ligands stably interact with the lipid heads of both leaflets, see figure (31f).

In the work of Federica Simonelli *et al.* [27], using the just described CG MARTINI models of the NP ligands and of biological membrane, with the standard MARTINI FF, they found and characterize a possible path way mechanism of the interaction process with membrane at low curvature. Then they perform metadynamics simulations for characterize the thermodynamic of the process.

#### 3.3.1 Three-stage process

The authors found a three-stage mechanism that regulate the insertion of the AuNP into a low curvature membrane core: from the adsorbed state, figure (31a), to the snorkeling configuration, figure (31f). When the NP in the water phase is approaching the surface of the membrane it enters in the Stage 1 or in the adsorbed state in which the charged ligands interact with the polar lipid heads. Then, in Stage 2, a key role is played by a lipid tail protrusion out of the hydrophobic region that stably interact with the hydrophobic beads of the NP ligands, see figure (31b-c). In order to minimize the contacts between water and the tails of the protruding lipid it pull the OT ligands into the membrane core and the NP enter in the hydrophobic contact state, figure (31d). Then due the thermal fluctuation a charged bead is involved to cross the membrane core and stably interact with the head region of the opposite leaflet, figure (31e). Since the anchored state is thermodynamically favorable respect to the hydrophobic contact, more and more ligands drop the charged beads to the head region of the opposite leaflet, approaching step-by-step the snorkeling configuration, figure (31f). Moreover, compatibly with other works in literature, they found that the energy cost associated with the extraction of the NP out of the membrane core is very high, making the anchoring process, and the snorkeling configuration, almost irreversible.

The authors performed MD simulations for the three different NP configurations showed in figure (26): the striped (MUS:OT 1:1), the random (MUS:OT 1:1) and the random (MUS:OT 2:1). They found that the described behavior is common to the three configurations. Moreover, the lipid tail protrusion mechanism is validate form the kinematic of the simulations: the energy barrier to be overcome to move from stage 1 to stage 2 is of the order of the estimated energy cost of a lipid



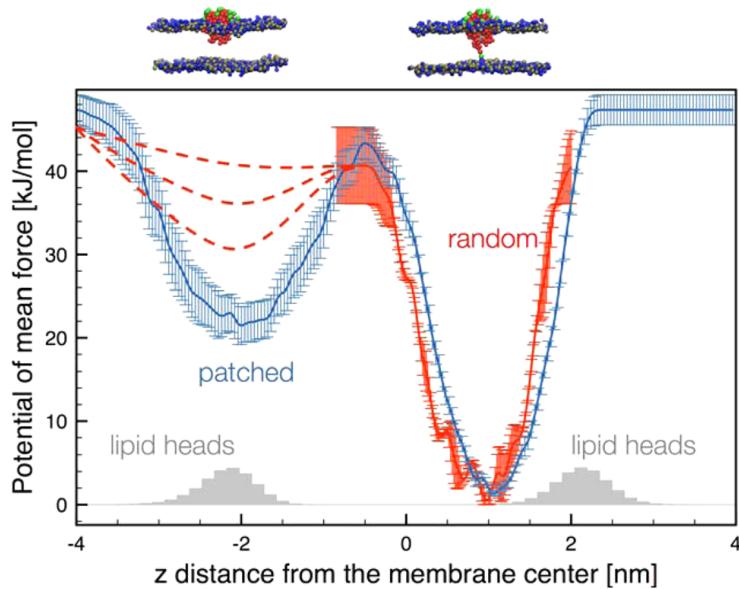
**Figure 31:** AuNP–membrane interaction. (a) Stage 1, adsorption of the NP at membrane surface; (b) to (d) Stage 2, the protrusion of a lipid tail initiates the hydrophobic contact that leads to partial embedding of the NP in the membrane core; (e) the NP binds to the opposite leaflet throwing a charged ligand; (f), Stage 3, snorkeling configuration (five anchors shown). The hydrophobic beads of the ligands are shown in red and the charged beads in green. Lipid heads are blue (choline) and tan (phosphate), lipid tails are not shown, except for (b) and (c), where the protruding lipid is shown with yellow tails. Water beads are not shown. All snapshot refer to a random (MUS:OT 1:1) configuration. Taken from [27].

tail protrusion. Nevertheless, after the hydrophobic contact is reached, its lifetime depends on the surface ligand arrangements. For the random configurations the time lag between the hydrophobic contact and the first anchor is on the order of few nanoseconds. Instead, the striped configuration can linger in the stage 2 for several microseconds. This suggest that the energy barrier to be overcome to move from stage 2 to a one anchor state is less for the former than that for the latter.

### 3.3.2 Preliminary metadynamics results

To quantify the thermodynamic behavior, metadynamics calculations are performed. In particular, in order to estimate the energy barrier for the anchor process, the FES for only one anchored ligand, is computed for the three NP configurations. The choice CV is the  $z$  component of the distance between the center of the charged beads and the COM of POPC. Then, starting from the hydrophobic state the metadynamics simulations begin biasing the charged bead. Since the achievement of the convergence it was not possible, the authors decide to perform different statistically independent metadynamics runs and stop each simulation at the recrossing process of the biased ligand. A recrossing is determined when the charged bead is 0.5 nm above the COM of the membrane in the leaflet in which the NP stay. The resulting FES profile, shown in figure (32), is obtained averaging the statistically

independent runs. The errorbars are the standard error of the independent runs, as described in 1.6.2.



**Figure 32:** Free energy profile related to the transfer of a negatively charged ligand from the entrance leaflet to the opposite one. The blue is related to the striped (MUS:OT 1:1) configuration while the red to the random (MUS:OT 1:1) configuration. Metadynamics data are shown with errorbars, while the red dashed line are hypothesized profiles for the stage 2 to stage 3 transition of random NPs. Gray shades show the position of the polar head beads (AU). Taken from [27].

For the striped arrangement, the crossing barrier to reach the saddle point, located at 0.5 nm off the COM of POPC, is of the order of  $\sim 22$  kJ/mol. While the recrossing energy barrier is approximately twice as high. The same is for the recrossing barrier of the random configuration. The absent of data for the random NPs in the hydrophobic contact is due to the difficulty to sample the stage 2 to the anchor state transition for only one charged ligand. This is because, while the metadynamics is running on a specific charged ligand, other charged ligands spontaneously anchor to the opposite leaflet, making the free energy sampling of the first anchor impossible. Hence, together with the kinematic, for which the average lifetime of stage 2 for random NPs is three order of magnitude shorter than for striped, the authors confirm the very low energy barrier, of the order of few  $k_B T$ , for the anchor process of the random NPs.

# 4

## RESULTS FROM METADYNAMICS SIMULATIONS

4.1 METADYNAMICS SET-UP

4.2 MODELS COMPARE

4.3 CONVERGENCE PROBLEMS

4.4 HEAD LIPIDS DRAGGING

4.5 MEMBRANE LEAFLET ENGULFMENT



# 5 | RESULTS FROM UNBIASED SIMULATIONS



# 6 | SUMMARY AND CONCLUSIONS



## BIBLIOGRAPHY

- [1] Mark Abraham et al. *GROMACS Reference Manual Version 5.1*. 2015. URL: <http://manual.gromacs.org/documentation/5.1/manual-5.1.pdf>.
- [2] 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>.
- [3] Davide Bochicchio et al. “Calculating the free energy of transfer of small solutes into a model lipid membrane: Comparison between metadynamics and umbrella sampling”. In: *The Journal of Chemical Physics* 143.14, 144108 (2015). DOI: <http://dx.doi.org/10.1063/1.4932159>. URL: <http://scitation.aip.org/content/aip/journal/jcp/143/14/10.1063/1.4932159>.
- [4] 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>.
- [5] 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>.
- [6] 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>.
- [7] 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>.
- [8] 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>.
- [9] Hannu Häkkinen. “The gold-sulfur interface at the nanoscale”. In: *Nature Chemistry* 4 (), pp. 443–455. DOI: <10.1038/nchem.1352>. URL: <http://dx.doi.org/10.1038/nchem.1352>.
- [10] Jochen S. Hub, Bert L. de Groot, and David van der Spoel. “g\_wham-A Free Weighted Histogram Analysis Implementation Including Robust Error and Autocorrelation Estimates”. In: *Journal of Chemical Theory and Computation* 6.12 (2010), pp. 3713–3720. DOI: <10.1021/ct100494z>. URL: <http://dx.doi.org/10.1021/ct100494z>.

- [11] iGem. *Membrane Modeling for XylE-transporter*. URL: [http://2015.igem.org/Team:TU\\_Darmstadt/Project/Bio/Modeling/sec3](http://2015.igem.org/Team:TU_Darmstadt/Project/Bio/Modeling/sec3).
- [12] Johannes Kästner. "Umbrella sampling". In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 1.6 (2011), pp. 932–942. ISSN: 1759-0884. DOI: 10.1002/wcms.66. URL: <http://dx.doi.org/10.1002/wcms.66>.
- [13] Alessandro Laio and Francesco L. Gervasio. "Metadynamics: a method to simulate rare events and reconstruct the free energy in biophysics, chemistry and material science". In: *Reports on Progress in Physics* 71.12 (2008), p. 126601. URL: <http://stacks.iop.org/0034-4885/71/i=12/a=126601>.
- [14] Alessandro Laio and Michele Parrinello. "Escaping free-energy minima". In: *Proceedings of the National Academy of Sciences* 99.20 (2002), pp. 12562–12566. DOI: 10.1073/pnas.202427399. URL: <http://www.pnas.org/content/99/20/12562.abstract>.
- [15] Alessandro Laio et al. "Assessing the Accuracy of Metadynamics". In: *The Journal of Physical Chemistry B* 109.14 (2005), pp. 6714–6721. DOI: 10.1021/jp045424k. URL: <http://dx.doi.org/10.1021/jp045424k>.
- [16] A. R. Leach. *Molecular Modelling: Principles and Applications*. Pearson Education. Prentice Hall, 2001. ISBN: 9780582382107. URL: <https://books.google.it/books?id=kB7jsbV-uhkC>.
- [17] Cesar A. López et al. "Martini Coarse-Grained Force Field: Extension to Carbohydrates". In: *Journal of Chemical Theory and Computation* 5.12 (2009), pp. 3195–3210. DOI: 10.1021/ct900313w. URL: <http://dx.doi.org/10.1021/ct900313w>.
- [18] Olga Lopez-Acevedo et al. "Structure and Bonding in the Ubiquitous Icosahedral Metallic Gold Cluster Au<sub>144</sub>(SR)<sub>60</sub>". In: *The Journal of Physical Chemistry C* 113.13 (2009), pp. 5035–5038. DOI: 10.1021/jp8115098. URL: <http://dx.doi.org/10.1021/jp8115098>.
- [19] 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>.
- [20] 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. URL: <http://dx.doi.org/10.1039/C3CS60093A>.
- [21] 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>.
- [22] M. Parrinello and A. Rahman. "Crystal Structure and Pair Potentials: A Molecular-Dynamics Study". In: *Phys. Rev. Lett.* 45 (14 Oct. 1980), pp. 1196–1199. DOI: 10.1103/PhysRevLett.45.1196. URL: <http://link.aps.org/doi/10.1103/PhysRevLett.45.1196>.

- [23] M. Parrinello and A. Rahman. "Polymorphic transitions in single crystals: A new molecular dynamics method". In: *Journal of Applied Physics* 52.12 (1981), pp. 7182–7190. DOI: <http://dx.doi.org/10.1063/1.328693>. URL: <http://scitation.aip.org/content/aip/journal/jap/52/12/10.1063/1.328693>.
- [24] 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](https://doi.org/10.1039/C0SM00481B). URL: <http://dx.doi.org/10.1039/C0SM00481B>.
- [25] Benoît Roux. "The calculation of the potential of mean force using computer simulations". In: *Computer Physics Communications* 91.1 (1995), pp. 275–282. ISSN: 0010-4955. DOI: [http://dx.doi.org/10.1016/0010-4955\(95\)00053-I](http://dx.doi.org/10.1016/0010-4955(95)00053-I). URL: <http://www.sciencedirect.com/science/article/pii/001046559500053I>.
- [26] Federica Simonelli. "A computational study of the interaction between metal nanoparticles and model cell membranes". MA thesis. Università degli Studi di Genova, 2015.
- [27] Federica Simonelli et al. "Monolayer-Protected Anionic Au Nanoparticles Walk into Lipid Membranes Step by Step". In: *The Journal of Physical Chemistry Letters* 6.16 (2015), pp. 3175–3179. DOI: [10.1021/acs.jpclett.5b01469](https://doi.org/10.1021/acs.jpclett.5b01469). URL: <http://dx.doi.org/10.1021/acs.jpclett.5b01469>.
- [28] Sabina Tatur et al. "Effect of Functionalized Gold Nanoparticles on Floating Lipid Bilayers". In: *Langmuir* 29.22 (2013), pp. 6606–6614. DOI: [10.1021/la401074y](https://doi.org/10.1021/la401074y). URL: <http://dx.doi.org/10.1021/la401074y>.
- [29] M. Tuckerman. *Statistical Mechanics: Theory and Molecular Simulation*. Oxford Graduate Texts. OUP Oxford, 2010. ISBN: 9780191523465. URL: <https://books.google.it/books?id=Lo3Jqc0pgrcC>.
- [30] Reid C. Van Lehn and Alfredo Alexander-Katz. "Pathway for insertion of amphiphilic nanoparticles into defect-free lipid bilayers from atomistic molecular dynamics simulations". In: *Soft Matter* 11.16 (2015), pp. 3165–3175. ISSN: 1744-683X. DOI: [10.1039/C5SM00287G](https://doi.org/10.1039/C5SM00287G). URL: <http://xlink.rsc.org/?DOI=C5SM00287G>.
- [31] Reid C. Van Lehn et al. "Effect of Particle Diameter and Surface Composition on the Spontaneous Fusion of Monolayer-Protected Gold Nanoparticles with Lipid Bilayers". In: *Nano Letters* 13.9 (2013), pp. 4060–4067. DOI: [10.1021/nl401365n](https://doi.org/10.1021/nl401365n). URL: <http://dx.doi.org/10.1021/nl401365n>.
- [32] Reid C. Van Lehn et al. "Lipid tail protrusions mediate the insertion of nanoparticles into model cell membranes." In: *Nature communications* 5 (2014), p. 4482. ISSN: 2041-1723. DOI: [10.1038/ncomms5482](https://doi.org/10.1038/ncomms5482). URL: <http://www.ncbi.nlm.nih.gov/pubmed/25042518>.
- [33] 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](https://doi.org/10.1103/PhysRev.165.201). URL: <http://link.aps.org/doi/10.1103/PhysRev.165.201>.

- [34] P.L. Yeagle. *The Membranes of Cells*. Elsevier Science, 2016. ISBN: 9780128004869.  
URL: <https://books.google.it/books?id=hAPSBgAAQBAJ>.
- [35] Semen O. Yesylevskyy et al. "Polarizable Water Model for the Coarse-Grained MARTINI Force Field". In: *PLoS Comput Biol* 6.6 (June 2010), pp. 1–17. DOI: 10.1371/journal.pcbi.1000810. URL: <http://dx.doi.org/10.1371%2Fjournal.pcbi.1000810>.