Introduction to Molecular Dynamics Simulations of Monolayers at Liquid/Air Interfaces

Ángel Piñeiro: http://www.angel.pineiro.sitew.com

1st June 2009

Contents

1   Introduction

The structure of macromolecules and supramolecular assemblies is not rigid and static; they experience local and global movements and rearrangements at different time scales due to thermal diffusion/fluctuations, or in response to external perturbations such as pressure, concentration, pH, or temperature changes. This dynamic behaviour and the dependence of the structure on external factors are key for technological applications. However, it is difficult to obtain experimental structural information on individual molecules as a function of time, as well as to get fine structural information of supramolecular assemblies. During the last years atomistic molecular dynamics (MD) simulations based on classical force fields have contributed considerably to the understanding of macromolecules and supramolecular assemblies. By employing this approach, simulated systems can be controlled and manipulated in previously unthinkable ways, providing data on properties and features that can be compared to experimental results, together with other significant information not available from experiments. Among many other applications of this powerful computational technique, it is now possible to describe the complete folding pathway of peptides about 20 residues long, to suggest reliable structures of protein-ligand aggregates that take the solvent into account explicitly, to observe micelle, lipid bilayer or even vesicle formation processes starting from random aqueous mixtures of the solutes, or to study even more complex systems like membrane proteins considering specific lipid compositions of the bilayer in addition to the intra and extracelullar environments, all this in atomic detail. During this mini-course, the theoretical bases of **classical** MD simulations **at atomic level** are summarized, and a couple of simple applications to monolayer systems are studied in detail.

2   Historical Evolution

Nearly thirty years ago, the first molecular dynamics (MD) simulation of a protein at atomic level was performed by J. Andrew McCammon, when he was a research fellow at Harvard collaborating with Martin Karplus. The simulated protein was the small bovine pancreatic trypsin inhibitor (BPTI), with approximately 500 atoms, in vacuum, and the simulation time was only 9.2 ps. Even with these limitations, the significant atomic fluctuations showed by the BPTI in that pioneer work contributed to change the classical view of biological macromolecules as rigid structures. Several MD simulations of macromolecules in vacuum were reported in the following years. The Simple Point Charge (SPC) water model published in 1981 by Berendsen et al. made possible the implementation of biomolecular dynamics simulations with explicit solvent. A variety of water models arose in the early 80's. These simulations made evident the importance of the solvent for the study of biological systems. Proteins associated with lipid membranes constitute approximately 30 % of genomes and form more than half of all drug targets. Thus, researchers realized that a correct modelization of lipids and related molecules was needed to mimic realistic biological environments. The former simulations in solution were performed with small proteins or DNA polynucleosides in the ps time scale due to the high computational cost of the resulting systems, with several thousands of water molecules in addition to the solute. The transition from the ps to the ns and μs timescales for systems with nearly 10^4 atoms

had place almost simultaneously nearly ten years ago ($\sim$1998). Currently, atomistic MD simulations with many thousands of particles (up to 10^5) and tens of ns are common. Large time and size scales (μs and μm) were achieved with programs and algorithms able to take advantage of the accessibility of many processors simultaneously. Those resources are not currently available to everybody. However, considering the progress on both hardware and algorithms implemented in molecular dynamics engines, and taking the Moore's law[1] as a reference, it is reasonable to guess that what today is practically unattainable tomorrow will be routine. A conservative extrapolation of representative time scales, following the historical evolution of average time scales of single trajectories, is presented in Figure 1. By assuming that typical simulations of medium-sized systems in 1990, 1995, 2000, and 2005 were 0.1, 0.5, 5, and 50 ns long, respectively, the average time scale of single trajectories expected for 2020 is longer than 20 μs, longer than 6 s for 2050, and nearly 7 min for 2060. Figure 3 shows a similar estimation performed for the average number of simulated atoms. It considers the 1977 BPTI simulation with 500 atoms, and typical-size simulations of 10^4 and 10^5 atoms in 1998 and 2005, respectively.
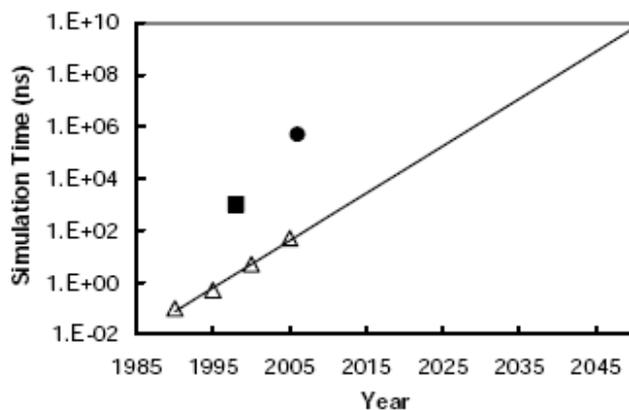

Figure 1: Typical molecular dynamics simulation times of supra or macromolecular systems between 1990 and 2005 (triangles). Linear fit extrapolated till 2050 (line). Simulation times reported for the α-helical villin headpiece in 1998 (solid square) and 2006 (solid circle). The simulations performed for the villin subdomain suggest that the extrapolation underestimates the actual progress in this field.

3   Theory

In order to perform a MD simulation several elements need to be combined:

- The initial conditions that include mainly the composition of the system, atomic coordinates, and initial velocities.

- The topology of the molecules and the potential function that, together, define the interactions of all atoms with each other.

- The conditions under which the simulation will be performed: algorithms to be employed to evaluate the different types of interactions, constraints and restraints to the

---

[1]Moore's law describes a long-term trend in the history of computing hardware. Since the invention of the integrated circuit in 1958, the number of transistors that can be placed inexpensively on an integrated circuit has increased exponentially, doubling approximately every two years.

Figure 2: Typical number of atoms of systems simulated between 1977 and 2005 (triangles). Linear fit extrapolated till 2050 (line). Number of atoms of a system simulated by Freddolino et al. (solid square). The simulation of the complete satellite tobacco mosaic virus performed in 2006, with more than 10^6 atoms, suggests that the extrapolation underestimates the actual progress in this field.
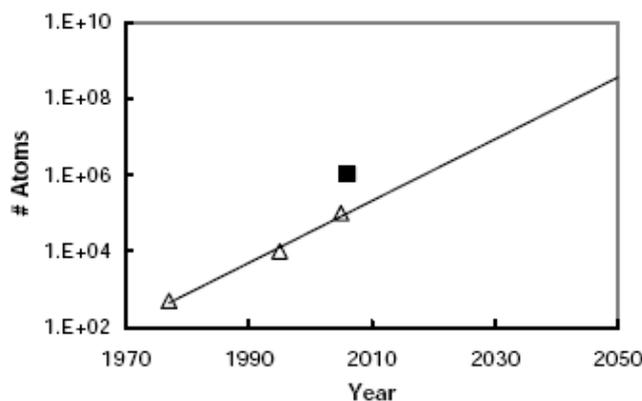
movement of atomic groups, external forces, time scale of the simulation, time step, frequency at which the different data will be saved, etc.

- A molecular dynamics engine, i. e., the program and the electronic devices that will compute the dynamic evolution of the system and will store the results.

- Finally the results should be analyzed.

These items are briefly described in this section.

3.1   Initial Conditions

Typical molecular dynamics simulations describe the movement of every atom in the system being studied, by solving numerically the Newton or Lagrange equations of motion. This means that quantum effects such as proton transfer, high frequency degrees of freedom, or low temperatures are not commonly treated. To perform a MD simulation, an initial set of the atomic coordinates and velocities is required. Protein coordinates are generally taken from the Protein Data Bank if the structure of the molecule was previously determined by NMR or X-ray spectroscopy, or from computational models when the experimental structure is unavailable. The coordinates of smaller molecules such as lipids can be easily modelled using simple molecular viewers or other computational programs developed for this purpose. The initial conformation of the molecule to be studied, defined by such coordinates, is then rotated and translated as a whole in a convenient way, to introduce it in a virtual mathematical box bigger than the molecule itself. If needed, the solute molecule is replicated in a more or less ordered way, depending on the specific aim of the simulation. When the solvent employed is homogeneous, pre-equilibrated small boxes of the solvent are successively copied throughout the main simulation box and, in a following step, the added molecules which overlap any solute atom are removed from the coordinates file. The resulting system is a box completely filled with the solute and the solvent atoms. Sometimes it is convenient to include also ions or different kinds and amounts of molecules; this is performed similarly to the addition of the main solvent. A special case is the simulation of membrane proteins since such macromolecules have to be introduced

4

in a lipid bilayer with a certain orientation before adding the solvent as described above. Periodic boundary conditions are usually employed to avoid artifacts caused by unrealistic walls. This means that replicas of the box, also termed the unit cell of the simulation, are infinitely copied in three dimensions in such a way that the Euclidean space is completely filled. The use of periodic boundary conditions restricts the geometry of the simulation box. Classical boxes are cubic or rectangular, although nowadays octahedron or dodecahedron boxes are common in order to optimize the size of the system and the isotropy of any artifact due to the use of periodic boundary conditions. If the simulation unit cell is expanded in one direction, an interface of the system with vacuum (or air) is created allowing to simulate adsorption processes, monolayers, thin films and related systems. The initial velocities of every atom in the final system are usually randomly assigned under the restriction that they follow a Maxwell distribution corresponding to the temperature at which the MD simulation will be performed.

3.2 Potential Function

In addition to the coordinates and velocities, to determine the dynamics of the system, the force on every atom has to be calculated. This is done by means of the following expression:

$$F = -\vec{\nabla} V \tag{1}$$

where $\vec{\nabla}$ is the well known vector differential operator defined by

$$\vec{\nabla} = \frac{\partial}{\partial x} \vec{i} + \frac{\partial}{\partial y} \vec{j} + \frac{\partial}{\partial z} \vec{k}$$

and V is a semi-empirical potential function for which several expressions and sets of parameters, altogether called the force field, have been proposed. A typical molecular force field has the following form:

$$V = V_{bonded} + V_{non\text{-}bonded} + V_{special} \tag{2}$$

where

$$V_{bonded} = V_{bonds} + V_{angles} + V_{improper} + V_{proper} \tag{3}$$

and

$$V_{nonbonded} = V_{L\text{-}J} + V_{Coulomb} \tag{4}$$

and $V_{special}$ is an optional potential function artificially imposed to bias the behaviour of the system in a special or particular way. Typically, $V_{special}$ is used to restrain the positions or distances of particular groups of atoms. V_{bonded} includes the covalent bond-stretching (V_{bonds}), angle-bending (V_{angle}), and improper dihedrals or torsions ($V_{improper}$) interactions, which are, all of them, modelled by means of harmonic potentials. Proper dihedrals (V_{proper}) interactions, also named rotations of molecular groups, are typically defined by a sinusoidal function. The four contributions to V_{bonded} act between atoms separated by less than three covalent bonds. $V_{non\text{-}bonded}$ is pair-additive for all the atoms in the system. Typically, it contains a Lennard-Jones potential (V_{L-J}) for the dispersion and repulsion terms, and a Coulomb potential ($V_{Coulomb}$) for (partially) charged atoms. V_{L-J} is a short range interaction and only the contribution of the atoms contained in a sphere with a radius shorter

than 1 nm is significant. However, $V_{Coulomb}$ has important effects even at long molecular-scale distances. Since it is evaluated by means of a pair-additive potential, the number of contributions for a N-particles system is N(N−1)/2, so it is said to scale as N^2. Many algorithms, grouped in lattice summation techniques and fast multipole methods, have been developed to save computational time in the evaluation of this latter contribution, because it represents the highest percentage of the resources employed in a molecular dynamics simulation. The most popular of these techniques are the artificial use of some cut-off for the long range interactions, the Ewald summation, and its efficient discrete refinements: the Particle Particle - Particle Mesh (PPPM or P3M), and the Particle Mesh Ewald (PME) methods. P3M and PME scale as N ln(N). Each of the potential contributions needs a series of atom-dependent parameters like spring constants and equilibrium distances to be included in the harmonic potentials, the amplitude and reference angle in the proper dihedral, the L-J radii, and the partial charges for the Coulomb term. Several works focusing in the comparison of different force fields and parametrizations have been published. They conclude that in many cases at the time scales studied, the results do not depend on the combination of equations and parameters employed. CHARM, AMBER, GROMOS, and OPLS are among the most employed force fields in MD simulations of protein systems.

3.3   Running Molecular Dynamics Simulations

The set of parameters defining all the interactions in a molecular system is named "topology" and contains all the information needed to calculate the force on each particle as a function of the coordinates of all the atoms in the system. The trajectory of every atom is calculated from the force associated to it, by integration of the movement equation during very small time steps (Δt) typically of 1 or 2 fs (1 fs = 10^{-15} s). Then, the coordinates and velocities of all the atoms in the system are updated. The integration is performed by using specific algorithms such as the popular Verlet or some of its variants, which assume that the force on every atom is constant during the time step. The time step by itself is also a critical parameter that determines both the stability and the speed at which the MD simulation is performed. Stable simulations require a time step at least 4 times lower than the minimum period, corresponding to the fastest degree of freedom, of the system. The degrees of freedom of the simulated system can be grouped as a function of their frequency. For instance, it is well-known that bond-length vibrations happen at frequencies within the quantum limit and they are usually disregarded in classical simulations by considering fix bond lengths. It has been proved that this assumption does not change the dynamics of the system, in contrast with restrictions imposed on angles. Fastest degrees of freedom in biological MD simulations are those involving hydrogen atoms due to their low mass and several approaches have been proposed to make them slower. Under the previous conditions, the total energy of the simulated system should be constant, but cumulative errors, mainly due to integration schemes and computer precision, produce drift energies. A number of algorithms are employed to perform the simulation under a particular thermodynamic ensemble. The most common of these algorithms are thermostats and barostats which correct the velocities and coordinates of the atoms to keep constant the temperature and pressure, respectively, of the system. Several molecular dynamics engines are currently available, the best known being CHARM, AMBER, GROMACS, GROMOS, NAMD, Q,

and TINKER. By using these computational packages, molecular dynamics simulations of biological systems are accessible even for non specialized people. However, considering the multiple options, algorithms and parameters involved in any simulation, it is strongly recommendable to know the theoretical basis of this technique before working with it.

3.4   Trayectory analysis

During the simulation, several data, such as the coordinates and velocities of all the atoms, the temperature, pressure, and energy contributions, are stored for further analysis. The resulting files are usually very large due to the number of particles involved. Thus, only data corresponding to one of every 10^2 to 10^4 steps are typically stored. The analysis performed on the trajectory will depend on the objectives of the simulation itself. Usually it is specific for each work although the analysis of several properties is standard. In general, before starting to determine structural or energetic properties, one should visualize the trajectory by means of some molecular viewer, like for example RASMOL, PyMOL, or VMD. Such programs allow the representation of proteins in different ways and colours, generating high quality images and/or movies. In the case of macromolecules, their structural deviation from a reference conformation, usually obtained by X-ray spectroscopy, as a function of the time is normally quantified by means of the Root Mean Square positional Deviation (RMSD). RMSD plots appear in almost every publication dealing with molecular dynamics simulations of proteins systems in solution or at water/air interfaces. Typically such plots present a sharp slope during the first few nanoseconds and oscillate around a constant average value during the rest of the simulation. Another typical analysis for these systems is the calculation of Root Mean Square positional Fluctuations (RMSF) averaged for different groups of atoms. RMSF is defined as the standard deviation of the RMSD values throughout the trajectory. RMSF values are closely related to the crystallographic B-factors of protein residues. Macromolecules radii of gyration can also be easily calculated from MD simulations and are specially useful when comparing the dependence of MD simulations on different conditions such as force fields or water models as well as in polymer folding simulations. An important and very common analysis in MD simulations of protein systems is the identification of secondary structure blocks of amino acids. The Dictionary of Secondary Structure of Proteins (DSSP) defines different kinds of local structure as a function of geometrical parameters like intramolecular angles and distances. Such definitions are employed to identify residues with common characteristics. Then, plots with a colour code for every type of secondary structure are displayed as a function of time. Visual analysis, RMSD, RMSF, radius of gyration and determination of secondary structure are just examples of the most typical analysis performed on simulations of macromolecular and protein systems. For MD simulations of monolayers, bilayers or micelles there are other interesting parameters to be calculated such as the surface pressure, the molecular area for the different phases that can be observed, the order parameters of the hydrocarbon chain, and the distribution of molecules along the axis perpendicular to the plane of the interface. In general, from the information contained in a MD trajectory, a very large number of properties can be calculated including fluctuation correlations between different parts of the system, calculus of energetic contributions, temperature, pressure, determination of relative orientations and distribution of molecules, and estimation of thermodynamic properties such as binding or

adsorption energies.

4 Simulating a DPPC Monolayer

4.1 Introduction

The aim of this section is to show you how to start a molecular dynamics simulation of a lipid monolayer, as well as how to analyze the resulting trajectory, by using the GROMACS package. This is a very basic course and we will not go into fine details of algorithms and commands but following this tutorial you are expected to be able to run and analyze a basic monolayer MD simulation.

GROMACS (acronym for GROningen MAchine for Chemical Simulations) is one of the most powerful programs package for general purpose MD simulations. It has been recognized to be the fastest engine for MD simulations. Additionally it is free, supports a wide variey of force fields, and it is very flexible. Other programs packages are available but we have adopted GROMACS for this course.

A dipalmitoyl phosphatidyl choline (DPPC) monolayer was chosen as the victim for your first MD simulation. Amongst surfactant lipids, 80-90 % are phospholipids, 70-80 % of which are phosphatidylcholines, approximately half of which is DPPC. It is doubtless the primary component of lung surfactant and it is though to be the main responsible for the reduction of surface tension in the lungs to near-zero. DPPC is also a standard compound of biological membranes such as those of cells and vesicles. If you do not know this lipid and are curious, just type DPPC on your preferred web browser and will find plenty of information about this molecule.
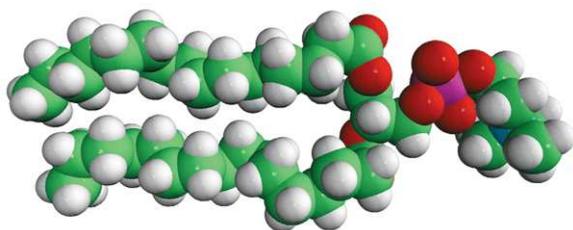

Figure 3: Dipalmitoyl Phosphatidyl Choline (DPPC) structure.

4.2 Generate the Monolayer Model

An initial monolayer model can be obtained in different ways. The most common methods are the formation of the monolayer from a random aqueous mixture in the presence of an interface or the separation of a bilayer into two leaflets exposed to the air. Here we will start from an already preformed structure obtained using the second method. The file will be downloaded from a public server to a directory called "monolayer", which you should create by using the command

```
mkdir monolayer
```

Then go that directory
```
cd monolayer
```

and type:

```
wget web.usc.es/~fangel/monodppc.pdb
```

to get the pdb file from the server. This is a standard pdb-file which contains the list of atoms consisting your system and their spatial coordinates. Confirm that "monodppc.pdb" has been successfully downloaded by listing the files contained in the current directory:

```
ls -lh
```

A pdb-file can be read using any simple text editor. Type, for instance:

```
kwrite monodppc.pdb
```

You should see a 2-lines head with some comments, then a line which starts with CRYST1 followed by a list of 6 numbers. These numbers define the size and shape of the unit cell. Then you will find a large list of rows with 10 fields each. The most relevant fields are those in columns 6, 7 and 8 which contain the coordinates of the atom indicated in the third column.

Take a look to the monolayer structure using a viewer like rasmol or PyMOL:

```
rasmol monodppc.pdb
```
or
```
pymol monodppc.pdb
```

Rasmol is more suitable for quick things while PyMOLis a more powerful molecular viewer with which you can take high quality images of molecules or molecular assemblies. We will return to it later. Within the rasmol environment type, in the command line:

```
set unitcell true
```

and zoom out the structure using the shift key and the left mouse button. You will see the unit cell, i.e. the recipient where the molecular assembly is contained. When you are convinced that you have a wide water layer with one DPPC monolayer per interface close the viewer environment.

4.3   System Topology

Next, we will perform the simulation of an adsorption isotherm... yes! the same you can measure with a tensiometer or a Langmuir balance. Why are we going to simulate something that we can measure? because the simulations provide much more information than

experiments. In addition to the surface-pressure vs area-per-molecule curves we will be able to get any structural, dynamic or energetic information of the same system at atomic level. In order to do that we need the elements listed in the Theory section of this document: (i) initial coordinates and velocities; (ii) topology and potential function; (iii) define simulation conditions; (iv) a MD engine; and (v) analysis tools. The initial coordinates are ready in the pdb-file and the velocities will be randomly assigned with the above-mentioned temperature restrictions. The analytical form of the potential function is quite standard as indicated in the section 3.2 of this document. Within this function, the bonded parameters are specific for each molecule and any mistake will be critical for the simulation. The generation of these parameters is not a matter for beginners but it is worth to take a look to the files containing this information. Download the corresponding files from the server:

```
wget web.usc.es/~fangel/lipid.itp
wget web.usc.es/~fangel/dppc.itp
wget web.usc.es/~fangel/topol.top
```

Open the files using your preferred text editor (perhaps kwrite?) and take a look to all the information in there. "lipid.itp" contains some atom-type definitions together with the Lennard-Jones parameters which define their mutual interaction, "dppc.itp" contains the bonded parameters specific for DPPC and "topol.top" is a shorter file including all the parameter files together with the number and type of molecules of the system. Next you need to define the simulation conditions.

4.4 Minimize the System

Before starting the MD simulation, the system should be minimized to avoid unrealistic interactions. This will be done using the steepest descent algorithm that slightly moves the atomic positions optimizing the interatomic interactions and the total potential energy of the system. For this you should create a text file (call it "em.mdp") with the following information:

title	= ENERGY MINIMIZATION
integrator	= steep
nsteps	= 500
constraints	= none
nstlist	= 1
ns-type	= Grid
pbc	= xyz
rlist	= 0.8
coulombtype	= cut-off
rcoulomb	= 1.4
vdw-type	= cut-off
rvdw	= 1.4
nstenergy	= 20

Next, a binary file will be generated combining the coordinates file "monodppc.pdb", the topology files "lipid.itp" and "dppc.itp" included in "topol.top", and the minimization pa-

rameters file "em.mdp". This is done using the following GROMACS command:

```
grompp -f em.mdp -p topol.top -c monodppc.pdb -o em.tpr
```

Read carefully the output of this program to see if there is any error. If not, you are ready to perform an energy minimization. It is better to do this in an independent folder:

```
mkdir Rmin
mv em.tpr Rmin
cd Rmin
mdrun -v -s em.tpr
```

The minimization should finish after 500 steps, as indicated in the "em.mdp" file. This should take as much a couple of minutes. As a result several files should be created, including "confout.gro" that contains the coordinates of all the atoms of the system -the same information that the pdb file. Now the minimization is finished and we can move back to the previous directory.

4.5   Production Simulation

The system is now prepared to run the actual MD simulation. The process is similar to the previous one, the only changes are the simulation parameters file and the initial coordinates file. Our aims are to reproduce an adsorption isotherm and to peform a basic structural characterization of the liquid-expanded and liquid-condensed phases. In order to get the adsorption isotherm, exactly in the same way as in an experiment, we need to perform a series of MD simulations, each at a given area per molecule. In a laboratory experiment the independent variable is usually the concentration. Some recent studies proved that the best way of obtaining an adsorption isotherm from MD simulations is to use the surface tension or the lateral pressure as an independent variable and to determine the corresponding area per molecule just dividing the area of the interface by the number of molecules, once the system reaches the equilibrium. For a system with two interfaces, the surface tension, γ, is determined from the pressure tensor as:

$$\gamma = (P_N - P_L)\frac{L}{2} \tag{5}$$

where P_N and P_L are the pressure components of the pressure tensor in the direction normal to the interface and in the plane of the interface, respectively. L is the size of the box in the normal dimension. For instance, if the interface is parallel to the XY plane:

$$P_N = P_{zz} \qquad P_L = (P_{xx} + P_{yy})/2 \qquad \text{and} \qquad L = L_z$$

GROMACS has specific algorithms already implemented to determine the surface tension from a constant area (or volume) MD simulation and, alternatively, to perform a MD simulation at a certain surface tension value allowing to change the interfacial area.

To perform a constant surface tension simulation, the following parameters (put them in a file called "run.mdp"!) will be employed:

```
constraints        = all-bonds
integrator         = md
dt                 = 0.002
nsteps             = 1000 ; 5000000 for 10 ns
nstcomm            = 1
nstxout            = 100
nstvout            = 100
nstfout            = 100
nstenergy          = 100
nstlist            = 5
ns_type            = grid
rlist              = 1.0
rcoulomb           = 1.0
vdwtype            = Cut-off
rvdw               = 1.4
fourierspacing     = 0.15
coulombtype        = PME
ewald_geometry     = 3dc
pme_order          = 4
ewald_rtol         = 1e-5
optimize_fft       = yes

Tcoupl             = berendsen
tau_t              = 0.1 0.1
tc-grps            = DPPC SOL
ref_t              = 298 298

Pcoupl             = berendsen
tau_p              = 2.0 2.0
compressibility    = 4.5e-5 0
ref_p              = 1000 1.0
pcoupltype         = surface-tension

gen_vel            = yes
gen_temp           = 298
gen_seed           = -1
```

where the surface tension value of the system (see the first value of the "ref_p" parameter) would be $1000\,bar\,nm = 100\,mN\,m^{-1}$, i. e. $50\,mN\,m^{-1}$ per interface. Using this parameter file, the Z dimension of the box will be constant and the volume variations will be due only to changes of the interface area. Change the "ref_p" value to perform a simulation with a different surface pressure. Try to choose always reasonable surface tension values or the simulation will be meaningless!

As in the previous case, the binary file needed to perform the simulation is created by combining the coordinates, topology and simulation parameters files:

```
grompp -f run.mdp -p topol.top -c Rmin/confout.gro -o run.tpr
```

and the simulation is ready to start:

```
mkdir Rrun
cd Rrun
cp ../run.tpr .
mdrun -v -s run.tpr
```

Now you can go back again to the previous directory.

4.6   Trajectory Analysis

With the simulation parameters file specified above you have obtained a too short trajectory to be analyzed. I have performed 30-ns-long MD simulations of this system at different surface tension values. Copy in your working directory one of these trajectories ("trajout.xtc" and "ener.edr" files) from the shared path (**ask to the teacher for this information**) and you will be ready to start the analysis.

A preliminary visual analysis is performed by extracting some protein conformations from the trajectories. Use the gromacs-tool "trjconv" to do that:

```
trjconv -f trajout.xtc -o 1ns.pdb -dump 1000 -s -pbc mol
trjconv -f trajout.xtc -o 30ns.pdb -dump 30000 -s -pbc mol
```

Use rasmol to compare those structures and, if you want, any other intermediate structure just indicating to the "trjconv" program the number of picoseconds of the selected frame after the "-dump" flag (remember to change also the output file after the "-o" flag).

Next you will take high quality pictures of the final structure with pymol. First open the pdb by typing "pymol 30ns.pdb". Then write the following commands in the pymol command line:

```
set orthoscopic, on
hide all
create water, resn SOL
create DPPC, not SOL
set sphere_scale=0.8
show spheres, DPPC
cmd.zoom("DPPC",animate=-1)
show surface, water
set transparency=0.5, water
bg_color white
color marine, water
util.cba(6,"DPPC")
```

```
ray 300, 300
png 30ns_top.png
turn x, -90
ray 150, 300
png 30ns_side.png
```

You have created two images in png format which can be used for your application, report, web page, thesis, presentation, article, screensaver, etc (see Figure below). Feel free to modify colors and/or representations to your taste.
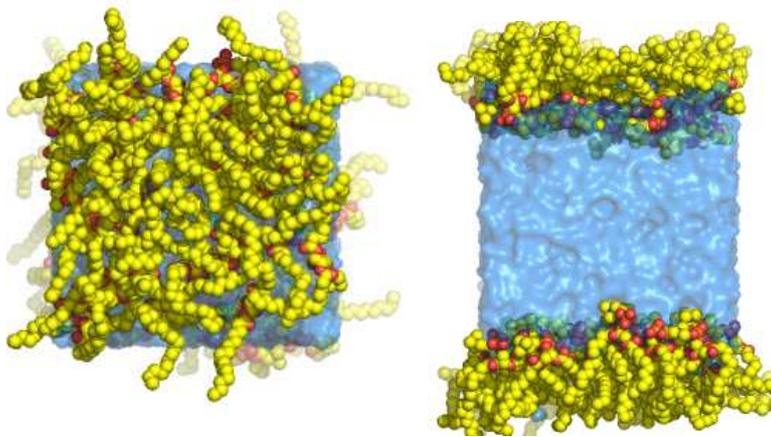

Figure 4: Dipalmitoyl Phosphatidyl Choline (DPPC) monolayer. Top and lateral views of a Liquid-Expanded phase (79.5 Å^2 per molecule and $65\,mN\,m^{-1}$).

Lets assume your simulation was performed at $XX\,mN\,m^{-1}$. Which is the corresponding area per molecule? One could just open the "30ns.pdb" file using a text editor and look at the box dimensions in the third line. Since the monolayer is parallel to the XY plane, the area of the interface expressed in Å^2 is just the product of the X and Y dimensions indicated in the pdb file. This simulation box contains 32 lipids in each of the two interfaces. So it is direct to determine the area per molecule for a given box. However, it would be interesting (actually necessary!) to know whether or not this area is stable. Since the pathway through which the simulation box reached those dimensions is stored in the trajectory files, that information should be available. In order to observe the box-size evolution type:

```
g_energy -f -o xbox.xvg
```

and choose the options 17, 18 and 19. Then press twice the key "enter" and you will see some statistical information in the screen about the X, Y and Z dimensions of the box. X and Y vary exactly in the same way while the Z box dimension is constant. It is even more interesting to follow any of these variables as a function of time. In order to do this we will employ the "xmgrace" program:

```
xmgrace xbox.xvg
```

This is a powerful plotting tool about which you can find many tutorials in internet. It is quite intuitive for simple things. You could try to modify the axis legends, etc, to your

taste. The most interesting here is to observe the box dimension changes as a function of time. Do you think the simulation time is long enough to determine the area per lipid for the selected surface tension value?
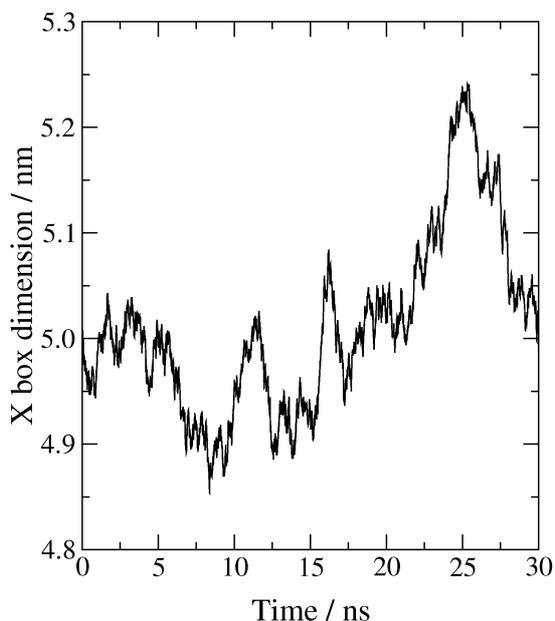

Figure 5: X box-dimension as a function of time for a DPPC monolayer MD simulation.

Note that the "g_energy" program can supply much information including a number of energetic contributions that could be interesting to monitor. Using the "-b" and "-e" flags you can perform the same analysis for a part of the trajectory (read the output of the "g_energy -h" command). Repeat the previous process for the 10 trajectories provided by the teacher and build the isotherm, i.e., plot the surface pressure values (water surface tension minus sample surface tension) versus the final area per molecule value for each simulation. Identify the liquid-expanded, liquid-condensed and equilibrium regions. Take pictures of representative conformations using pymol and think about further analysis you would like to do on this system.

It is worth to determine also the so-called order parameters, defined as:

$$S_z = \frac{3}{2} < \cos^2(\theta_z) > - \frac{1}{2} \tag{6}$$

where θ_z is the angle between the Z-axis of the simulation box and the molecular axis under consideration. The latter is defined as the vector from C_{n-1} to C_{n+1}. The brackets imply averaging over time and molecules. Order parameters can vary between 1 (full order along the interface normal) and -1/2 (full order perpendicular to the normal), with a value of zero in the case of isotropic orientation. In order to do this you will need two index files which you can download from the server:

15

```
wget web.usc.es/~fangel/cola1.ndx
wget web.usc.es/~fangel/cola2.ndx
```

and then type:

```
g_order -f 30ns.pdb -s topol.tpr -n cola1.ndx -o cola1.xvg -od cola1od.xvg
g_order -f 30ns.pdb -s topol.tpr -n cola2.ndx -o cola2.xvg -od cola2od.xvg
```

You can see the results using xmgrace:
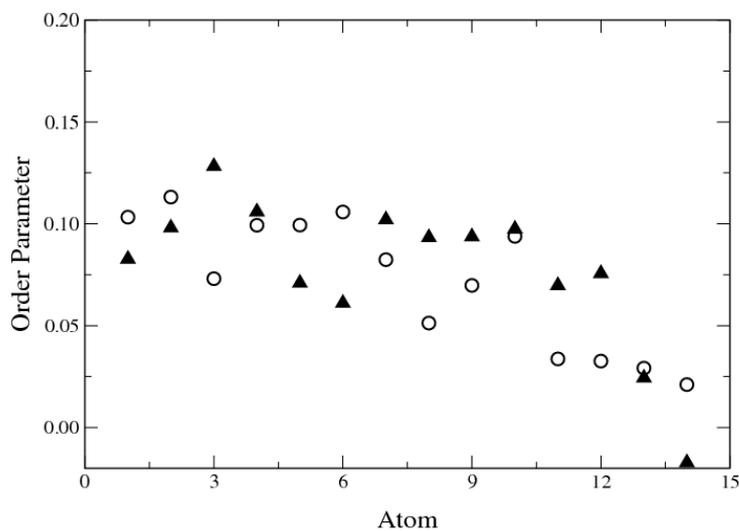
```
xmgrace cola1od.xvg cola2od.xvg
```


Figure 6: Order parameters for the aliphatic carbon atoms of each of the DPPC palmitoyl chains for the last frame of the same simulation shown in the previous two figures.

It would be interesting to compare the order parameters for the lipids in the liquid-extended against those in the liquid-condensed phase.

To finish, lets determine how the molecules distribute along the Z axis. This will be done with the "g_density" GROMACS program:

```
echo 1 | g_density -f 30ns.pdb -s ../../MD/kk/topol.tpr -o ddppc.xvg
echo 2 | g_density -f 30ns.pdb -s ../../MD/kk/topol.tpr -o dwater.xvg
```
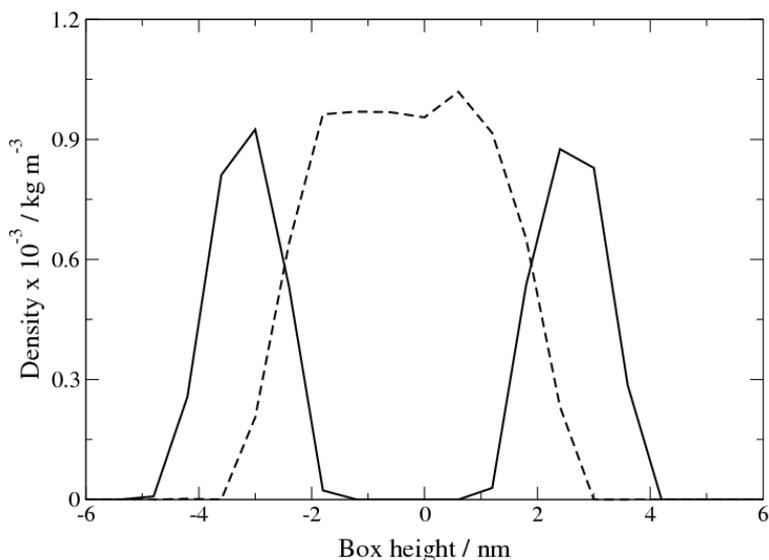
which, for the case that it is being analyzed, gives:

Figure 7: Density of DPPC and water molecules along the Z-axis of the simulation box. A value of z = 0 is asigned to the center of the box.

5 References

- Ángel Piñeiro, Norma Díaz-Vergara, Eduardo Jardón-Valadez, Aned de Leon and Sergio Rodríguez-Morales. Applications of Molecular Dynamics Simulations to Protein Systems. In Advances in Protein Physical Chemistry, 2008: ISBN: 978-81-7895-324-3.

- D. van der Spoel, E. Lindahl, B. Hess, A. R. van Buuren, E. Apol, P. J. Meulenhoff, D. P. Tieleman, A. L. T. M. Sijbers, K. A. Feenstra, R. van Drunen and H. J. C. Berendsen, Gromacs User Manual version 3.3 (2005). www.gromacs.org.

- David van der Spoel. Gromacs exercises (2004). http://www.csc.fi/english/research/sciences/chemistry/courses/gmx2004/exercises/index_html.

- Erick Lindahl. Pre-Workshop Hands-On Tutorial. Gromacs Workshop 2007 CSC, Espoo/Helsinki, Finland. https://extras.csc.fi/chem/courses/gmx2007/tutorial1/tutorial1.pdf.

- Susan L. Duncan, Ronald G. Larson. Comparing Experimental and Simulated Pressure-Area Isotherms for DPPC. Biophysical Journal, 2008 (94) 2965-2986.

- Egbert Egberts, Siewert-Jan Marrink, Herman J. C. Berendsen. Molecular Dynamics Simulation of a Phospholipid Membrane. European Biophysics Journal, 1994 (22) 423-436.

- Svetlana Baoukina, Luca Monticelli, Siewert J. Marrink, D. Peter Tieleman. Pressure-Area Isotherm of a Lipid Monolayer from Molecular Dynamics Simulations. Langmuir, 2007 (23) 12617-12623.

- Siewert J. Marrink, Erik Lindahl, Olle Edholm, Alan E. Mark. Simulation of the Spontaneous Aggregation of Phospholipids into Bilayers. Journal of the American Chemical Society, 2001 (123) 8638-8639.

- Peter J. Bond, Jonathan M. Cuthbertson, Sundeep S. Deol, Mark S. P. Sansom. MD Simulations of Spontaneous Membrane Protein/Detergent Micelle Formation. Journal of the American Chemical Society, 2004 (126) 15948-15949.

- Jorge Hernández-Pascacio, Cristina Garza, Xavier Banquy, Norma Díaz-Vergara, Alfredo Amigo, Salvador Ramos, Rolando Castillo, Miguel Costas, Ángel Piñeiro. Cyclodextrin-Based Self-Assembled Nanotubes at the Water/Air Interface. The Journal of Physical Chemistry B, 2007 (111) 12625-12630.

- Norma Díaz-Vergara, Ángel Piñeiro. Molecular Dynamics Study of Triosephosphate Isomerase from Trypanosoma cruzi in Water/Decane Mixtures. The Journal of Physical Chemistry B, 2008 (112) 3529-3539.

- Ángel Piñeiro, Gerardo Prieto, Juan M. Ruso, Pedro V. Verdes, Félix Sarmiento. Surface films of short fluorocarbon-hydrocarbon diblocks studied by molecular dynamics simulations: Spontaneous formation of elongated hemimicelles. Journal of Colloid and Interface Science, 2009 (329) 351-356.

- Elena Blanco, Ángel Piñeiro, Reinhard Miller, Juan M. Ruso, Gerardo Prieto, Félix Sarmiento. Langmuir Monolayers of a Hydrogenated/Fluorinated Catanionic Surfactant: From the Macroscopic to the Nanoscopic Size Scale. Langmuir, 2009 (in press).

- Siewert J. Marrink, Alan E. Mark. The Mechanism of Vesicle Fusion as Revealed by Molecular Dynamics Simulations. Journal of the American Chemical Society, 2003 (125) 11144-11145.

- Alex H. de Vries, Alan E. Mark, Siewert J. Marrink. Molecular Dynamics Simulation of the Spontaneous Formation of a Small DPPC Vesicle in Water in Atomistic Detail. Journal of the American Chemical Society, 2004 (126) 4488-4489.