



Πανεπιστήμιο Δυτικής Μακεδονίας  
Τμήμα Μηχανολόγων Μηχανικών

Διπλωματική Εργασία

Υπολογιστική Μοντελοποίηση Μοριακής Δυναμικής  
Ενός Μονοστρώματος Λιπιδίων

Η Περίπτωση Του Επιφανειοδραστικού Μονοστρώματος  
Μιας Πνευμονικής Κυψελίδας

Μιχαήλ Παλαιοκώστας Αβραμίδης  
Α.Ε.Μ.: 946

Επιβλέπων Καθηγητής:  
Δρ. Αντώνιος Τουρλιδάκης

Κοζάνη, Σεπτέμβριος 2013

Πανεπιστήμιο Δυτικής Μακεδονίας  
Τμήμα Μηχανολόγων Μηχανικών

Διπλωματική Εργασία

Η εργασία αυτή πραγματοποιήθηκε στα πλαίσια του Ευρωπαϊκού Προγράμματος Κινητικότητας Erasmus και του μεταπτυχιακού προγράμματος MSc. in Computational Fluid Dynamics σε συνεργασία με το Cranfield University στο Ηνωμένο Βασίλειο. Το Cranfield University διατηρεί τα πνευματικά δικαιώματα της εργασίας αυτής, όπως αναγράφεται παρακάτω.

Υπολογιστική Μοντελοποίηση Μοριακής Δυναμικής  
Ενός Μονοστρώματος Λιπιδίων

Η Περίπτωση Του Επιφανειοδραστικού Μονοστρώματος  
Μιας Πνευμονικής Κυψελίδας

Μιχαήλ Παλαιοκώστας Αβραμίδης  
A.E.M.: 946

Επιβλέπων Καθηγητής για το Πανεπιστήμιο Δυτικής Μακεδονίας:  
Δρ. Αντώνιος Τουρλιδάκης

Επιβλέποντες Καθηγητές για το Cranfield University:  
Prof. Dimitris Drikakis  
Dr. Sandip Pal  
Dr. Nikolaos Asproulis

Κοζάνη, Σεπτέμβριος 2013

©Cranfield University, 2013.

All rights reserved. No part of this publication may be reproduced  
without the written permission of the copyright holder.



Στοχάσου.  
Πόσον κράτησε η κακιά στιγμή;  
Πόσον το θαύμα;  
- Μάρκος Μεσκός



## Περίληψη

Το Πνευμονικό Επιφανειοδραστικό Μονόστρωμα (Alveolar Surfactant Monolayer) που βρίσκεται στις πνευμονικές κυψελίδες (Alveoli), είναι ένα πολύ σημαντικό συστατικό για την διαδικασία της αναπνοής και του οποίου η απουσία είναι υπεύθυνη για τον θάνατο του 1% των πρόωρα γεννημένων νεογνών. Καθώς το μονόστρωμα συμμετέχει σε κάθε διαδικασία ανταλλαγής αερίων στους πνεύμονες, η κατανόηση της συμπεριφοράς του είναι ιδιαίτερης σημασίας και στην παρούσα εργασία η διάχυση του Οξυγόνου καθώς το διασχίζει, επιχειρήθηκε να μετρηθεί.

Προκειμένου να μοντελοποιηθεί το μονόστρωμα, έγινε χρήση της μεθόδου της Μοριακής Δυναμικής. Δύο διαφορετικές τοπολογίες (ένα μονόστρωμα και το συμμετρικό του) δημιουργήθηκαν και δοκιμάστηκαν για διαφορετικές "Επιφάνειες Προς Λιπίδια" (Area Per Lipids) και Θερμοστάτες (Thermostats). Όλα τα συστήματα εξετάστηκαν προσεκτικά υπό το πρίσμα πέντε σημαντικών παραμέτρων (Θερμοκρασίας, Κατασκευαστικής συνοχής, Πυκνότητας, Πλευρικής Διάχυσης και Υπολογιστικού Κόστους).

Για τη μέτρηση του συντελεστή διάχυσης του Οξυγόνου, η τεχνική Umbrella Sampling χρησιμοποιήθηκε και έπειτα από έναν μεγάλο αριθμό προσομοιώσεων η μέθοδος ανάλυσης των Weighted Histograms εφαρμόστηκε προκειμένου να εξαχθεί το διάγραμμα ελεύθερης ενέργειας. Με βάση αυτό, αναλύθηκε η συμπεριφορά του οξυγόνου και μετρήθηκε ο συντελεστής διάχυσης.

Εν κατακλείδι, η διαφορά της Ελεύθερης Ενέργειας της διαδικασίας διάχυσης μετρήθηκε ως 16.674 [kJ/mol] και ο συντελεστής ως  $D_{O_2} = 1.557 \cdot 10^{-5} [cm^2/s]$  ο οποίος ήταν μικρότερος από ότι σε αντίστοιχες μελέτες που αφορούσαν διπλοστρώματα-μεμβράνες (bilayers).

---

## Abstract

Alveolar Surfactant plays a major role in the respiration process with its absence being responsible for 1% of prematurely-born infants mortality. As the surfactant is involved in every process of gas exchange in the lungs, the understanding of its working mechanism is of great importance and in this thesis, the diffusion of oxygen during the respiration process was tried to be measured.

In order to model the alveolar surfactant the Molecular Dynamics method was used; two different topologies (a physical monolayer and its double mirror) were created and were tested under different Areas per lipid (50,60,70 [ $\text{\AA}^2$ ]) and for two temperature coupling methods (Nose-Hoover and V-Rescale). All the systems were tested carefully and compared in five important parameters; Temperature Convergence, Deuterium Order Parameters, Density Profiles, Lateral Diffusion and Simulation Cost.

For the diffusion coefficient of Oxygen through the monolayer, the Umbrella Sampling Technique was used and after a large number of biased simulations, the Weighted Histogram Analysis Method was utilized to extract the unbiased Free-Energy Diagram. Based on these results, the behaviour of the Oxygen through the monolayers was predicted and the Diffusion Coefficient was calculated.

Overall, the Free-Energy Difference of the diffusion process was measured to be 16.674 [kJ/mol] and Diffusion coefficient was  $D_{O_2} = 1.557 \cdot 10^{-5} [cm^2/s]$  which was smaller in comparison to studies of bilayers, possible because of the more chaotic movement of lipid chains due to the existence of vacuum and therefore no density in their tips.

## Acknowledgements

This thesis for me was not a simple dissertation project but rather an initiation to real research. It taught me how hard is to produce even the simplest of results, how intimidating is swimming in unknown waters and how revealing is the far sight of the coast. And after two and a half months of exhausting work, I can finally say that I survived. And this adventure taught me a lot and made me better. This however would have not been possible without the assistance and support of some people.

First of all I would like to thank my supervisors Prof. Dimitris Drikakis, Dr. Sandip Pal and Dr. Nikolaos Asproulis for the opportunity that they gave me to discover an exciting (and brave) new world, far beyond the continuum assumption. Their assistance all these three months was invaluable and only by their guidance and advices I managed to complete successfully this work.

Furthermore, since this Thesis is part of my double degree program, I would like to sincerely thank my supervisor in University of Western Macedonia Dr. Antonios Tournlidakis because without him I would have never come to Cranfield University. Also, his teaching of Fluid Mechanics was the reason that I chose to continue my studies in Computational Fluid Dynamics.

Also, I would like to thank my friends, old and new, because they made me who I am and because they made this difficult year a bit easier with their company. Most of my gratitude however goes to my parents Olga and Giannis, my sister Katerina and my love Artemis for all their support, love and understanding all this time, even at its most difficult. I wouldn't have made it without you!





# Contents

Περίληψη	v
Acknowledgements	vii
Nomenclature	xvii
<b>1 Introduction &amp; Motivation</b>	<b>1</b>
1.1 Motivation . . . . .	3
1.2 Thesis Aims & Objectives . . . . .	4
1.3 Thesis Overview . . . . .	6
<b>2 Physiology and Diffusion Mechanisms</b>	<b>9</b>
2.1 Physiology of Pulmonary Surfactant . . . . .	11
2.2 Controversy and Alternative Models . . . . .	11
2.2.1 Physiology vs Physics . . . . .	11
2.2.2 Model of Scarpelli . . . . .	14
2.2.3 Model of Hills . . . . .	15
2.2.4 Model of Bangham . . . . .	15
2.2.5 Model of Kashchiev - Exerowa . . . . .	16
2.2.6 Model of Kitaoka . . . . .	16
2.3 Diffusion in Lipid Monolayers . . . . .	17
<b>3 Molecular Dynamics Theory</b>	<b>19</b>

---

3.1	Modelling beyond the Continuum . . . . .	21
3.2	Historic Evolution of Molecular Dynamics . . . . .	23
3.3	Main Idea of Simulation Process . . . . .	24
3.3.1	Simulation Assumptions . . . . .	24
3.3.2	Limitations of the Method . . . . .	26
3.4	Examining the Simulation Parameters . . . . .	27
3.4.1	Force Field Types . . . . .	27
3.4.2	Potential Functions . . . . .	28
3.4.3	Integration of Equations of Motion . . . . .	31
3.4.4	Ensembles . . . . .	34
3.4.5	Temperature Coupling . . . . .	35
3.4.6	Pressure Coupling . . . . .	36
3.4.7	Boundary Conditions . . . . .	37
<b>4</b>	<b>Molecular Dynamics for Lipid Monolayers</b>	<b>39</b>
4.1	Research on Lipid Monolayers . . . . .	41
4.1.1	Lipid Molecules . . . . .	41
4.1.2	MD Simulations for Alveolar Surfactant . . . . .	43
4.2	Molecular Dynamics Parameters . . . . .	45
4.2.1	Time and Length Scales . . . . .	45
4.2.2	Force Fields . . . . .	46
4.2.3	Ensembles for Interfaces . . . . .	47
4.2.4	Temperature and Pressure Control . . . . .	48
4.2.5	Electrostatic Interactions and Cut-off . . . . .	49
4.3	Studies on $O_2$ diffusion through the surfactant . . . . .	50
4.4	Validation of Simulations . . . . .	51
4.4.1	Model Validation . . . . .	51
4.4.2	Diffusion Coefficient Validation . . . . .	53
<b>5</b>	<b>Simulation Approach and Analysis</b>	<b>55</b>
5.1	Tools and Resources . . . . .	57
5.1.1	Software . . . . .	57
5.1.2	Hardware . . . . .	58

---

5.2	Topologies and Force Fields Preparation . . . . .	58
5.2.1	CHARMM36 Installation . . . . .	58
5.2.2	Monolayer and Symmetry Topologies . . . . .	58
5.2.3	Domain of Solution . . . . .	60
5.2.4	Oxygen Molecule . . . . .	61
5.3	Simulation Parameters . . . . .	62
5.3.1	General Simulation Parameters . . . . .	62
5.3.2	Thermostat for Temperature Coupling . . . . .	63
5.3.3	Area per Lipid . . . . .	63
5.4	Simulation Process . . . . .	64
5.4.1	Construction of the Lipid Monolayer . . . . .	64
5.4.2	Calculation of the Diffusion Coefficient . . . . .	69
<b>6</b>	<b>Simulation Results and Discussion</b>	<b>73</b>
6.1	Surfactant Models . . . . .	75
6.2	Energy Minimization . . . . .	78
6.3	NVT Equilibration . . . . .	80
6.3.1	Temperature Convergence . . . . .	82
6.3.2	Deuterium Order Parameters . . . . .	86
6.3.3	Density Profiles . . . . .	91
6.3.4	Mean Square Displacement . . . . .	95
6.3.5	Computational Cost and Effectiveness . . . . .	99
6.3.6	First Stage Conclusions . . . . .	100
6.4	Umbrella Sampling Simulations . . . . .	100
6.4.1	Effect of Harmonic Force Constant . . . . .	101
6.4.2	Effect of Sampling Space . . . . .	102
6.4.3	Calculation of PMF . . . . .	105
6.4.4	Calculation of $D_{O_2}$ . . . . .	107
<b>7</b>	<b>Conclusions and Future Work</b>	<b>109</b>
7.1	Conclusions . . . . .	111
7.1.1	Model Creation and Comparison . . . . .	111
7.1.2	Energy Minimization . . . . .	112

---

7.1.3	NVT Equilibration . . . . .	112
7.1.4	Diffusion Coefficient . . . . .	114
7.2	Future Work . . . . .	116
<b>Bibliography</b>		<b>117</b>
<b>A Pre-processing Scripts</b>		<b>129</b>
A.1	Creation of Monolayer Topology . . . . .	130
A.2	Creation Symmetry Topology . . . . .	131
A.3	Geometrical Parameters for Monolayer . . . . .	132
A.4	Geometrical Parameters for Symmetry . . . . .	136
A.5	Calculation of Umbrella Sampling Distances . . . . .	141
A.6	Translation of relative distances . . . . .	142
<b>B GROMACS Parameters Files</b>		<b>143</b>
B.1	Energy Minimization . . . . .	144
B.2	NVT Equilibration . . . . .	146
B.3	Steered MD . . . . .	149
B.4	Umbrella Simulation . . . . .	152
B.5	Required Files . . . . .	155
<b>C Post-Processing Codes</b>		<b>163</b>
C.1	Energy Minimization Error Analysis . . . . .	164
C.2	Temperature Convergence Error Analysis . . . . .	167
C.3	Deuterium Order Parameters Error Analysis . . . . .	172
C.4	MSD Post-Processing . . . . .	177
C.5	PMF and Diffusion Coefficient Calculation . . . . .	179

## List of Tables

4.1	Force Fields used on Lipid Monolayers Simulations . . . . .	47
4.2	Pressure-Temperature Coupling Techniques used in Lipid Monolayers Simulations . . . . .	48
4.3	Similar Researches to provide Deuterium Order Parameters . . . . .	52
4.4	Diffusion coefficients of $O_2$ in water . . . . .	53
4.5	Diffusion coefficients of $O_2$ in lipid bilayers . . . . .	53
5.1	Wall Properties . . . . .	62
5.2	Validation Simulations . . . . .	62
6.1	Number of atoms for each model . . . . .	78
6.2	Error Analysis of Temperature Convergence for $50\text{\AA}^2$ . . . . .	84
6.3	Error Analysis of Temperature Convergence for $60\text{\AA}^2$ . . . . .	84
6.4	Error Analysis of Temperature Convergence for $70\text{\AA}^2$ . . . . .	84
6.5	Error Analysis of Deuterium Order Parameters for the $50\text{\AA}^2$ APL . . . . .	88
6.6	Error Analysis of Deuterium Order Parameters for the $60\text{\AA}^2$ APL . . . . .	90
6.7	Error Analysis of Deuterium Order Parameters for the $70\text{\AA}^2$ APL . . . . .	90
6.8	Lateral Diffusion Coefficient of Phosphorus for $50\text{\AA}^2$ APL . . . . .	95
6.9	Lateral Diffusion Coefficient of Phosphorus for $60\text{\AA}^2$ APL . . . . .	98
6.10	Lateral Diffusion Coefficient of Phosphorus for $70\text{\AA}^2$ APL . . . . .	98
6.11	Required Simulation Time for each system examined . . . . .	99
6.12	Number of Simulations ran for each spacing of umbrella sampling . . . . .	104



## List of Figures

1.1	Infant with Respiratory Distress Syndrome . . . . .	4
1.2	Project Motivation . . . . .	4
2.1	A Typical Surfactant Molecule . . . . .	12
2.2	The Complete Respiratory System . . . . .	12
2.3	The Alveolar Unit . . . . .	13
2.4	Alveolus Topology . . . . .	13
2.5	Scarpelli's and Conventional Model . . . . .	14
2.6	Alveolar Surface Network proposed by Scarpelli . . . . .	14
2.7	Hills and Conventional Model . . . . .	15
2.8	Bangham and Conventional Model . . . . .	15
2.9	Exerowa Model . . . . .	16
2.10	Kitakoa Computational Model . . . . .	17
2.11	Lateral Diffusion Schematic . . . . .	17
3.1	Flow Regimes . . . . .	22
3.2	Molecular Dynamics Algorithm . . . . .	25
3.3	Types of intramolecular connections . . . . .	26
3.4	The Hard-Sphere Potential . . . . .	29
3.5	The Square-Well Potential . . . . .	29
3.6	The Lennard-Jones 12-6 Potential . . . . .	30
3.7	The Weeks-Chandler-Andersen Potential . . . . .	30
3.8	The Buckingham Potential . . . . .	31
3.9	The Coulombic Potential . . . . .	31
3.10	Thermodynamics Ensembles commonly used in MD . . . . .	34
3.11	Periodic Boundary Conditions . . . . .	38



4.1	DPPC Molecule Structure . . . . .	42
4.2	Phase Changes on the Surface Pressure-Area Isotherm . . . . .	43
4.3	DPPC Molecule Topology . . . . .	44
5.1	Topology Generation for each case . . . . .	59
5.2	Domains of Solution . . . . .	60
5.3	Oxygen Molecule . . . . .	61
5.4	Stage 1 Algorithm . . . . .	65
5.5	Stage 2 Algorithm . . . . .	65
5.6	Pulling simulations Schematic . . . . .	71
6.1	The final Models after 1ns of NVT Equilibration . . . . .	76
6.2	Timelapse of 1ns for both models . . . . .	77
6.3	Specific Potential Energy during Energy Minimization . . . . .	81
6.4	Normalized Specific Potential Energy during Energy Minimization . . . . .	81
6.5	Temperature Convergence for 50Å <sup>2</sup> APL . . . . .	84
6.6	Temperature Convergence for 60Å <sup>2</sup> APL . . . . .	85
6.7	Temperature Convergence for 70Å <sup>2</sup> APL . . . . .	85
6.8	Deuterium Order Parameters $S_{CD}$ for 50Å <sup>2</sup> APL . . . . .	88
6.9	Deuterium Order Parameters $S_{CD}$ for 60Å <sup>2</sup> APL . . . . .	89
6.10	Deuterium Order Parameters $S_{CD}$ for 70Å <sup>2</sup> APL . . . . .	90
6.11	Mass Density Profile for the 50Å <sup>2</sup> APL . . . . .	92
6.12	Electron Density Profile for the 50Å <sup>2</sup> APL . . . . .	92
6.13	Mass Density Profile for the 60Å <sup>2</sup> APL . . . . .	93
6.14	Electron Density Profile for the 60Å <sup>2</sup> APL . . . . .	93
6.15	Mass Density Profile for the 70Å <sup>2</sup> APL . . . . .	94
6.16	Electron Density Profile for the 70Å <sup>2</sup> APL . . . . .	94
6.17	MSD of Phosphorus atoms for 50Å <sup>2</sup> APL . . . . .	96
6.18	MSD of Phosphorus atoms for 60Å <sup>2</sup> APL . . . . .	96
6.19	MSD of Phosphorus atoms for 70Å <sup>2</sup> APL . . . . .	97
6.20	Comparative Histogram analysis for $k_h=1000$ and $k_h=3000$ . . . . .	101
6.21	Histogram analysis for $k_h=1000$ and $d=0.4,0.2,0.1\text{nm}$ . . . . .	102
6.22	Histogram analysis for $k_h=1000$ and $d=0.05\text{nm}$ . . . . .	103
6.23	Histogram analysis for $k_h=1000$ and $d=0.025\text{nm}$ . . . . .	103
6.24	Potential of Mean Force for different spacings . . . . .	104
6.25	Potential of Mean Force for $d=0.025\text{nm}$ . . . . .	106

## Nomenclature

### Latin Symbols

$\theta_i$	Angle Between Methyl Chain and Monolayer Normal
$a$	Acceleration
$b^0$	Equilibrium constant for Bonds
$D$	Diffusion Coefficient
$d$	Travelling distance $d$ of the diffusing particle
$d_{ch}$	Characteristic Diameter
$D_{lat}$	Lateral Diffusion Coefficient
$D_{O_2}$	Diffusion Coefficient of Oxygen
$E_P$	Potential Energy
$e_P$	Specific Potential Energy
$e_{abs}$	Absolute error
$e_{rel}$	Relative error
$F$	Applied Force
$f$	Friction Coefficient
$F_{fr}$	Frictional Force
$k(t)$	Random Force due to Solvent
$k^b, k^\theta, k^\phi$	Force Constants for Bonds, Angles and Dihedrals
$k_B$	Boltzmann's Constant

$k_B$	Boltzmann's constant
$k_h$	Force Constant of Harmonic Potential
$L_\infty$	Infinity Norm
$L_\infty$	The L infinity Norm
$L_{ch}$	Characteristic Length of the System
$M$	Total Mass of the system
$MW_{lipids}$	Molecular weight of DPPC
$MW_{water}$	Molecular weight of Water
$N_A$	Avogadro's Number
$n_{dih}$	Dihedral multiplicity
$N_{lipids}$	Number of DPPC Atoms
$N_{water}$	Number of Water Atoms
$P$	Pressure
$P_0$	Reference Pressure
$q_i$	Partial Charge of Particle i
$Q_M$	Mass Parameter of the System
$r$	Position
$r_c$	Truncation Distance
$r_{curv}$	Radius of Curvature
$r_{ij}$	Inter-particle Distance
$S_{CD}$	Deuterium Order Parameter
$S_{ij}$	Generic Order Parameter Tensor
$T$	Temperature
$t$	Time or Time Instance
$T_0$	Reference Temperature
$U(t)$	Harmonic Potential for Umbrella Sampling
$V$	System Potential
$V_{LJ}$	Lennard-Jones Potential

---

$W$	Work Required for a Particle to Move
$Kn$	Knudsen Number
$N$	Number of Particles
$v$	Velocity

**Greek Symbols**

$\delta_{ij}$	Kronecker's Delta
$\Delta G$	Free-Energy Difference
$\Delta P_{int}$	Pressure Difference of Interface
$\epsilon_0$	Electrical Permittivity of Space
$\epsilon$	Strength of Particle interactions
$\gamma$	Surface Tension
$\lambda$	Molecular Free-Mean Path
$\phi^0$	Equilibrium constant for Dihedrals
$\sigma$	Particle interaction Radius
$\tau_p$	Pressure Relaxation Factor
$\tau$	Time Relaxation Factor
$\theta^0$	Equilibrium constant for Angles
$\xi$	Frictional Term

**Abbreviations**

APL	Area Per Lipid
ASL	Alveolar Surface Liquid
ASN	Alveolar Surface Network
CFD	Computational Fluid Dynamics
CPAP	Continuous Positive Airway Pressure
DPPE	Dipalmitoylphosphatidylcholine
FD	Finite Differences
Fig.	Figure
GROMACS	GRONingen MACHine for Chemical Simulations

HMD	Hyaline Membrane Disease
HPC	High Performance Computer
IRDS	Infant Respiratory Distress Syndrome
MC	Monte Carlo
MD	Molecular Dynamics
MEMS	Micro-Electro-Mechanical-Systems
MSD	Mean Square Displacement
MTTK	Martyna-Tuckerman-Tobias-Klein Barostat
NH	Nose-Hoover Thermostat
NMR	Nuclear Magnetic Resonance
NPT	Isothermal-Isobaric Ensemble
NVE	Micro-Canonical Ensemble
NVT	Canonical Ensemble
PBC	Periodic Boundary Conditions
PC	Phosphatidylcholine
PEM	Particle Ewald Mesh
PG	Phosphatidylglycerol
PMF	Potential of Mean Force
RMSD	Root Mean Square Deviation
VMD	Visual Molecular Dynamics
VR or Vres	Velocity Rescale Thermostat
WHAM	Weighted Histogram Analysis Method

**Introduction & Motivation**

**Contents**

---

<b>1.1</b>	<b>Motivation . . . . .</b>	<b>3</b>
<b>1.2</b>	<b>Thesis Aims &amp; Objectives . . . . .</b>	<b>4</b>
<b>1.3</b>	<b>Thesis Overview . . . . .</b>	<b>6</b>

---



## Chapter Summary

Alveolar Surfactant is a substance covering the alveoli of lungs with a very thin layer and is composed by lipids, proteins and other minor components. The lack of surfactant is responsible for the Infant Respiratory Distress Syndrome (IRDS) which is developed in infants that are prematurely born. The syndrome, also known as, neonatal respiratory distress syndrome or respiratory distress syndrome of newborn or Hyaline Membrane Disease (HMD) is the main reason of death for prematurely born infants, affecting approximately 1% of newborn babies [1]. In this chapter the motivation to examine the diffusion through the surfactant is explained, the main objectives of the thesis are presented and in the end, the thesis structure is analysed.

### 1.1 Motivation

Scientific community has developed until today, three ways of dealing with IRDS. The first method is preventive where in case a mother is expected to give birth prematurely then is given glucocorticoids hormones which increase the production of surfactant and in most cases prevent or at least decrease IRDS effects. After the infant is born, there are two ways to treat the syndrome.

*Continuous Positive Airway Pressure (CPAP)* provides oxygen to the infant through constant positive pressure difference, while in the worst cases the use of endotracheal tube is necessary to mechanically support breathing. Also, animal (usually extracted from cow) or synthetic surfactant is given through the tube which according to Schwartz et al. [2] has decreased low-weight infant's death by 30%.

An alternative but more complex method is *Extracorporeal membrane oxygenation* in which the gas exchange process of the lungs is imitated mechanically outside the body [3]. However several limitation parameters should be taken into account before it is used in infants and this is the reason it is not yet widespread.

It is therefore clear, that the complete understanding of the surfactant function as part of the alveolus system is of great importance not only for engineers, scientists and doctors that produce synthetic surfactants but also for respiratory therapists who apply treatments to newborn children.





Figure 1.1: Infant with Respiratory Distress Syndrome.



Figure 1.2: Project Motivation.

## 1.2 Thesis Aims & Objectives

The aim of the Thesis is to help improve the understanding of oxygenation process in the lung's air/water interface inside each alveolus. Potentially, this could lead to better drug delivery methods and treating approaches of respiratory problems.

A way to examine oxygenation is by examining the diffusion process of Oxygen through the alveolar monolayer and to observe how it is affected by the lipid molecules so **the most important aim of this Thesis is to calculate the Diffusion Coefficient of the Oxygen through the monolayer and observe its performance.**

Another aim is to explore further, the use of Molecular Dynamics as a method of modelling biological systems and to **provide extra feedback in the scientific community such as optimum topologies, force field adequacy, Thermostats performance and structural analysis data.** Since Molecular Dynamics is a relatively new method of modelling, the studies of alveolar surfactant are quite limited and the effects of various simulation parameters still under investigation.

Ultimately, this Thesis, aims in the **development of a consistent simulation methodology of researching oxygen diffusion** through the surfactant which can be used from future researches to continue the exploration of the surfactant system.

The above-mentioned aims will be achieved by a series of objectives which can be summarized to:

1. **Literature Review.** Before the initiation of the actual modelling, several things have to be determined both for the physical properties but also for the numerical parameters. Therefore an extended literature review will be performed in order to prepare sufficiently to approach computationally the system:

- (a) The alveolar surfactant is part of the greater system of alveolus which is the final step of the respiratory branch. Its form however is still under debate and many scientists disagree with the classic physiology. For this reason, the classical and the alternative alveolar forms will be examined in order to determine the possible effects they might have on our computational model.
- (b) Molecular Dynamics is an exceptional tool to examine systems in the molecular level. In order to correctly utilize it however, the basic operating parameters have to be examined thoroughly and understood fully.
- (c) How Molecular Dynamics are applied to lipid systems and especially lipid monolayers such as the alveolar surfactant is another important factor that has to be researched. All the parameters that were used in past papers will be researched, such as:
  - i. The topology of the model in comparison to the physical form.
  - ii. The parameters set-up such as numerical integration of the governing equations, Thermodynamic ensembles, boundary conditions, force fields, number of lipids, area per lipid, domain size etc.
- (d) Furthermore, studies of Diffusion of Oxygen through alveolar surfactant or other lipid monolayers will be researched to examine the tools and methodology that were used. It will also be ensured that not exactly same work as in this Thesis has been performed in the past.
- (e) All the above mentioned parameters will be also used as validation data for the produced results of our model.
- (f) Finally, the optimum simulation software for our case will be searched in order to ensure the most accurate and efficient results will be produced.

2. **Topology Creation.** After the literature review the system will be attempted to be modelled based on what will be found:

- (a) In general not many free-to-use monolayer models exist in contrast to bilayers and probably a model of the latter will be used to construct the monolayer.

- (b) The domain of solution will be defined based on the topology and the important parameters that will be found.
  - (c) The oxygen molecule has to be constructed and imported in the system so the appropriate way has also to be defined.
3. **Results Evaluation.** The evaluation of the produced results will happen in two stages:
- (a) The created models will be examined for their structural integrity and their general correct behaviour during simulation as well as their total computational cost. For this, validation data found during the literature review will be used. Based on the evaluation of the models, the most appropriate will be used for the calculation of the diffusion coefficient.
  - (b) The diffusion coefficient will be calculated according to the most reliable methods found in the literature. Also the behaviour of the oxygen molecule as it penetrates the monolayer will be examined.

### 1.3 Thesis Overview

In Chapter 2, an introduction to the classical physiology of the alveoli region is made and the alveolus surfactant position is presented. Then, the different models of the alveolus topology are presented as well as the still open controversies between the physiology and physics scientific communities. Finally, the Diffusion Mechanisms that most commonly appear in the biological systems are presented.

Chapter 3 introduces the reader to the fundamental concepts behind the Molecular Dynamics as a tool to simulate fluid systems where the Continuum hypothesis is invalid. Initially, an introduction is made to the various modelling methods of fluid systems from the classic Computational Fluid Dynamics to the Molecular Dynamics. Then is presented a brief historic review of the key-points of MD evolution since the development of the first Computer Systems. Also the main idea behind MD, the fundamental physical assumptions on which it is based and of course its limitations are discussed. Finally, the rest of the chapter is dedicated entirely to the most important aspects of MD, such as the various force fields, the potential functions commonly used, the numerical methods of integrating the equations of motion, the thermodynamic ensembles, the temperature and pressure coupling techniques and finally the boundary conditions.

In Chapter 4, the Literature Review of the Thesis is presented. At first, the lipid molecules of the alvolar surfactant and especially the DPPC molecule are analysed. Also the main research areas regarding the surfactant are mentioned and then the key-parameters during lipid monolayer molecular dynamics studies are examined. The end of the chapter is about the studies done regarding the diffusion of  $O_2$  through the surfactant and also the validation cases for the first and second stage of the Thesis are mentioned.

In Chapter 5, the Solution Approach that was followed in the Thesis is presented. First, the computational tools that were used and also the computational resources on which the problem was solved are analysed. Then, the necessary pre-processing procedures as well as the most important parameters of the problem are presented. In the end of the chapter the whole simulation process is explained in detail for both the stages of the Thesis, the construction of the model and the calculation of the Oxygen diffusion coefficient. A brief introduction to the Umbrella Sampling method is also made.

In Chapter 6 the analysis and the results produced by the two stages of the thesis are presented and discussed in order to extract meaningful conclusions. Initially the final models and their special characteristics are shown. Then the more important parameters of the first stage are analysed, such as the Energy Minimization and for the NVT Equilibration, the Temperature Convergence, the Deuterium Order Parameters, the Density profiles, the Mean Square Displacement of the Phosphorus atoms and the total computational cost. Then the results of the Umbrella Sampling Simulations such as the calculated Potential of Mean Force and Diffusion coefficient of Oxygen are presented and discussed.

Finally in Chapter 7, the conclusions of this Thesis are summarized and recommendations for future studies are made.



## Physiology and Diffusion Mechanisms

### Contents

---

<b>2.1</b>	<b>Physiology of Pulmonary Surfactant . . . . .</b>	<b>11</b>
<b>2.2</b>	<b>Controversy and Alternative Models . . . . .</b>	<b>11</b>
2.2.1	Physiology vs Physics . . . . .	11
2.2.2	Model of Scarpelli . . . . .	14
2.2.3	Model of Hills . . . . .	15
2.2.4	Model of Bangham . . . . .	15
2.2.5	Model of Kashchiev - Exerowa . . . . .	16
2.2.6	Model of Kitaoka . . . . .	16
<b>2.3</b>	<b>Diffusion in Lipid Monolayers . . . . .</b>	<b>17</b>

---



## Chapter Summary

In this Chapter, an introduction to the classical physiology of the alveoli region is made and the alveolus surfactant position is presented. Then, the different models of the alveolus topology are presented as well as the still open controversies between the physiology and physics scientific communities. Finally, the Diffusion Mechanisms that most commonly appear in the biological systems are presented.

## 2.1 Physiology of Pulmonary Surfactant

The respiration system of human beings is one of the most impressive and "fine-tuned" body mechanisms, that even today there are unknown parts of it under research by the scientific community.

The whole respiratory system from the entrance (mouth/nose) to alveoli can be observed in Figure 2.2. The Alveoli area is presented in more detail in Figure 2.3 where the alveoli sacks (the hollows on which each alveolus is attached) can be observed and finally a single alveolus is presented in Figure 2.4. The alveolar surfactant which is modelled in this Thesis is the thin interface layer composed by lipids and proteins that separates the alveolus air from the alveolus fluid.

Surfactant by its definition is a surface active substance which main purpose is to reduce the surface tension on water. This is the reason why surfactants are used in soaps and detergents, to reduce the surface tension of water and aid in the cleaning of interfaces. In the lungs, the surfactant has exactly the same purpose. Reduces the water surface tension in order to make expansion and contraction of alveoli during respiration, easier. In the molecular level, a surfactant is an amphiphilic molecule with a water attracting (soluble) head and a water repelling (insoluble) chain (Fig.2.1).

## 2.2 Controversy and Alternative Models

### 2.2.1 Physiology vs Physics

As it can be noticed by Figure 2.4, classical physiology considers the alveolus to be an "open sphere" which the alveolar surfactant covers in its whole surface. Moreover, the surfactant is connected to the surfactants of all other alveoli and all this



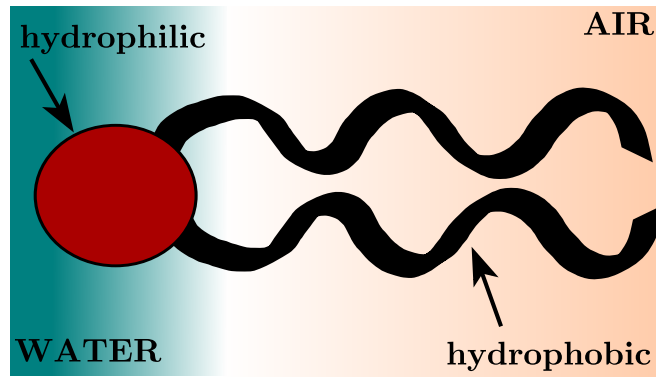


Figure 2.1: A Typical Surfactant Molecule

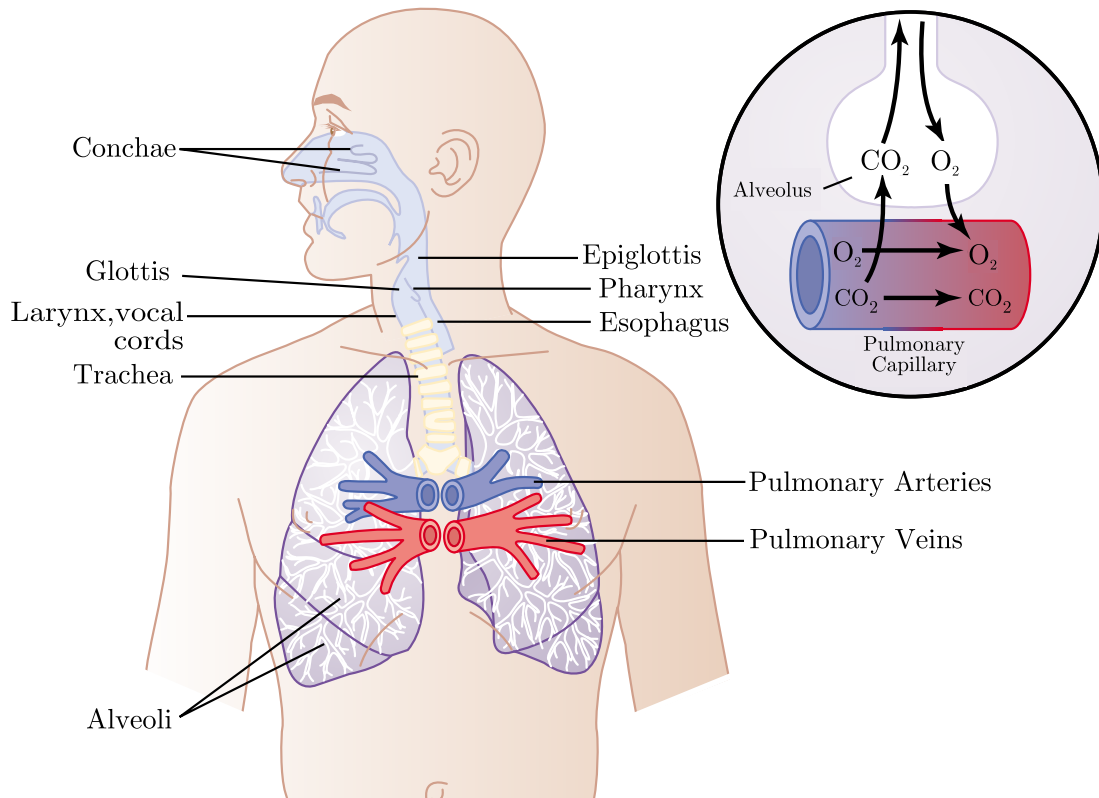


Figure 2.2: The Complete Respiratory System from the entrance until alveoli [4]

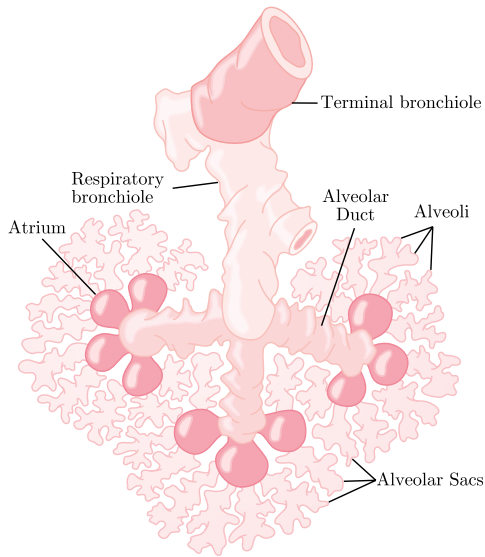


Figure 2.3: The Alveolar Unit [4]

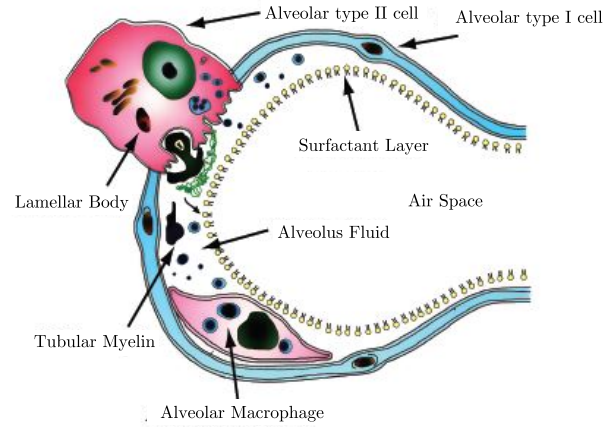


Figure 2.4: Alveolus Topology. The surfactant is the thin lipid layer in the air/liquid interface [5]

surfactant surface reaches the lower parts of the bronchiole. As it can be observed by Figure 2.3, if such is the case, this huge surfactant surface, should have a very complex geometry, consisting of semi-spheres and other geometries.

From a physics point of view however this cannot be true and many physiologists have mentioned that in the past. According to Young-Laplace's Equation the pressure difference across a fluid-fluid interface that is subject to capillary forces is:

$$\Delta P_{int} = -2\gamma/r_{curv} \quad (2.1)$$

where  $\gamma$  is the surface tension and  $r_{curv}$  is the radius of curvature.

The above equation, is based on the Second Law of Thermodynamics which says that all surfaces will tend to minimize their surface energy (this is the reason a pressure difference is created) and therefore such a complex geometry as the alveolar unit would collapse instantly during exhalation and would not retain its initial shape.

To address this controversy between established physiology and physics, several scientists have proposed alternative models but even today, the physiology community has not decided which is the most appropriate. In the following section a brief overview of the alternative proposed models is made.

### 2.2.2 Model of Scarpelli

Scarpelli in his model [6] [7], proposed a complete alternative to the existing controversial model [Fig.2.5]. According to Scarpelli, alveolar liquid is not an open surface layer that covers the whole acinus (from each alveolus to the conducting airways), but rather a liquid (Alveolar Surface Liquid - ASL) that is moving in small channels created by ultra thin films of surfactant. This network of continuous liquid that is connected by all these channels was referred by Scarpelli as the Alveolar Surface Network (ASN)[Fig.2.6].

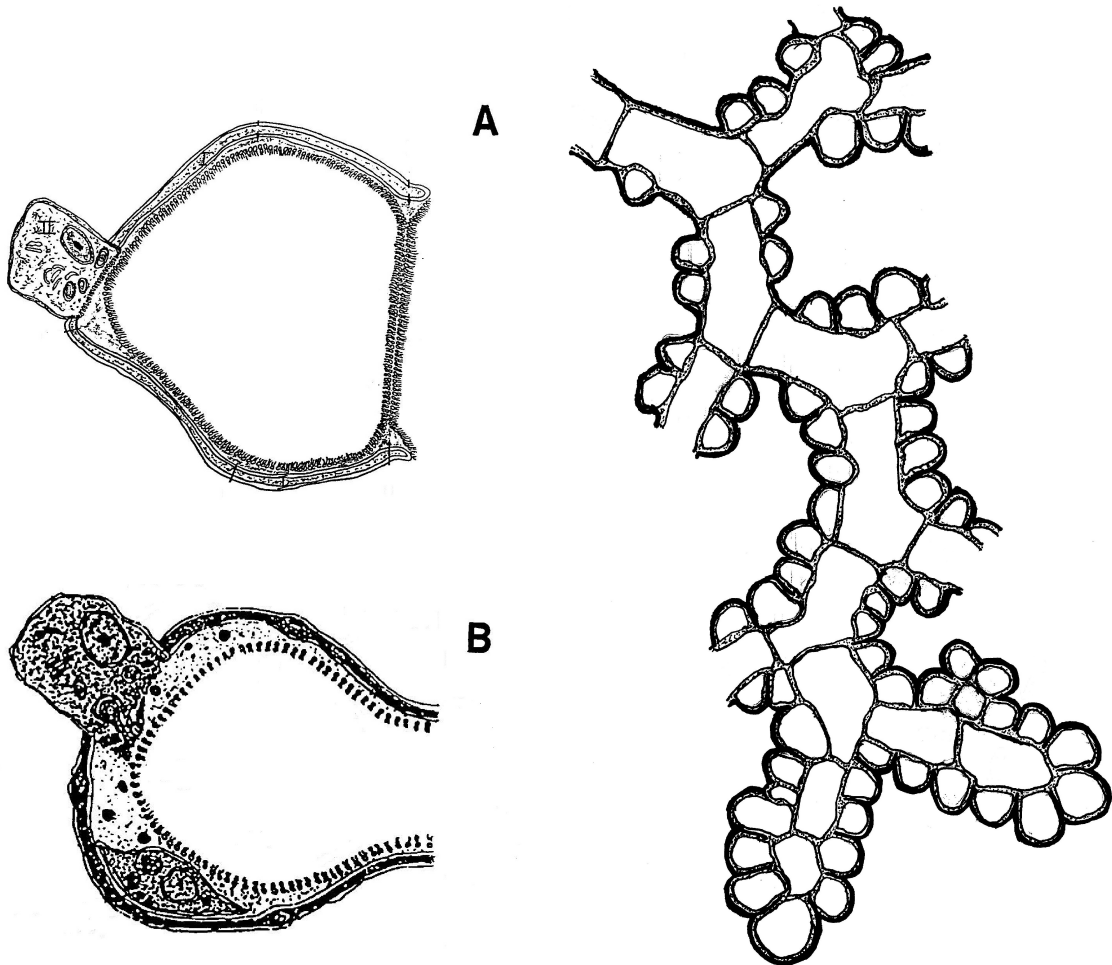


Figure 2.5: Scarpelli's Model (A) in comparison to the Conventional Model (B)[6]

Figure 2.6: The Alveolar Surface Network proposed by Scarpelli[7]

In his research, Scarpelli claimed that other researchers that were based in vitro studies, destroyed the whole network, due to the conventional preservation methods, which he was able to observe and study in vivo [8] [9] [10].

One of the open questions regarding Scarpelli's model was the ability of air to

diffuse through this network of bubbles and foam films and the resistance they would impose.

### 2.2.3 Model of Hills

In his model, Hills [11] proposed that the surfactant adsorbs directly to the epithelial surfaces cells without any liquid between. The integrity of the alveolus is maintained due to pits and pools of liquid that exist in certain parts and all together result to a polyhedral alveolus geometry [Fig.2.7].

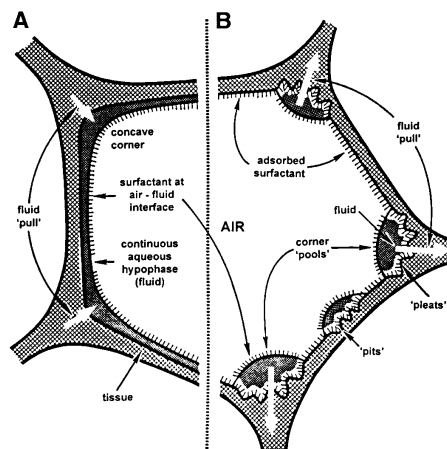


Figure 2.7: Hills Model (B) in comparison to the Conventional Model (A) [11]

### 2.2.4 Model of Bangham

Bangham's model [12], was based in the main hypothesis that in order for the alveoli structures to be stable, they should not be connected by a continuous liquid. Again the open alveolus hypothesis was used but the surfactant was modelled as solid plaques that were not connected, but instead created a liquid geodesic plane.

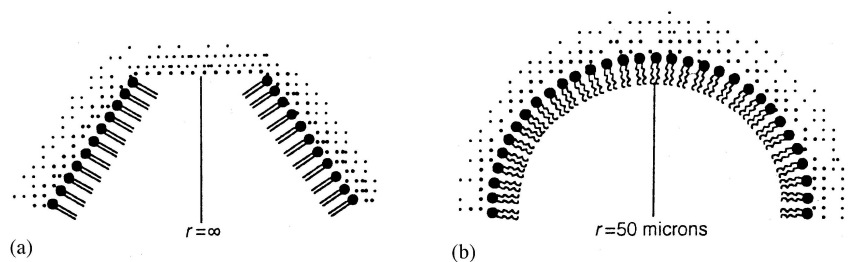


Figure 2.8: Bangham Model (A) in comparison to the Conventional Model (B)(r refers to the radius of the lining) [12]

### 2.2.5 Model of Kashchiev - Exerowa

Exerowa et al. [13], performed a series of in vitro experiments concerning the formation and stability of surfactant surfaces (essentially bilayers). In a later research, Kaschiev and Exarowa [14] and Lalchev et al. [15], proposed an open model for alveolus in which the surfactant monolipid is adsorbed in a stack of lipid bilayers immersed in the hypophase liquid [Fig.2.9].

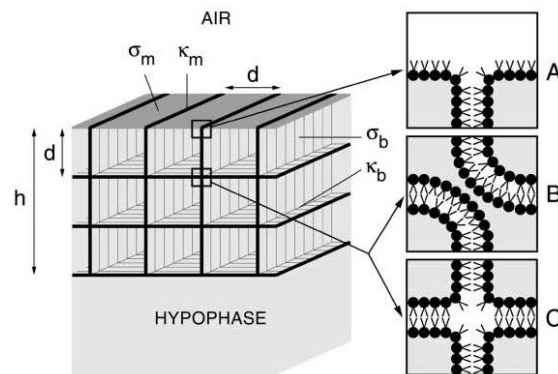


Figure 2.9: Exerowa model with the monolayer-bilayer model of alveolar surfactant layer of thickness  $h$ . The monolayer (*the shaded plane*) at the hypophase/air interface is supported by a network of bilayers (*the hatched planes*) at a distance  $d$  from each other. Cross Sections of Monolayer-Bilayer (MB) and Bilayer-Bilayer (BB) contact lines (*the small squares*) are shown enlarged on the right. Square A represents the molecular structure of the MB contact line (the *circles* and the *lines* schematize, respectively, the heads and the tails of the lipid molecules). Squares B and C illustrate two possible molecular structures of the BB contact line. [Taken directly by [14]]

### 2.2.6 Model of Kitakoa

Kitakoa et al. [16] [17] [18] proposed initially on 2007 a computational model of alveolus as an alternative to the existing which then tried to validate with in vivo experiments in male adult rats. Their model was constructed as a mechanical system composed by springs and hinges, simulating the entire acinus and including alveoli mechanics. An important hypothesis of their computational model, was the existence of an a alveolar lining surfactant film, when the alveolus was at the minimum possible volume, which was necessary to keep the structure from collapse. However, their in vivo experiments were unable to validate this assumption. Although similar to Scarpelli's closed model, this approach was different, since Kitakoa et al. considered the closed alveoli isolated from the respiration process.

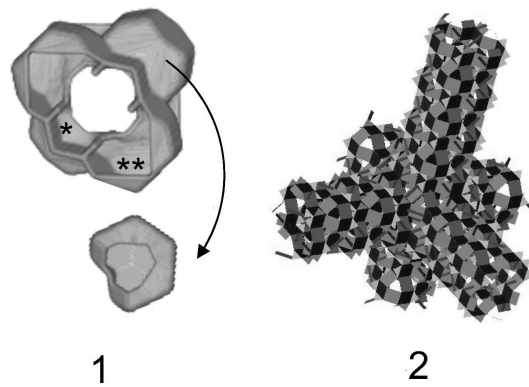


Figure 2.10: Kitakoa computational model. The 3D cubic alveolar duct model is shown, as well as the single alveolus structure(1). In (2) is shown the final 3D branching duct system of hinges/springs [16].

## 2.3 Diffusion in Lipid Monolayers

In lipid monolayer systems, diffusion can be defined as the result of the Brownian Motion of the lipid molecules. Brownian Motion or Pedesis, named after the British botanist Robert Brown who first observed the phenomenon, is the random movement of a substance's particles created by their interaction with the fluid particles in which they are included (air/water) [19]. In comparison to lipid bilayer systems where molecules can diffuse transversely (from one leaflet to the other), in lipid monolayers the diffusion is only lateral [Fig.2.11] and happens in small time scales. Lateral Diffusion of each molecule can be described by Einstein-Smoluchowski equa-

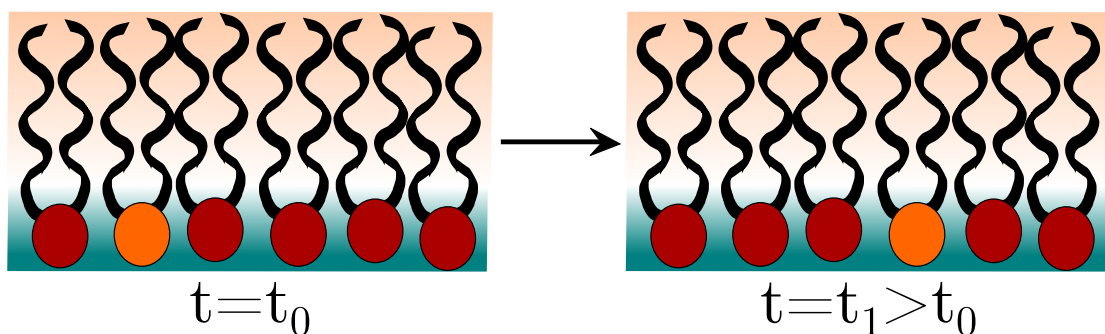


Figure 2.11: Lateral Diffusion Schematic. The lipid molecule (orange) moves laterally from a position at time  $t = t_0$  in an adjacent position at time  $t = t_1$

tion [20] [21] [22]:

$$\langle r(t)^2 \rangle = 4D_{lat}t \quad (2.2)$$

$$\Rightarrow D_{lat} = \frac{1}{4t} \langle r(t)^2 \rangle \quad (2.3)$$

where  $\langle r(t)^2 \rangle$  is the Mean Square Displacement of the molecule (the average of the squared distance covered in time  $t$ ) from its center of mass,  $D_{lat}$  is the diffusion coefficient which is basically the slope of MSD [23] and finally  $t$  is the time. From the above equation it is clear that the larger the slope, the higher will be the diffusion coefficient and therefore the speed of the diffusion. Mean Square Displacement can be also defined as:

$$\langle r(t)^2 \rangle = \langle |r(t + t_0) - r(t_0)|^2 \rangle \quad (2.4)$$

with  $t_0$  being the initial time and so in the end equation 2.3 becomes:

$$D_{lat} = \frac{1}{4t} \langle |r(t + t_0) - r(t_0)|^2 \rangle \quad (2.5)$$

The lateral diffusion can be divided in three types: the Single-file, the Fickian and the Ballistic. In the Single-File Diffusion the particles do not jump on top of the other but rather move like they are inside a quasi-one dimensional channel. On the other hand, during Normal Fickian Diffusion, molecules overcome each other. The main characteristic of Single-File is that MSD is analogous to the square root of time [24]:

$$MSD_{\text{Single File}} \sim \sqrt{t} \quad (2.6)$$

while in the Normal Fickian Diffusion is [25]:

$$MSD_{\text{Fickian}} \sim t \quad (2.7)$$

and in ballistic motion is [26]:

$$MSD_{\text{Ballistic}} \sim t^2 \quad (2.8)$$

## Molecular Dynamics Theory

### Contents

---

<b>3.1</b>	<b>Modelling beyond the Continuum</b>	<b>21</b>
<b>3.2</b>	<b>Historic Evolution of Molecular Dynamics</b>	<b>23</b>
<b>3.3</b>	<b>Main Idea of Simulation Process</b>	<b>24</b>
3.3.1	Simulation Assumptions	24
3.3.2	Limitations of the Method	26
<b>3.4</b>	<b>Examining the Simulation Parameters</b>	<b>27</b>
3.4.1	Force Field Types	27
3.4.2	Potential Functions	28
3.4.2.1	Pairwise Intermolecular Potentials	29
3.4.2.2	Many-Bodies Intermolecular Potentials	31
3.4.3	Integration of Equations of Motion	31
3.4.3.1	Finite Difference Methods	31
3.4.3.2	Predictor-Corrector Methods	33
3.4.3.3	Required Properties of Integration Schemes	33
3.4.4	Ensembles	34
3.4.5	Temperature Coupling	35
3.4.5.1	Berendsen Thermostat	35
3.4.5.2	Noose-Hoover Thermostat	35
3.4.5.3	Velocity Rescale Thermostat - VRes	36
3.4.6	Pressure Coupling	36
3.4.6.1	Berendsen Pressure Coupling	36
3.4.6.2	Parrinello-Rahman Pressure Coupling	36
3.4.6.3	MTTK Pressure Coupling	37
3.4.7	Boundary Conditions	37

---





## Chapter Summary

This chapter introduces the reader to the fundamental concepts behind the Molecular Dynamics as a tool to simulate fluid systems where the Continuum hypothesis is invalid. Initially, an introduction is made to the various modelling methods of fluid systems from the classic Computational Fluid Dynamics to the Molecular Dynamics. Then is presented a brief historic review of the key-points of MD evolution since the development of the first Computer Systems. Also the main idea behind MD, the fundamental physical assumptions on which it is based and of course its limitations are discussed. Finally, the rest of the chapter is dedicated entirely to the most important aspects of MD, such as the various force fields, the potential functions commonly used, the numerical methods of integrating the equations of motion, the thermodynamic ensembles, the temperature and pressure coupling techniques and finally the boundary conditions.

### 3.1 Modelling beyond the Continuum

From the first years of development of Computers, scientific community attempted to utilise them to solve complex physical phenomena that their difficult mathematical expressions could only be approximated and approached with simplistic assumptions. One group of problems were those involving fluid motion which were mathematically explained by the Navier-Stokes Equations [27] and their computational simulation is studied by the Computational Fluid Dynamics (CFD) scientific community. A lot of progress have been done both in the understanding of the flow physics and the development of accurate and robust numerical methods to model them but CFD still is considered to be at its first steps.

Despite, its general applicability, CFD is unable to model flows accurately under a specific length scale due to the fact that viscosity is starting to dominate velocity in the flow characteristics and therefore is involved only with the continuum mechanics of the flow while other tools are developed to model smaller scales than this. In order to be able to understand the regime of a flow (and the modelling method that it should be followed), the use of the dimensionless Knudsen Kn number [28] is necessary:

$$\text{Kn} = \frac{\lambda}{L_{ch}} \tag{3.1}$$

with  $\lambda$  being the free mean path of the molecules (average distance between two collisions) and  $L$  is the characteristic length of the examined system. The mean free path is dependant on the Temperature and Pressure and is defined as:

$$\lambda = \frac{k_B T}{\sqrt{2} \pi d^2 P} \quad (3.2)$$

where  $k_B$  is Boltzmann constant [J/K],  $T$  is the Temperature [K],  $d$  is the molecule diameter [m] and  $P$  is the Pressure [Pa]. It is clear that as  $Kn$  becomes larger,

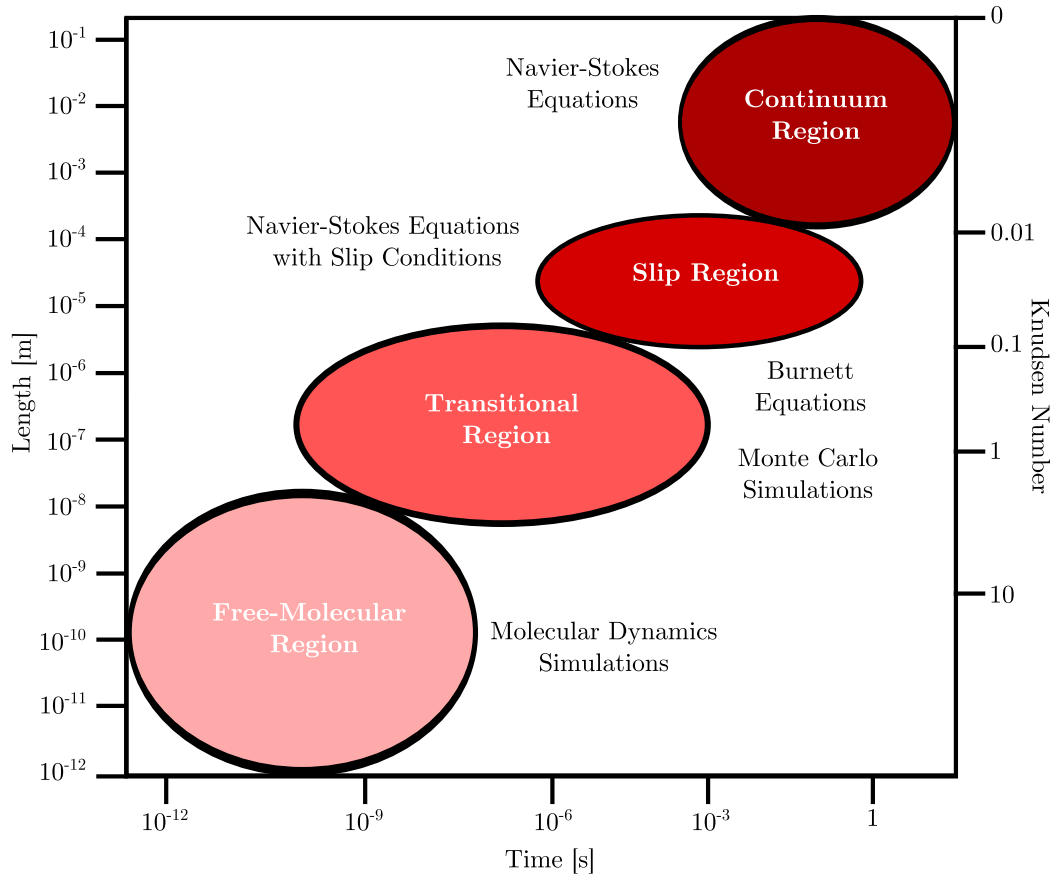


Figure 3.1: Schematic of the flow regimes in relation to Length scale, Time scale and Knudsen number. Next to each regime, the modelling tool that is usually used is presented.

molecular effects influence the system more. Figure 3.1 shows the various flows regimes and the respective Knudsen numbers. As it can be seen just below the

continuum ( $Kn > 0.01$ ), the slip flow regime ( $0.1 > Kn > 0.01$ ) exists where Navier-Stokes are still valid but with the introduction of slip boundary conditions. Lower than that ( $1 > Kn > 0.1$ ), exists the Transition area where the Burnett equations [29, 30] are used. In the lowest parts of the Transition area ( $10 > Kn > 0.1$ ) the Direct Simulation Monte Carlo method is used [31], where the ensemble average of the properties of interest is calculated through a series of random sampling schemes [32]. Finally, in the Free-Molecular area ( $Kn > 10$ ), Molecular Dynamics are utilized to model the system which is basically consisted of interacting molecules. Although MD is a deterministic approach, further improvements have been done to include quantum effects in the system.

## 3.2 Historic Evolution of Molecular Dynamics

According to Allen and Tildesley [33], the first attempts to model liquids in a molecular manner was in 1936 by Morrell and Hildebrand [34] where they physically tried to represent liquid molecules with gelatine balls. The method was so successful that continued to be used even on 1968 when Bernal and King further developed that technique.

Computationally, the first attempt to model liquids was on Los Alamos National Laboratories of U.S.A with the project Metropolis on 1953 [35] using the computer MANIAC. This first attempt was based on the Monte Carlo method using the hard-sphere potential. The more accurate Lennard-Jones potential was implemented on 1957 by Wood and Park [36] which allowed direct comparison with experimental data.

At the same time, Molecular Dynamics were proposed as a simulation method by Alder and Wainwright [37, 38] solving the Newton's equations of motion for a system of hard spheres. Lennard-Jones Potential was not introduced until 1964 by Rahman [39] due to the difficulty of a constantly updating force field.

After 1970s the study of molecular systems started to develop rapidly, with the most important achievement being the MD simulations of water from Rahman and Stillinger on 1971 [40], large protein molecules by McCammon et al. on 1977 [41], phase transition and interfaces behaviour on 1974 [42], 1977 [43] and 1980 [44]. On 1980s were also introduced the first quantum phenomena which made simulations more accurate in many physical problems (on 1982 [45] and 1986 [46]). On 1990s and 2000s as the computational resources increased exponentially in comparison to

the past, MD simulations became more complex, for larger systems and for longer times.

Also, by the end of the 20th century, the interest of the scientific community for micro and nano scale flows increased. Initially oriented to the development of military purposes and new type computers, micro-flows gained another major role in the next decades on the development of Micro-Electro-Mechanical-Systems (MEMS) [47], drug delivery, new-materials development etc. Today, numerous engineering applications for medical, security, aerospace, defence and other industries, require the investigation of such scale phenomena, where the fundamental macroscopic flow physics break down and depending on the order of magnitude, adjustments or even new approaches are required.

### 3.3 Main Idea of Simulation Process

Despite the fact that Molecular Dynamics might be intimidating for the newcomer to the field, in reality the method is trivial. The main idea is to examine with classical mechanics, the interactions of every particle included in a system. Particles can be atoms, molecules or even groups of molecules as will be explained in more detail below. Through the calculation of interactions, the trajectories of the particles are extracted and the final form of the whole system at each timestep is calculated. Figure 3.2 displays the main steps of the method in a flow chart.

#### 3.3.1 Simulation Assumptions

As a computational simulation approach, MD is used based on the following assumptions [49]:

**Newtonian Physics.** Molecular Dynamics ignore Heisenberg's uncertainty principle and Quantum Physics and are based on the classical mechanics introduced by Newton's Laws of Motion [50]:

1. **1st Law.** An object will not change its condition (moving or stationary) unless acted upon by a force.
2. **2nd Law.** The acceleration of a body is in the same direction and propor-

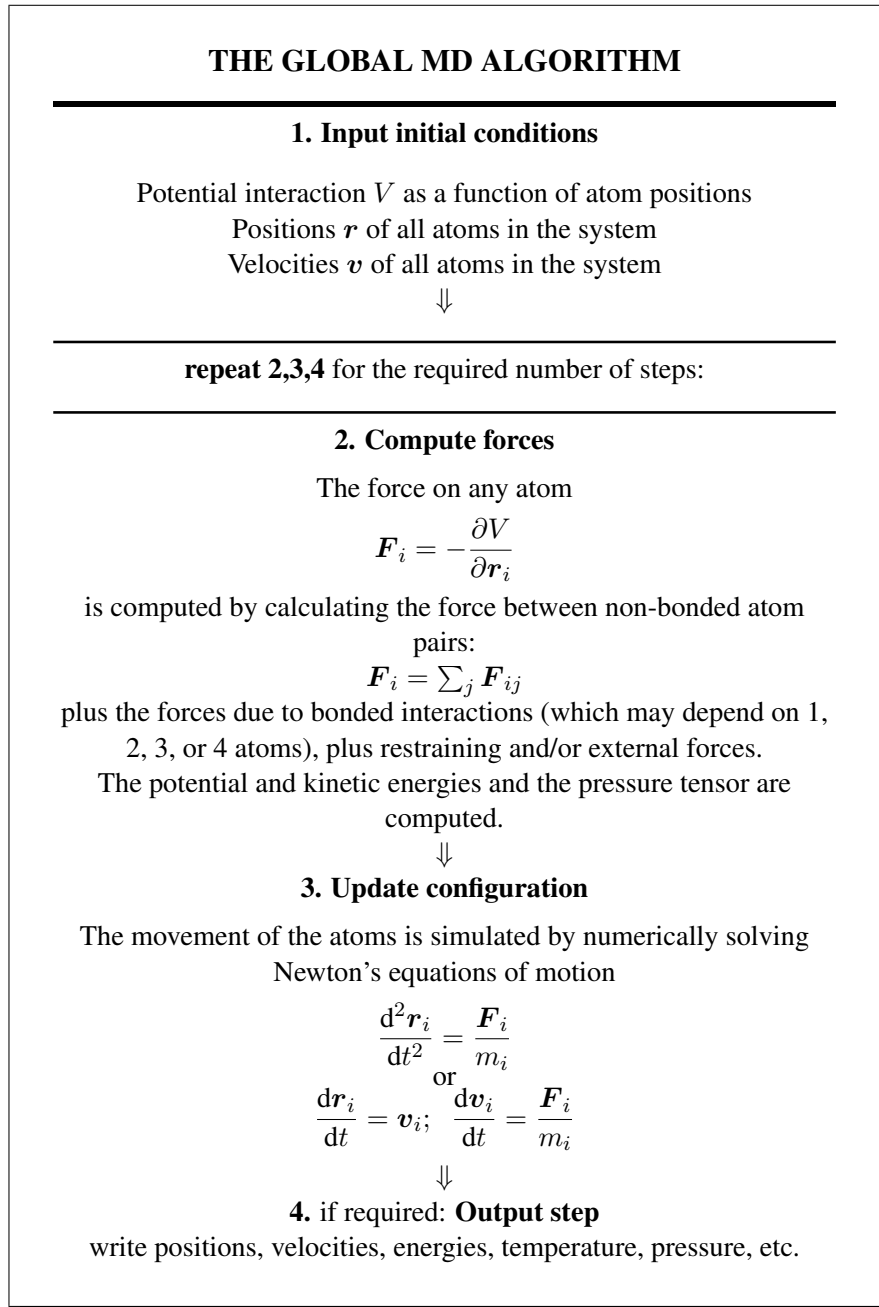


Figure 3.2: The global Molecular Dynamics Algorithm [48]

tional to the force acting on it and inversely proportional to its mass:

$$\vec{F} = m\vec{a} = m\frac{d^2\vec{r}}{dt^2} \quad (3.3)$$

3. **3rd Law.** For every action there is an equal and opposite reaction

**Atoms.** Atoms are considered the smallest elements of the system and are represented as particles with 99.9% of their mass concentrated in their cores since the summation of electrons is averaged out.

**Molecules.** Atoms can be combined to form molecules which are considered as larger particles. Usually these atoms are hold together by three types connections, pairwise *bonds*, *angles* (two bonds with a common particle) and *dihedrals* (two angles with a common bond) [Fig. 3.3].

**Interactions.** Particles interact with each other by four possible ways. Individual pairs of particles interact through a potential which is the result of the summation of all the forces applied on them. These interactions are called *non-bonded* and are usually of two types, Van der Waals (short range) and Electrostatic (long range). Also, intra-molecular or *bonded* interactions are based on the connections described above.

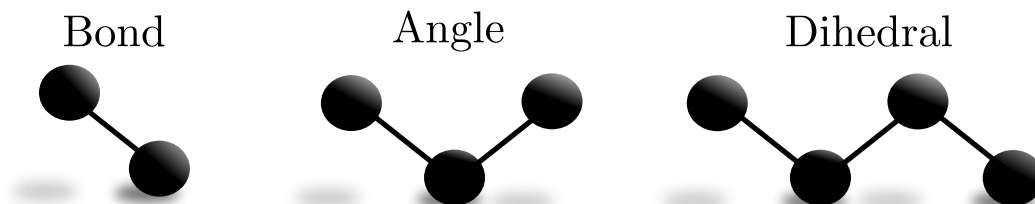


Figure 3.3: The three types of intramolecular connections.

### 3.3.2 Limitations of the Method

As any other simulation method, Molecular Dynamics has its own limitations [23]. First of all, its simplistic approach of interactions and the neglect of realistic quantum physics, leads to an inadequate potential function. Despite the fact that the various parameters can be extracted from experiments, their use does not necessarily produce good results. Further uncertainties such as the omission of atomic

polarizability and the loss of detailed effects, or the difficulty in the calculation of Van der Waals dispersion and barrier's height of dihedral potentials, makes necessary the treatment of parameters as empirical values.

Secondly, the fact that inter- and intra- atomic interactions have a time scale of a few picoseconds makes compulsory the use of very small timesteps, e.g. a few femto-seconds, in the solution of the equations of motions. As a results, even for a solution of a few nanoseconds, millions of iterations are required. By taking also into account that in each iteration all the particle interactions have to be calculated, leads to the conclusion that simulations are limited both in time duration and in spatial dimensions because of the prohibitive computational cost.

The third major limitation of Molecular Dynamics is the use of classical physics which makes it impossible to calculate chemical reactions correctly.

## 3.4 Examining the Simulation Parameters

### 3.4.1 Force Field Types

As it was mentioned before, the most important and most time-consuming stage of a Molecular Dynamics simulation is the accurate calculation of the total system potential. The solution approach can be divided to two large categories according to the force field type that is chosen:

*Atomistic Models.* Atomistic models can be divided in two sub-categories. The first is *All-atoms Force Field* which as the name implies, considers all the models of the system as separate particles, including the non-polar hydrogen, which are taken into account in the calculation of the total system potential. On the other hand, *United-Atoms Force Field* considers most of the atoms of the system separate particles, except from the hydrogen atoms (e.g. hydrocarbon chains in the tails of the lipids) which are considered as one "particle". Atomistic models provide exceptional detail but have a huge computational cost [32].

*Coarse Grain Models.* In this approach, interaction areas are considered inside the domain and the atoms are grouped into them. In that way, the degrees of freedom are reduced leading to higher minimum timescales that can be selected to integrate the equations of motion. Usually, Coarse Grain Simulations are chosen when the physical phenomena examined, require longer time to "unfold". Although



the details in the atomic level are lost, the speed-up in the computational time is noteworthy [32].

In the next pages, the potential functions that are commonly used will be presented and explained as well as the way to calculate the inter-particle interactions.

### 3.4.2 Potential Functions

According to Karniadakis et al. [47] the total potential of a system, consisting of N interacting Particles, can be defined as:

$$\begin{aligned}
 V = & \sum_i V_1(r_i) \text{ Potential due to external field} \\
 & + \sum_i \sum_{j>i} V_2(r_i, r_j) \text{ Potential between pairs of particles} \\
 & + \sum_i \sum_{j>i} \sum_{k>j>i} V_3(r_i, r_j, r_k) \text{ Potential between particle triplets} \\
 & + \dots \text{ Potential of n-particle interactions}
 \end{aligned} \tag{3.4}$$

According to Tieleman et al. [23] the total potential of the system can be defined as:

$$\begin{aligned}
 V = & \sum_{i<j} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} + \sum_{i<j} \left( \frac{A_{ij}}{r_{ij}} \right)^{12} - \left( \frac{B_{ij}}{r_{ij}} \right)^6 \\
 & \text{Electrostatic} \qquad \qquad \qquad \text{Van der Waals} \\
 & + \sum_{\text{Bonds}} \frac{1}{2} k_{ij}^b (r_{ij} - b_{ij}^0)^2 \\
 & + \sum_{\text{Angles}} \frac{1}{2} k_{ijk}^\theta (\theta_{ijk} - \theta_{ijk}^0)^2 \\
 & + \sum_{\text{Dihedrals}} \frac{1}{2} k^\phi (1 + \cos(n_{dih}(\phi - \phi^0)))
 \end{aligned} \tag{3.5}$$

where  $r_{ij}$  is the distance between particles i and j,  $q_i$  is the partial charge of particle i,  $A_{ij}$  and  $B_{ij}$  are Lennard-Jones parameters (will be further explained later),  $k^b$ ,  $k^\theta$ ,  $k^\phi$  are force constants for bonds, angles and dihedrals, n is the dihedral multiplicity and  $b^0$ ,  $\theta^0$ ,  $\phi^0$  are equilibrium values for the bond lengths, angles and dihedral angles.

Usually however, for both equations only the first two terms are used and the rest

are truncated since they are sufficient to represent the basic physical principles.

### 3.4.2.1 Pairwise Intermolecular Potentials

In this paragraph some of the most common potential functions are presented but many more exist. For more details regarding the inter-atomic interactions and potential functions can be found in the literature [51, 52].

**Hard-Sphere Potential.** Initial Molecular Dynamic Simulations [37] were conducted with the use of the hard-sphere model for the approximation of the inter-atomic interactions. The concept of the method is the simplest possible; the collisions happen only when the inter-particle distance is equal to the particle radius  $\sigma$  and are considered completely elastic [53]. Therefore, the potential function [Fig. 3.4] can be defined [33] as :

$$V_{ij} = \begin{cases} 0 & , r_{ij} \geq \sigma \\ \infty & , r_{ij} \leq \sigma \end{cases} \quad (3.6)$$

**Square-Well Potential.** An alternative and slightly more accurate simple model is the square-well potential, where the interactions are separated in three proximity areas, a further  $\sigma_2$ , the radius  $\sigma_1$  and the intermediate of strength  $\epsilon$ . The potential is displayed in Fig. 3.5 and is defined [33] as:

$$V_{ij} = \begin{cases} 0 & , r_{ij} \geq \sigma_2 \\ -\epsilon & , \sigma_1 \leq r_{ij} \leq \sigma_2 \\ \infty & , r_{ij} \leq \sigma_1 \end{cases} \quad (3.7)$$

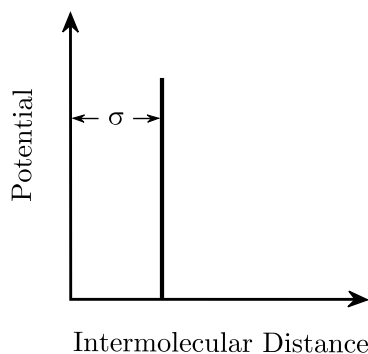


Figure 3.4: The Hard-Sphere Potential

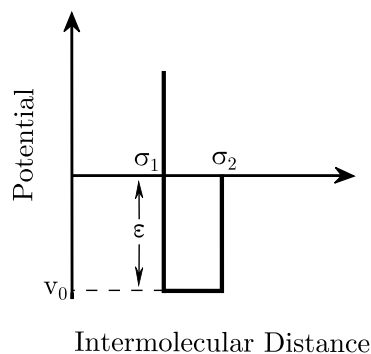


Figure 3.5: The Square-Well Potential

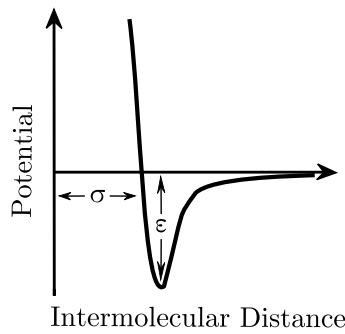


Figure 3.6: The Lennard-Jones 12-6 Potential

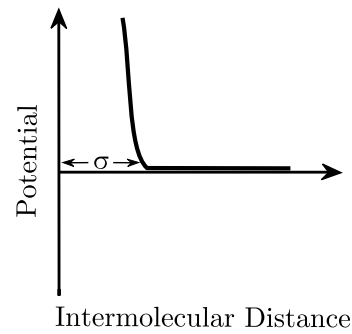


Figure 3.7: The Weeks-Chandler-Andersen Potential

**Lennard-Jones 12-6 Potential.** It is one of the most widely used potentials for nonpolar molecules [Fig. 3.6]. It represents both the short-range repulsive (first part of equation) and attractive (second part of equation) forces during particle interactions but also retains a simple and computationally inexpensive function:

$$V_{ij} = 4\epsilon \left[ \left( \frac{\sigma}{r_{ij}} \right)^{12} - \left( \frac{\sigma}{r_{ij}} \right)^6 \right] \quad (3.8)$$

**WCA Potential.** The Weeks-Chandler-Andersen Potential or Lennard-Jones Truncated [Fig. 3.7], is a modification of the original Lennard-Jones where the interactions are only repulsive and the forces that exceed a truncation distance are neglected:

$$V_{ij} = \begin{cases} 0 & , r_{ij} \leq r_c \\ 4\epsilon \left[ \left( \frac{\sigma}{r_{ij}} \right)^{12} - \left( \frac{\sigma}{r_{ij}} \right)^6 \right] - V_{LJ}(r_c) & , r_{ij} \geq r_c \end{cases} \quad (3.9)$$

**Buckingham Potential.** A more accurate but also much more expensive potential than Lennard-Jones, is the Buckingham Potential [Fig. 3.8]. The main difference is the exponential dependence on distance of the repulsive term which is more realistic [54]:

$$V_{ij} = A \exp(-Br) - \frac{C}{r_{ij}^6} \quad (3.10)$$

with  $A, B, C$  to be empirical constraints.

**Coulomb Potential.** As it can be understood, Coulomb Potential [Fig. 3.9] is

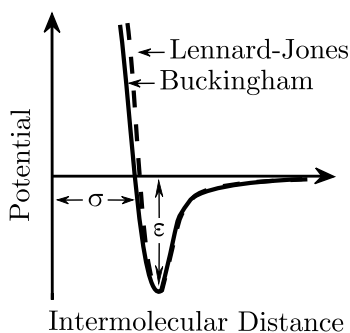


Figure 3.8: The Buckingham Potential. The Lennard-Jones is also shown for clarity

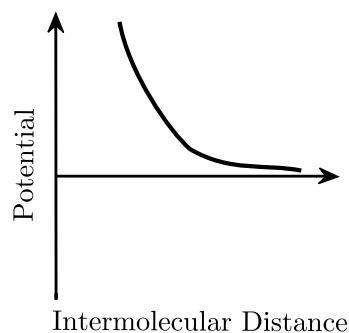


Figure 3.9: The Coulombic Potential of long range electrostatic forces

used for the calculation of the electrostatic interactions between particles:

$$V_{ij} = \frac{1}{4\pi\epsilon_r\epsilon_0} \frac{q_1 q_2}{r_{ij}} \quad (3.11)$$

where  $q_1, q_2, \epsilon_0$  are the charges of the two particles and the vacuum permittivity.

### 3.4.2.2 Many-Bodies Intermolecular Potentials

In some cases the use of more than two particles interactions is necessary in the potential calculation. For such cases, many-bodies potentials, e.g. Tersoff, exist in the literature [47], but their examination is out of the scope of this thesis.

## 3.4.3 Integration of Equations of Motion

All the above-mentioned potentials have a continuous analytical mathematical expression. However, their implementation in a computer solver is impossible in their continuous form because of the coupling of all the particles of the system together. In order to bypass this problem the equations are integrated using finite differences [53].

### 3.4.3.1 Finite Difference Methods

The main idea behind the use of Finite Differences is the separation of the Newton's equation integration in discrete time steps  $dt$ . Then for each time instance  $t$ , the interactions of each particle with the rest is calculated and the summary is the total force acting on it. As seen previously, from the total force the acceleration

is obtained, which then is combined with the known position and velocity of the previous timestep, leading to the position and velocity in the time instance  $t + dt$ . Many methods exist but in this thesis the three most known are going to be presented, Verlet, Leap-frog and Velocity Verlet. In all approaches, the main concept is that the position, velocities and accelerations can be approximated as Taylor Series expansions as:

$$\mathbf{r}(t + dt) = \mathbf{r}(t) + dt \mathbf{v}(t) + \frac{1}{2}dt^2 \mathbf{a}(t) + \frac{1}{6}dt^3 \mathbf{b}(t) + \dots \quad (3.12)$$

$$\mathbf{v}(t + dt) = \mathbf{v}(t) + dt \mathbf{a}(t) + \frac{1}{2}dt^2 \mathbf{b}(t) + \frac{1}{6}dt^3 \mathbf{c}(t) + \dots \quad (3.13)$$

$$\mathbf{a}(t + dt) = \mathbf{a}(t) + dt \mathbf{b}(t) + \frac{1}{2}dt^2 \mathbf{c}(t) + \frac{1}{6}dt^3 \mathbf{d}(t) + \dots \quad (3.14)$$

$$\mathbf{b}(t + dt) = \mathbf{b}(t) + dt \mathbf{c}(t) + \frac{1}{2}dt^2 \mathbf{d}(t) + \frac{1}{6}dt^3 \mathbf{e}(t) + \dots \quad (3.15)$$

**Verlet Algorithm.** The Verlet Algorithm, introduced first by Verlet on 1967 [55] is one of the most common methods. In this scheme, the positions and accelerations at time  $t$  and the positions at time  $t - dt$  give the new positions at  $t + dt$ .

$$\mathbf{r}(t + dt) = 2 \mathbf{r}(t) - \mathbf{r}(t - dt) + dt^2 \mathbf{a}(t) \quad (3.16)$$

Although velocity is not directly involved with the Verlet algorithm, it can be calculated for either time  $t$  or time  $t + \frac{1}{2}$  as:

$$\mathbf{v}(t) = \frac{\mathbf{r}(t + dt) - \mathbf{r}(t - dt)}{2 dt} \quad (3.17)$$

$$\mathbf{v}(t + \frac{1}{2}dt) = \frac{\mathbf{r}(t + dt) - \mathbf{r}(t)}{dt} \quad (3.18)$$

**Leap-frog Algorithm.** The Leap-frog Algorithm was based on the Verlet algorithm and developed by Hockney on 1970 [56]. In this method, the velocities at time  $t + \frac{1}{2}dt$  are calculated from the velocities at time  $t - \frac{1}{2}dt$  and accelerations at time  $t$ . The positions  $\mathbf{r}(t + dt)$  are then calculated at  $t + dt$  by the positions at  $t$  and the velocities at  $t + \frac{1}{2}dt$ , giving in the method the name Leap-frog (...-Velocity-Position-Velocity-...):

$$\mathbf{v}(t + \frac{1}{2}dt) = \mathbf{v}(t - \frac{1}{2}dt) + dt \mathbf{a}(t) \quad (3.19)$$

$$\mathbf{r}(t + dt) = \mathbf{r}(t) + dt \mathbf{v}(t + \frac{1}{2}dt) \quad (3.20)$$

**Velocity Verlet Algorithm.** Another variation of the Verlet algorithm which gives the positions, velocities and accelerations at the same time with high accuracy is the Velocity Verlet, introduced by Swope et al. on 1982 [57]. The method can be separated in three steps; initially the positions at  $t + dt$  are calculated, secondly the velocities at  $t + \frac{1}{2}dt$ , then the acceleration at  $t + dt$  through the new forces (which are based on the calculated positions) and finally the velocities at  $t + dt$ :

$$\mathbf{r}(t + dt) = \mathbf{r}(t) + dt \mathbf{v}(t) + \frac{1}{2}dt^2 \mathbf{a}(t) \quad (3.21)$$

$$\mathbf{v}(t + \frac{1}{2}dt) = \mathbf{v}(t) + \frac{1}{2}dt \mathbf{a}(t) \quad (3.22)$$

$$\mathbf{v}(t + dt) = \mathbf{v}(t + \frac{1}{2}dt) + \frac{1}{2}dt \mathbf{a}(t + dt) \quad (3.23)$$

$$\mathbf{v}(t + dt) = \mathbf{v}(t) + \frac{1}{2}dt [\mathbf{a}(t) + \mathbf{a}(t + dt)] \quad (3.24)$$

In the literature more FD schemes can be found, as well as the analytical derivations of those mentioned before [53].

### 3.4.3.2 Predictor-Corrector Methods

An alternative category of integration algorithms includes the predictor-corrector methods mainly developed by Gear on 1971 [58] but also used by Rahman on 1964 [39]. Predictor-Corrector methods are generally 'slower' than the previously mentioned and their use is limited nowadays but nevertheless provide accurate results. The main idea of these methods can be summarized in 4 steps:

1. Predict, according to Taylor expansion, new positions, velocities, accelerations
2. Calculate  $\mathbf{a}(t+dt)$  from forces since positions are known
3. Compare the calculated accelerations with the predicted and measure the difference
4. Correct positions, velocities and accelerations according to this difference

### 3.4.3.3 Required Properties of Integration Schemes

In order for integration algorithms to be utilised in Molecular Dynamics Simulations, some requirements should be fulfilled [49]:

1. **Efficiency.** Since Molecular Dynamics simulations are by their nature computationally expensive, efficient usage of the resources is very important for keeping simulation times small.
2. **Stability.** Every integration method should conserve the energy of the system and retain its stability..
3. **Accuracy.** The results of the integration algorithm (e.g the trajectories) should represent physically meaningful quantities.

### 3.4.4 Ensembles

Ensembles are basically the macroscopic thermodynamic conditions under which the system is simulated. There are four commonly used ensembles in Molecular Dynamics and their name comes from the properties that each time are kept constant [59].

One of the most used (especially in the cases of lipid monolayers) is the **Canonical Ensemble** or **NVT** because the Number of particles, the Volume and the Temperature are kept constant. Another is the **Isothermal-Isobaric Ensemble** or **NPT** where the Pressure instead of the Volume is constant. There is also the **Microcanonical Ensemble** or **NVE** where the total Energy of the system is kept constant and basically the system can be considered isolated. The above-mentioned ensembles are also shown for clarity in Figure 3.10. Finally there is the **Grand Canonical Ensemble** which is basically an NVT with also constant chemical Potential.

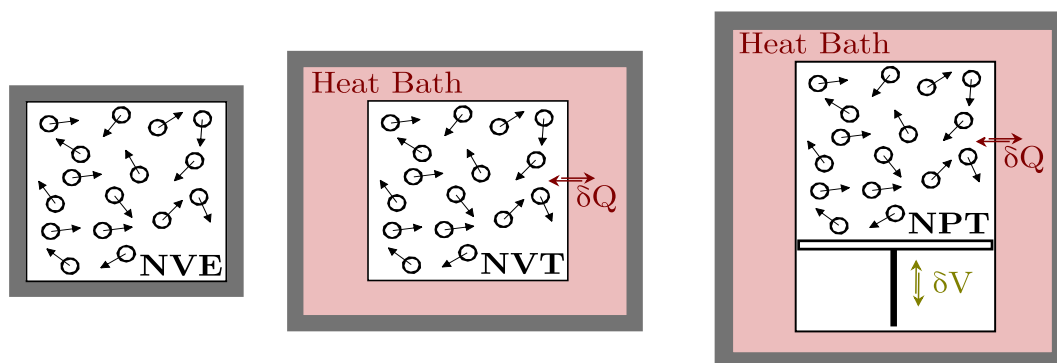


Figure 3.10: Thermodynamics Ensembles commonly used in MD simulations. The isolated NVE, the constant Volume NVT and the constant Pressure NPT.

### 3.4.5 Temperature Coupling

The Temperature of the system is directly related to the time average of the kinetic energy. Therefore, in order to control it during an MD simulation, it is necessary to couple it to an external "heat bath" of fixed temperature. In order to approximate this procedure, algorithms simulating a thermostat are used, with the most common described below.

#### 3.4.5.1 Berendsen Thermostat

The thermostat introduced by Berendsen et al. on 1984 [60] basically simulates a weak coupling of the heat-bath with the given Temperature through first-order kinetics. This is achieved by slowly correcting the deviation of system Temperature according to the following equation [61]:

$$\frac{dT}{dt} = \frac{T_0 - T}{\tau} \quad (3.25)$$

This leads to an exponential decay of Temperature deviation according to a time constant  $\tau$  which makes the method very efficient in relaxing the system to the required Temperature but on the other hand, the generated states are not in a canonical ensemble.

#### 3.4.5.2 Noose-Hoover Thermostat

To bypass the inability of the Berendsen Thermostat to use canonical ensemble, the extended-ensemble approach was proposed by Nose on 1984 [62] and Hoover on 1985 [63]. In this approach, extra terms are introduced in the equations of motion representing a "thermal reservoir" and a frictional term  $\xi$ :

$$\frac{d^2\mathbf{r}_i}{dt^2} = \frac{\mathbf{F}_i}{m_i} - \xi \frac{d\mathbf{r}_i}{dt} \quad (3.26)$$

with the equation of motion for  $\xi$  being:

$$\frac{d\xi}{dt} = \frac{1}{Q_M}(T - T_0) \quad (3.27)$$

where  $T_0$  is the reference Temperature,  $T$  is the instantaneous Temperature of the

---



system and  $Q_M$  is mass parameter of the system which together with  $T_0$  define the strength of the coupling (similar to  $\tau$  in the Berendsen). Although Nose-Hoover thermostat enables canonical ensemble simulations, the relaxation of the temperature has an oscillatory behaviour.

### 3.4.5.3 Velocity Rescale Thermostat - VRes

Another weak coupling method, based on the Berendsen is the Velocity-Rescale proposed by Bussi on 2007 [64]. Essentially, the method is the same as in Berendsen with the addition of a stochastic term that gives a correct kinetic energy distribution. The great advantage of this method is that it has a first order decay of temperature deviations without the oscillations of Nose-Hoover and also produces a correct Canonical Ensemble.

## 3.4.6 Pressure Coupling

Very similarly to Temperature Coupling, the system can be coupled to a "Pressure bath" with "barostats". Many algorithms exist to perform this coupling and the most commonly used are presented below.

### 3.4.6.1 Berendsen Pressure Coupling

The Pressure coupling Berendsen algorithm [60] is the same as in Temperature coupling but transformed to rescale the coordinates and the box vectors every specific number of steps. The method is a first-order kinetic relaxation of the pressure towards  $P_0$  (reference Pressure):

$$\frac{d\mathbf{P}}{dt} = \frac{\mathbf{P}_0 - \mathbf{P}}{\tau_p} \quad (3.28)$$

The main disadvantage of the method is that it does not result in the exact NPT ensemble.

### 3.4.6.2 Parrinello-Rahman Pressure Coupling

This method is the analogous of Nose-Hoover in Temperature coupling, providing the correct NPT ensemble and in reality is usually used in conjunction to Nose-Hoover Thermostat. It was developed by Parrinello and Rahman on 1981 [65] and

improved by Nose on 1983 [66]. The equations of motion of the particles change similarly:

$$\frac{d^2\mathbf{r}_i}{dt^2} = \frac{\mathbf{F}_i}{m_i} - \mathbf{M} \frac{d\mathbf{r}_i}{dt} \quad (3.29)$$

with M being:

$$\mathbf{M} = \mathbf{b}^{-1} \left[ \mathbf{b} \frac{d\mathbf{b}'}{dt} + \frac{d\mathbf{b}}{dt} \right] \mathbf{b}'^{-1} \quad (3.30)$$

where b is the box matrix representations [48]. As in Nose-Hoover, the disadvantage of the method is that it has oscillatory behaviour if the initial temperature is very far from the target and also needs 4 to 5 times more time to converge to the reference Pressure.

### 3.4.6.3 MTTK Pressure Coupling

In Parrinello-Rahman method, there is an extra disadvantage when using the leap-frog integrator that the necessary Pressure information is available only after the full timestep. Therefore, Martyna et al. [67] and Tuckerman [68] introduced an alternative method for Pressure coupling that provides a correct NPT ensemble using the Velocity-Verlet integrator. For more details on the topic, the reader is directed to the relative literature [48, 69].

### 3.4.7 Boundary Conditions

In contrast to the classical Computational Fluid Dynamics where there is a plethora of boundary conditions on which the system can be defined, in Molecular Dynamics, the options are quite limited. Since a major problem in MD simulations are the edge effects on the molecules, the most widely used boundary conditions are Periodic.

In that sense an infinite number of images of the main system are created around it mimicking "an infinite computational domain with a finite number of molecules" [49]. Figure 3.11 shows how PBC work, which can be also summarised in the following conditions:

1. An atom leaving from one boundary, should re-enter directly from the exact opposite boundary.
2. If an atom is close to the boundaries, it interacts with the atom images of the opposite site of the simulation domain. Since however that would mean the calculation of all interactions between all boxes when calculating the potential

of particles, the usage of **minimum image convention** is necessary. With this convention, which is based on the cut-off distance of Van der Waals forces, only the closest periodic image is used. For the long range electrostatic interactions, the Ewald Sum, PME and other methods are used.

In the case where wall boundary conditions are necessary, usually a layer of non-reactant atoms (e.g. Argon) is added in the boundary, with a weak Lennard-Jones potential, so that when a particle reach this "wall", the weak potential will rebound it.

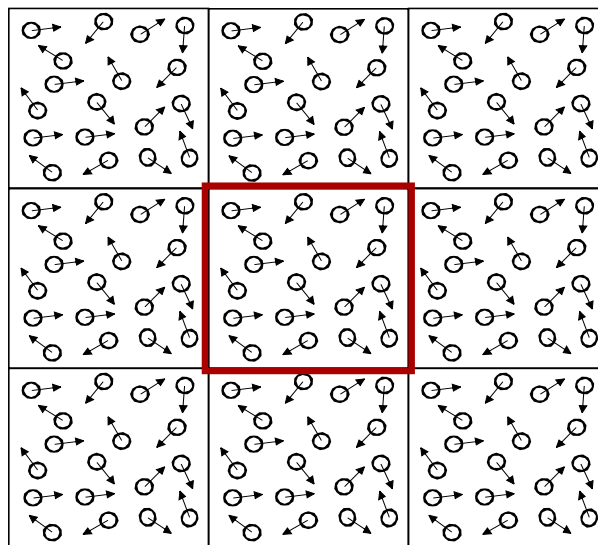


Figure 3.11: Periodic Boundary Conditions schematic for two dimensions. In the center is the actual solution domain (red box) and around it, the first of every of its images.

## Molecular Dynamics for Lipid Monolayers

### Contents

---

<b>4.1</b>	<b>Research on Lipid Monolayers . . . . .</b>	<b>41</b>
4.1.1	Lipid Molecules . . . . .	41
4.1.1.1	DPPC Lipid Molecule . . . . .	42
4.1.2	MD Simulations for Alveolar Surfactant . . . . .	43
<b>4.2</b>	<b>Molecular Dynamics Parameters . . . . .</b>	<b>45</b>
4.2.1	Time and Length Scales . . . . .	45
4.2.2	Force Fields . . . . .	46
4.2.3	Ensembles for Interfaces . . . . .	47
4.2.4	Temperature and Pressure Control . . . . .	48
4.2.5	Electrostatic Interactions and Cut-off . . . . .	49
<b>4.3</b>	<b>Studies on <math>O_2</math> diffusion through the surfactant . . . . .</b>	<b>50</b>
<b>4.4</b>	<b>Validation of Simulations . . . . .</b>	<b>51</b>
4.4.1	Model Validation . . . . .	51
4.4.2	Diffusion Coefficient Validation . . . . .	53

---



## Chapter Summary

In this Chapter, the Literature Review of the Thesis is presented. At first, the lipid molecules of the alveolar surfactant and especially the DPPC molecule are analysed. Also the main research areas regarding the surfactant are mentioned and then the key-parameters during lipid monolayer molecular dynamics studies are examined. The end of the chapter is dedicated to the studies done regarding the diffusion of  $O_2$  through the surfactant and also the validation cases for the first and second stage of the Thesis are mentioned.

### 4.1 Research on Lipid Monolayers

Lipid Systems and in particular lipid monolayers have been examined both with Molecular Dynamics and Monte Carlo Simulations. They exist mainly in air/water interfaces and have special importance in the modelling of biological interfaces such as the tear film of eyes and the gas exchange interface of lungs (pulmonary surfactant).

#### 4.1.1 Lipid Molecules

Lipids are a combination of compounds such as fatty acids, glycerolipids, glycerophospholipids etc. [70] with a characteristic structure of a hydrophobic tail (apolar) and a hydrophilic headgroup (polar). The composition of the tail is usually a combination of hydrocarbon chains while the headgroup can be phosphatidylcholine (PC), Phosphatidylglycerol (PG) and others. Geometrically, lipid molecules are elongated and asymmetric and in average their length is about  $2.5nm$  with a cross-sectional area of  $0.6nm^2$  [32]. Their affinity for water is strongly dependant on their geometrical characteristics and can vary from hydrophobic to amphiphilic [71]. The latter, adsorb in the air/water interface, forming a lipid monolayer also known as Langmuir film, which reduces the surface tension of the interface from approximately  $70[mN/m]$  [72] to almost zero.

Alveolar Surfactant is a mixture of lipids and proteins [32], mainly consisting of PC lipids (80% by weight) and especially DPPC (40% by weight). There are also PG and PI lipids (8-15% by weight), cholesterol (5-10% by weight) and other components. Also, four proteins are included in the monolayer but only two are associated with its surfactant function, SP-B and SP-C, although their amount does not ex-

ceed 3% by weight. Since the main component of the lipid monolayer is the DPPC lipid (which is also the only component examined in this Thesis) a more thorough analysis is necessary.

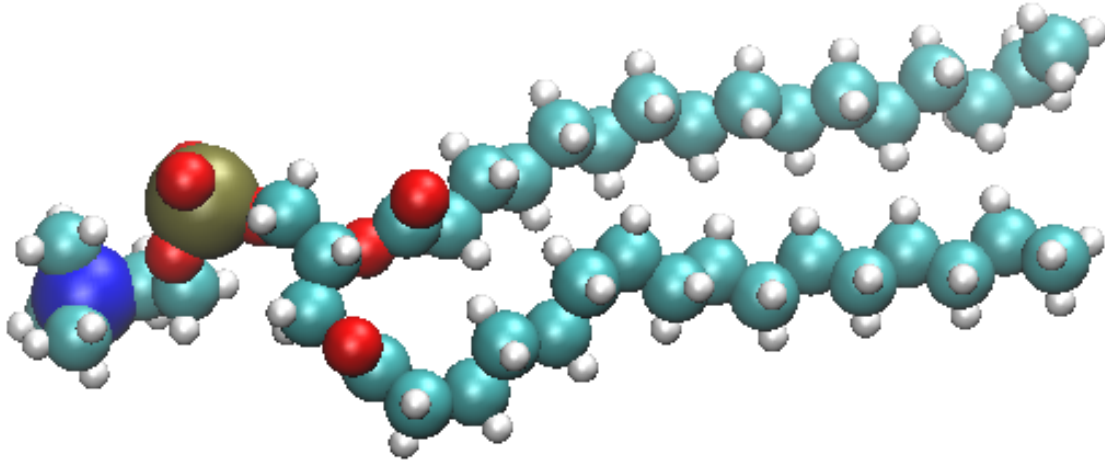


Figure 4.1: A 3D Structure representation of the DPPC lipid Molecule as it was extracted by the bilayer of Klauda et al. [73]. Legend: Blue=Nitrogen, Gold=Phosphorus, Red=Oxygen, Cyan=Carbon and White=Hydrogen.

#### 4.1.1.1 DPPC Lipid Molecule

DPPC or Dipalmitoylphosphatidylcholine is a zwitterionic (neutral charge) phospholipid found in many biological structures such as cell membranes, bacterial membranes and of course the pulmonary surfactant. In Figure 4.1, its structure can be observed with its phosphoric head and its two acid chains (Sn-1 and Sn-2). The structure is extracted from a DPPC lipid bilayer that was used in this Thesis. In particular, the DPPC molecule was constructed by Klauda et al. [73] and its exact topology is presented in Figure 4.3.

According to Baoukina et al. [32] a lipid monolayer changes many phases at different pressures and temperatures. In Figure 4.2 the surface pressure-area isotherm shows this wide variety of possible phase changes. The phase however is also strongly dependant in the Temperature of the system (for steady surface pressure-area). For instance the DPPC layer transforms from the solid Condensed (C) phase to the gel-like Liquid-Condensed (LC). Due to the complex phenomena that appear during phase changing, such as the creation of aggregates and bilayers, in this Thesis the monolayer was simulated in the solid Condensed phase at approximately 50 °C.

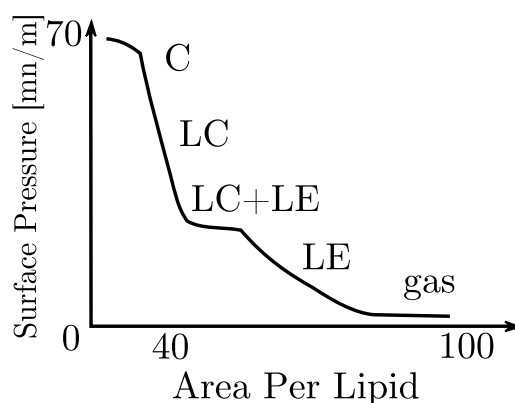


Figure 4.2: Phase Changes of the alveolar surfactant on the Surface Pressure-Area Isotherm at 24 °C [73] [74]. Legend: C=Condensed, LC=Liquid Condensed, LE= Liquid Expanded

#### 4.1.2 MD Simulations for Alveolar Surfactant

Alveolar Surfactant was one of the first lipid systems examined with Molecular Dynamics. Due to the complexity of the system, simulations focused in a wide variety of topics related to it.

An active research area with very recent research is the investigation of **phase transition** of alveolar surfactant. Tieleman and his co-workers in 2012 [75, 76], examined the system with increased complexity adding as many details as possible in the model such as three phospholipid components (DPPC, DOPC, POPG), cholesterol (2:1:1:1 composition) and surfactant proteins SP-B and SP-C. Because the phase transition needed a larger timescale, coarse grain simulations were used. That research was one of the many conducted by Tieleman’s research group regarding the alveolar surfactant. Previous works include the investigation of folding mechanics during compression by Baoukina et al. [77] with a DPPC & POPG monolayer and CG simulations. Duncan et al. [78] have examined the phase transition of a surfactant, made of DPPC and other components, with CG simulations. Shushkov et. al [79] have also examined the changes during phase transition.

Another important area is **the structure and the collapse** of surfactant during expansion/compression. Collapse leading to the forming of surfactant bilayers in the water subphase (more details regarding the physiology can be found in literature [80]) was examined by Tieleman and his co-workers along with the phase transition in the same research as mentioned before [75, 76]. The structural behaviour of a DOPC surfactant monolayer after oxidation of non-saturated lipids, was simulated



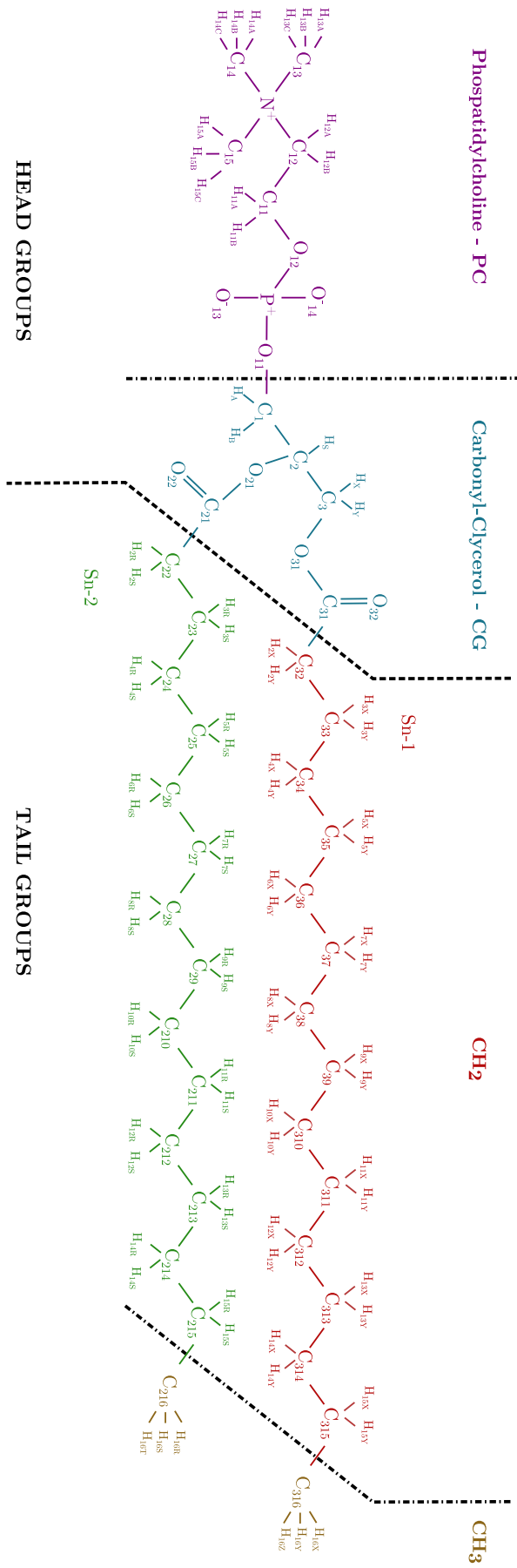


Figure 4.3: The topology of the DPPC lipid molecule based on the generic atomic structure of Ma and Allen [74] and the specific atomic labels utilized by Klauda et al. [73] in their model.

by Khabiri et al. [81] with an all-atoms simulation. Duncan and Larson [82] explored how proteins SP-B and SP-C affected the monolayer mixture (mainly DPPC) folding during compression/expansion with the use of the Coarse Grain Model. A general structural research of the alveolar surfactant was conducted by Shushkov et al. [79]. Apart from the phase transition that was mentioned before, they examined the organizational patterns for different areas per lipid and the film folding. Finally, Freitas et al. [83] researched the structure and properties of a PA lipid monolayer with protein SP-B and Laing et al. [84] the effect of cholesterol on the properties of lipid monolayers at low surface tensions. The latter research utilized both CG and atomistic models as well as a combination of DPPC, POPG and cholesterol.

Other researches include the examination of the **electric and dielectric properties** of the surfactant by Tzvetanov et al. [85] with an atomistic approach and the calculation of the **lateral pressure profiles** for different interfaces (including air/water), lipid compositions and surface tensions by Baoukina et al. [86]. Also research has been conducted to examine the disturbance created to pulmonary surfactant after inhalation of air with diesel exhaust components by Sosnowski et al. [87], as well as, to measure surfactant's **ability to reduce surface tension** for different composition and areas per lipid, by Rose et al. [88].

Another very important area was the examination of diffusion which will be further discussed in the end of the chapter.

## 4.2 Molecular Dynamics Parameters

### 4.2.1 Time and Length Scales

As described already in Chapter 3 regarding the theory of Molecular Dynamics, the high computational cost of the method, comes from the need to model phenomena that take place in very small timescales.

As described by Tieleman et al. [23], in lipid systems, the fastest motions are the vibrations of the bonds, angles and dihedrals inside the molecules and also the diffusion of water and small particles, which are in the magnitude of *a few picoseconds* ( $10^{-12}s$ ). Then, the trans-gauche isomerization takes place in tens of picoseconds for the lipid tails and in hundreds of picoseconds in the headgroups.

Then, are the bulk-molecule phenomena such as the rotation of phospholipids

around their long axis (*nanoseconds*  $10^{-9}$  s), lateral diffusion or lipids position exchange (*tens of nanoseconds*) and finally phase transitions, protein-lipid interaction and others. Especially, cases of gas diffusion through the lipids monolayer would take an enormous amount of time because gases' density is so low that is practically impossible to simulate with an MD system. Therefore, various techniques are used to bypass that problem and are explained further, later in this chapter.

## 4.2.2 Force Fields

In the simulation of the air/water interface, the air phase is modelled as vacuum [32] due to low density of gas and negligible intermolecular interactions. In Molecular Dynamics Simulations in the past, both atomistic and coarse grain models have been extensively utilized as it can be also observed by Table 4.1. The most commonly used are further explained in the rest of the section.

CHARMM (Chemistry at HARvard Macromolecular Mechanics) is a set of atomistic force fields and also the name of a Molecular Dynamics software developed at Harvard University [89]. For lipid simulations usually the CHARMM27 force field is used [90] and is also combined with the "all-atom" CHARMM22 protein force field [91] when there is a lipid-protein simulation. It is not uncommon however the "united-atom" CHARMM19 to be selected instead of the all-atom field, depending on the case. In 2012 the CHARMM36 [92] was released as an improvement of CHARMM22 for protein modelling and studies such as protein folding, assembly etc. but it introduced also improvements for lipid mono/bi-layers modelling [73]. When CHARMM force fields are used for lipids/proteins, usually the TIP3P water model [93] is utilized, since it is parametrized for them.

GROMOS is a united-atoms atomistic force field developed by the University of Groningen and ETH Zurich [94]. In most of the Molecular Dynamics simulations is implicitly used through its evolution, Berger's model. The SPC and SPC/E water models are used in conjunction to GROMOS force fields, with the first being better for interfaces and the latter being better for bulk water simulations [23].

An alternative model was developed based on GROMOS by Berger et. al [95] to be used with DPPC Lipids. All the bonds, angles and dihedrals (improper and dihedral angles of headgroups) are the same as in GROMOS87 but the Ryckaert-Bellemans potential [96] is used for the hydrocarbon chains. Also the non-bonded interactions is based on the OPLS and GROMOS models and alternative Lennard-

Jones parameters.

Chiu et.al [97] proposed an improvement of Berger’s model by using a different set of charges.

Finally, Calgary model [84] uses Berger Model for Bonded and Non-Bonded parameters with modified charges for the head groups. As in Berger’s, bonded are based on GROMOS and non-bonded are based on OPLS united-atom model.

Category	Lipid F.F.	Research
Atomistic	CHARMM27	[98], [85], [99]
	CHARMM22	[83]
	Burger	[81], [88],
	Chiu	[86]
	Calgary	[84]
Coarse Grain	MARTINI	[75] , [76], [82], [86] ,[84]

Table 4.1: Force Fields used on Lipid Monolayers Simulations

### 4.2.3 Ensembles for Interfaces

Ensembles used in Lipid Monolayers simulations vary depending on the phenomenon examined but mainly the NPT [81] and NVT [79, 83, 88] ensembles are used. For example, in the simulation of structural characteristics such as compression and expansion, surface tension is the major parameter because lipid monolayers, in contrast to bilayers, can have a wide range of molecular densities or areas per lipid and therefore different surface tensions at the interface [32]. Surface tension is directly connected to interface area and for that reason, selected statistical ensembles are proposed for the simulations by Zhang et.al [100] and Feller et.al [101] with constant:

1. Normal Pressure and Interface Area
2. Lateral Pressure and Length Normal to the Interface
3. Volume and Surface Tension
4. Normal Pressure and Surface Tension

Another unique characteristic of high importance on lipid monolayers simulations, is the fixed normal size of the box in order to negate its shrinkage due to the existence

of vacuum [32] and in that way imposing a fixed dimension. If both dimensions are considered fixed and therefore the Volume of the box is constant, the monolayer area is considered also constant and the surface tension can be adjusted by applying lateral or normal pressure to the boundaries and consequently expand or compress the monolayer.

#### 4.2.4 Temperature and Pressure Control

In order to control the pressure and temperature of the simulation in lipid systems simulations, two main methods are used [23]:

*Weak Coupling Scheme.* Initially proposed by Berendsen et al. in 1984 [60], in this method, the system is coupled to a "bath" of constant pressure and temperature allowing the system to be easily controlled. Conceptually the method is very simple and causes little perturbation but on the other hand, introduces an unknown statistical mechanical ensemble which makes impossible the analysis of thermodynamic fluctuations.

*Extended System.* The method was developed during 1990s by Martyna et al. [102] [67] and Feller et al. [103]. In this approach, the system is controlled by using a piston for pressure coupling and a thermostat for temperature coupling. In that way, the statistical mechanical ensemble is well defined (since the operating parameters of a piston or a thermostat are well known) but also the complexity of the system increases.

The most commonly used Thermostats and Pistons in lipid monolayer simulations are presented in Table 4.2.

Coupling	Method	Research
Thermostat	Nose-Hoover	[86], [98], [83], [81]
	Berendsen	[78], [88]
Pistons	Langevin	[98]
	Parinello-Rahman	[86], [81]
	Berendsen	[86], [78]

Table 4.2: Pressure-Temperature Coupling Techniques used in Lipid Monolayers Simulations

### 4.2.5 Electrostatic Interactions and Cut-off

Calculation of the system potential in a Molecular Dynamics simulation is based on the calculation of all atomic interactions. However, not all of these interactions have a noteworthy effect after a certain distance, a cut-off is introduced to limit the computational cost. The type of cut-off is related to the type of forces examined but also to the system properties. For example, Van Der Waals interactions are very small after a short distance but electrostatic Coulomb forces between dipoles have a long range [23].

According to Tieleman et al. [23], the most common cutoffs can be distinguished in four main categories:

*Cylindrical Cut-off.* As the name suggests, only the interactions inside a cylinder which is perpendicular to the lipid/water interface are calculated. The rest can be calculated by solving the Poisson Equation but researches showed that are negligible and can be omitted [104, 105]. The method however does not work for systems that do not have a cylindrical symmetry and also becomes very expensive in large systems.

*Spherical Double Cut-off.* In this method, all the interactions within a certain radius are calculated. The Double refers to the separation of interactions into two groups. The Van Der Waals are calculated in each timestep for approximately 1nm [23] and the electrostatic interactions every 10 timesteps for 1.5-2nm. In general it was a popular method which gave good results [95, 106, 107] but has been replaced by the more accurate Ewald summation and its various improvements.

*Multipole Expansion Method.* The method was introduced by the research team of Schulten in the 1990s [108, 109] and it was used in conjunction to stochastic boundary conditions to simulate the transfer of energy from the boundaries to the areas outside the system, by imposing a friction force to random boundary atoms (based on Langevin dynamics). This allowed the correct treatment of the Coulomb forces but the method was later abandoned.

*Ewald Summation method.* This method developed by P.Ewald in 1921 [110], was initially used for the calculation of electrostatic interactions in crystals but has recently become very popular in the MD community with the *Particle Ewald Mesh (PEM) method*, mainly because of its high accuracy. It is used to calculate the short and long range interactions by solving the Poisson Summation formula. In

particular the method splits the interactions to two parts, the short-range which are solved in the real space for each particle and the long-range which are solved in the Fourier Space. In that way the speed is severely increased in comparison to Poisson summation in real space [111] but in order to be accurate the method requires Periodic Boundary Conditions. The last years many improvements have been developed such as the the *Smooth Particle Mesh Ewald* method (SPME) [112], *Fast Particle Ewald Mesh* Method [113], the *Particle-Particle Particle Mesh Ewald* (P<sup>3</sup>ME) [114] and others.

### 4.3 Studies on $O_2$ diffusion through the surfactant

In the literature, several studies regarding the permeation of oxygen have been found for lipid bilayer systems. Even for the cases where lipid monolayer studies existed, the majority of them were based on experiments or proposed mathematical/analytical models and in reality only a few similar (but not exactly the same) cases were found involving the measurement of the  $O_2$  diffusion coefficient with the use of Molecular Dynamics.

According to Baoukina et al. [32] lipid monolayers can be simulated with two ways, either by having a vacuum-monolayer-water-vacuum topology or by having a symmetrical, vacuum-monolayer-water-monolayer-vacuum which is essentially a bilayer. To apply the first, a wall potential should be used in the water bottom interface, to avoid diffusion of water particles to the top of the box and therefore on the top of lipid tails. Although it is more natural to model a monolayer with a monolayer, the use of the symmetry provides equal surface tension to both surfaces and also more sampling in the monolayer phase space.

McKinnon et al., in 1992 [99] examined the diffusion of Oxygen through a hexadecane monolayer with different cholesterol concentrations. To simulate the system, they used non equilibrium MD and in order to examine the diffusion, they pulled the Oxygen molecules through the monolayer by forcing a harmonic restraint potential and running umbrella sampling simulations. Their studies indicated increased diffusion with high cholesterol concentrations and their results were in close relation to the experimental values obtained by Quenched Pyrene Fluorescence by Fishkoff and Vanderkooi [115].

Similarly to Oxygen, Choe et al. [98] examined the interaction of alveolar surfactant and a carbonaceous nanoparticle. The composition of the surfactant was modelled

as DPPC lipids with SP-B protein. The nanoparticle was dragged through the surfactant using a potential (similarly to McKinnon et al. [99]) and it was found that it can easily penetrate it but cannot translocate to the water phase. Further research and MD simulations on the effect of nanoparticles to lipid membranes can be found in the literature [116].

Apart from the computational studies mentioned above, also some theoretical work exists in the literature. Aberg et al. [117] on 2010, proposed an alternative theoretical model to calculate the diffusion of  $O_2$  and  $CO_2$  not only for the alveolar surfactant but for the whole system from surfactant to capillary.

Experimentally, the work of Olmeda et al. [118] on 2010 showed the pulmonary surfactant facilitates oxygenation especially as it folds inside the liquid subphase. On the other hand, Ivanov et al. on 2004 [119] examined the permeability of lipid monolayer to dioxygen and they obtained three to four orders of magnitude smaller permeation coefficients concluding that lipid monolayers by their own consist a noteworthy barrier and they suggested the existence of an alternative way of Oxygen diffusion through special channels.

## 4.4 Validation of Simulations

Validation of the simulation results is a crucial part of the Thesis and a lot of effort was spent in finding appropriate and reliable values to compare the produced results. The validation process was separated in two steps; initially, the constructed models were checked for their structural integrity and topology with the use of the Deuterium Order Parameters and also for appropriate density profiles, Temperature Convergence and Mean Square Displacement (MSD) of the phosphorus atoms. Finally, the second stage of the thesis, which involved the calculation of the Diffusion Coefficient of Oxygen, was validated by comparing the calculated coefficients with data from experiments and other simulations.

### 4.4.1 Model Validation

**Deuterium Order Parameters** is a typical method of evaluating the structural order of lipid systems because it is trivial to measure it experimentally with Nuclear Magnetic Resonance (NMR) and infrared spectroscopic techniques by deuterating molecules. Although the methods are widely used for experimental studies of lipid



bilayers [120–122], according to Gericke et al. [123], they are not as sensitive as required for monolayer studies and in their work used Infrared Reflection-Absorption spectroscopy. Their results showed qualitatively that for DPPC monolayers, the head region was more ordered than the tail and that compression increased the order of the system.

Quantitatively, no experimental values were found on the literature and further research of molecular dynamics studies on the topic revealed a lack on experimental validation data. In some cases, the values of Douliez et al. [124] for the  $S_{CD}$  parameters of Sn-2 chain were used although they are based on a bilayer. However, according to Kaznessis et al. [125] due to the fact that acyl chains of a monolayer are not constrained by other chains (as in a bilayer), their deuterium order results would be different.

Moreover, order parameters of the system are dependant on Temperature, phase, Area per lipid and other. Therefore, although some molecular dynamic simulations have been conducted, not an exact same case as in this thesis, was found to compare. In particular, validation results for  $S_{CD}$  of chains sn-1 and sn-2 were researched, for a DPPC monolayer at Solid Condensed phase at 323.15K and in 50, 60 and 70 Å<sup>2</sup> Area per Lipid. In Table 4.3 are presented the most similar studies that could provide possible validation data for the examined case with the case of Pickholz et al. [126] to be the most relative. In their studies they examined the effect of Neutral Chlorpromazine in Zwitterionic Phospholipid monolayers where they also constructed a pure DPPC lipid monolayer.

Publication	Temperature [K]	Model	APL [Å <sup>2</sup> ]
Dominguez 1999 [127]	323	Monolayer	55
Kaznessis 2002 [125]	323.15	Monolayer	55,60,70
Pickholz 2006 [126]	323	2x Monolayers	50,60,70
Rose et al. 2008 [88]	310	Monolayer	50,60,70
Shushkov 2010 [79]	300	2x Monolayers	50,60,70
Lucas 2012 [128]	323	2x Monolayers	64

Table 4.3: Similar studies found in the literature that could provide validation data of Deuterium Order Parameters for the evaluation of the produced results.

**Density Profile** and **Mean Square Displacement** are parameters to be examined in conjunction with other simulations but due to the nature of the data the comparisons can only be qualitative.

Finally, the simulations should converge to the **reference Temperature** of 323.15K

and the sufficiency of convergence is one of possible validation criteria.

#### 4.4.2 Diffusion Coefficient Validation

Since not exactly same studies as in this Thesis were found on the literature, in order to validate the results from the umbrella sampling simulations, a variety of experimental and computational values were used.

The diffusion coefficient can be separated in three different areas, the water slab, the lipids and the vacuum. For the latter the diffusion coefficient is not of interest since the oxygen molecule can move freely and its motion is ballistic.

For the water slab, several studies have been conducted and some of them are presented in Table 4.4. Also for lipid bilayers the diffusion coefficient for different cases, are shown in Table 4.5.

Publication	D · 10 <sup>-5</sup> [cm <sup>2</sup> /s]	Notes
Ferrell and Himmelblau [129]	3.33	$T = 313.15K$
Krieger et al. [130]	3.49	$T = 302.75K$
Han and Bartels [131]	2.78	$T = 313.12K$

Table 4.4: Diffusion coefficients of  $O_2$  in water

Study	D · 10 <sup>-5</sup> [cm <sup>2</sup> /s]	Notes
Marrink and Berendsen [132]	2	$T = 350K$ , DPPC
McKinnon et al. [99]	2.6	$T = 300K$ , hexadecane
Fishkoff and Vanderkooi [115]	6.8	$T = 318K$ , hexadecane
Al-Abdul-Wahid et al. [133]	3.7	$T = 318K$ , MLMPC
Kowert and Dang [134]	2.66	$T = 298.7K$ , hexadecane
Ju and Ho [135]	2.49	$T = 295.15K$ , hexadecane

Table 4.5: Diffusion coefficients of  $O_2$  in lipid bilayers



## Simulation Approach and Analysis

### Contents

---

<b>5.1</b>	<b>Tools and Resources . . . . .</b>	<b>57</b>
5.1.1	Software . . . . .	57
5.1.2	Hardware . . . . .	58
<b>5.2</b>	<b>Topologies and Force Fields Preparation . . . . .</b>	<b>58</b>
5.2.1	CHARMM36 Installation . . . . .	58
5.2.2	Monolayer and Symmetry Topologies . . . . .	58
5.2.3	Domain of Solution . . . . .	60
5.2.4	Oxygen Molecule . . . . .	61
<b>5.3</b>	<b>Simulation Parameters . . . . .</b>	<b>62</b>
5.3.1	General Simulation Parameters . . . . .	62
5.3.2	Thermostat for Temperature Coupling . . . . .	63
5.3.3	Area per Lipid . . . . .	63
<b>5.4</b>	<b>Simulation Process . . . . .</b>	<b>64</b>
5.4.1	Construction of the Lipid Monolayer . . . . .	64
5.4.2	Calculation of the Diffusion Coefficient . . . . .	69
5.4.2.1	Umbrella Sampling method . . . . .	70

---



## Chapter Summary

In this Chapter, the Solution Approach that was followed in the Thesis is presented. First, the computational tools that were used and also the computational resources on which the problem was solved are analysed. Then, the necessary pre-processing procedures as well as the most important parameters of the problem are presented. In the end of the chapter the whole simulation process is explained in detail for both the stages of the Thesis, the construction of the model and the calculation of the Oxygen diffusion coefficient. A brief introduction to the Umbrella Sampling method is also made.

## 5.1 Tools and Resources

### 5.1.1 Software

For pre- and post- processing of the simulation data, the **Visual Molecular Dynamics (VMD - version 1.9.1)** software was used. Being developed since 1996 by the Theoretical and Computational Biophysics Group of University of Illinois [136], VMD is one of the best available software for visualisation and manipulation (through additional packages) of molecular geometries and simulation results.

Simulations were conducted using the **GROMACS 4.6.1 (GRoningen MAchine for Chemical Simulations)** Molecular Dynamics Code which is primarily designed for biochemical molecules such as lipids and proteins and is ideal for the examined lipid monolayer system. Also GROMACS is optimized and has extremely high performance in parallel simulations while by version 4.6 [48] supports CUDA-based GPU acceleration. GROMACS was initially designed by University of Groningen in the Netherlands but at the moment many institutions internationally are participating in the development and especially Uppsala University in Sweden and many publications exist regarding its development [137–142].

Several other software were used during the solution process. For instance, in order to construct the  $O_2$  Oxygen Molecule and correctly import it in the lipid topology, the **Avogadro v1.1.0** molecule editor was utilized. Finally, the error analysis and statistical manipulation of data was made with user defined codes in **MATLAB 2012a** package and for the plots the **GNU PLOT v4.6** program was used.

## 5.1.2 Hardware

Molecular Dynamics is a significantly expensive method of simulating problems and for this reason the most state-of-the-art facilities of Cranfield University were used to solve the various cases. That is (as for Summer 2013) the ASTRAL 2 - High Performance Computer (HPC) which is consisted of 80 compute nodes of two Intel E5-2660 (Sandy Bridge) CPUs each resulting in 16 cores for each node. In total ASTRAL 2 has 1280 cores that can be used for calculations, giving in total a theoretical peak processor performance  $R_{max}$  of 22.5 TFlops (and 19.9 TFlops best measured)[143].

## 5.2 Topologies and Force Fields Preparation

### 5.2.1 CHARMM36 Installation

For the simulation of the system it was decided to use the newest version of the CHARMM force field optimised for lipid-water systems. However, GROMACS 4.6.1 by default does not include the newest CHARMM36 Force Field and therefore a manual installation was required. The file was downloaded by the official GROMACS website<sup>1</sup> and after extraction was placed in the appropriate directory<sup>2</sup>. No further configurations were needed.

### 5.2.2 Monolayer and Symmetry Topologies

Based on the literature review (see Section 4.3), two methods to model lipid monolayer systems have been examined. The first is the naturally observed lipid leaflet with a water slab below it and the second was the "reverse bilayer model" where two lipid leaflets enclosed a water slab.

Also, every lipid molecule that is simulated by molecular dynamics, has to follow certain rules of atomic topology in order to be compatible with the applied force field, e.g. for CHARMM36 is required to have 130 atoms (50 heavy and 80 hydrogens). However, since the construction from scratch of a new lipid molecule compatible with CHARMM forcefield would be out of the aims of that Thesis, an already pre-constructed lipid system was used. Unfortunately, no monolayer

---

<sup>1</sup>[http://www.gromacs.org/@api/deki/files/184/=charmm36.ff\\_4.5.4\\_ref.tgz](http://www.gromacs.org/@api/deki/files/184/=charmm36.ff_4.5.4_ref.tgz)

<sup>2</sup>[/gromacs-4.6.1/GMX\\_INSTALL/share/gromacs/top](http://www.gromacs.org/@api/deki/files/184/=charmm36.ff_4.5.4_ref.tgz)

topology compatible with CHARMM36 was available and therefore, based on a pre-simulated bilayer (323.15K,NPT,40ns,72 lipids) system from Klauda's website<sup>3</sup> [144], two monolayer systems were constructed.

The **monolayer topology** was created by deleting the water molecules and the top lipid leaflet of the Klauda's bilayer. The process was done in VMD with the Monolayer.tcl script (Appendix A.1).

The **symmetry topology** was created similarly by deleting the water molecules and translating the upper leaflet 10nm lower than the bottom (exact coordinates were calculated by *Geometrical\_Parameters\_SYM.m* - Appendix A.4 ), in order to make space for the new water slab. Once again, the process was made in VMD by using the Symmetry.tcl script (Appendix A.2)

Figure 5.1 shows the pre-processing of the initial file for both cases.

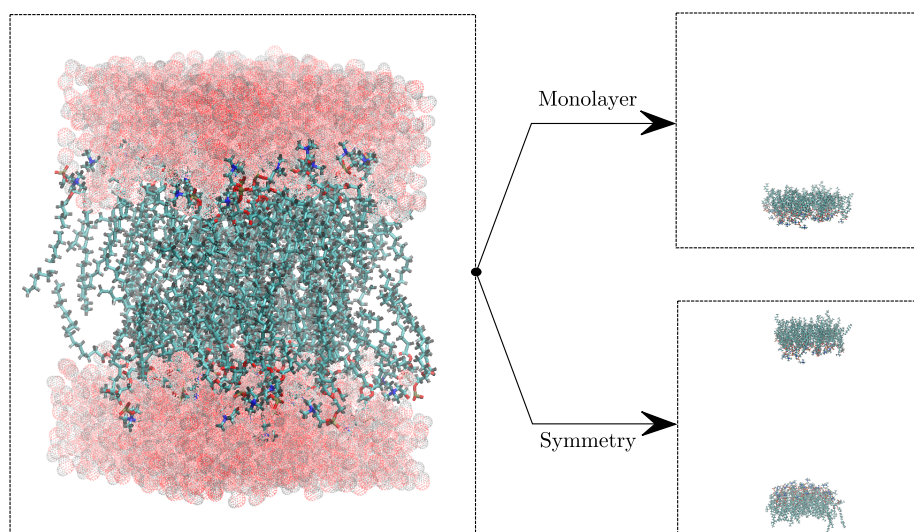


Figure 5.1: Process of topology generation for each model. On the left is the original Klauda's model. On the top right is the generated monolayer and on the bottom the symmetrical. In both generated, water molecules have been omitted.

<sup>3</sup><http://terpconnect.umd.edu/~jbklauda/research/download/dppc72-c36npt.pdb>



### 5.2.3 Domain of Solution

The domain of the two different models can be seen in Figure 5.2. The x-y dimensions are varying according to APL (see Section 5.3.3) and the z is taken as 150nm to make sure that no interaction between the lipid chains takes place, even for the long range electrostatic forces since the length of the box does not affect the solution time in terms of extra calculations. However, it can make it slightly inefficient in parallelization due to the fact that the PME load per node will be asymmetric (Fourier Grid Spacing from Ewald summation will be too large in one dimension). As it can be observed, in order to be assured for the consistency of the comparisons of the systems, the symmetric case is an exact mirrored double of the monolayer.

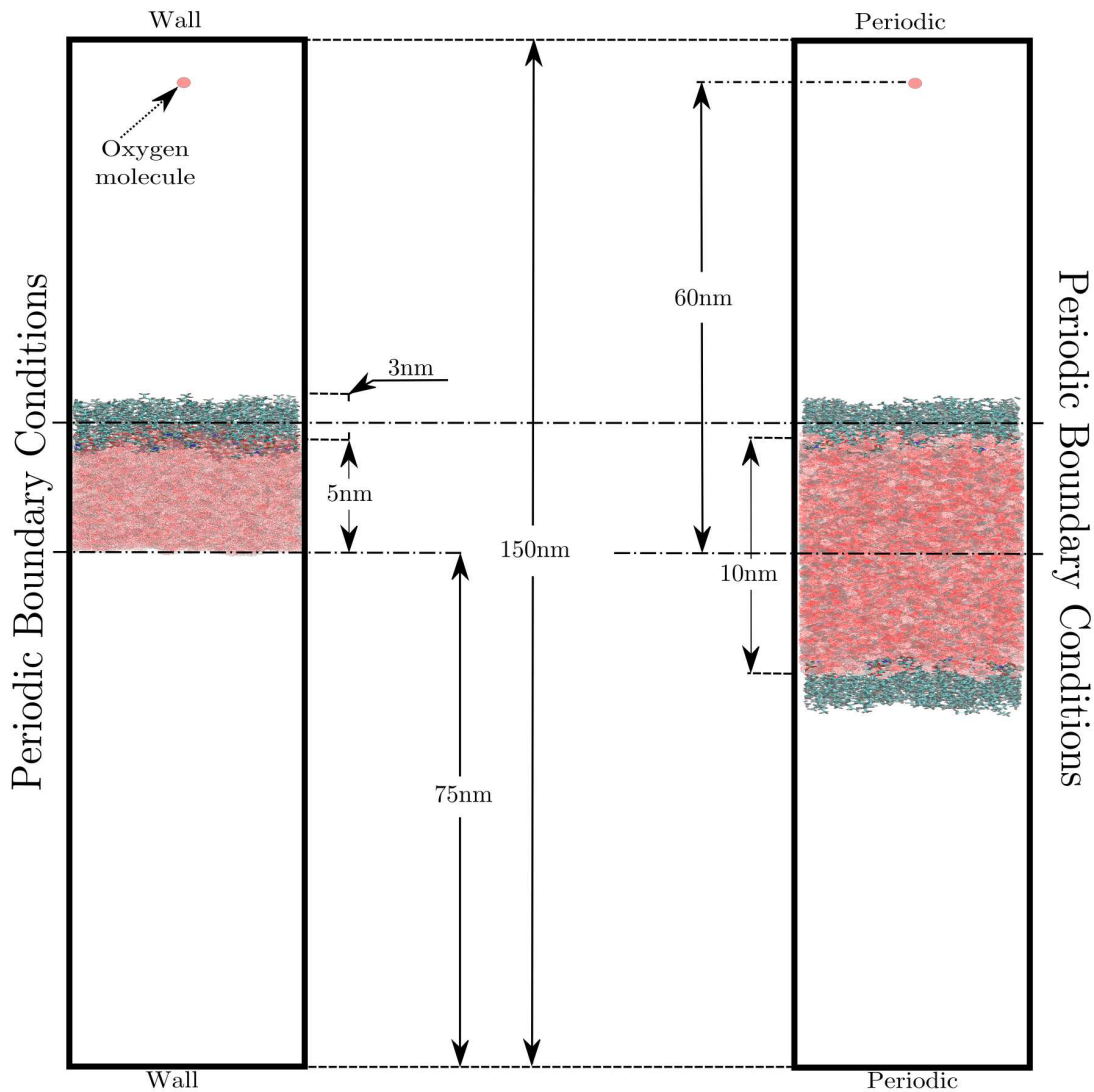


Figure 5.2: Schematics of the Solution Domains. On the left is the monolayer topology and on the right the symmetrical. The different boundary conditions on z axis are also shown. Scales are not realistic for clarity.

Another characteristic of both systems is the existence of vacuum around the topologies. Based on literature and previous works on monolayer simulations, that was considered the most appropriate and cost-effective way to model the low density of air.

Finally, the boundary conditions of each model were different. Usually in MD problems, all boundaries are considered periodic. In this case, that was true only for the symmetric topology where all the water molecules were constrained by the lipid leaflets. For the simple monolayer on the other hand, there was a possibility that water molecules would escape the slab and travel through vacuum to interact with the hydrophobic chains. To reduce this possibility, wall boundary conditions were used in the z boundaries, simulated as single layer of Argon atoms. Table 5.1 shows the specific details.

#### 5.2.4 Oxygen Molecule

The oxygen molecule before inserted in the lipid topologies in GROMACS, had to be first created. With the use of the Avogadro molecule editor, the process was quite trivial. First two oxygen molecules were placed arbitrarily in space and then their geometry was optimized to the minimum energy state (Fig. 5.3). Then the geometry was exported as a .pdb file from which the coordinates were used as input in the Geometry Parameters MATLAB Codes, to calculate the translated coordinates for the implementation in the final system (see Section 5.4 for details).

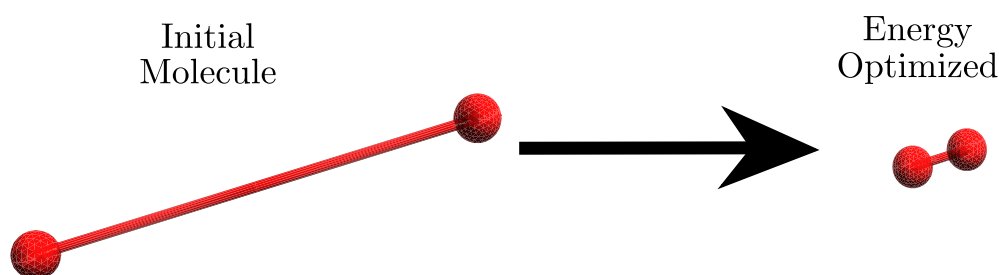


Figure 5.3: Transformation of Oxygen Molecule during its construction with the Avogadro Molecule Editor.

In order to import this molecule in GROMACS without adding a new molecule type in the CHARMM36 force field, it was decided to use the physical properties of an

existing Oxygen molecule. Since not peer-reviewed literature was found regarding such a specific case, in the end the guidelines proposed in the official CHARMM Development Project forum<sup>4</sup> to use the HEME Oxygen molecule type with the charges set to zero, were used.

Wall Properties	
Positions	$z = 0$ and $z = z_{max}$
Atom Type	Argon (Ar)
Potential	Lennard-Jones 9-3
Density	20 atoms
Scaling	$3 \cdot z_{max}$

Table 5.1: Properties of the Argon atom layers used as wall boundary conditions in the z direction

Validation Simulation		
Model	T Coupling	APL
Monolayer	Nose-Hoover	0.5
		0.6
		0.7
Monolayer	V-Rescale	0.5
		0.6
		0.7
Symmetry	Nose-Hoover	0.5
		0.6
		0.7
Symmetry	V-Rescale	0.5
		0.6
		0.7

Table 5.2: Table of simulations that ran to validate the constructed models.

## 5.3 Simulation Parameters

The first stage of the Thesis aimed to the construction of a correct and consistent model of pulmonary surfactant system which could be considered the basis for other researches. Three main parameters were examined during the first stage and the complete simulation plan that was followed is displayed in Table 5.2.

### 5.3.1 General Simulation Parameters

The simulated system was consisted of 36 DPPC lipid molecules per leaflet and according to literature, an increase of the number of lipids above 36 would not lead to a significant change in the results [145, 146]. Also, the Temperature was kept constant at 325.15K in order to avoid phase changing from the solid crystalline.

<sup>4</sup><http://www.charmm.org/ubbthreads/ubbthreads.php?ubb=showflat&Number=28482>

For all simulations the Verlet cut-off scheme was used. The Van der Waals and electrostatic interactions were limited to a cut-off range of 1.0nm. In particular the Lennard-Jones potential was used for the Van der Waals and the Fast-Smooth Particle-Mesh Ewald Electrostatics (PME) for the Coulombic forces. The Fourier grid spacing was 0.15nm and the interpolation of PME was of fourth order. Also, long-range dispersion correction of Energy and Pressure was applied. The atomic bonds were constrained with LINCS algorithm for every iteration by using fourth order expansion. Furthermore, in every case and for every iteration, the Center of Mass translation was removed for every group.

For the **Energy Minimization** of the system the steepest descent algorithm was used with 50,000 as the maximum number of iterations and the 100 kJ/mol/nm as the convergence criteria.

In the **NVT Equilibration**, the Leap-frog integrator was chosen for the Newton's Equations of motion, with a time step of 1fs (0.001ps) and total simulation time of 1ns or 1,000,000 time steps. Data were saved every 0.5ps to keep the amount of trajectory data less than 3GB. Finally, the velocity was generated according to Maxwell distribution for a Temperature at 323.15K.

### 5.3.2 Thermostat for Temperature Coupling

The second parameter that was examined during the model construction, was the thermostat for the Temperature coupling and the effect on the results. Two different thermostats were used to keep the temperature constant, V-Rescale and Nose-Hoover, both explained in Section 3.4.5. Both the thermostats were examined during the NVT Equilibration and only the Nose-Hoover for the Production Molecular Dynamics simulations. Every system component (DPPC,  $O_2$ , SOL) was coupled separately.

### 5.3.3 Area per Lipid

In the literature APL ranges from 0.40 to 0.80  $nm^2$ . Especially for monolayers composed by DPPC, for APL less than 0.40 $nm^2$ , the compression is so high that lipid chains almost touch each other, in the solid crystalline form [147] and start to squeeze out of the leaflet structure. On the other hand for expansion above 0.80  $nm^2$ , the monolayer becomes too coarse, holes are created and finally water penetrates it. In this thesis three different APL were examined at 0.50,0.60 and

$0.70 \text{ nm}^2$  which were controlled by changing the x and y dimensions of the box and consequently scaled appropriately the lipid topology inside that. Dimensions were extracted from the Area Per Lipid definition:

$$\text{APL} = \frac{\text{Area}}{N_{\text{lipids}}} \quad (5.1)$$

where  $N_{\text{lipids}}$  is the number of lipid molecules. Since the x and y dimensions are equal, equation 5.1 becomes:

$$\text{APL} = \frac{x^2}{N_{\text{lipids}}} \quad (5.2)$$

$$\Rightarrow x = \sqrt{\text{APL} \cdot N_{\text{lipids}}} \quad (5.3)$$

which gives the dimensions of the box for each case. The above calculation for every APL was included in the codes *Geometrical\_Parameters\_MN.m* or *Geometrical\_Parameters\_SYM.m* for the two types of models, presented both at the Appendices A.3 and A.4.

Finally, in some cases, e.g. for APL equal to  $0.50 \text{ nm}^2$ , after the geometry was imported to GROMACS, under-scaling was necessary in order to avoid rapid changes of the dimensions that could cause extreme forces (and divergence in energy minimization). Therefore, according to the required x-y dimensions, a Scaling Factor (SF) was used, to reduce the initial geometry to the desired area. The calculations were part of the Geometrical Parameters codes.

## 5.4 Simulation Process

### 5.4.1 Construction of the Lipid Monolayer

The simulation process of the first stage is presented in Figure 5.4. After the two **lipid models were constructed** in two different topologies, they were **imported to GROMACS** by using the `pdb2gmx` command in order to transform it to a GROMACS coordinate (`.gro`) and topology (`.top`) file. The **CHARMM36** force field was selected for the lipids and the **TIPS3P** for the water (SOL) molecules. An extra restrain file (`.itp`) was created for the topology but was not used in the next steps.

```
pdb2gmx -f lipid_geom.pdb -o dppc_nw.gro -p topol.top
```

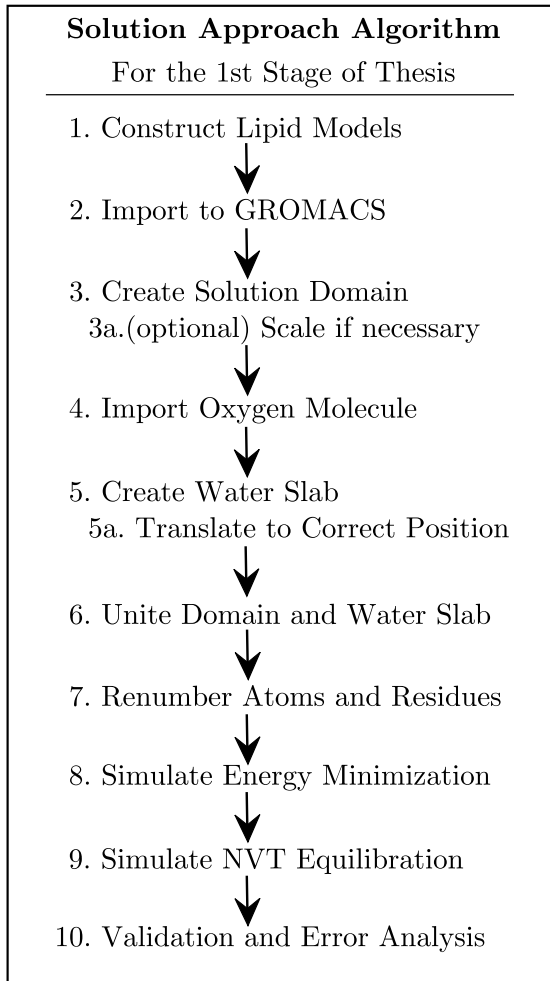


Figure 5.4: Algorithm of the steps followed in the first stage of the thesis regarding the construction of the lipid monolayer models

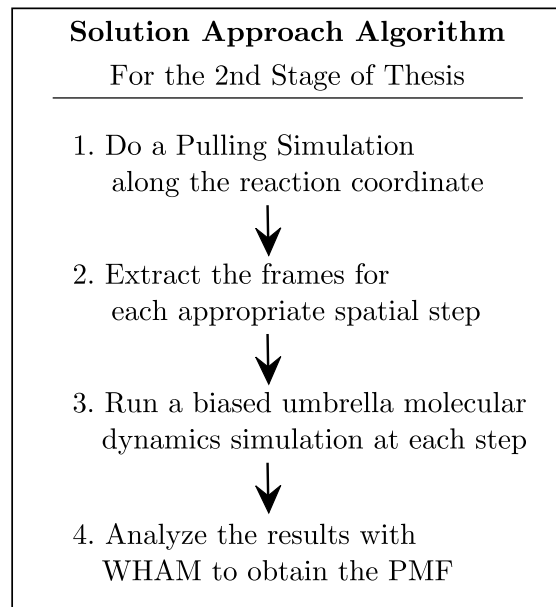


Figure 5.5: Algorithm of the steps followed in the second stage of the thesis where the Umbrella Sampling Technique was used to obtain the Free-Energy Difference and Diffusion Coefficient of the Oxygen Molecule

After that, the **solution box** was defined with dimensions based on the chosen APL (for details see Section 5.3.3)

```
editconf -f dppc_nw.gro -o dppc_box.gro -box x_mat y_mat 150 -center
      xc_mat yc_mat zc_mat
```

In the case of  $0.50nm^2$  however a scaling of the geometry was also necessary because the reduce of the box was too rapid and the atomic forces were not able to minimize. To fix this problem a scaling of the lipid geometry took place before, based on the scaling factors calculated in the Geometrical Parameters MATLAB Codes:

```
editconf -f dppc_nw.gro -o dppc_scale.gro -scale SFx SFy SFz
```

To **import the Oxygen molecule** correctly, both the coordinate and topology files had to be modified<sup>5</sup>. In order to avoid movement of the oxygen molecule and any interaction with the lipid-water system before the second stage, the oxygen molecule was placed 60nm away from the lipid leaflet. Since the charge of the molecule was zero, no interaction of the oxygen and the lipid leaflet was observed.

### Coordinate File Alterations (*rename dppc\_box.gro to dppc\_o2.gro*)

1. In the total atom number on the top, two more were added
2. After the end of the DPPC atoms (normally in the end of the file) and just before the box vectors, the following were added:

```
atom+1 01 atom+1 01x 01y 01z
atom+2 02 atom+2 02x 02y 02z
```

where x,y,z were based on the original Avogadro coordinates and their translation in the box (Geometrical Parameters MATLAB Codes).

### Topology File Alterations

1. In the end of the [atoms] section the following lines were added (the 0.0 in the seventh column denotes the zero charge of the atoms):

---

<sup>5</sup>It has to be noted that extra care should be given when alterations of GROMACS files are done. For instance the correct placement of data in the appropriate column number is crucial, since even one or two spaces away can cause fatal errors in the next steps.

```
atom+1 OM residue+1 O2 O1 atom+1 0.00 15.999
atom+2 OM residue+2 O2 O2 atom+2 0.00 15.999
```

2. In the end of the [bonds] section the following single line was added:

```
atom+1 atom+2 1
```

The next step was the **creation of the water slab**. Initially, the water slab was created in a box of height  $z$  (5nm for the monolayer or 10 for the symmetrical) and  $xy$  equal to the previous steps. Then the box was translated to the appropriate position (Geometrical Parameters MATLAB Codes) into a 150nm-height box. In that way the coordinates of the water molecules became correct and a unification with the lipid solution box was possible:

```
genbox -cs spc216.gro -o solv.gro -box x_mat y_mat z("5 or 10")
>
editconf -f solv.gro -o solv_final.gro -box x_mat y_mat 150 -center
xc_water_mat yc_water_mat zc_water_mat
```

The solvent was the spc216 which was the same water type for GROMOS, TIP3P and TIPS3P force fields.

Then the two **files were united** by adding the contents of the solv\_final.gro to the dppc.o2.gro:

### Coordinate File Alterations (*rename dppc\_o2.gro to dppc\_united.gro* )

1. In the top, the water atom numbers (easily found by the end of solv\_final.gro) were added in the DPPC and oxygen atom numbers.
2. The water atoms were copied and pasted after the second oxygen atom line and above the box vector lengths<sup>6</sup>.

### Topology File Alterations

---

<sup>6</sup>Caution should be given, not to add an extra space line or move the new lines in different columns



1. In the end of the [molecules] section, in the end of file, the number of molecules (observed by the last residue number in the solv\_final.gro) were added:

```
[molecules]
...
SOL      (number_of_mols)
```

It is obvious that the current file unification would create problems as the water residue and atom numbers restarted from 1. Therefore, to **renumber the new file**, the following command was used, to finally obtain the lipid-oxygen-water topology:

```
genconf -f dppc_united.gro -o dppc_final.gro -renumber
```

In order to **minimize the energy state** of the final system, an energy minimization simulation was ran (for details see section 5.3.1). To perform it, the parameter file em.mdp was necessary (see Appendix B.1) to create the run input file .tpr and then the minimization was executed:

```
grompp -f em.mdp -c dppc_final.gro -p topol.top -o em.tpr
>
mdrun -v -deffnm em
```

Finally, the minimized system was **equilibrated under the canonical (NVT) ensemble** (see section 5.3.1), for 1ns, to loose the correlation with the initial conditions and to provide the final states that the comparison were based on. The nvt.mdp (Appendix B.2) parameters file was used to create the nvt.tpr and then run the simulation:

```
grompp -f nvt.mdp -c em.gro -p topol.top -o nvt.tpr
>
mdrun -v -deffnm nvt
```

After that, a comparative analysis of the simulated cases was made (see Section 6.3) in order to check the validity of the model before initiating the production simulations.

### 5.4.2 Calculation of the Diffusion Coefficient

In order to calculate the diffusion coefficient a correlation to a quantity measurable with Molecular Dynamics had to be made. To achieve that, initially the Einstein equation was used:

$$D = \frac{k_B T}{f} \quad (5.4)$$

where  $D$  was the Diffusion Coefficient,  $k_B$  was the Boltzmann's constant,  $T$  was the temperature and  $f$  was the friction coefficient of the oxygen molecule. Also, according to Langevin Equation the friction coefficient was related to the Friction force  $F_{fr}$  as:

$$\frac{d\langle v \rangle}{dt} = \langle F_{fr} \rangle - f\langle v \rangle + \langle k(t) \rangle \quad (5.5)$$

with  $\langle v \rangle$  being the average velocity,  $\langle F_{fr} \rangle$  being the Frictional Force averaged over all the simulation and  $\langle k(t) \rangle$  being the averaged random force due to the solvent which was zero by definition. Also the average acceleration of the oxygen particle was zero because during the Steered MD simulation the oxygen molecule travelled through the monolayer at a constant velocity. Finally, equation 5.5 became:

$$\langle F_{fr} \rangle = f\langle v \rangle \quad (5.6)$$

By substituting Equation 5.5 to 5.4:

$$D = \frac{k_B T \langle v \rangle}{\langle F_{fr} \rangle} \quad (5.7)$$

where  $\langle v \rangle$  was the the travelling distance  $d$  of the diffusing particle over the time  $t$ :

$$\langle v \rangle = \frac{d}{t} \quad (5.8)$$

and  $F_{fr}$  was the work  $W$  required for the particle to move, over the travel distance  $t$ :

$$\langle F_{fr} \rangle = \frac{W}{d} \quad (5.9)$$

Finally the Diffusion coefficient became:

$$D = \frac{k_B T d^2}{W t} \quad (5.10)$$

where  $W$  was an implicitly measurable quantity. To get the Work, the Jarzynski Equation [148–150] was used where the work was related to the free-energy difference

$\Delta G$ :

$$\Delta G = -k_B \cdot T \cdot \log \langle e^{-\frac{1}{k_B T} \cdot W} \rangle \quad (5.11)$$

To obtain the free-energy difference  $\Delta G$ , the Potential of Mean Force (PMF) was used as:

$$\Delta G = PMF_{max} - PMF_{min} \quad (5.12)$$

which was an explicitly measurable quantity.

In reality in order to calculate  $\Delta G$ , the Free-Energy Differences had to be measured and several methods to achieve that could be utilized, such as the Umbrella Sampling, the Thermodynamic Integration, the Slow Growth, the Steered Molecular Dynamics or the Fast-Growth technique [151]. As it was noticed during the literature review, not many Molecular Dynamics studies involving the diffusion of Oxygen through a lipid monolayer existed. One of the most similar cases was the study of McKinnon et al. on 1992 [99] where they utilized the Steered Molecular Dynamics and the Umbrella Sampling Technique to extract the diffusion coefficient of  $O_2$  through a hexadecane monolayer.

Since this method is relatively easy on its preparation and the extraction of data is quite straightforward, it was decided to be followed also in this Thesis.

#### 5.4.2.1 Umbrella Sampling method

The main concept of the Umbrella Sampling Technique is to create numerous biased simulations along a reaction coordinate (e.g.  $z$  axis) and extract the unbiased potential of mean force by using the Weighted Histogram Analysis Method [152]. In every of these simulations the oxygen molecule is restrained in its position by using a simple harmonic potential  $U(t)$  which behaves similar to a spring, allowing the continuous sampling of the area near the reference position. The potential is defined as:

$$U(t) = \frac{1}{2} k_h (z(t) - z_{con}(t))^2 \quad (5.13)$$

with  $z(t)$  being the current position of Oxygen Center of Mass,  $z_{con}$  the center of the restraining potential [99] and  $k_h$  the Force Constant of Harmonic Potential.

In Figure 5.5 the basic algorithm of the umbrella sampling is displayed. These exact steps were also followed during the second part of this thesis. After the validation of the created models in stage 1, the Symmetry model with APL  $60\text{\AA}^2$  and Nose-Hoover Thermostat was selected. Because in the first stage, the oxygen molecule

was far above the lipid-water system, a fast pulling simulation was performed for 530ps and pull rate equal to 0.104[nm/ps] to bring it close to the acyl chains of the monolayer.

```
grompp -f md_pull.mdp -c nvt.gro -p topol.top -t nvt.cpt -o md_pull.tpr
>
mdrun -v -deffnm md_pull
```

Then another pulling simulation took place for 500ps with pull rate equal to 0.01 [nm/ps] (*md\_pull.mdp*, App.B.3) which covered approximately 5nm and pulled the oxygen from the top vacuum until inside the water slab. In Figure 5.6 the two simulations are shown as well as the whole distance covered by the oxygen molecule. The pulling was made relatively to the reference group (DPPC double monolayer) and the pulled group ( $O_2$  molecule).

```
grompp -f md_pull.mdp -c md_pull.gro -p topol.top -t md_pull.cpt -o
  md_pull_final.tpr
>
mdrun -v -deffnm md_pull_final
```

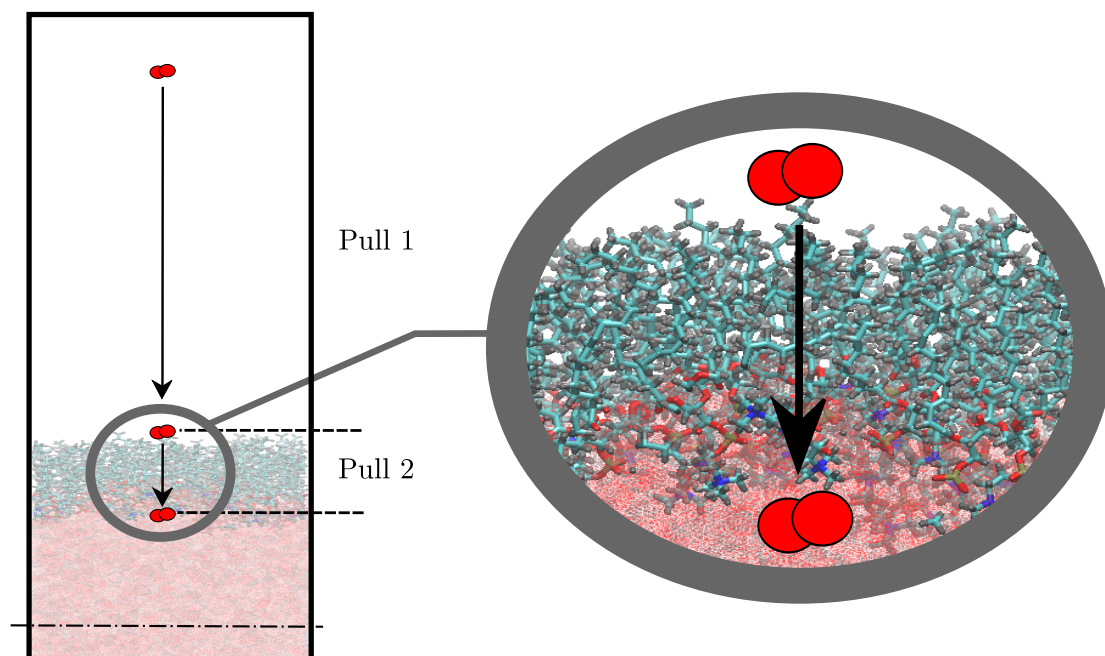


Figure 5.6: Schematic of the two pulling simulations. For clarity the scales are not realistic and the lower part of the system is omitted.

After the Steered Molecular Dynamic Simulations were completed, all the frames

of the last pull were extracted (in total 1001).

```
trjconv -s md_pull_final.tpr -f md_pull_final.xtc -o frame.gro -sep
```

A Perl Code (*distances.pl*, App.A.5) created by Justin Lemkul<sup>7</sup> was utilized to extract the distances of each frame. Since the pulling was relative to the two groups, the results had to be translated to the actual domain distances and the MATLAB code *Translate\_Distances.m* (App.A.6) was used.

```
perl distances.pl
```

Two different types of harmonic force constants were used,  $k_h = 3000$  and  $k_h = 1000$ . However, the  $k_h = 3000$  made the potential very stiff (which reduced the samples taken at each position, for more details see in the Results Chapter) and therefore the complete set of umbrella sampling simulations were performed for the  $k_h = 1000$ .

The Weighted Histogram Analysis Method (WHAM), in order to produce reliable results, requires the histograms created by each biased umbrella simulation to be overlapping. To ensure that a sufficient number of samples would be taken, in total **128 simulations** of 1ns were ran with spacing between frames equal to 0.025nm. **For every simulation** the parameters file *md.umbrella.mdp* (App.B.4) was utilized to produce the input *umbrella.tpr* for the simulation and then was executed with *mdrun*<sup>8</sup>:

```
grompp -f md_umbrella.mdp -c frameXXX.gro -p topol.top -o umbrella.tpr
>
mdrun -v -s umbrella.tpr
```

After all the 128 simulations finished, all the *pullf.xvg* and *umbrella.tpr* were gathered in a folder with the order of the frames on which they ran. The filenames of both *pullf* and *umbrella* were added in two *dat* files which then were used by the WHAM tool to produce the final histograms and PMF Plots.

```
g_wham -it tpr-files.dat -if pullf-files.dat -o -hist -unit kJ
```

---

<sup>7</sup>The code can be found at <http://www.bevanlab.biochem.vt.edu/Pages/Personal/justin/gmx-tutorials/umbrella/Files/distances.txt> and instructions of use can be found in <http://www.bevanlab.biochem.vt.edu/Pages/Personal/justin/gmx-tutorials/umbrella/tutorial.html>

<sup>8</sup>The *-s* flag is used to prevent GROMACS from merging *pullf* and *pullx* files together.

## Simulation Results and Discussion

### Contents

---

<b>6.1</b>	<b>Surfactant Models</b>	<b>75</b>
<b>6.2</b>	<b>Energy Minimization</b>	<b>78</b>
	6.2.0.2 Analysis	79
	6.2.0.3 Results	80
<b>6.3</b>	<b>NVT Equilibration</b>	<b>80</b>
6.3.1	Temperature Convergence	82
	6.3.1.1 Analysis	82
	6.3.1.2 General Observations	83
	6.3.1.3 Model behaviour	83
	6.3.1.4 Thermostat Efficiency	83
6.3.2	Deuterium Order Parameters	86
	6.3.2.1 Analysis	86
	6.3.2.2 General Observations	87
6.3.3	Density Profiles	91
6.3.4	Mean Square Displacement	95
6.3.5	Computational Cost and Effectiveness	99
6.3.6	First Stage Conclusions	100
<b>6.4</b>	<b>Umbrella Sampling Simulations</b>	<b>100</b>
6.4.1	Effect of Harmonic Force Constant	101
6.4.2	Effect of Sampling Space	102
6.4.3	Calculation of PMF	105
6.4.4	Calculation of $D_{O_2}$	107

---



## Chapter Summary

In this chapter the analysis and the results produced by the two stages of the thesis are presented. Based on the previous chapters of the literature review and the simulation approach, the results are discussed in order to extract meaningful conclusions. Initially the final models and their special characteristics are shown. Then the more important parameters of the first stage are analysed, such as the Energy Minimization and for the NVT Equilibration, the Temperature Convergence, the Deuterium Order Parameters, the Density profiles, the Mean Square Displacement of the Phosphorus atoms and the total computational cost. Then the results of the Umbrella Sampling Simulations such as the calculated Potential of Mean Force and Diffusion coefficient of Oxygen are presented and discussed.

### 6.1 Surfactant Models

As it was previously explained in Chapter 5, two different models were constructed according to previous research studies, in order to examine their properties, compare their behaviour and finally choose the most appropriate for the calculation of the Oxygen diffusion. Figure 6.1 shows the final models for the case of  $60\text{\AA}^2$  area per lipid, Nose-Hoover Thermostat and after  $1ns$  NVT Equilibration; the cases for  $50\text{\AA}^2$  and  $70\text{\AA}^2$  are almost identical and are omitted.

Although the Monolayer model has a better physical meaning (as it represents the real topology of the alveolar surfactant), it was rather unstable during the simulation in comparison to the Symmetry model which remained in the same position in the center of the box. The simulation time-lapses for both cases can be observed in Figure 6.2. Analytically, in the first picoseconds the stabilization period is clearly noticed, as the system passed from the energy minimized separated topologies, into the united DPPC-water system at  $323.15K$ . For the symmetry cases, the system did not differentiate much more after the  $50ps$  except from the change of lateral diffusion mechanism that dominated in the system (more details on that are in section 6.3.4). On the other hand, the monolayer after components stabilization, begun to move upwards.

This movement started after  $50ps$  and it was created by the interaction of escaped water molecules with the lower part of the water slab. Because the water slab was not constrained by anything in its lower boundary, some water molecules escaped and started moving in the vacuum below. As they moved downwards, they inter-



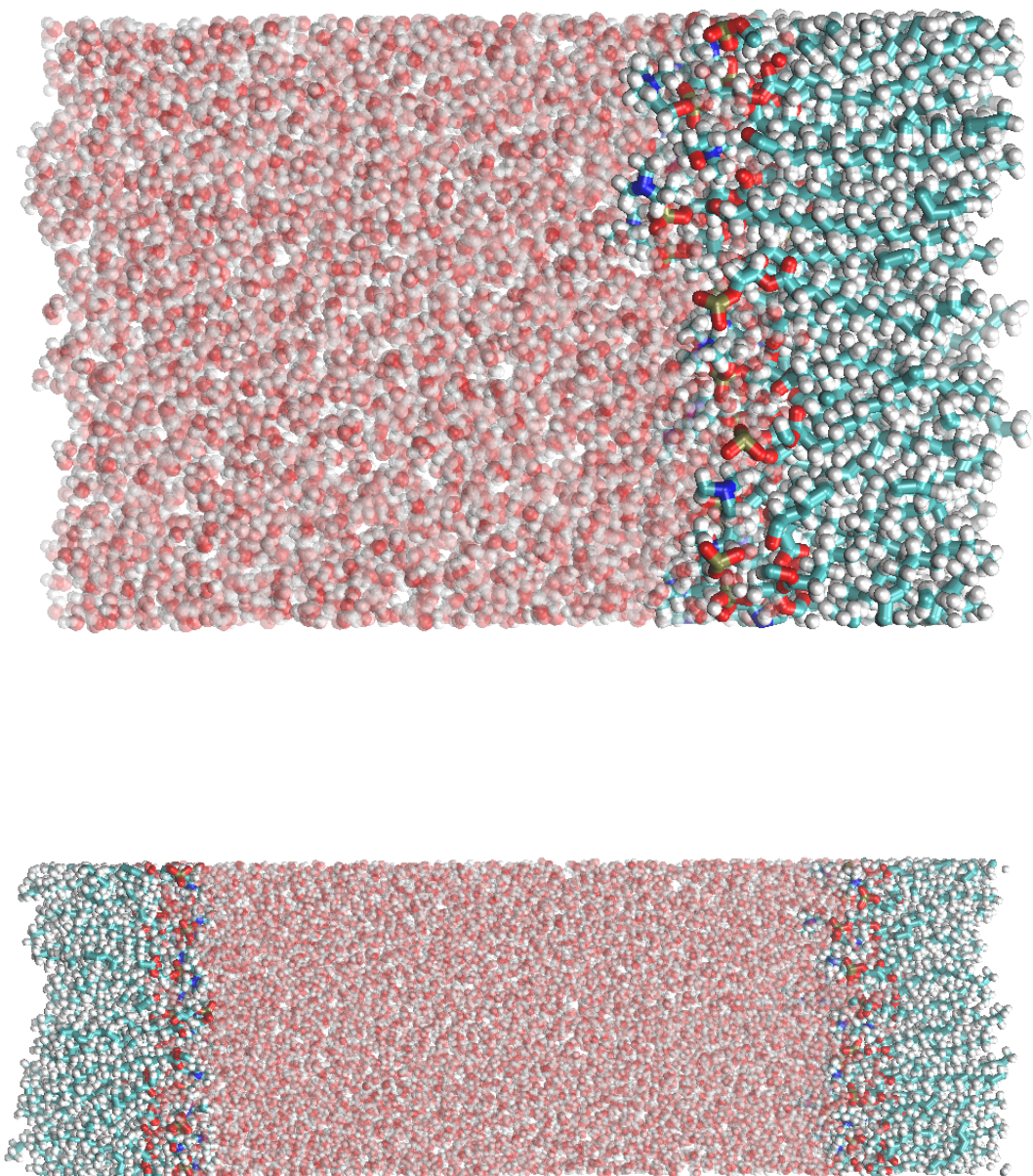


Figure 6.1: The final Models after 1ns of NVT Equilibration. On the left is the monolayer system and on the right is the symmetry. The water molecules (bottom or middle) are shown with partial transparency to make clear the submerging of the Phosphatidylcholine heads inside them. Legend: Blue=Nitrogen, Gold=Phosphorus, Red=Oxygen, Cyan=Carbon and White=Hydrogen.

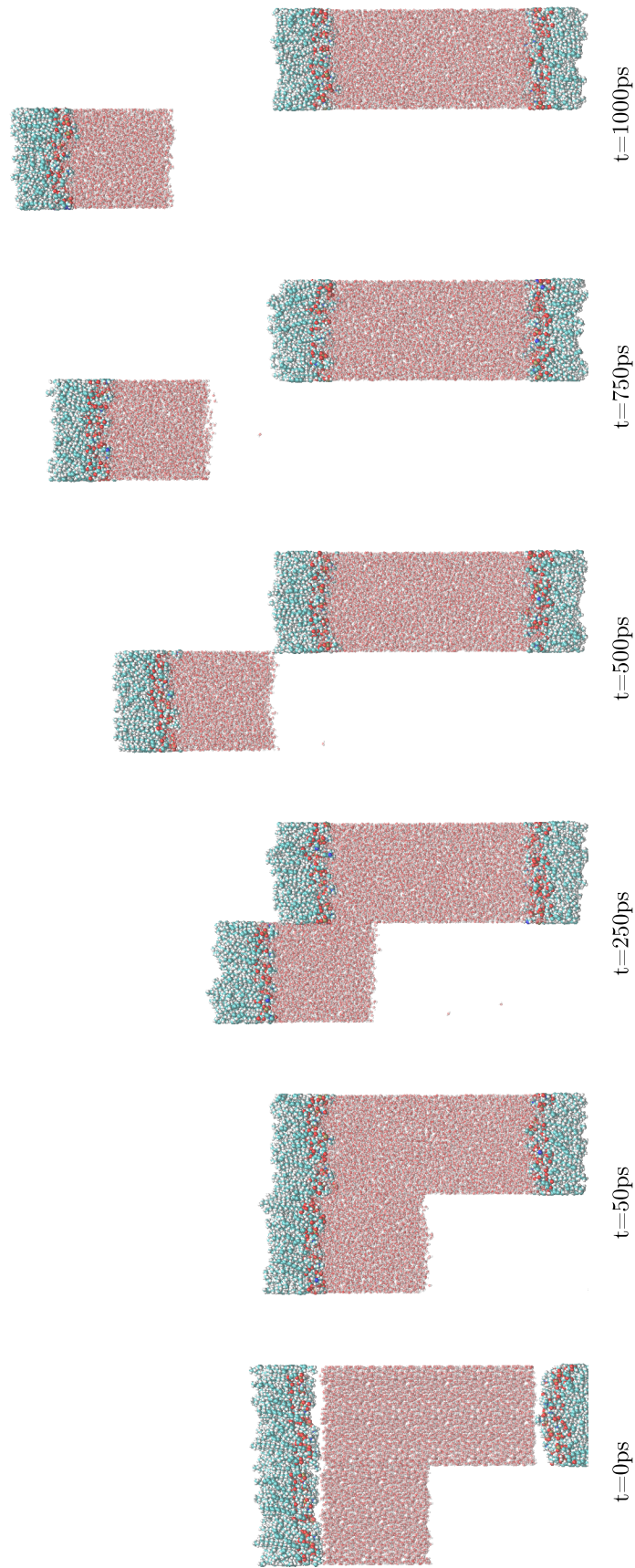


Figure 6.2: Timelapse of Ins for both models. The two systems are shown in conjunction to make clear the upward movement of the monolayer system (the left in each picture). Legend: Blue=Nitrogen, Gold=Phosphorus, Red=Oxygen, Cyan=Carbon and White=Hydrogen.

acted with the Argon "wall" layer, increased their energy state and got reflected back to the vacuum. The moment they reached again the water slab, they applied a small pressure that if summed for the total number of escaped particles, was adequate to move the system upwards. It is possible that at some point the system would reach a steady equilibrium but that would probably require a very long simulation. It was also possible however that the system would never stabilize but rather reach the upper wall where the lipid chains would interact with it and be suppressed, distorting their structure and therefore the whole simulation purpose.

Another important parameter for each case, was the number of atoms it used. In order to be consistent in every simulation, the number of atoms in the symmetry systems had to be ideally the double of the monolayer. As it can be observed by Table 6.1, that was true, since in all cases the Symmetry models were almost 100% the size of the monolayers, ensuring the fact that similar things would be compared. Table 6.1 also shows the increase of atom numbers as the A.P.L. increased, which was expected since the solution box increased in the xy plane.

Model	APL $\text{\AA}^2$	Lipid	Water	Total	Diff. %
Monolayer	50	4680	9045	13725	-
	60	4680	10788	15468	-
	70	4680	12465	17145	-
Symmetry	50	9360	17964	27324	99.08
	60	9360	21411	30771	98.93
	70	9360	24756	34116	98.99

Table 6.1: Number of atoms for each model used and for each A.P.L.. Also the percentage of difference between Monolayer and Symmetry is shown in the last column

## 6.2 Energy Minimization

In the Energy Minimization process, the molecular topology is "relaxed" from possible overlapping atoms and extreme forces that are created when the system is very far from equilibrium.

In theory, the Potential Energy function of a system is a complex surface which has a global minimum and maximum as well as many local minima and maxima. As one can recall from Mathematical Analysis the first derivatives of a function (in this case the potential energy) at a minima, is zero and all second derivatives positive or zero. The Hessian matrix is the matrix of the second derivatives which has

generally positive or zero eigenvalues, except from the specific points called saddle points where a negative eigenvalue exists. These points are the transition paths of the system to other minima [48].

Calculation of all the minima of the system is impossible due to the big complexity of the system and therefore from a starting configuration usually the closest local minimum (moving down the steepest local gradient) is calculated. GROMACS is using the "steepest descent or saddle-point" method to calculate the local minimum by taking a step every time towards the direction of the negative gradient, ignoring the previous steps. Although the method is very simple, it can be slow in convergence in the area around the local minimum. If all the coordinates (position) are defined by the vector  $\mathbf{r}$  with an initial maximum displacement of  $h_0$  then after the forces and potential energy are computed, the new positions are defined as:

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \frac{\mathbf{F}_n}{\max(|\mathbf{F}_n|)} h_n \quad (6.1)$$

with  $h_n$  being the maximum displacement and  $\mathbf{F}_n$  being the force (negative gradient) of the potential  $V$ . Then if  $V_{n+1} < V_n$  then new positions are accepted and  $h_{n+1} = 1.2h_n$  or if  $V_{n+1} \geq V_n$  the new positions are rejected and  $h_n = 0.2h_n$ . In the end, the algorithm stops either when reach the maximum number of steps or when  $\max(|\mathbf{F}_n|)$  reaches pre-defined value  $\epsilon$  [48].

### 6.2.0.2 Analysis

The Potential Energy during each simulation was extracted by using the GROMACS tool *g\_energy*. By default, the results are in [kJ/mol] units. As it was presented in the previous section, the systems have a different number of atoms. Therefore a direct comparison of the results was not possible and the data had to be transformed to something comparable, such as, the specific potential energy with units [kJ/kg]:

$$e_P = \frac{E_P}{M_{tot}} \left[ \frac{kJ}{kg} \right] \quad (6.2)$$

In order to calculate each system's total mass the following relation was used:

$$M_{tot} = \frac{N_{lipids} \cdot MW_{lipids} + N_{water} \cdot MW_{water}}{N_A} \quad (6.3)$$

with  $N$  being the Number of atoms (see Table 6.1),  $MW$  the molecular weight ( $MW_{lipids} = 0.73404$  [kg/mol] and  $MW_{water} = 0.018015$  [kg/mol]) and the Number

of Avogadro  $N_A$  ( $N_A = 6.022 \cdot 10^{23}[\text{atoms/mol}]$ ).

Then the Specific Potential Energy was normalized to check the behaviour of the systems under the same scale according to:

$$e_{P_{norm}} = \frac{e_P - e_{P_{min}}}{e_{P_{max}} - e_{P_{min}}} \quad (6.4)$$

The above procedure was performed with the MATLAB code *Error\_Analysis\_EM.m* (App.C.1)

### 6.2.0.3 Results

Figure 6.3 shows the Specific Potential Energy for every case over the required minimization time. Observing the figure, it is clear that the cases of  $50\text{\AA}^2$  were the slowest to converge, requiring more than 5ns in order to reach the local minimum. On the other hand, the  $60\text{\AA}^2$  were the fastest case to converge. Another observation is that the minimum (final) Specific Potential Energy became more negative as the APL became bigger. Nevertheless for all simulations, the results seem to be consistent, having the same minimum  $e_P$  regardless of the model used. For the cases of  $60\text{\AA}^2$  and  $70\text{\AA}^2$  the monolayers took slightly more time to converge but that was not the case for the  $50\text{\AA}^2$  cases and therefore no clear conclusion could be extracted regarding the connection of model type with convergence time.

Figure 6.4 shows the normalized Specific Potential Energy over the required simulation time. By examining all the systems under the same scale, the speed of convergence becomes much clearer. The cases of  $70\text{\AA}^2$  is reaching almost instantly the minimum potential energy but then needs a lot of time to reach the final minimum. Furthermore, for the  $60\text{\AA}^2$  the monolayer has a faster convergence rate than the symmetry. The exact opposite happens in the  $50\text{\AA}^2$  as already mentioned before. This time however, the very slow convergence of the symmetry model is obvious with the monolayer ending at  $2.5 - 3ns$ .

## 6.3 NVT Equilibration

For the validation of the models, it was decided to compare the produced results after the NVT Equilibration step. Since the system would not ran under the NPT ensemble, and not a real change in the system properties would occur, after 1ns

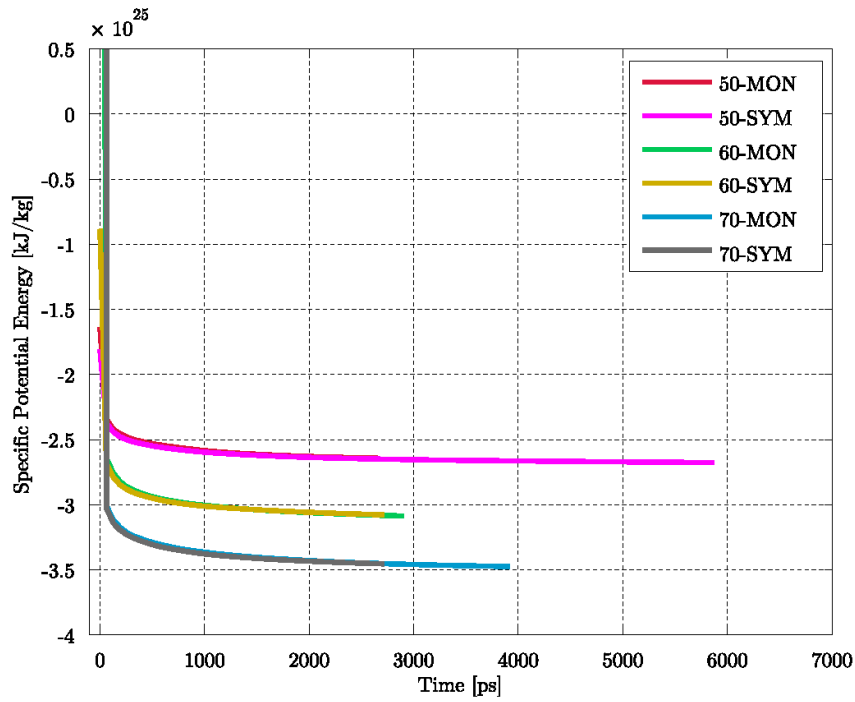


Figure 6.3: Specific Potential Energy during Energy Minimization for all cases. The y-axis ranges from 0 to 1 but only until 0.1 is shown, for clarity.

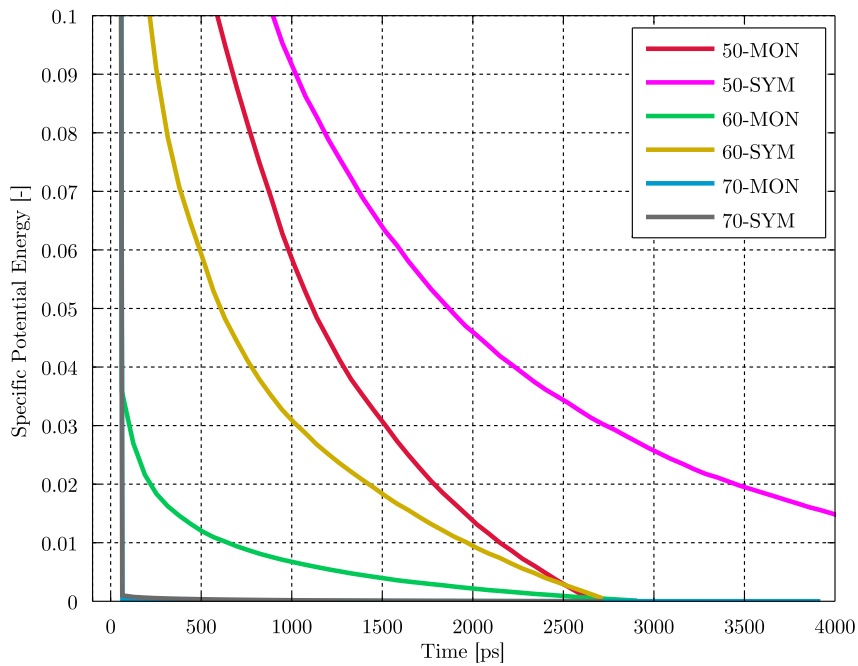


Figure 6.4: Normalized Specific Potential Energy during Energy Minimization for all cases.

each system had ran enough time to lose the initial conditions and get its unique characteristics. In the following section a detailed presentation of five comparison parameters will be attempted and based on their results, the selection of the final model for the calculation of the diffusion coefficient will be made.

### 6.3.1 Temperature Convergence

One of the most important parameters was the Temperature convergence and the ability of the Temperature Coupling method to equilibrate the system at  $T_{ref} = 323.15K$ .

#### 6.3.1.1 Analysis

In order to produce the Temperature results, the *g\_energy* tool of GROMACS was utilized. Then the data were imported to MATLAB Code *Error\_Analysis\_Temp.m* (see App. C.2) where for all timesteps the mean value of the fluctuations from the reference value was measured as:

$$T_{ave} = \frac{1}{N_{steps}} \sum_{N_{steps}} T_i \quad (6.5)$$

and the variance was:

$$Variance = \frac{\sum (T_i - T_{ave})^2}{N_{steps}} \quad (6.6)$$

The  $L_\infty$  Norm was calculated as:

$$L_\infty = \max(|T_i - T_{ave}|) \quad (6.7)$$

and the Root Mean Square Deviation(RMSD) was measured as:

$$RMSD = \sqrt{\frac{\sum (T_i - T_{ave})^2}{N_{steps}}} \quad (6.8)$$

The  $L_\infty$ , Variance and RMSD, were calculated only for the last 200 values or 100ps in order to check the system at its final stage and exclude the initial values until stabilization. Also the absolute and relative errors were calculated according to:

$$e_{abs} = |T_{ref} - T_{ave}| \quad (6.9)$$

$$e_{rel} = \frac{|T_{ref} - T_{ave}|}{|T_{ref}|} \cdot 100\% \quad (6.10)$$

Figures 6.5, 6.6 and 6.7 show the Temperature of the  $50\text{\AA}^2$ ,  $60\text{\AA}^2$  and  $70\text{\AA}^2$  system respectively, as a relation to the total simulation time. In the background (light colors) the fluctuations of the Temperature and in the front (bold lines) the ensemble averages are shown. The ensemble average was calculated with 20 windows of 100 values each. Also, in every figure the Phase transition temperature is drawn to check whether the systems remain in the solid condensed phase.

### 6.3.1.2 General Observations

As it can be observed both visually and numerically, in all the systems, the reference Temperature of 323.15K was achieved very fast, in **less than 100ps or 10% of the total simulation time**. Another common characteristic of all was that the fluctuations of the system remained well above the phase transition temperature of 314K except from a negligible number of points in the  $50\text{\AA}^2$  and a single point in  $60\text{\AA}^2$  system. For both cases however that did not happen during the last 200ps and therefore it can be considered that from the temperature point of view, **the system remained in the solid condensed phase**. Finally, for all simulations **the performance was exceptional** with the error of the final Temperature in comparison to the reference to be less than 0.2%.

### 6.3.1.3 Model behaviour

From Tables 6.2, 6.3 and 6.4 can be observed that regardless of each case's final error, **the monolayer systems had a more fluctuating behaviour**. This is clear not only by the maximum fluctuation shown from the  $L_\infty$  norm but also by the variance which is approximately 50% bigger for monolayer systems indicating a increased deviation.

### 6.3.1.4 Thermostat Efficiency

Just by observing the Temperature, no clear conclusion can be made and also the examination of the Temperature Coupling method is necessary. In all APL cases, **the Nose-Hoover performed better** giving a smaller error in comparison to the V-Rescale method. However, in almost every model was noticed **a smaller variance with the V-Rescale**. Such results are in agreement with the theory and were expected since the V-Rescale method is based on the Berendsen thermostat which couples the Temperature with a non-oscillatory first order exponential decay which is more stable than Nose-Hoover (see Section 3.4.5).



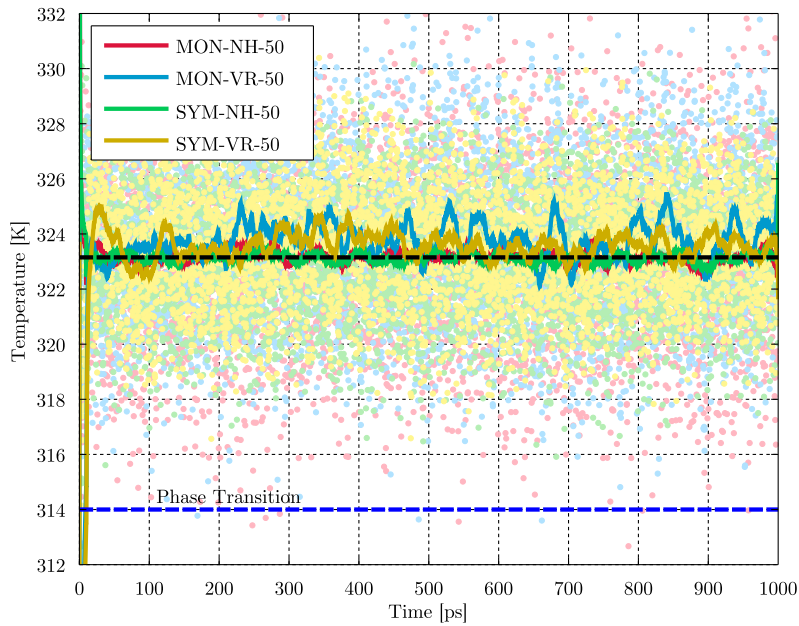
Case	$T_{ave}$ [K]	Variance [K]	$L_{\infty}$ [K]	$RMSD$ [K]	$e_{abs}$ [K]	$e_{rel}\%$
MON-NH-50	323.185	9.257	8.928	0.609	0.035	0.011
MON-VR-50	323.586	7.137	8.631	0.565	0.436	0.135
SYM-NH-50	323.147	3.926	5.649	0.397	0.003	0.001
SYM-VR-50	323.384	3.175	5.273	0.361	0.234	0.072

 Table 6.2: Error Analysis of Temperature Convergence for  $50\text{\AA}^2$ 

Case	$T_{ave}$ [K]	Variance [K]	$L_{\infty}$ [K]	$RMSD$ [K]	$e_{abs}$ [K]	$e_{rel}\%$
MON-NH-60	323.096	7.918	8.212	0.890	0.054	0.017
MON-VR-60	323.329	5.798	7.708	0.765	0.179	0.055
SYM-NH-60	323.009	3.916	5.650	0.631	0.141	0.044
SYM-VR-60	323.218	2.946	5.565	0.557	0.068	0.021

 Table 6.3: Error Analysis of Temperature Convergence for  $60\text{\AA}^2$ 

Case	$T_{ave}$ [K]	Variance [K]	$L_{\infty}$ [K]	$RMSD$ [K]	$e_{abs}$ [K]	$e_{rel}\%$
MON-NH-70	323.180	5.441	7.707	0.468	0.030	0.009
MON-VR-70	323.224	5.762	7.320	0.488	0.074	0.023
SYM-NH-70	323.116	3.195	5.125	0.565	0.034	0.010
SYM-VR-70	323.200	3.144	4.555	0.566	0.050	0.016

 Table 6.4: Error Analysis of Temperature Convergence for  $70\text{\AA}^2$ 

 Figure 6.5: Temperature Convergence for  $50\text{\AA}^2$  APL. Legend: NH=Nose-Hoover, VR=V-Rescale, MON=Monolayer, SYM=Symmetry.

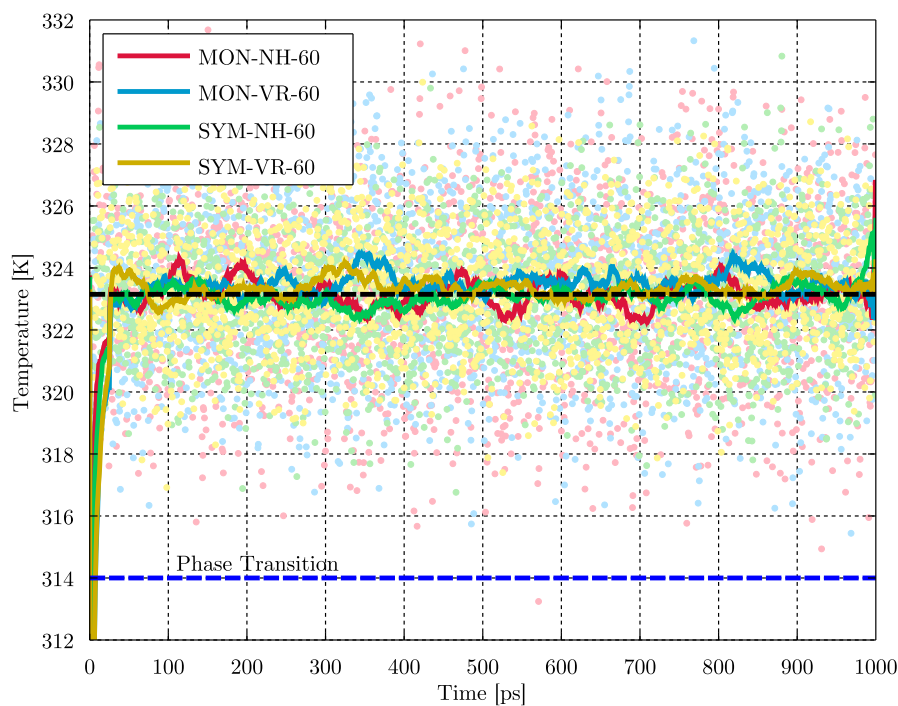


Figure 6.6: Temperature Convergence for  $60\text{\AA}^2$  APL. Legend: NH=Nose-Hoover, VR=V-Rescale, MON=Monolayer, SYM=Symmetry.

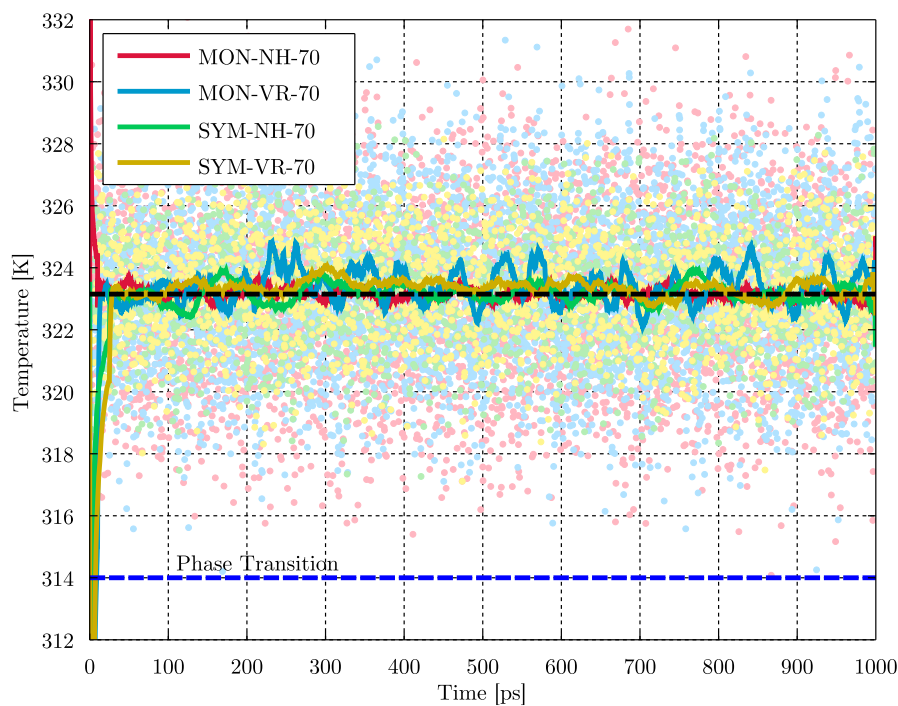


Figure 6.7: Temperature Convergence for  $70\text{\AA}^2$  APL. Legend: NH=Nose-Hoover, VR=V-Rescale, MON=Monolayer, SYM=Symmetry.

### 6.3.2 Deuterium Order Parameters

The Deuterium Order Parameters, also examined in Section 4.4, are a measure of the structural and dynamical characteristics of the lipid system. They are related to the average orientation of the Carbon-deuterium chains and show the overall anisotropy of the molecules' chains. The generic order parameter tensor  $S$  is defined as [108]:

$$S_{ij} = \frac{1}{2} \langle 3 \cos \theta_i \cos \theta_j - \delta_{ij} \rangle \quad (6.11)$$

where  $\theta_i$  is the angle between the  $i$ th molecular axis and monolayer normal and  $\delta_{ij}$  is the Kronecker's Delta. The average refers to all the simulation time and number of molecules.

The Deuterium Order Parameters are defined as [104]:

$$S_{CD} = \frac{2}{3} S_{xx} + \frac{1}{3} S_{yy} \quad (6.12)$$

and the range of values (for the  $-S_{CD}$ ) is from 0.5 which means complete alignment of chains with the monolayer normal (perfect order) to 0 for complete disorder.

#### 6.3.2.1 Analysis

In order to calculate the Deuterium order Parameters for all the examined cases, the GROMACS tool *g\_order* was used. Then the results were imported to MATLAB (*Error\_Analysis\_Deuterium\_ave.m*, App. C.3) where the statistical analysis of the results was conducted.

Since *g\_order* exports the  $-S_{CD}$  for each methyl chain and most of the validation data existed for the average of both, initially the results at each point were averaged:

$$S_{CD_i}^{\text{final}} = \frac{S_{CD_{sn-1}} + S_{CD_{sn-2}}}{2} \quad (6.13)$$

The absolute error at each atom  $i$  was measured as:

$$e_{abs_i} = |S_{CD_i}^{\text{reference}} - S_{CD_i}^{\text{final}}| \quad (6.14)$$

Due to different validation sources, only the Carbon atoms 2 to 14 were selected for comparison ( $C_{32}$  to  $C_{315}$  for Sn-1 and  $C_{22}$  to  $C_{215}$  for Sn-2 - Fig. 4.3). Also, the

$L_\infty$  norm of the absolute error at each atom, was calculated as:

$$L_\infty = \max(e_{abs_i}) \quad \text{for } i = [2, 14] \quad (6.15)$$

The RMS of  $e_{abs}$  was measured as:

$$RMS = \sqrt{\frac{(S_{CD_i}^{\text{reference}} - S_{CD_i}^{\text{final}})^2}{N_{\text{atoms}}}} \quad (6.16)$$

and also the average absolute and average relative errors (for all atoms) were calculated as:

$$e_{abs_{ave}} = \frac{1}{13} \sum_{i=2}^{14} |S_{CD_i}^{\text{reference}} - S_{CD_i}^{\text{final}}| \quad (6.17)$$

$$e_{rel_{ave}} = \frac{1}{13} \sum_{i=2}^{14} \frac{|S_{CD_i}^{\text{reference}} - S_{CD_i}^{\text{final}}|}{|S_{CD_i}^{\text{reference}}|} \quad (6.18)$$

Finally, for easy comparison the average deuterium parameters for all atoms was also computed:

$$S_{CD_{ave}} = \frac{1}{13} \sum_{i=2}^{14} S_{CD_i}^{\text{final}} \quad (6.19)$$

### 6.3.2.2 General Observations

Close examination of the results for all APL systems, indicates a very **similar behaviour** regardless of the model or the Thermostat utilized, especially in the atoms 6 to 14 which are placed in the final part of the chains. Another observation is the theoretically expected **decrease of  $-S_{CD}$  by the increase of APL**; the more compressed system has the highest order parameters, approximately 0.30 [Table 6.5] and for the rest it lowers. In total, an increase in Area Per Lipid of 40% led to a decrease in order of 50%. Finally, the overall  $-S_{CD}$  is reduced from the atoms next to heads (2,3,...) to the atoms in the end of the chains (... ,13,14), indicating that the **molecules are more ordered in the head area than in the upper tail area**. The observed reduction in  $-S_{CD}$  from atom 2 to atom 14 is almost 0.08 and is the same in absolute numbers for all systems. Relatively however, the  $-S_{CD}$  drops 26% in the  $50\text{\AA}^2$ , 38% in the  $60\text{\AA}^2$  and 57% in the  $70\text{\AA}^2$ , meaning that as the system expands, the disorder increase in the tails gets higher.

Figure 6.8 shows the  $-S_{CD}$  for every case of the  $50\text{\AA}^2$  system and Table 6.5 the

statistical analysis of it. As it can be observed visually, the performance of the most compressed system was not very good when it was compared with the simulation results of Pickholz et al. [126] ( $T=323\text{K}$ ) while it was excellent when compared to results from Rose et al. [88], although the latter were on another Temperature ( $T=310\text{K}$ ). Quantitatively, the relative error of the system in comparison to Pickholz et al. is around 25% and it was chosen to be compared only with these results because it had the most similar simulation parameters.

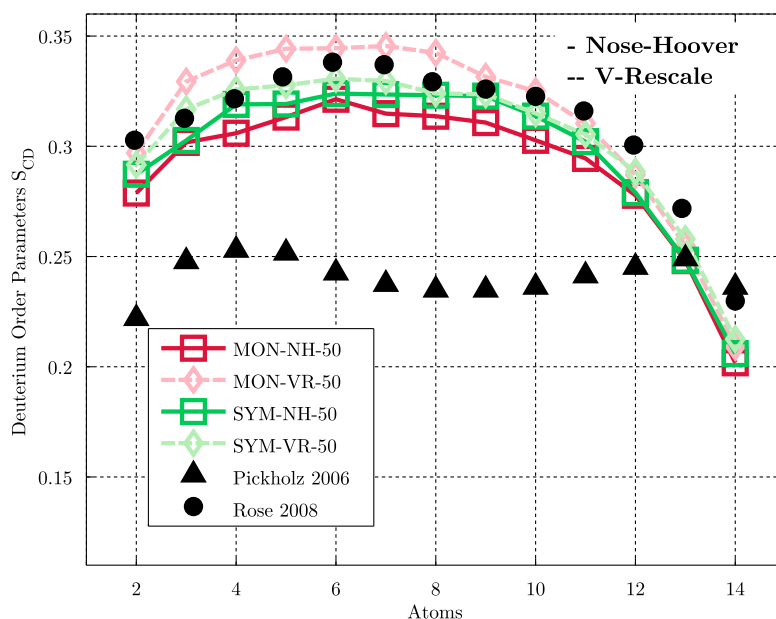


Figure 6.8: Deuterium Order Parameters  $S_{CD}$  for  $50\text{\AA}^2$  APL. The validation data of Pickholz et al. [126] at  $T=323\text{K}$  and Rose et al. [88] at  $T=310\text{K}$ . Legend: NH=Nose-Hoover, VR=V-Rescale, MON=Monolayer, SYM=Symmetry.

Case	$S_{CD_{ave}}$	$L_\infty$	RMS	Av.Abs	Av.Rel %
MON-NH-50	0.291	0.079	0.060	0.056	23.253
MON-VR-50	0.312	0.108	0.082	0.076	31.486
SYM-NH-50	0.298	0.088	0.067	0.062	25.719
SYM-VR-50	0.304	0.092	0.071	0.067	27.722

Table 6.5: Error Analysis of Deuterium Order Parameters for the  $50\text{\AA}^2$  APL. The comparison was made with Pickholz et al. [126] at  $T=323\text{K}$ .

One might thought that since the results from Rose et al. are closer to this system, there might be a transition of the system from the Solid Condensed Phase to the Liquid Condensed due to the extra compression. However, if that was the case, the results of Rose et al. would not compare well in the other two systems. A simple

observation of Figures 6.9 and 6.10 proves that is not happening and the values of Rose et al. are close to those calculated in this Thesis. It can be considered therefore that the **Deuterium order parameters do not alter severely from the Solid Condensed to the Liquid Condensed phase.**

Examination of Table 6.5 shows that the Nose-Hoover Thermostat did better than V-Rescale in both models. Also Symmetry model had a smaller variation in the results (25.7-27.7) in comparison to monolayer (23.3-31.5), something which is also validated qualitatively.

In the system of APL  $60\text{\AA}^2$ , the direct comparison with the results from Pickholz et al. gave very satisfactory values. The same was observed also for the comparison with the results of Rose et al. [88] and Lucas et al. [128]. The relative error did not exceed the 6.1% and the average  $-S_{CD}$  was around 0.2. It is noteworthy that although there is a difference in the errors between Nose-Hoover and V-Rescale for both models, the average  $-S_{CD}$  is almost identical at 0.2. This shows the relativity that different references introduce especially since there is a lack on the literature regarding experimental values for Deuterium Order Parameters. Overall, all the models and thermostats **performed very satisfactory in the case of  $60\text{\AA}^2$  APL.**

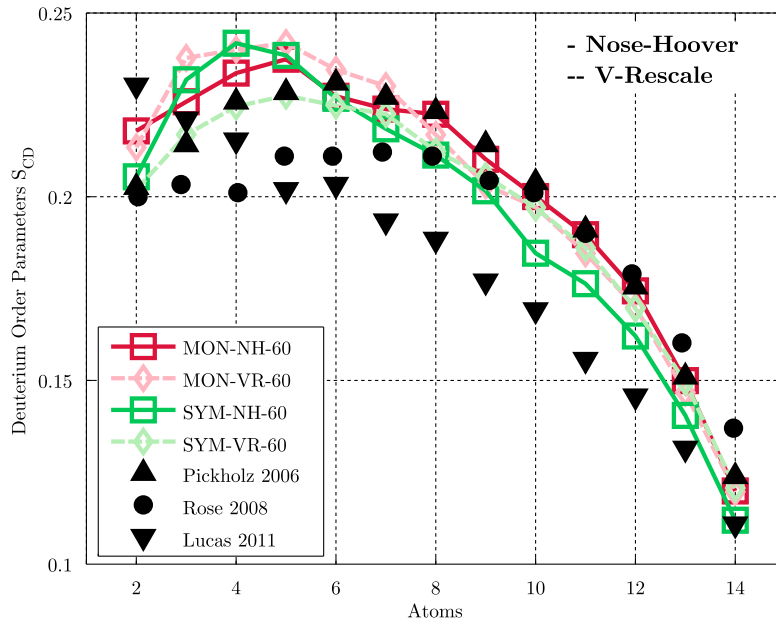


Figure 6.9: Deuterium Order Parameters  $S_{CD}$  for  $60\text{\AA}^2$  APL. The validation data of Pickholz et al. [126] at  $T=323\text{K}$ , Rose et al. [88] at  $T=310\text{K}$  and Lucas et al. [128] at  $323\text{K}$ . Legend: NH=Nose-Hoover, VR=V-Rescale, MON=Monolayer, SYM=Symmetry.

Case	$S_{CD_{ave}}$	$L_\infty$	RMS	Av.Abs	Av.Rel %
MON-NH-60	0.203	0.015	0.007	0.005	2.515
MON-VR-60	0.203	0.024	0.010	0.009	4.262
SYM-NH-60	0.196	0.019	0.013	0.012	6.094
SYM-VR-60	0.197	0.011	0.005	0.004	2.222

Table 6.6: Error Analysis of Deuterium Order Parameters for the  $60\text{\AA}^2$  APL. The comparison was made with Pickholz et al. [126] at T=323K.

Case	$S_{CD_{ave}}$	$L_\infty$	RMS	Av.Abs	Av.Rel %
MON-NH-70	0.129	0.013	0.006	0.005	3.912
MON-VR-70	0.117	0.022	0.014	0.013	10.754
SYM-NH-70	0.125	0.015	0.008	0.007	6.475
SYM-VR-70	0.127	0.011	0.007	0.006	5.633

Table 6.7: Error Analysis of Deuterium Order Parameters for the  $70\text{\AA}^2$  APL. The comparison was made with Pickholz et al. [126] at T=323K.

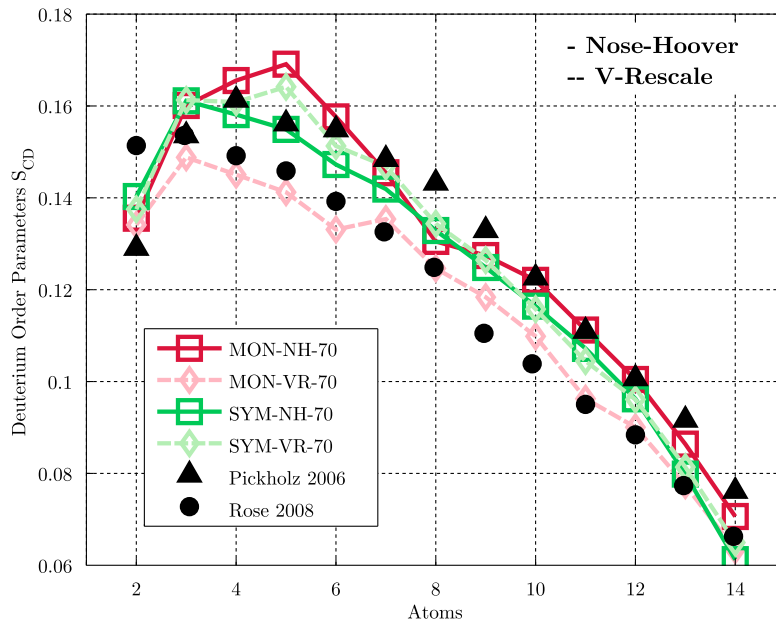


Figure 6.10: Deuterium Order Parameters  $S_{CD}$  for  $70\text{\AA}^2$  APL. The validation data of Pickholz et al. [126] at T=323K and Rose et al. [88] at T=310K. Legend: NH=Nose-Hoover, VR=V-Rescale, MON=Monolayer, SYM=Symmetry.

For the system of  $70\text{\AA}^2$  the results were compared only with Pickholz et al. and Rose et.al similarly to the  $50\text{\AA}^2$ . **The obtained comparison results were again quite satisfactory [3.9%-10.8%]**. The  $-S_{CD}$  values for the most expanded system were lower than the previous systems (as explained in the general observations) and the average calculated  $S_{CD_{ave}}$  was approximately 0.125.

### 6.3.3 Density Profiles

Density profiles are the tools to understand the structural characteristics of the whole system and were calculated both on  $[e/nm^3]$  and on  $[kg/m^3]$  by using the GROMACS tool *g\_density* and the index files of all the groups of the system (Fig.4.3 and App.B.5):

```
g_density -s nvt.tpr -f nvt.trr -n GROUP.ndx -o dens_elec_GROUP.xvg -d z
-dens electron -ei electrons.dat -sl 1500
>
g_density -s nvt.tpr -f nvt.trr -n GROUP.ndx -o dens_mass_GROUP.xvg -d z
-dens mass -sl 1500
```

For the electron density the file *electrons.dat* was also used, which includes the electric charges of each atom in the DPPC molecule based on Klauda's [73] model. Also, apart from the DPPC and Water molecules, the DPPC was analyzed in its main groups and for each, the densities were calculated.

During the calculation of the Density profiles for the monolayers, a problem was appeared. The upward motion of the system, was taken by GROMACS as an extended area of atoms which then averaged to obtain the density profile. The consequence was the lower than expected density values both for  $[e/nm^3]$  and  $[kg/m^3]$ . Examination of each specific frames over the whole 1ns trajectory validated the above statement as the values were physically correct but contained the unavoidable oscillations that are eliminated only by averaging<sup>1</sup>.

On the other hand the results of the symmetry systems for all APLs were excellent. The physically expected submerging of the head parts as well as the correct values for the density of the water and DPPC molecules are shown in Figures 6.11 to 6.16.

<sup>1</sup>Since no method to address this issue was found, it was considered meaningless to present the density profiles of the monolayers



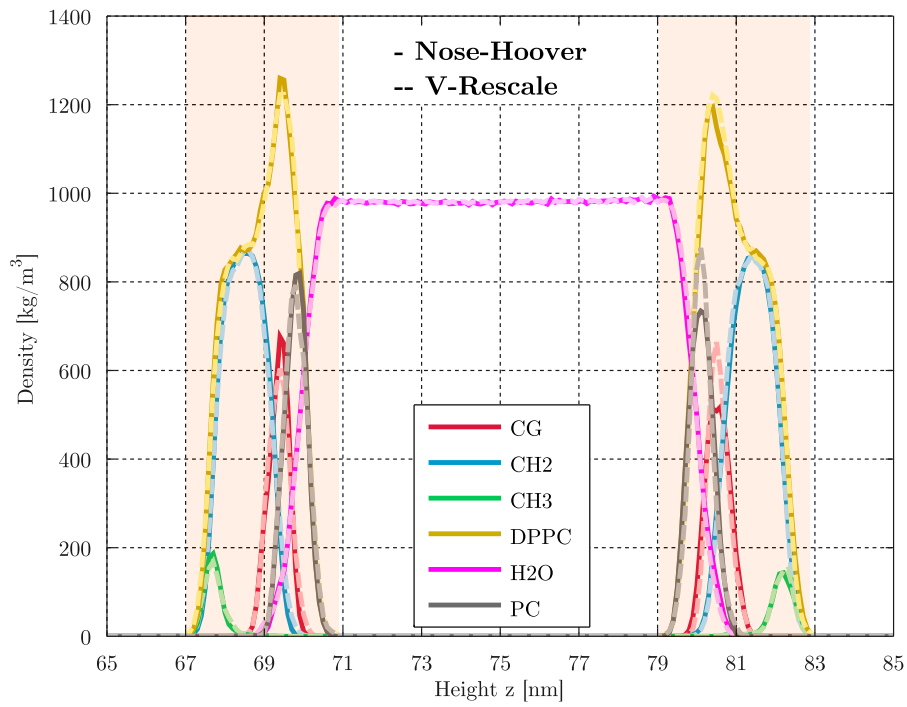


Figure 6.11: Mass Density Profile for the  $50\text{\AA}^2$  APL.

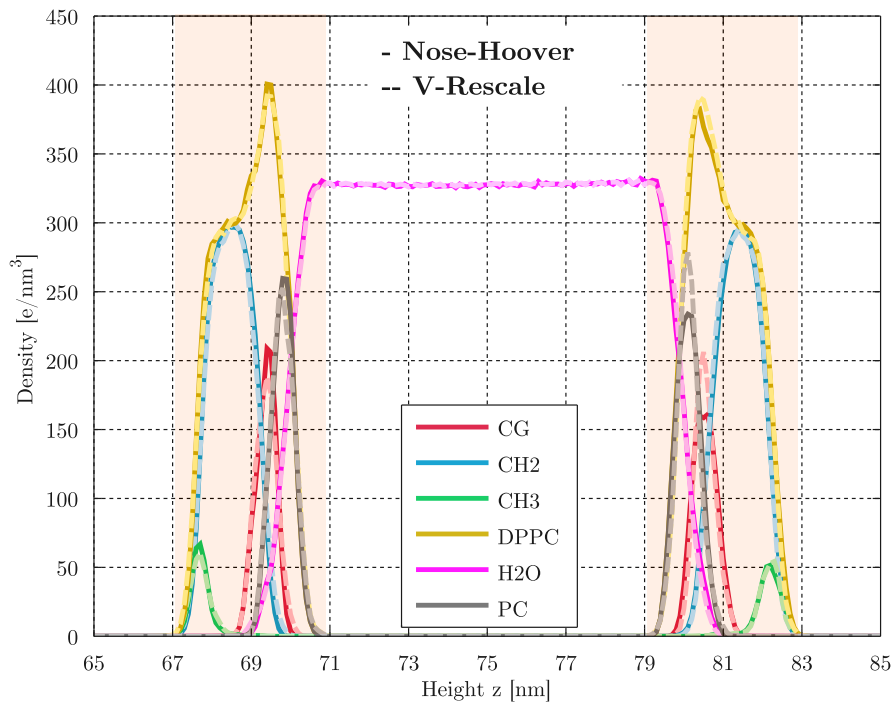
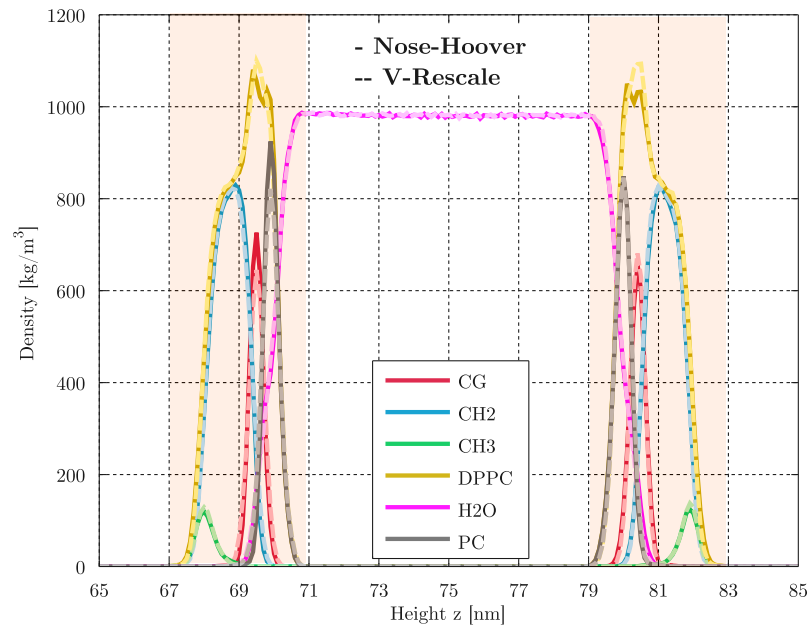
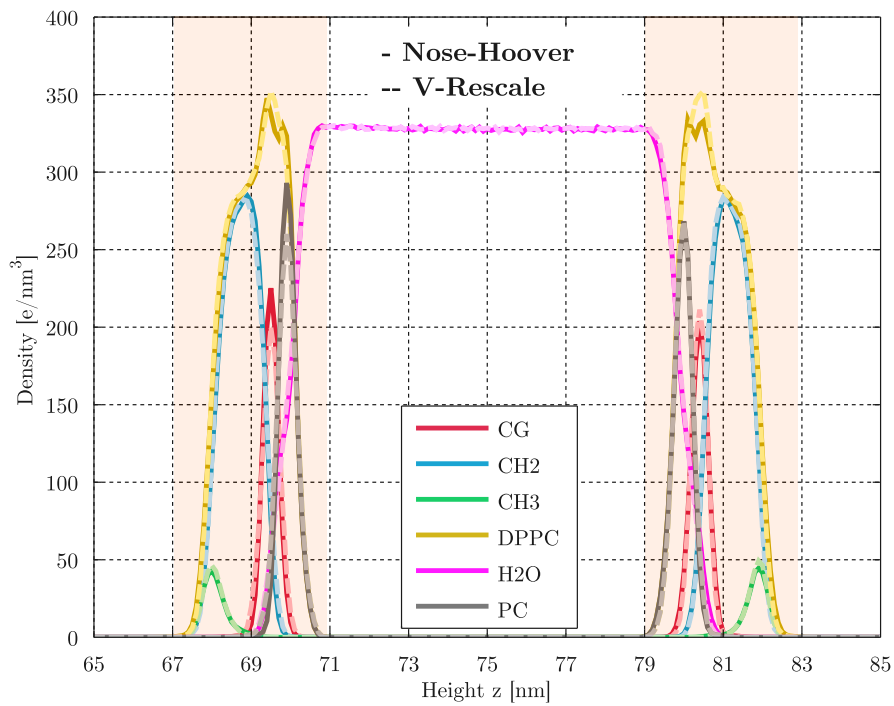


Figure 6.12: Electron Density Profile for the  $50\text{\AA}^2$  APL.

Figure 6.13: Mass Density Profile for the 60Å<sup>2</sup> APL.Figure 6.14: Electron Density Profile for the 60Å<sup>2</sup> APL.

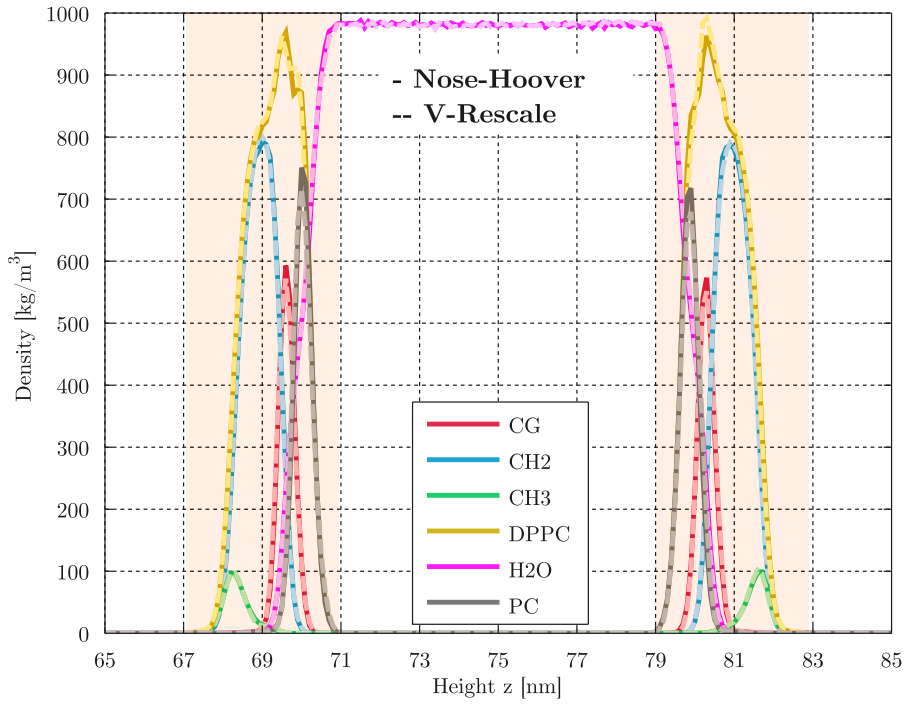


Figure 6.15: Mass Density Profile for the  $70\text{\AA}^2$  APL.

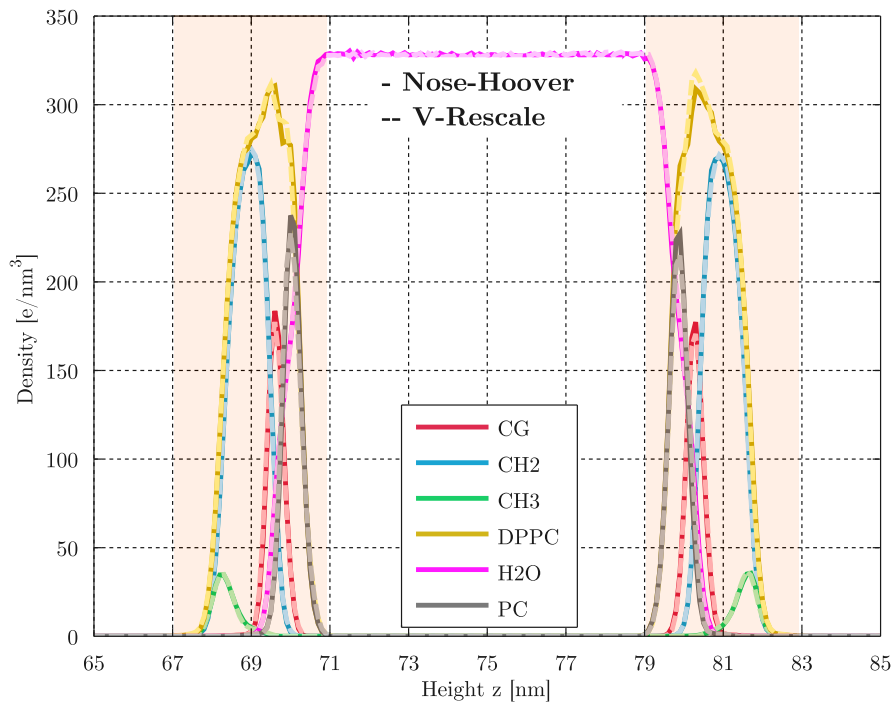


Figure 6.16: Electron Density Profile for the  $70\text{\AA}^2$  APL.

### 6.3.4 Mean Square Displacement

The Mean Square Displacement of the DPPC lipid molecules (MSD) is another important parameter of the system that was examined after the 1ns NVT Equilibration. In order to specify the lateral lipid movement, only the Phosphorus atoms are used since they are unique for each molecule and exist overall in the same level. The theory behind the calculation of MSD was explained in Section 2.3.

GROMACS tool *g\_msd* was used to do the MSD analysis of the simulation trajectory, along with the Phosphorus index file *P.ndx* (App.B.5):

```
g_msd -s nvt.tpr -f nvt.trr -n P.ndx -lateral z -o msd_lateral.xvg
```

and then the results were post-processed with MATLAB (*Error\_Analysis\_MSD.m*, App.C.4). During that step, also the three Diffusion type lines were introduced (Single-File, Fickian, Ballistic), according to equations in section 2.3.

Observing Figures 6.17, 6.18 and 6.19, for the symmetry model the lateral diffusion of phosphorus can be divided in three regions. Initially, for the first 10ps, the diffusion is locally restrained to single-file. As time passes, the slope increases and from 100ps until 800ps, the diffusion is in the anomalous region between normal-Fickian and ballistic motion. After the 800ps, in all the systems, a drop is observed where the simulation seems to take a slope similar to Fickian, especially around 1ns.

Case	$D_{P_{xy}} \cdot 10^{-5} [cm^2/s]$	$\pm$	$e_{abs}$	$e_{rel}\%$
Shushkov et al. [79]	0.2	-	-	-
MON-VR-50	0.014	0.006	0.186	92.800
MON-NH-50	0.021	0.003	0.179	89.700
SYM-VR-50	0.033	0.775	0.167	83.650
SYM-NH-50	1.204	1.449	1.004	501.850

Table 6.8: Lateral Diffusion Coefficient of Phosphorus atoms  $D_{P_{xy}}$  for the 50Å<sup>2</sup> APL system.

The above observations are also justified by the numerical results presented in Tables 6.8, 6.9 and 6.10. The  $D_{P_{xy}}$  values of Symmetry systems are varying severely indicating an unstable lateral diffusion and a possible need for extended simulation time until diffusion stabilization. This possibility is further increased after the comparison with the  $D_{P_{xy}}$  value after 3ns and the fact that the value was closer to

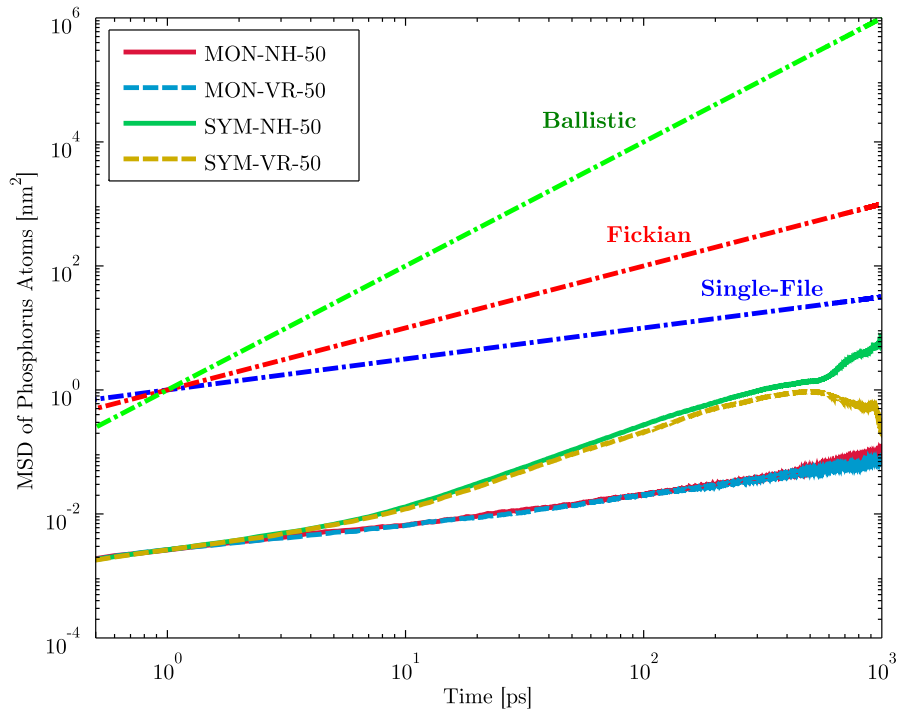


Figure 6.17: Mean Square Displacement of Phosphorus atoms for  $50\text{\AA}^2$  APL. Legend: NH=Nose-Hoover, VR=V-Rescale, MON=Monolayer, SYM=Symmetry.

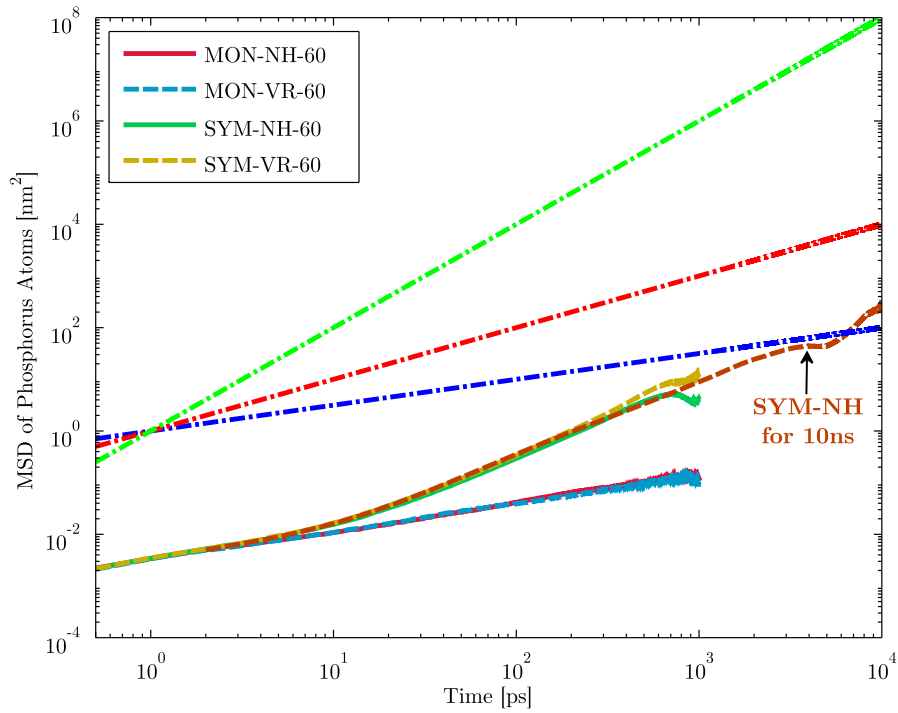


Figure 6.18: Mean Square Displacement of Phosphorus atoms for  $60\text{\AA}^2$  APL. Also, the 10ns simulation of the Symmetry-Nose Hoover Thermostat case is shown. Legend: NH=Nose-Hoover, VR=V-Rescale, MON=Monolayer, SYM=Symmetry.

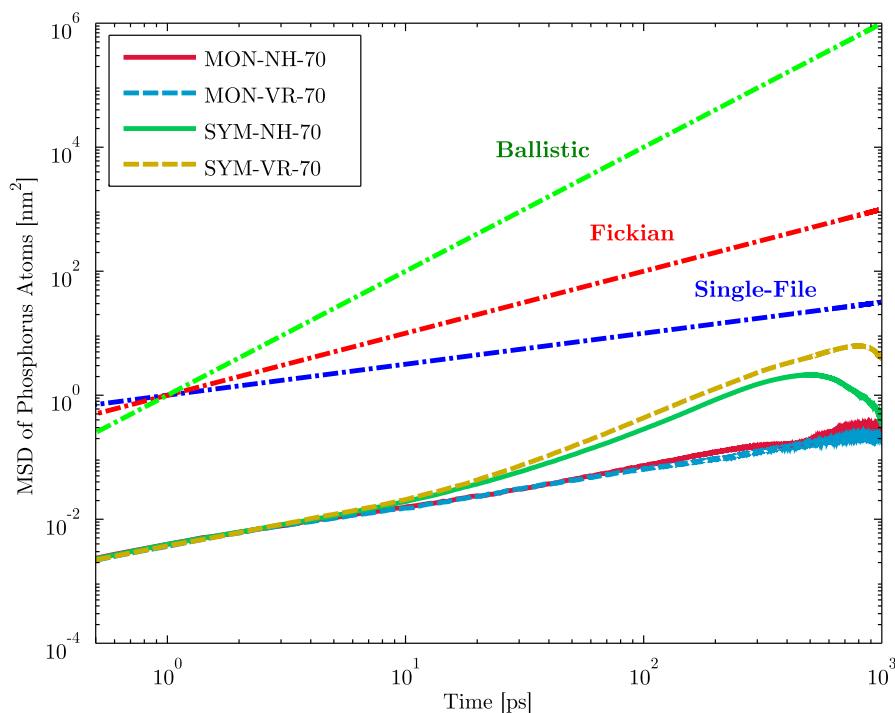


Figure 6.19: Mean Square Displacement of Phosphorus atoms for  $70\text{\AA}^2$  APL. Legend: NH=Nose-Hoover, VR=V-Rescale, MON=Monolayer, SYM=Symmetry.

the value of Shushkov et al. [79] for  $T=300\text{K}$ . Although the value from Shushkov is in a different phase, the magnitude is indicative of what is expected. Therefore by comparing the data relatively, a hypothesis was made that **extra simulation time was necessary to obtain a stable normal lateral diffusion for the symmetry system.**

To evaluate the validity of the above mentioned hypothesis, a simulation of 10ns was run for the case of  $60\text{\AA}^2$  APL and Nose-Hoover Thermostat. Figure 6.18 shows the results of the MSD after 10ns and in Table 6.9 the  $D_{P_{xy}}$  is presented. Observation of the figure shows that the anomalous Fickian-Ballistic behaviour that was observed for 1ns, continuous also in the 10ns simulations and the calculated diffusion coefficient does not seem to approach the reference or at least to converge in a specific value.

An alternative explanation for this behaviour is the fact that since the **lipid chains were not constrained by other lipid chains (as in a bilayer)** but were rather free to move in a zero-density vacuum, made the lateral movement of the molecules more chaotic.

Case	$D_{P_{xy}} \cdot 10^{-5} [cm^2/s]$	$\pm$	$e_{abs}$	$e_{rel}\%$
Shushkov et al. [79]	0.25	-	-	-
MON-VR-60	0.029	0.010	0.221	88.360
MON-NH-60	0.028	0.013	0.222	88.880
SYM-VR-60	3.447	0.370	3.197	1278.920
SYM-NH-60 (1ns)	1.624	2.032	1.374	549.520
SYM-NH-60 (3ns)	0.175	0.019	0.075	29.920
SYM-NH-60 (10ns)	5.318	8.6845	5.068	2027.160

Table 6.9: Lateral Diffusion Coefficient of Phosphorus atoms  $D_{P_{xy}}$  for the 60Å<sup>2</sup> APL system.

According to Hofsass et al. [153], for Bilayers even 100ns are not enough for the calculation of diffusion but there is at least a linear behaviour from 5ns to 100ns. This is also validated by previous Thesis work on a DPPC bilayer [154] where the diffusion coefficient remained the same after 3-4ns. Considering that bilayers are naturally more stable than the monolayers, there might be a possibility that the required stabilization time to be bigger. Unfortunately no literature was found to confirm or not this hypothesis. In any case, due to the fact that the time to complete this Thesis was very limited compared to the required simulation time for an exhaustive research and because the oxygen molecule according to the literature is moving almost unhindered through the monolayer (and therefore the lateral lipids movement will not probably affect it), the systems of 1ns were considered enough to proceed with the umbrella simulations.

Case	$D_{P_{xy}} \cdot 10^{-5} [cm^2/s]$	$\pm$	$e_{abs}$	$e_{rel}\%$
Shushkov et al. [79]	2.6	-	-	-
MON-VR-70	0.051	0.020	2.549	98.054
MON-NH-70	0.078	0.032	2.522	97.012
SYM-VR-70	1.999	1.188	0.601	23.115
SYM-NH-70	0.080	2.269	2.520	96.931

Table 6.10: Lateral Diffusion Coefficient of Phosphorus atoms  $D_{P_{xy}}$  for the 70Å<sup>2</sup> APL system.

The cases of the monolayer model were different than the symmetry although the MSD results were problematic. Due to the unstable-upward movement of the system, the calculated MSD was only relative to a Single-File diffusion. It was expected however that after the initialization of the simulation the diffusion would become normal, and that was not observed in the first 1ns. This results however were consistent for all monolayer cases, giving similar  $D_{P_{xy}}$  values.

### 6.3.5 Computational Cost and Effectiveness

The final parameter which is of great importance not only for that Thesis but for all Molecular Dynamics studies, is the computational cost and the actual physical time required for the simulation of a system. The final time that is required is usually a function of the number of atoms included in the system, the physics that are involved in it and of course the resources that are used to solve it. Table 6.11 shows all these parameters and the total physical time that was required in minutes.

Model	APL $\text{\AA}^2$	T Coupling	Atoms	CPUS	$T_{sim}[ns]$	$T_{phys}[min.]$
Monolayer	50	Nose-Hoover	13725	32	1	288
	50	V-Rescale	13725	32	1	288
	60	Nose-Hoover	15468	32	1	290
	60	V-Rescale	15468	32	1	293
	70	Nose-Hoover	17145	32	1	348
	70	V-Rescale	17145	32	1	347
Symmetry	50	Nose-Hoover	27324	32	1	209
	50	V-Rescale	27324	32	1	208
	60	Nose-Hoover	30771	32	1	218
	60	V-Rescale	30771	32	1	216
	70	Nose-Hoover	34116	32	1	242
	70	V-Rescale	34116	32	1	241
Symmetry	60	Nose-Hoover	30771	64	10	1213

Table 6.11: Required Simulation Time for each system examined

Examination of the table shows an unexpected behaviour. The monolayer systems did 33% to 43% more time to finish than the respective symmetry systems that had double the atoms. This can be explained by the upward movement of the monolayers which apparently made the computation of neighbouring particles more complicated.

Another observation that can be made is for the symmetry/Nose-Hoover system that ran for 1 and 10ns. In the 10ns simulation the CPUS utilized were doubled and in the end, the double number of CPUS, solved a 10 times longer problem in only 5.6 times the physical time. This is expected because the ASTRAL 2 High Performance Computer that was used, as well as the GROMACS solver, are optimized for massively parallel problems.

Finally, from the Table 6.11 is clear that the Temperature Coupling method did not alter the required time for solution in any particular way.



### 6.3.6 First Stage Conclusions

By taking into consideration all the previously examined parameters, it was decided to proceed to the second stage of the Thesis with the Symmetry model, the Nose-Hoover Thermostat and the  $60\text{\AA}^2$  APL. The reasons were:

1. **Area Per Lipid.** The case of  $60\text{\AA}^2$  was the one in the middle of the choices, were no extreme phenomena were expected and with the most validation data in the literature.
2. **Temperature Convergence.** The system Temperature converged perfectly to the reference value of 323.15K without any problems, even after the 50ps.
3. **Deuterium Order Parameters.** The case also had very good results regarding the validation of the deuterium order parameters of the lipid chains with a relative error of 6%.
4. **Density Profile.** The produced Density Profile was normal with its values in accordance to the expected from theory.
5. **MSD.** The lateral diffusion of the symmetry systems was in general more reliable than of the monolayers. Also the system had the most stability of all the other APL and T-Coupling methods and was also tested until 10ns, confirming in that way, its behaviour.
6. **Computational Cost.** The system performed very well also in the required physical time as it was the second most fast solution and 33% faster than its respective monolayer.

## 6.4 Umbrella Sampling Simulations

Umbrella Simulations were used to extract the Potential of Mean Force (PMF) of the Oxygen as it was restrained in different positions through the upper monolayer leaflet. The dragging procedure, as well as, the set up of each umbrella sampling simulation was explained in detail in section 5.4.2.1. That PMF was used later to obtain the final diffusion coefficient of Oxygen and the results of the procedure are explained in the end of the chapter. For all the procedures explained in this section the MATLAB Code *Diffusion\_Coefficient.m* (App.C.5) was used.

### 6.4.1 Effect of Harmonic Force Constant

Initially, because the effect of the harmonic force constant of equation 5.13 was unknown, an extreme value of  $k_h=3000$  was used to restrain the oxygen molecule at each position. In general, the higher the value of  $k_h$ , the more restrained the oxygen would be as the "oscillator" would become more stiff. That would result in more samples and a "clearer" value at each position but also smaller areas of sampling. Figure 6.20 shows the histogram analysis for  $k_h=3000$  and spacing of both  $d=0.1\text{nm}$  and  $d=0.2\text{nm}$  (top chains area, next to vacuum).

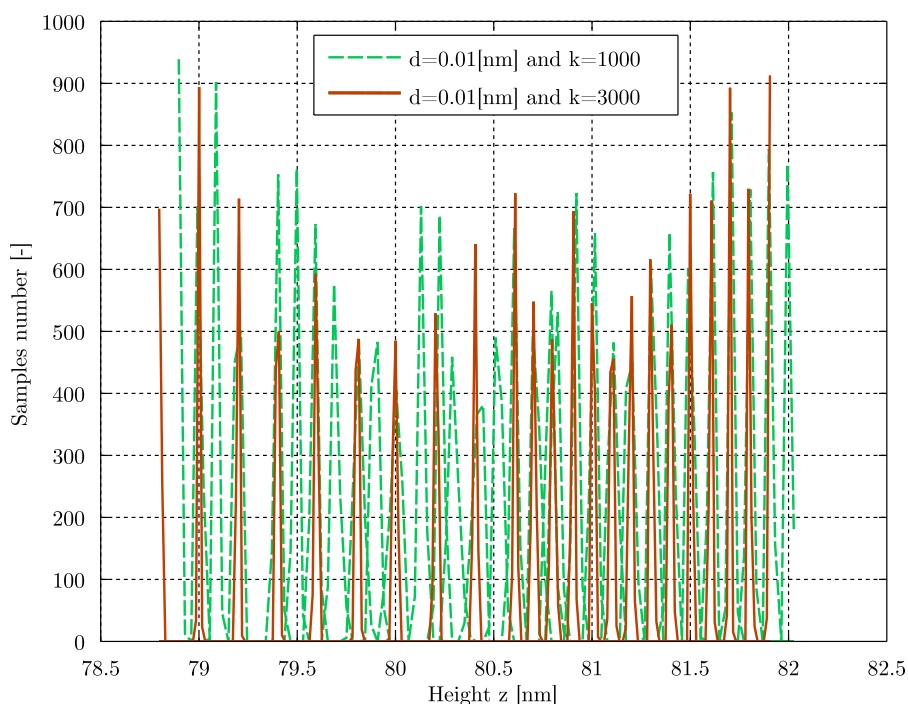


Figure 6.20: Comparative Histogram analysis for  $k_h=1000$  and  $k_h=3000$

From Figure 6.20 is clear that the number of samples is high [500-1000] for each position but on the other hand the histograms do not start to overlap even for 0.1nm.

With that in mind, a reduce of the stiffness was attempted by changing the harmonic force constant at  $k_h=1000$  to increase the sampling areas. Figure 6.20 shows a the histogram analysis for  $d=0.1$  and  $k_h=1000$ . It is clear that although overlapping has not started, the histograms are much closer to each other. As it was expected however, that reduced the number of samples at each position [200-700].

Since the number of umbrella simulations was large and efficient management of

the time was crucial, it was decided to continue the study with the harmonic force constant as 1000 as it would give adequate results for larger spacing, the effect of which is explained in the next paragraph.

### 6.4.2 Effect of Sampling Space

Another parameter that was of great importance was the spacing between umbrella sampling positions. As it was seen before, even for the case of  $k_h=1000$ , the spacing of 0.1nm was not enough to create adequate sampling of the whole movement of Oxygen. In total, 5 different spacings were examined: 0.4, 0.2, 0.1, 0.05 and 0.025nm and the produced PMF of each is shown in Figure 6.24.

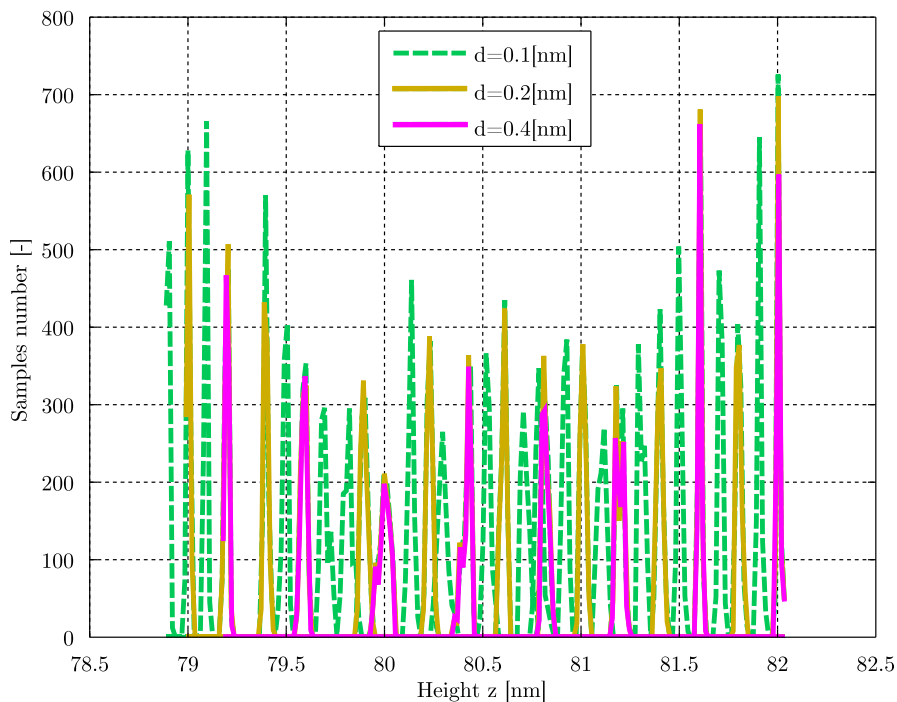
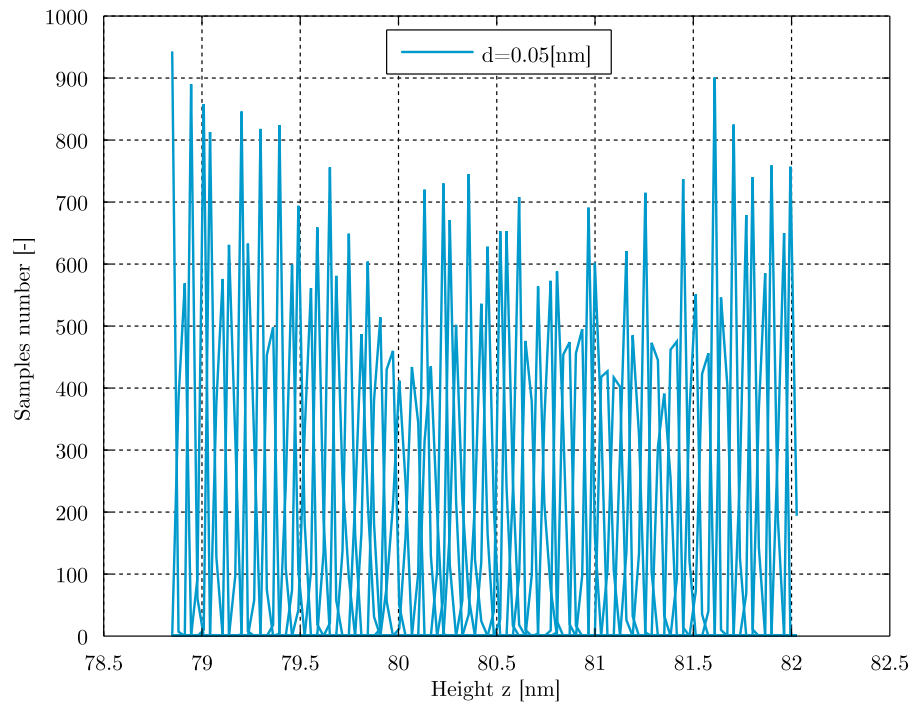
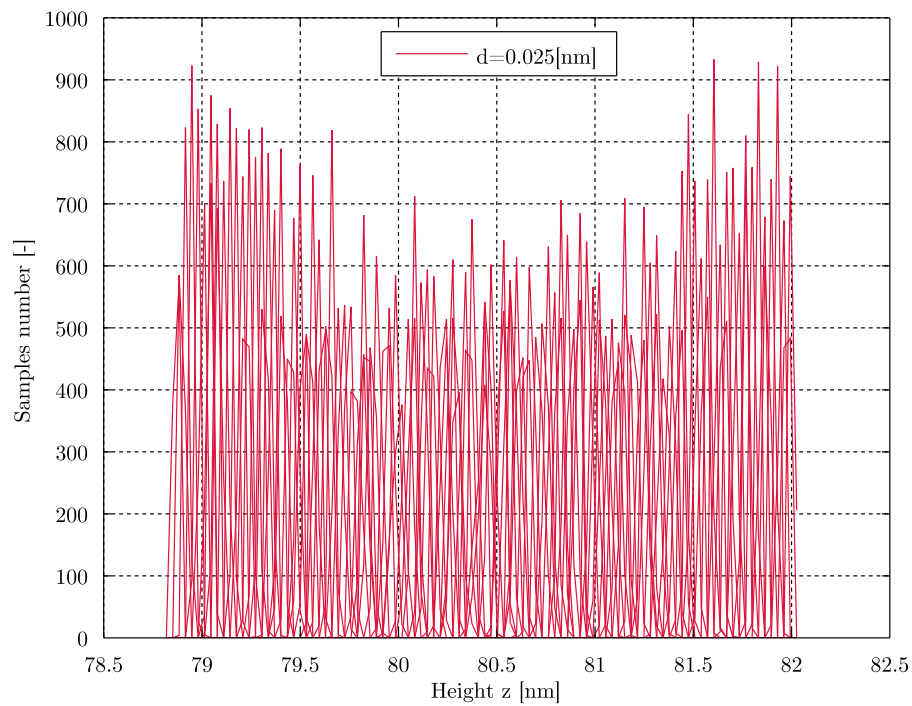


Figure 6.21: Histogram analysis for  $k_h=1000$  and  $d=0.4, 0.2, 0.1$ nm.

The analysis of the histograms showed how different spacings covered the required area. For the first three, the data are far from overlapping [Fig.6.21]. Decrease of spacing to 0.05nm [Fig.6.22] shows that histograms begin to overlap but as it will be analysed later, still the results are not considered sufficient for the reliable construction of the PMF. Finally, the spacing of  $d=0.025$  is presented in Figure 6.23 where the histograms overlap sufficiently in the whole length of the examined geometry (vacuum-monolayer-water).

Figure 6.22: Histogram analysis for  $k_h=1000$  and  $d=0.05$ nm.Figure 6.23: Histogram analysis for  $k_h=1000$  and  $d=0.025$ nm.

Also, Table 6.1 shows the required number of simulations for each of the cases and is clear that for every halving of spacing, the required number of simulations was doubled. Since every simulation ran for 1ns, the last case of 128 simulations required over a week to be produced.

Spacing [nm]	Simulations
0.4	8
0.2	16
0.1	32
0.05	64
0.025	128

Table 6.12: Number of Simulations ran for each spacing of umbrella sampling

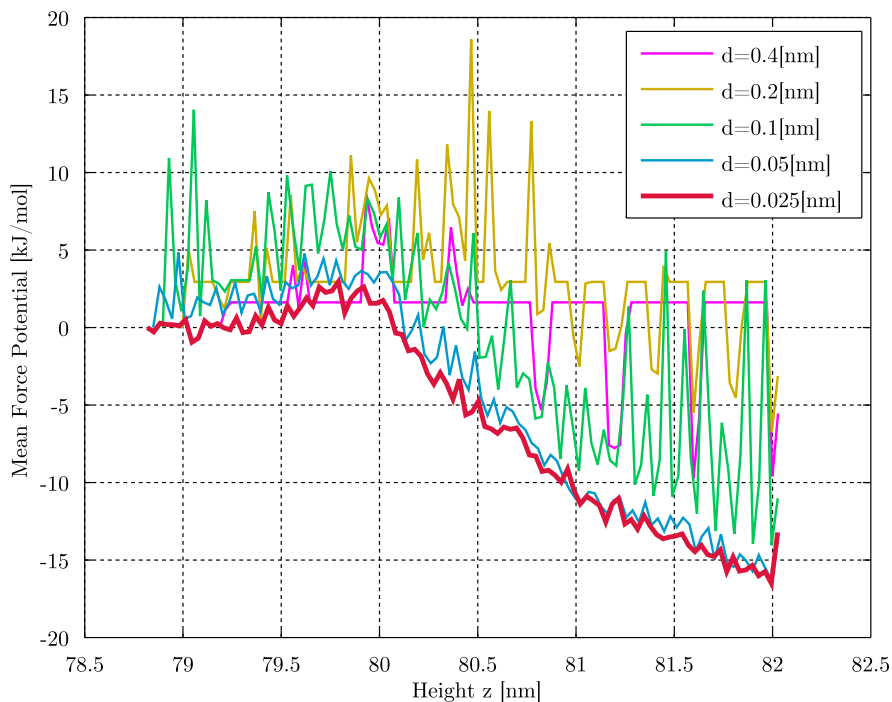


Figure 6.24: Potential of Mean Force as produced for umbrella sampling simulations of different spacings.

Examination of Figure 6.24 leads to a couple of observations. Firstly, in the three initial cases of 0.4, 0.2 and 0.1nm the fact that histograms do not overlap, especially in the lipid chains area (82-80.5nm) is clear, since the PMF curve for every single position creates small plateaus equal to zero. Of course in the case of 0.1nm are spikes and not plateaus but still there is not an actual continuous PMF curve. In the head area (80.5-79.5nm), this behaviour is observed only for the 0.4 and 0.2 and in the water area (79.5-78.8nm) this is again observed for all three.

For the smaller spacing of 0.05nm, things are better but still the PMF curve is very oscillatory with larger than usual spikes in the areas where before there were plateaus. Another problematic region was the water area where the PMF was very oscillatory.

In the final case of  $d=0.025$ , the PMF curve obtains an acceptable form with physical values and no abnormal oscillations which were considered sufficient and no smaller spacing was attempted, especially since the number of simulations would once again double, leading in an extreme total of 256.

### 6.4.3 Calculation of PMF

Figure 6.25 shows the Potential of Mean Force of the Oxygen Molecule as it was created by the Weighted Histogram Analysis Method from the 128 umbrella simulations of spacing  $d=0.025\text{nm}$  as well as the areas of the system that are sampled. Examination of the system from above the leaflet (far-right) until the water slab (far-left) leads to the following observations:

1. **Vacuum(air) area [ $z \geq 82.0\text{nm}$ ]**. The selected points were exactly in the boundaries between ethyl part of DPPC molecule and the vacuum and therefore no actual sampling of the vacuum area was made. Apart from the obvious gain in computational time, the vacuum area is of no interest anyway because the PMF there would be zero. This is also noticeable by the spike on 82.01nm that moves with a high gradient towards zero.
2. **Lipid Chains area [80.2-82nm]**. In this area the PMF is gradually increasing with the same slope from the initial value of  $-16[\text{kJ/mol}]$  to zero at 80.2nm. In the whole movement through the lipids, the movement was unhindered as the negative values of the PMF are indicating.
3. **Lipid Heads area [79.4-80.2nm]**. In contrast to the lipid chains, the lipid heads posed an energy barrier to the oxygen diffusion as the PMF had positive values something which was also observed by previous studies (will be further explained later).
4. **Water area [ $z \leq 79.4\text{nm}$ ]**. In the final area of water slab, the PMF of oxygen is zero. The oscillations that are shown around  $z=0.79\text{nm}$ , are probably an indication for the need of further reduction of spacing and better sampling.

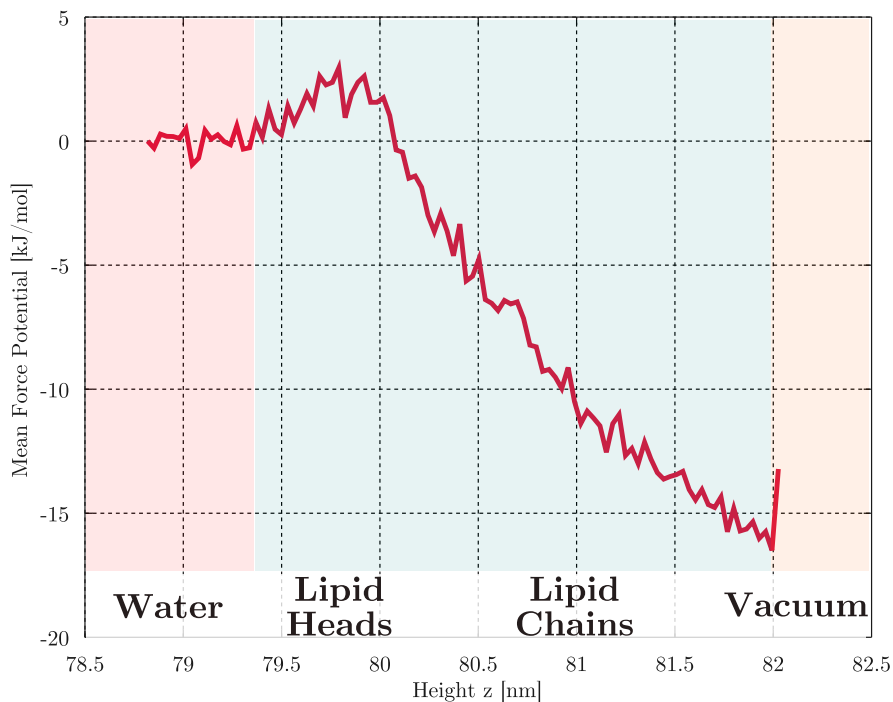


Figure 6.25: Potential of Mean Force for  $d=0.025\text{nm}$ . In the background the different areas of the system are shown.

As it was mentioned in the literature review, no other molecular dynamics studies of  $O_2$  diffusion through a monolayer were found. Therefore, it was decided to compare the results with published data for bilayers since a lot of literature exists there and also use a recent Master Thesis of the University of Tampere in Finland [155] where the permeation of Oxygen through the alveolar surfactant was studied with the method of Thermodynamic Integration but their publications are still under preparation.

The *g\_wham* tool of GROMACS calculated  $\Delta G$  both in [kJ/mol] (which is presented in Figure 6.25) and in [ $k_B T$ ]. The latter was used also for the calculation of the diffusion coefficient since it made its calculation very easy. Based on that, **the  $\Delta G$  was calculated as 16.674 [kJ/mol] or 6.7294( $k_B T$ ).**

The created PMF curve although similar, was not exactly the same as Javanainen found it on his Thesis, where he defined PMF in a different way that *g\_wham* produces it. Nevertheless some characteristics were common. First of all, the area before the head was of much lower potential which increased fast and then in the head area an Energy barrier appeared. In this Thesis, it was calculated around 4[kJ/mol] while Javanainen found it for his bilayer at around 4 as well. Shinoda et

al. [156] measured the energy barrier at 3 [kJ/mol] for a DPPC bilayer at 323K. Also Hub and de Groot [152], for a POPE bilayer at 300K measured it at 4[kJ/mol]. Sugii et al. [157] also calculated the energy barrier at 4 [kJ/mol] for a DPPC bilayer at 320K. **From the above mentioned studies it is clear that the lipid head energy barrier is a common characteristic in bilayers and monolayers and their values are fairly the same at around 4[kJ/mol], a value that was also predicted by our simulations.**

A noticeable difference with the bilayers was the lowest value of PMF. Usually in the middle of the cores, other studies predicted values of -7 [152] or -7.3 [155] to -8 [kJ/mol] [156, 157]. **Our simulations however, predicted a value almost the double at around -15[kJ/mol].** That was expected because the vacuum was an area of no density (while in the bilayers lipid molecules would exist) and the lipid chains in their ends, reached that area, reducing in that way their free-energy [155].

#### 6.4.4 Calculation of $D_{O_2}$

Based on Equation 5.10 and the calculated value of  $\Delta G$ , **the diffusion coefficient of Oxygen as it passed the lipid monolayer, was measured to be  $D_{O_2} = 1.5565 \cdot 10^{-5} [cm^2/s]$ .** In particular, apart from the work W that was considered equal to  $\Delta G$ , the distance was equal to the dragging distance ( $d=3.2364nm$ ) and t equal to 1ns.

Once again the validation of the results was made with bilayer studies since no values were found for monolayers as even Javanainen in his Thesis did not wrote the actual values of his simulations. Based on the literature review, Marrink and Berendsen [132] predicted a value of  $2 \cdot 10^{-5} [cm^2/s]$ , McKinnon et al. [99] for his study with hexadecane, predicted a value of 2.6 and Ju and Ho [135] calculated it at 2.49. Then Kowert and Dang [134] measured it at 2.66, Al-Abdul-Wahid et al. [133] at 3.7 and finally Fishkoff and Vanderkooi [115] at  $6.8 \cdot 10^{-5} [cm^2/s]$ .

By comparing this Thesis results with what studies of bilayers gave, it is clear that the monolayer had a relatively smaller diffusion coefficient. A possible reason for that, might be the anomalous, almost ballistic movement of the lipid molecules which may add an extra barrier in the movement of the Oxygen molecule in comparison to the more normal lateral movement of the bilayer molecules.





## Conclusions and Future Work

### Contents

---

<b>7.1</b>	<b>Conclusions . . . . .</b>	<b>111</b>
7.1.1	Model Creation and Comparison . . . . .	111
7.1.2	Energy Minimization . . . . .	112
7.1.3	NVT Equilibration . . . . .	112
7.1.4	Diffusion Coefficient . . . . .	114
<b>7.2</b>	<b>Future Work . . . . .</b>	<b>116</b>

---



## 7.1 Conclusions

Alveolar Surfactant plays a major role in the respiration process with its absence being responsible for 1% of prematurely-born infants mortality and its destruction by cigarettes the reason why smokers have respiration problems. As the surfactant is involved in every process of gas exchange in the lungs, the understanding of its working mechanism is of great importance.

**In this thesis, the diffusion of oxygen during the respiration process was tried to be measured as it would provide information on how the gas exchange works in the molecular level.** To achieve that, a monolayer model had to be constructed from scratch and evaluated for its validity. The conclusions of this work will be summarised in the following pages.

### 7.1.1 Model Creation and Comparison

The research of alveolar physiology showed that several different proposed forms of the alveolus area exist. In all the cases however apart (partially) of the Hills model, the surfactant-water interface was common. For this reason, regardless of the alveolar form, **the topologies that were constructed were universally valid.** Especially in the case of Scarpelli and Kitakoa, the existence of two lipid monolayer leaflets restraining a water slab, had also a physical meaning as well.

The composition of the surfactant in reality is a mixture of different lipid molecules (POPG, DPPC, POPC) and proteins (SP-A, SP-B, SP-C and SP-D). The modelling of the full composition of the actual alveolar surfactant would be too complex for the purposed of this Master Thesis especially since a previous model to be based on, did not exist. Therefore, **only the main component of the surfactant(40% by weight), the DPPC lipid molecule, was utilized for the created topologies.**

Based on the literature, two different topologies exist to model a lipid monolayer. The first is the physically meaningful lipid monolayer-water slab and the other is the lipids-water slab-lipids. The latter has the ability to keep the surface tension of the two leaflets the same as well retain the system more stable. In this Thesis, **both models were created using a bilayer topology constructed for the CHARMM36 force field by Klauda et al. [73]** and compared in order to decide which is the best for the simulations of Oxygen diffusion.

The pre-processing was made using the VMD software and the GROMACS code

since the latter **is the fastest and better optimized solver for biological systems**. The oxygen molecule that was also introduced in the system was designed in Avogadro molecular editor and manually inserted in the domain.

The two systems were examined for two different thermostats, Nose-Hoover and V-Rescale and for three different Areas per lipid, 50,60 and 70Å<sup>2</sup>.

### 7.1.2 Energy Minimization

During the Energy Minimization simulations the system reached the local (or ideally the global) minimum of their energy state.

In that phase, the systems converged in different rates to their minima. The 50Å<sup>2</sup> APL system had the slowest convergence speed and longest convergence time. Also, the 60Å<sup>2</sup> APL models had an average convergence speed but they were the fastest to find their minima. Finally, the 70Å<sup>2</sup> APL, had an incredibly fast convergence speed in their local minima area but it took a lot of time to finally converge. This has happened due to the steepest decent algorithm which is possible to show this kind of behaviour when the gradients around the local minima are very small. Finally, the system type did not affect with a consistent manner the minimization process and no conclusions could be extracted from that.

Regarding the numerics, **energy minimization had physically meaningful negative values** for all cases which allowed us to proceed to the next step.

### 7.1.3 NVT Equilibration

After the system was minimized, the Equilibration of the system was attempted in order to lose the initial conditions introduced by Maxwell distribution. Because the systems were modelled with the existence of vacuum around them, **only the NVT Ensemble was used during Equilibration**. In the case of NPT, pressure would shrink the system and the model would collapse. The systems ran for 1ns to ensure the full equilibration and the final state **after the end of 1ns was used to verify and validate the system** because it was necessary to determine the best system for the umbrella sampling simulations.

The first conclusion of the equilibration is that the **monolayer system was very unstable** with a continuous upward movement that was created by the escaped wa-

ter molecules bouncing on the lower walls of the system and then pushing the lower part of the water slab upwards. On the other hand, **symmetry was completely stable all the time** after the initial stabilization of the system ( $t \geq 50ps$ ).

Concerning the **Temperature convergence**, all cases (Models, Thermostats, APL) were equilibrated perfectly in the appropriate temperature of 323.15K with errors less than 0.1%. In fact, the fluctuations of temperature stayed well above the phase transition at 314K and that ensured that the DPPC molecules retained their Solid-Condensed form. The Nose-Hoover although it had a slightly more fluctuating behaviour (expected by its nature), performed overall better than V-Rescale.

The structure and order of the lipid chains was examined with the **deuterium order parameters**. It was observed that the smallest the APL, the highest the order of the lipid chains. **An increase of 40% of APL led to a decrease in order of 50%**. Another observation was that molecules were **more ordered in the head region** than in the upper tail area. The order observed in the monolayers was **lower in the tails than in the bilayers** due to the existence of vacuum instead of other lipid chains. The validation of the systems showed an error of 25% for the most compressed system, 5% for the  $60\text{\AA}^2$  APL and around 7% for the most expanded. Another conclusion that Deuterium Order Parameters did not alter severely from the Solid Condensed to the Liquid Condensed.

Systems' structure was examined by density profiles of electrons and mass. For the all the cases of the symmetry model the results were excellent and the systems had the theoretically expected behaviour. However, the upward movement of the monolayer created a distorted density profile where the actual values were lower due to the extended area covered.

Regarding the lateral diffusion of the lipid molecules calculated by the Mean Square Displacement of Phosphorus atoms, the two systems behaved differently. Monolayer systems retained a single-file diffusion in all the simulation time of 1ns without passing in the Fickian diffusion. That was considered wrong and a possible explanation is that **the MSD was affected by the normal movement of the system**. On the other hand, the symmetry models had a single-file diffusion for the initial stabilization time (10ps) and then the **diffusion changed gradually to Fickian and Fickian/Ballistic**, which was retained for the rest of the simulation. Overall, the lateral diffusion coefficients were not stabilized and it was concluded that the **existence of vacuum in the one side of the lipids, enhanced their movement**

**and thus made the whole behaviour more chaotic.**

The final parameter that was examined was the **Computational cost** for each simulation. Surprisingly enough, it was found that despite the fact of the double system size of the **symmetry systems, the simulation time was 30% to 40% shorter**. It was concluded that the upward movement of the monolayer apart from the other disadvantages, also made the system more slow to solve.

**Based on the previous conclusions, the best model to examine the diffusion of oxygen was the Symmetry model of the 60Å<sup>2</sup> APL with the Nose-Hoover thermostat and in any case the monolayer topology was too problematic to be used in any other study. Possibly special care would be necessary in order to make it behave correctly but despite all the effort no actual solutions were found in this Thesis.**

#### 7.1.4 Diffusion Coefficient

To measure the Diffusion coefficient the chosen Symmetry, 60Å<sup>2</sup>, Nose-Hoover system was used along with the Umbrella Sampling Technique. First the oxygen molecule was dragged through the lipid monolayer with a steered MD simulation and then every 0.025nm, a 1ns umbrella simulation was ran. In total 128 simulations were used to obtain the unbiased Mean Force Potential by using the Weighted Histogram Analysis Method and consequently the Diffusion Coefficient of Oxygen.

The effect of the harmonic force constant was examined by comparing a 1000 and 3000 value. It was found that **the highest the force constant, the harmonic movement became stiffer and therefore the sampling area became smaller while the samples taken at each point, more**. Since however, a small sampling area would require an enormous amount of points and simulations, the force constant of 1000 was chosen as the most appropriate.

The sampling spacing effect was also examined as in total five different sizes were simulated. It was concluded that **in order to obtain a reliable result from the umbrella sampling simulations, for a total length of 3.2nm, a total of 128 simulations at least** were necessary or a spacing of 0.025nm.

In the end, the produced PMF curve was compared with previous studies and several observations were made. First of all, **the energy barrier of 4[kJ/kg] in the head area**, found in many previous researches of bilayers and for a few

monolayers, **was also validated in this study.** On the other hand, the values of PMF in the chains and especially in the far end, were much lower than in the bilayers literature, **around -15[kJ/mol] in contrast to -7 to -8 [kJ/mol].** The reason of these behaviour can be explained by the huge gradient of density from the water slab to the vacuum showing apparently, that the energy required for the oxygen molecule to travel through the monolayer chains was more than in the bilayer. **The overall Free-Energy Difference of the oxygen diffusion was calculated as  $\Delta G = 16.674 [kJ/mol]$  or  $\Delta G = 6.729 [k_B T]$ .**

In the end **the Diffusion coefficient of Oxygen was calculated at  $1.557 \cdot 10^{-5} [cm^2/s]$  which was lower than the values found for bilayers in the literature.** This value is also lower than the  $2.6 \cdot 10^{-5} [cm^2/s]$  calculated by McKinnon et al. [99] where they examined the Oxygen diffusion through a hexadecane monolayer. In their studies however they did examined the system in 300K which is a liquid condensed/expanded phase where obviously the diffusion will be higher. The reason of lower than literature values is assumed to be the increased lateral movement of lipids (as seen by the MSD studies) that might imposed an extra barrier in the movement of oxygen. On the other hand, the fact that the oxygen molecule is very small and an almost unhindered movement was expected as well as the lack of other studies as source of validation data, makes the extraction of a definite conclusion regarding the oxygen diffusion rather difficult. **In any case our simulations predicted a smaller diffusion value than in the bilayer systems and future studies on the topic are necessary to validate the produced results.**

To conclude, this thesis was an initial attempt to build a simple surfactant model, validate it and then calculate the diffusion coefficient of oxygen as it was passing through it. All the objectives that were set during the beginning of the Thesis have been achieved successfully and the basis for future studies on the oxygenation of the alveolar surfactant has been built.



## 7.2 Future Work

As it was mentioned during the work, this system is a simplistic model of the actual alveolar surfactant. Future researchers should add more components in the system and examine their effect on the diffusion coefficient. Especially cholesterol is expected to increase significantly the diffusion of oxygen, especially in high concentrations.

Another parameter that should be examined in the future is the Temperature of the system. In realistic conditions, the human body is at  $36.6^{\circ}C$  and not at  $50^{\circ}C$  as it was assumed for this study. With the reduce of the temperature however new phenomena will be observed and the dynamics of the system will change.

Alternative methods to calculate the Diffusion Coefficient are also a possible research area with Thermodynamic integration to be a strong candidate. Also the lateral diffusion of Oxygen inside the lipids can be examined.

Overall, many research projects can be based on this Thesis since many questions are still open regarding the alveolar surfactant and its behaviour in the molecular level.

## Bibliography

- [1] R. •Rodriguez, R. Martin, and A. Fanaroff, *Neonatal-perinatal medicine: Diseases of the fetus and infant*, ch. Respiratory distress syndrome and its management, pp. 1001–1011. St.Louis: Mosby, 7 ed., 2002.
- [2] R. Schwartz, A. Luby, J. Scanion, and R. Kellogg, “Effect of surfactant on morbidity, mortality and resource use on newborn infants weighting 500 to 1500g.,” *New England Journal of Medicine*, vol. 330, pp. 1476–1480, 1994.
- [3] D. Thakar, A. Sinha, and O. Wenker, “Concepts of neonatal ecmo,” *The Internet Journal of Perfusionists*, vol. 1, no. 2, 2001.
- [4] A. Guyton and J. Hall, *Textbook of Medical Physiology*. USA: Elsevier Saunders, 11 ed., 2006.
- [5] A. Pramanik, “Respiratory distress syndrome,” September 2012. <http://emedicine.medscape.com/article/976034-overview>.
- [6] E. Scarpelli, “The alveolar surface network: A new anatomy and its physiological significance,” *The Anatomical Record*, vol. 251, pp. 491–527, 1998.
- [7] E. Scarpelli, “Physiology of the alveolar surface network,” *Comparative Biochemistry and Physiology*, vol. 135, pp. 39–104, 2003.
- [8] E. Scarpelli, “Lung surfactant: in vitro vs. in vivo,” *J. Appl. Physiol.*, vol. 94, pp. 1290–1292, 2003.
- [9] E. Scarpelli and B. Hills, “Opposing views on the alveolar surface, alveolar models and the role of surfactant,” *Journal of Applied Physiology*, vol. 89, pp. 408–412, 2000.
- [10] H. Bachofen, U. Gerber, and S. Schurch, “Effects of fixatives on function of pulmonary surfactant,” *J. Applied Physiology*, vol. 93, pp. 911–916, 2002.

- [11] B. Hills, "An alternative view of the role of surfactant and the alveolar model," *J. Appl. Physiology*, vol. 87, no. 5, pp. 1567–1583, 1999.
- [12] A. Bangham, "Geodesic planes to facilitate the extension of alveolar liquid/air interfaces," *Br. J. Anaesth.*, no. 87, pp. 519–520, 2001.
- [13] D. Exerowa, Z. Lalchev, and D. Kashchiev, "Stability of foam lipid bilayers of amniotic fluid," *Colloids Surf.*, no. 10, pp. 113–121, 1984.
- [14] D. Kashchiev and D. Exerowa, "Structure and surface energy of the surfactant layer on the alveolar surface," *Eur. Biophys.*, no. 30, pp. 34–41, 2001.
- [15] Z. Lalchev, R. Todorov, and D. Exerowa, "Thin liquid films as a model to study surfactant layers on the alveolar surface," *Current Opinion in Colloid and Interface Science*, vol. 13, pp. 183–193, 2008.
- [16] H. Kitaoka, G. Nieman, Y. Fujino, D. Carney, J. DiRocco, and I. Kawase, "A 4-dimensional model of the alveolar structure," *J. Physiol. Sci.*, vol. 57, no. 3, pp. 175–185, 2007.
- [17] H. Kitakoa and I. Kawase, "Novel interpretation of closing volume based on single-breath nitrogen washout curve simulation," *J. Physiol. Sci.*, vol. 57, no. 6, pp. 367–376, 2007.
- [18] H. Kitakoa, "The origin of frequency dependence of respiratory resistance: An airflow simulation study using a 4d pulmonary lobule model," *Respirology*, vol. 16, pp. 517–522, 2011.
- [19] P. Morters and P. Yuval, *Brownian Motion*. •, 2008. <http://www.stat.berkeley.edu/~peres/bmbook.pdf>.
- [20] A. Hac, *Diffusion Processes in Membranes Containing Coexisting Domains Investigated by Fluorescence Correlation Spectroscopy*. PhD thesis, University of Gottingen, 2003.
- [21] A. Einstein, "Über die von der molekularkinetischen theorie der warme geforderte bewegung von in ruhenden flussigkeiten suspendierten teilchen," *Annalen der Physik*, vol. 322, no. 8, pp. 549–560, 1905.
- [22] M. von Smoluchowski, "Zur kinetischen theorie der brownschen molekularbewegung und der suspensionen," *Annalen der Physik*, vol. 326, no. 14, pp. 756–780, 1906.
- [23] D. Tieleman, S. Marrink, and H. Berendsen, "A computer perspective of membranes: Molecular dynamics studies of lipid bilayer systems," *Biochimica et Biophysica Acta*, vol. 1331, pp. 235–270, 1997.
- [24] P. Almeida and W. Vaz, *Handbook of Biological Physics*, vol. 1, ch. Chapter 6. Lateral Diffusion in Membranes. Elsevier Science B.V., 1995.

- 
- [25] J. Sane, J. Padding, and L. A.A., “The crossover from single file to fickian diffusion,” *Faraday Discuss.*, vol. 144, pp. 285–299, 2010.
- [26] D. Bassolino-Klimas, H. Alper, and T. Stouch, “Mechanism of solute diffusion through lipid bilayer membranes by molecular dynamics simulation,” *J.Am.Chem.Soc.*, vol. 117, pp. 4118–4129, 1995.
- [27] J. Anderson, *Computational Fluid Dynamics - The Basics with Applications*. U.S.A: McGraw-Hill, 1995.
- [28] E. Cussler, *Diffusion: Mass Transfer in Fluid Systems*. Cambridge: Cambridge University Press, 1997.
- [29] L. Garcia-Colin, R. Velasco, and F. Uribe, “Beyond the navier-stokes equations: Burnett hydrodynamics,” *Physics Reports*, vol. 465, 2008.
- [30] R. Agarwal, K. Yun, and R. Balakrishnan, “Beyond navier-stokes: Burnett equations for flows in the continuum-transition regime,” *Physics of Fluids*, vol. 13, no. 10, pp. 3061–3085, 2001.
- [31] S. Baeurle, “Multiscale modeling of polymer materials using field-theoretic methodologies: A survey about recent developments,” *Journal of Mathematical Chemistry*, vol. 46, no. 2, pp. 363–426, 2009.
- [32] S. Baoukina, S. Marrink, and D. Tieleman, *Biomembrane Frontiers: Nanostructures, Models and the Design of Life*, ch. Structure and Dynamics of Lipid Monolayers: Theory and Applications, pp. 75–99. Handbook of Modern Biophysics, Humana Press, 2009.
- [33] M. Allen and D. Tildesley, *Computer Simulation of Liquids*. Oxford University Press, 1987.
- [34] W. Morell and J. Hilderbrand, “The distribution of molecules in a model liquid,” *J. Chem. Phys.*, vol. 4, pp. 224–227, 1936.
- [35] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *J. Chem. Phys.*, vol. 21, pp. 1087–1092, 1953.
- [36] W. Wood and F. Parker, “Monte carlo equation of state of molecules interacting with the lennard-jones potential. i. a supercritical isotherm at about twice the critical temperature,” *J. Chem. Phys.*, vol. 27, pp. 720–733, 1957.
- [37] B. Alder and T. Wainwright, “Phase transition for a hard-sphere system,” *Journal of Chemical Physics*, vol. 27, pp. 1208–1209, 1957.
- [38] B. Alder and T. Wainwright, “Studies in molecular dynamics. i. general method,” *Journal of Chemical Physics*, vol. 31, pp. 459–466, 1959.
- [39] A. Rahman, “Correlations in the motion of atoms in liquid argon,” *Phys. Rev.*, vol. 136A, pp. 405–411, ●.
-

- [40] A. Rahman and F. Stillinger, “Molecular dynamics study of liquid water,” *J. Chem. Phys.*, vol. 55, pp. 3369–3359, 1971.
- [41] J. McCammon, B. Gelin, and M. Karplus, “Dynamics of folded proteins,” *Nature*, vol. 267, pp. 585–590, 1977.
- [42] J. Lee, J. Barker, and G. Pound, “Surface structure and surface tension: perturbation theory and monte carlo calculation,” *J. Chem. Phys.*, vol. 60, pp. 1976–1980, 1974.
- [43] G. Chapela, G. Saville, S. Thompson, and J. Rowlinson, “Computer simulation of a gas-liquid surface. part 1,” *J. Chem. Soc. Faraday II*, 1977.
- [44] D. Frenkel, “Intermolecular spectroscopy and computer simulations. intermolecular spectroscopy and dynamical properties of dense systems,” in *Proceedings of the Enrico Fermi Summer School*, vol. 75, (Bologna), pp. 156–201, Soc. Italiana di Fisica, 1980.
- [45] N. Corbin and K. Singer, “Semiclassical molecular dynamics of wave packets,” *Mol. Phys.*, vol. 46, pp. 671–677, 1982.
- [46] D. Ceperlay and M. Kalos, “Quantum many-body problems.in monte carlo methods in statistical physics,” in *Topics in Current Physics* (K. Binder, ed.), vol. 7, (Berlin), pp. 145–194, Springer, 1986.
- [47] G. Karniadakis, A. Beskok, and N. Aluru, *Microflows and Nanoflows*. Springer, 2005.
- [48] D. van der Spoel, E. Lindahl, B. Hess, and the GROMACS development team, “Gromacs user manual version 4.6.1,” ●, 2013. [www.gromacs.com](http://www.gromacs.com).
- [49] N. Asproulis, *Hybrid Molecular and Continuum Fluid Dynamics Models for Micro and Nanofluidic Flows*. PhD thesis, Cranfield University, 2009.
- [50] T. Kibble and F. Berkshire, *Classical Mechanics*. Imperial College Press, 2004.
- [51] G. Maitland, M. Rigby, E. Smith, and W. Wakeham, *Intermolecular Forces: Their Origin and Determination*. Oxford: Clarendon Press, 1981.
- [52] A. Stone, *Theory of Intermolecular Forces*. Oxford: Clarendon Press, 1996.
- [53] A. Leach, *Molecular Modelling, Principles and Applications*. Pearson Education, 2 ed., 2001.
- [54] M. Born and J. Mayer, “Zur gittertheorie der ionenkristalle,” *Z. Physik*, vol. 75, pp. 1–18, 1932.
- [55] L. Verlet, “Computer experiments on classical fluids. i. thermodynamical properties of lennard-jones molecules,” *Physical Review*, vol. 159, pp. 98–103, 1967.

- 
- [56] R. Hockney, "The potential calculation and some applications," *Methods in Computational Physics*, vol. 9, pp. 136–211, 1970.
- [57] W. Swope, H. Anderson, P. Berens, and W. K.R., "A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters," *J. Chem. Phys.*, vol. 76, pp. 637–649, 1982.
- [58] C. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*. Englewood Cliffs, NJ: Prentice Hall, 1971.
- [59] A. Hinchliffe, *Molecular Modelling for Beginners*. England: Wiley, 2003.
- [60] H. Berendsen, J. Postma, W. van Gunsteren, A. DiNola, and J. Haak, "Molecular dynamics with coupling to an external bath," *J. Chem. Phys.*, vol. 81, pp. 3684–3689, 1984.
- [61] D. van der Spoel, E. Lindahl, B. Hess, A. van Buuren, E. Apol, P. Meulenhoff, D. Tieleman, A. Sijbers, K. Feenstra, R. van Drunen, and H. Berendsen, *Gromacs User Manual v3.2*.
- [62] S. Nose, "A molecular dynamics method for simulations in the canonical ensemble," *Mol. Phys.*, vol. 52, pp. 255–268, 1984.
- [63] W. Hoover, "Canonical dynamics: equilibrium phase-space distributions," *Phys. Rev. A.*, vol. 31, pp. 1695–1697, 1985.
- [64] G. Bussi, D. Donadio, and M. Parrinello, "Canonical sampling through velocity rescaling," *J. Chem. Phys.*, vol. 126, no. 014101, 2007.
- [65] M. Parrinello and A. Rahman, "Polymorphic transitions in single crystals: A new molecular dynamics method," *J. Appl. Phys.*, vol. 52, pp. 7182–7190, 1981.
- [66] S. Nose and M. Klein, "Constant pressure molecular dynamics for molecular system," *Mol. Phys.*, vol. 50, pp. 1055–1076, 1983.
- [67] G. Martyna, M. Tuckerman, D. Tobias, and M. Klein, "Explicit reversible integrators for extended system dynamics," *Mol. Phys.*, vol. 87, pp. 1117–1157, 1996.
- [68] M. E. Tuckerman, J. Alejandre, R. Lopez-Rendon, A. L. Jochim, and G. J. Martyna, "A liouville-operator derived measure-preserving integrator for molecular dynamics simulations in the isothermal-isobaric ensemble," *J. Phys. A.*, vol. 59, pp. 5629–5651, 2006.
- [69] T.-Q. Yu, J. Alejandre, R. Lopez-Rendon, G. J. Martyna, and M. E. Tuckerman, "Measure preserving integrators for molecular dynamics in the isothermal-isobaric ensemble derived from the liouville operator.," *Chem. Phys.*, vol. 370, pp. 294–305, 2010.

- [70] E. Fahy, S. Subramaniam, H. Brown, C. Glass, A. Merrill, R. Murphy, C. Raetz, D. Russell, Y. Seyama, W. Shaw, T. Shimizu, F. Spener, G. van Meer, M. Van Nieuwenhze, S. White, J. Witztum, and E. Dennis, "A comprehensive classification system for lipids," *Eur. J. Lipid Sci. Technol.*, vol. 107, no. 5, pp. 337–364, 2005.
- [71] O. Mouritsen, *Life - As a matter of fat*. Springer, 2005.
- [72] R. Aveyard and D. Haydon, *An introduction to the principles of surface chemistry*. New York: Cambridge UP, 1973.
- [73] J. Klauda, R. Venable, J. Freites, J. O'Connor, D. Tobias, C. Mondragon-Ramirez, I. Vorobyov, A. MacKerell Jr., and R. Pastor, "Update of the charmm all-atom additive force field for lipids: Validation on six lipid types," *Journal of Physical Chemistry B*, vol. 114, no. 23, pp. 7830–7843, 2010.
- [74] G. Ma and H. Allen, "Dppc langmuir monolayer at the air-water interface: Probing the tail and head groups by vibrational sum frequency generation spectroscopy," *Langmuir*, vol. 22, no. 12, pp. 5341–5349, 2006.
- [75] S. Baoukina, E. Mendez-Villuendas, and D. Tieleman, "Molecular view of phase coexistence in lipid monolayers," *Journal of the American Chemical Society*, vol. 134, no. 42, pp. 17549–17553, 2012.
- [76] E. Mendez-Villuendas, S. Baoukina, and D. Tieleman, "Challenges in analysing and visualizing large-scale molecular dynamics simulations: Domain and defect formation in lung surfactant monolayers," in *Journal of Physics: Conference Series*, vol. 385, 2012.
- [77] S. Baoukina, L. Monticelli, M. Amrein, and D. Tieleman, "The molecular mechanism of monolayer-bilayer transformations of lung surfactant from molecular dynamics simulations," *Biophysical Journal*, vol. 93, no. 11, pp. 3775–3782, 2007.
- [78] S. Duncan, I. Dalal, and R. Larson, "Molecular dynamics simulation of phase transitions in model lung surfactant monolayers," *Biochimica et Biophysica Acta - Biomembranes*, vol. 1808, no. 10, pp. 2450–2465, 2011.
- [79] P. Shushkov, S. Tzvetanov, M. Velinova, A. Ivanova, and A. Tadjer, "Structural aspects of lipid monolayers: Computer simulation analyses," *Langmuir*, vol. 26, no. 11, pp. 8081–8092, 2010.
- [80] J. Perez-Gil, "Structure of pulmonary surfactant membranes and films: The role of proteins and lipid-protein interactions," *Biochimica et Biophysica Acta*, vol. 1778, pp. 1676–1695, 2008.
- [81] M. Khabiri, M. Roeselova, and J. Cwiklik, "Properties of oxidized phospholipid monolayers: An atomistic molecular dynamics study," *Chemical Physics Letters*, vol. 519-520, pp. 93–99, 2012.

- 
- [82] S. Duncan and R. Larson, "Folding of lipid monolayers containing lung surfactant proteins sp-b(1-25) and sp-c studied via coarse-grained molecular dynamics simulations," *Biochimica et Biophysica Acta - Biomembranes*, vol. 1798, no. 9, pp. 1632–1650, 2010.
- [83] A. Freitas, Y. Choi, and D. Tobias, "Molecular dynamics simulations of a pulmonary surfactant protein b peptide in a lipid monolayer," *Biophysical Journal*, vol. 84, pp. 2169–2180, 2003.
- [84] C. Laing, S. Baoukina, and P. Tieleman, "Molecular dynamics study of the effect of cholesterol on the properties of lipid monolayers at low surface tensions," *Physical Chemistry Chemical Physics*, vol. 11, no. 12, pp. 1916–1922, 2009.
- [85] S. Tzvetanov, P. Shushkov, M. Velinova, A. Ivanova, and A. Tadjer, "Molecular dynamics study of the electric and dielectric properties of model dppc and dicaprin insoluble monolayers: Size effect," *Langmuir*, vol. 26, no. 11, pp. 8093–8105, 2010.
- [86] S. Baoukina, S. Marrink, and D. Tieleman, "Lateral pressure profiles in lipid monolayers," *Faraday Discussions*, vol. 144, pp. 393–409, 2009.
- [87] T. Sosnowski, M. Koliński, and L. Gradon, "Alteration of surface properties of dipalmitoyl phosphatidylcholine by benzo[a]pyrene: A model of pulmonary effects of diesel exhaust inhalation," *Journal of Biomedical Nanotechnology*, vol. 8, 2012.
- [88] D. Rose, J. Rendell, D. Lee, K. Nag, and V. Booth, "Molecular dynamics simulations of lung surfactant lipid monolayers," *Biophysical Chemistry*, vol. 138, no. 3, pp. 67–77, 2008.
- [89] B. Brooks, C. r. Brooks, A. J. Mackerell, L. Nilsson, R. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caffisch, L. Caves, Q. Cui, A. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. Pastor, C. Post, J. Pu, M. Schaefer, B. Tidor, R. Venable, H. Woodcock, X. Wu, W. Yang, D. York, and M. Karplus, "Charmm: The biomolecular simulation program," *Journal of Computational Chemistry*, vol. 30, no. 10, pp. 15445–1614, 2009.
- [90] J. A. MacKerell, N. Banavali, and N. Foloppe, "Development and current status of the charmm force field for nucleic acids," *Biopolymers*, vol. 56, no. 4, pp. 257–265, 2001.
- [91] J. A. MacKerell, D. Bashford, M. Bellott, R. Dunbrack Jr., J. Evanseck, M. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. Lau, C. Mattos, S. Michnick, T. Ngo, D. Nguyen, B. Prodhom, W. Reiher III, B. Roux, M. Schlenkrich, J. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus, "All-atom empirical potential for molecular modeling and dynamics studies of proteins," *J Phys Chem B*, vol. 102, no. 18, pp. 3586–3616, 1998.
-



- [92] R. Best, X. Zhu, J. Shim, P. Lopes, J. Mittal, M. Feig, and A. MacKerell Jr., "Optimization of the additive charmm all-atom protein force field targeting improved sampling of the backbone  $\phi$ ,  $\psi$  and side-chain  $\chi_1$  and  $\chi_2$  dihedral angles," *Journal of Chemical Theory and Computation*, vol. 8, no. 9, pp. 3257–3273, 2012.
- [93] W. Jorgensen, J. Chandrasekhar, D. Madura, I. R.W., and M. Klein, "Comparison of simple potential functions for simulating liquid water," *J. Chem. Phys.*, vol. 79, pp. 926–935, 1983.
- [94] W. Van Gunsteren, *GROMOS: Groningen molecular simulation program package*. University of Groningen, 1987.
- [95] O. Berger, O. Edholm, and F. Jahnig, "Molecular dynamics simulations of a fluid bilayer of dipalmitoylphosphatidylcholine at full hydration, constant pressure and constant temperature," *Biophysics journal*, vol. 72, no. 5, pp. 2002–2013, 1997.
- [96] J. Ryckaert and A. Bellemans, "Molecular dynamics of liquid alkanes," *Faraday Discuss. Chem. Soc.*, vol. 66, pp. 95–106, 1978.
- [97] S.-W. Chiu, M. Clark, V. Balaji, S. Subramaniam, H. Scott, and E. Jakobsson, "Incorporation of surface tension into molecular dynamics simulation of an interface: A fluid phase lipid bilayer membrane," *Biophysical Journal*, vol. 69, pp. 1230–1245, 1995.
- [98] S. Choe, R. Chang, J. Jeon, and A. Violi, "Molecular dynamics simulation study of a pulmonary surfactant film interacting with a carbonaceous nanoparticle," *Biophysical Journal*, vol. 95, no. 9, pp. 4102–4114, 2008.
- [99] S. McKinnon, S. Whittenburg, and B. Brooks, "Nonequilibrium molecular dynamics simulation of oxygen diffusion through hexadecane monolayers with varying concentrations of cholesterol," *J. Phys. Chem.*, vol. 96, pp. 10497–10506, 1992.
- [100] Y. Zhang, S. Feller, R. Brooks, and R. Pastor, "Computer-simulation of liquid/liquid interfaces,1:theory and application to octane/water," *J. Chem. Phys.*, vol. 103, no. 23, pp. 10252–10266, 1995.
- [101] S. Feller, Y. Zhang, and R. Pastor, "Computer-simulation of liquid/liquid interfaces,2:surface-tension area dependance of a bilayer and monolayer," *J. Chem. Phys.*, vol. 103, no. 23, pp. 10267–10276, 1995.
- [102] G. Martyna, D. Tobias, and M. Klein, "Constant pressure molecular dynamics algorithms," *J. Chem. Phys.*, vol. 101, pp. 4177–4189, 1994.
- [103] S. Feller, Y. Zhang, and R. Pastor, "Constant pressure molecular dynamics simulation: The langevin piston method," *J. Chem. Phys.*, vol. 103, no. 23, pp. 4613–4621, 1995.

- 
- [104] E. Egberts and H. Berendsen, “Molecular dynamics simulation of a smectic liquid crystal with atomic detail,” *J. Chem. Phys.*, vol. 89, pp. 3718–3732, 1988.
- [105] E. Egberts, S. Marrink, and H. Berendsen, “Molecular dynamics simulation of a phospholipid membrane,” *Eur. Biophys. J.*, vol. 22, pp. 423–426, 1994.
- [106] D. Tieleman and H. Berendsen, “Molecular dynamics simulations of a fully hydrated dipalmitoylphosphatidylcholine bilayer with different macroscopic boundary conditions and parameters,” *J. Chem. Phys.*, vol. 105, pp. 4871–4880, 1996.
- [107] H. Alper, D. Bassolino, and T. Stouch, “Computer simulation of a phospholipid monolayer-water system. the influence of long range forces on water structure and dynamics,” *J. Chem. Phys.*, vol. 98, pp. 9798–9807, 1993.
- [108] H. Heller, M. Schaefer, and K. Schulten, “Molecular dynamis simulations of a bilayer of 200 lipids in the gel and in the liquid-cristal phases,” *J. Phys. Chem.*, vol. 97, pp. 8343–8360, 1993.
- [109] F. Zhou and K. Schulten, “Molecular dynmaics study of a membrane-water interface,” *J. Phys. Chem.*, vol. 99, pp. 2194–2207, 1995.
- [110] P. Ewald, “Die berechnung optischer und elektrostatischer gitterpotentiale,” *Ann. Phys.*, vol. 369, pp. 253–287, 1921.
- [111] T. Schlick, *Molecular Modeling and Simulation: An Interdisciplinary Guide*, vol. 21 of *Interdisciplinary Applied Mathematics, Mathematical Biology*. New York: Springer-Verlag, 2002.
- [112] U. Essman, L. Perera, M. Berkowitz, T. Darden, H. Lee, and P. L.G., “A smooth particle mesh ewald method,” *J. Chem. Phys.*, vol. 103, pp. 8577–8593, 1995.
- [113] I. Yeh and M. Berkowitz, “Ewald summation for systems with slab geometry,” *J. Chem. Phys.*, vol. 111, pp. 3155–3162, 1999.
- [114] R. Hockney and J. Eastwood, *Computer Simulation using Particles*, ch. Particle-Particle-Particle-Mesh P3M Algorithms, pp. 267–304. CRC Press, 1988.
- [115] S. Fishkoff and J. Vanderkooi, “Oxygen diffusion in biological and artificial membranes determined by the fluorochrome pyrene,” *J. Gen. Physiology*, vol. 65, pp. 663–667, 1975.
- [116] L. Monticelli, E. Salonen, P. Ke, and I. Vattulainen, “Effects of carbon nanoparticles on lipid membranes: A molecular simulation perspective,” *Soft Matter*, vol. 5, no. 22, pp. 4433–4445, 2009.
-

- [117] C. Aberg, E. Sparr, M. Larsson, and H. Wennerstrom, "A theoretical study of diffusional transport over the alveolar surfactant layer," *J. R. Soc. Interface*, vol. 7, pp. 1403–1410, 2010.
- [118] B. Olmeda, L. Villen, A. Cruz, G. Orellana, and J. Perez-Gil, "Pulmonary surfactant layers accelerate  $O_2$  diffusion through the air-water interface," *Biochimica et Biophysica Acta*, vol. 1798, pp. 1281–1284, 2010.
- [119] I. Ivanov, G. Fedorov, R. Gus'kova, K. Ivanov, and A. Rubin, "Permeability of lipid membranes to dioxygen," *Biochemical and Biophysical Research Communications*, vol. 322, pp. 746–750, 2004.
- [120] A. Seelig and J. Seelig, "The dynamic structure of fatty acid chains in a phosphatidylcholine bilayer measured by deuterium magnetic resonance," *Biochemistry*, vol. 13, pp. 4839–4845, 1974.
- [121] J. Seelig and W. Niederberger, "Deuterium-labeled lipids as structural probes in liquid crystalline bilayers," *J. Am. Chem. Soc.*, vol. 96, pp. 2069–2072, 1974.
- [122] H. Schindler and J. Seelig, "Deuterium order parameters in relation to thermodynamic properties of a phospholipid bilayer. a statistical mechanical interpretation.," *Biochemistry*, vol. 14, 1975.
- [123] A. Gericke, D. Moore, R. Erukulla, R. Bittman, and R. Mendelsohn, "Partially deuterated phospholipids as structural probes of conformational order in bulk and monolayer phases," *Journal of Molecular Structure*, vol. 379, pp. 227–239, 1996.
- [124] J. Douliez, A. Leonard, and E. Dufourc, "Restatement of order parameters in biomembranes: Calculation of c-c bond order parameters from c-d quadrupolar splittings," *Biophysics Journal*, vol. 68, pp. 1727–1739, 1995.
- [125] Y. Kaznessis, S. Kim, and R. Larson, "Simulations of zwitterionic and anionic phospholipid monolayers," *Biophysics Journal*, vol. 82, pp. 1731–1742, 2002.
- [126] M. Pickholz, O. Oliveira, and M. Skaf, "Molecular dynamics simulations of neutral chlorpromazine in zwitterionic phospholipid monolayers," *J. Phys. Chem. B.*, vol. 110, pp. 8804–8814, 2006.
- [127] H. Dominguez, A. Smondyrev, and M. Berkowitz, "Computer simulations of phosphatidylcholine monolayers at air/water and  $CCl_4$ /water interfaces," *J. Phys. Chem. B*, vol. 103, no. 44, pp. 9582–9588, 1999.
- [128] T. Lucas, B. Bauer, J. Davis, and S. Patel, "Molecular dynamics simulation of hydrated dppc monolayers using charge equilibration force fields," *Journal of Computational Chemistry*, vol. 33, pp. 141–152, 2012.
- [129] R. Ferrell and D. Himmelblau, "Diffusion coefficients of nitrogen and oxygen in water," *J. Chem. Eng. D.*, vol. 12, pp. 111–115, 1967.

- 
- [130] I. Krieger, G. Mulholland, and C. Dickey, "Diffusion coefficients for gases in liquids from the rates of solution of small gas bubbles," *J. Phys Chem*, vol. 71, pp. 1123–1129, 1967.
- [131] P. Han and D. Bartels, "Temperature dependance of oxygen diffusion in  $h_2o$  and  $d_2o$ ," *J Phys Chem*, vol. 100, pp. 5597–5602, 1996.
- [132] S. Marrink and H. Berendsen, "Permeation process of small molecules across lipid membranes studied by molecular dynamics simulation," *J. Phys. Chem*, vol. 100, pp. 16729–16738, 1996.
- [133] M. Al-Abdul-Wahid, C. Yu, I. Batruch, F. Evancics, R. Pomes, and R. Prosser, "A combined nmr and molecular dynamics study of the transmembrane solubility and diffusion rate profile of dioxygen in lipid bilayers," *Biochemistry*, vol. 45, pp. 10719–10728, 2006.
- [134] B. Kowert and N. Dang, "Diffusion of dioxygen in n-alkanes," *J.Phys.Chem.A*, vol. 103, pp. 779–781, 1999.
- [135] H. Ju and C. Ho, "Oxygen diffusion coefficient and solubility in n-hexadecane," *Biotechnology and Bioengineering*, vol. 34, pp. 1221–1224, •.
- [136] H. Humphrey, A. Dalke, and K. Schulten, "Vmd - visual molecular dynamics," *J. Molec. Graphics*, vol. 14, no. 1, pp. 33–38, 1996.
- [137] H. Bekker, B. H. J. C., E. Dijkstra, S. Achterop, R. van Drunen, D. van der Spoel, A. Sijbers, H. Keegstra, B. Reitsma, and M. Renardus, "Gromacs: A parallel computer for molecular dynamics simulations," in *Physics Computing 92* (R. de Groot and J. Nadrchal, eds.), (Singapore), World Scientific, 1993.
- [138] H. Berendsen, D. van der Spoel, and R. van Drunen, "Gromacs: A message-passing parallel molecular dynamics implementation," *Computer Physics Communications*, vol. 91, pp. 43–56, 1995.
- [139] E. Lindahl, B. Hess, and D. van der Spoel, "Gromacs 3.0: A package for molecular simulation and trajectory analyss," *J. Mol. Mod.*, vol. 7, pp. 306–317, 2001.
- [140] D. van der Spoel, E. Lindahl, B. Hess, G. Groenhof, M. A.E., and H. Berendsen, "Gromacs: Fast, flexible and free," *J. Comp. Chem.*, vol. 26, pp. 1701–1718, 2005.
- [141] B. Hess, C. Kutzner, D. van der Spoel, and E. Lindahl, "Gromacs 4.0: Algorithms for highly efficient, load-balanced and scalable molecular simulation," *J. Chem. Theory Comp*, vol. 4, no. 3, pp. 435–447, 2008.
- [142] S. Pronk, S. Pall, R. Schulz, P. Larsson, P. Bjelkmar, R. Apostolov, M. Shirts, J. Smith, P. Kasson, D. van der Spoel, B. Hess, and E. Lindahl, "Gromacs 4.5: A high throughput and highly parallel open source molecular simulation toolkit," *Bioinformatics*, vol. 29, no. 7, pp. 845–854, 2013.
-

- [143] A. Gittings, *Cranfield University Astral HPC Cluster - User Guide - Issue 1.4*. IT Department, Cranfield University, March 2013.
- [144] J. Freites, J. O'Connor, D. Tobias, C. Mondragon-Ramirez, I. Vorobyov, A. MacKerell, J. Klauda, R. Venable, and P. J.W., "Update of the charmm all-atom additive force field for lipids: Validation of six lipid types," *Journal of Physical Chemistry B*, vol. 114, no. 7830-7843, 2010.
- [145] S. Duncan and R. Larson, "Comparing experimental and simulated pressure-area isotherms for dppc," *Biophysics Journal*, vol. 94, pp. 2965–2986, 2008.
- [146] F. Castro-Roman, R. Benz, S. White, and D. Tobias, "Investigation of finite system-size effects in molecular dynamics simulations of lipid bilayers," *Journal of Physical Chemistry B*, vol. 110, no. 47, pp. 24157–24164, 2006.
- [147] D. Marsh, *Handbook of Lipid Bilayers*. CRC Press, 2nd ed., 2013.
- [148] C. Jarzynski, "Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach," *Phys. Rev. E*, vol. 56, p. 5018, 1997.
- [149] C. Jarzynski, "Nonequilibrium equality for free energy differences," *Phys. Rev. Lett.*, vol. 78, p. 2690, 1997.
- [150] G. Crooks, "Nonequilibrium measurements of free energy differences for microscopically reversible markovian systems," *J. Stat. Phys.*, vol. 90, p. 1481, 1998.
- [151] J. Kastner, "Umbrella sampling." Universitat Stuttgart, March 2011.
- [152] J. Hub, B. de Groot, and D. van der Spoel, "g\_wham—a free weighted histogram analysis implementation including robust error and autocorrelation estimates," *J. Chem. Theory Comput.*, vol. 6, no. 12, pp. 3713–3720, 2010.
- [153] C. Hofsass, E. Lindahl, and O. Edholm, "Molecular dynamics simulations of phospholipid bilayers with cholesterol," *Biophysical J.*, vol. 84, no. 4, pp. 2192–2206, 2003.
- [154] S. Espinosa-Gutierrez, "Molecular dynamics cell membrane modelling, application to ultrasound cancer treatment," Master's thesis, School of Engineering, Cranfield University, UK, 2012.
- [155] M. Javanainen, "Oxygen-to lung surfactant and beyond," Master's thesis, Tampere University of Technology, Tampere, Finland, 2012.
- [156] W. Shinoda, M. Mikami, T. Baba, and M. Hato, "Molecular dynamics study on the effects of chain branching on the physical properties of lipid bilayers: 2. permeability," *J Phys Chem B*, vol. 108, pp. 9346–9356, 2004.
- [157] T. Sugii, S. Takagi, and Y. Matsumoto, "A molecular-dynamics study of lipid bilayers: effects of the hydrocarbon chain length on permeability.," *J. Chem. Phys.*, vol. 123, no. 184714, 2005.

*A*

**Pre-processing Scripts**

## A.1 Creation of Monolayer Topology - *Monolayer.tcl*

---

```
# MONOLAYER TOPOLOGY CREATION
# PROGRAMMER: M.PALAIOKOSTAS-AVRAMIDIS

#LOAD KLAUDA'S TOPOLOGY
mol delete all
mol new "dppc72-c36npt.pdb"

# SELECT LOWER PHOSPHORUS ATOMS
set P_lower "(name P and z<0)"

# SELECT LIPID ATOMS WITH SAME RESIDUE NUMBER AS P ATOMS
set lipids_lower [atomselect top "(lipid and same residue as $P_lower)"]

# SAVE LOWER LIPID LEAFLET TO .PDB
$lipids_lower writepdb dppc_lower.pdb

# LOAD NEW TOPOLOGY
mol delete all
mol new dppc_lower.pdb

# CHANGE DRAWING METHOD
mol rep vdw
mol addrep top
mol delrep 0 top
```

---

---

## A.2 Creation Symmetry Topology - *Symmetry.tcl*

---

```
# SYMMETRY TOPOLOGY CREATION
# PROGRAMMER: M.PALAIOKOSTAS-AVRAMIDIS

#LOAD KLAUDA'S TOPOLOGY
mol delete all
mol new "dppc72-c36npt.pdb"

# SELECT LIPID ATOMS WITH SAME RESIDUE NUMBER AS P ATOMS
set P_upper "(name P and z>0)"
set lipids_upper [atomselect top "(lipid and same residue as $P_upper)"]
$lipids_upper moveby {0 0 -157.340}
$lipids_upper delete

set lipids_all [atomselect top "(lipid)"]

# CHANGE DRAWING METHOD
mol rep points
mol selection "lipids"
mol addrep top
mol delrep 0 top

# SAVE LOWER LIPID LEAFLET TO .PDB
$lipids_all writepdb dppc_symmetry.pdb

# LOAD NEW TOPOLOGY
mol delete all
mol new dppc_symmetry.pdb
```

---



### A.3 Geometrical Parameters for Monolayer - *Geometrical\_Parameters\_MN.m*

---

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% DOMAIN GEOMETRICAL CHARACTERISTICS FOR A LIPID MONOLAYER
% AND OXYGEN ADDITION PARAMETERS
% MONOLAYER TOPOLOGY
%
% PROGRAMMER: MICHAEL PALAIOKOSTAS AVRAMIDS
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear
close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% USER INPUT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('-----')
disp('-MONOLAYER TOPOLOGY-')
disp('-----')

APL=0.50;      % Area per Lipid required [nm^2]
N_L=36;       % Number of Lipids per Lipid Leaflet

InitialLipid.x=5.95280; % Lipid Length [nm]
InitialLipid.y=6.14620; % Lipid Width [nm]
InitialLipid.z=3.3258; % Lipid Height [nm]

domain_z=150; % Domain of Solution Height [nm]
water_slab=5; % Water Slab Height [nm] - !Should be larger than cut-off!

dist_o2=60.0; % Distance between O2 molec. and lipid center [nm]

% Oxygen Atom Coordinates based on Avogadro output in Angstroms
P1.x= -1.3792;      % Atom 1 - X coordinate [A]
P1.y= +0.303;      % Atom 1 - Y coordinate [A]
P1.z= -1.477;      % Atom 1 - Z coordinate [A]
```

```

P2.x= -1.912; % Atom 1 - X coordinate [A]
P2.y= +0.346; % Atom 1 - Y coordinate [A]
P2.z= -0.316; % Atom 1 - Z coordinate [A]

disp('AREA PER LIPID [nm^2]:')
disp(APL)
disp('-----')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BOX DIMENSIONS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x=sqrt(APL*N_L); % x-y Length based on A.P.L

box.x=x; % Box x Vector [nm]
box.y=x; % Box y Vector [nm]
box.z=domain_z; % Box z Vector [nm]

disp('BOX DIMENSIONS [nm]:')
box
disp('-----')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LIPID SCALING
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ScaleFactor.x= box.x/InitialLipid.x;
ScaleFactor.y= box.y/InitialLipid.y;
ScaleFactor.z= 1;

disp('SCALE FACTORS [-]:')
ScaleFactor
disp('-----')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% GEOMETRICAL CENTERS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
lipid_center.x=box.x/2; % Center of Lipid leaflet x [nm]
lipid_center.y=box.y/2; % Center of Lipid leaflet y [nm]
lipid_center.z=(box.z/2)+water_slab+(InitialLipid.z/2); % Center of
Lipid leaflet z [nm]

```

```
water_center.x=box.x/2; % Center of Water Slab x [nm]
water_center.y=box.y/2; % Center of Water Slab y [nm]
water_center.z=(box.z/2)+(water_slab/2); % Center of Water Slab z [nm]

disp('CENTERS [nm]:')
lipid_center
water_center
disp('-----')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OXYGEN ADDITION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Conversion of Oxygen Coordinates from Angstroms to nm
P1.x=P1.x/10;
P1.y=P1.y/10;
P1.z=P1.z/10;
P2.x=P2.x/10;
P2.y=P2.y/10;
P2.z=P2.z/10;

Destin.x=lipid_center.x; % Destination - x Component [nm]
Destin.y=lipid_center.y; % Destination - y Component [nm]
Destin.z=(lipid_center.z)+dist_o2; % Destination - z Component [nm]

Translation.x=Destin.x-P1.x; % Translation Vector - x Component [nm]
Translation.y=Destin.y-P1.y; % Translation Vector - y Component [nm]
Translation.z=Destin.z-P1.z; % Translation Vector - z Component [nm]

% Translation of Oxygen Atoms
O1.x=(P1.x+Translation.x);
O1.y=(P1.y+Translation.y);
O1.z=(P1.z+Translation.z);

O2.x=(P2.x+Translation.x);
O2.y=(P2.y+Translation.y);
O2.z=(P2.z+Translation.z);

disp('OXYGEN ATOMS COORDINATES [nm]:')
O1
O2
```

```
disp('-----')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

---

## A.4 Geometrical Parameters for Symmetry - *Geometrical\_Parameters\_SYM.m*

---

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% DOMAIN GEOMETRICAL CHARACTERISTICS FOR A LIPID MONOLAYER
% AND OXYGEN ADDITION PARAMETERS
% SYMMETRY TOPOLOGY
%
% PROGRAMMER: MICHAEL PALAIOKOSTAS AVRAMIDS
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear
close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% USER INPUT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('-----')
disp('-SYMMETRICAL TOPOLOGY-')
disp('-----')

APL=0.60;      % Area per Lipid required [nm^2]
N_L=36;       % Number of Lipids per Lipid Leaflet

InitialLipid.x=6.18010; % Lipid Length [nm] - Based on pdb2gmX results
InitialLipid.y=6.29250; % Lipid Width [nm] - Based on pdb2gmX results
InitialLipid.z=17.15950; % Lipid Height [nm] - Based on pdb2gmX results

domain_z=150; % Domain of Solution Height [nm]
water_slab=10; % Water Slab Height [nm] - !Should be larger than cut-off!

dist_o2=70.0; % Distance between O2 molec. and top lipid center [nm]

abs_dist_o2=83; % To be used for Pulling Simulations

% Oxygen Atom Coordinates based on Avogadro output in Angstroms
P1.x= -1.3792;      % Atom 1 - X coordinate [A]
P1.y= +0.303;      % Atom 1 - Y coordinate [A]
```

```

P1.z= -1.477; % Atom 1 - Z coordinate [A]

P2.x= -1.912; % Atom 1 - X coordinate [A]
P2.y= +0.346; % Atom 1 - Y coordinate [A]
P2.z= -0.316; % Atom 1 - Z coordinate [A]

disp('AREA PER LIPID [nm^2]:')
disp(APL)
disp('-----')

% Lipid Geometry Min and Max (Output of VMD measure)
max_z=30.1; % Top leaflet maximum coordinate [A]
min_z=-27.24; % Bottom leaflet minimum coordinate [A]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TOPOLOGY GENERATION FOR VMD SCRIPT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Bilayer_height=max_z-min_z; % Bilayer height [A]
Offset=-(Bilayer_height+(water_slab*10)); % Offset to define in VMD [A]

disp('TOP LIPID OFFSET FOR VMD [nm]:')
disp(Offset)
disp('-----')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BOX DIMENSIONS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x=sqrt(APL*N_L); % x-y Length based on A.P.L

box.x=x; % Box x Vector [nm]
box.y=x; % Box y Vector [nm]
box.z=domain_z; % Box z Vector [nm]

disp('BOX DIMENSIONS [nm]:')
box
disp('-----')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LIPID SCALING
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
ScaleFactor.x= box.x/InitialLipid.x;
ScaleFactor.y= box.y/InitialLipid.y;
ScaleFactor.z= 1;

disp('SCALE FACTORS [-]:')
ScaleFactor
disp('-----')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% GEOMETRICAL CENTERS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

lipid_center.x=box.x/2; % Center of Lipid leaflet x [nm]
lipid_center.y=box.y/2; % Center of Lipid leaflet y [nm]
lipid_center.z=box.z/2; % Center of Lipid leaflet z [nm]

water_center.x=box.x/2; % Center of Water Slab x [nm]
water_center.y=box.y/2; % Center of Water Slab y [nm]
water_center.z=box.z/2; % Center of Water Slab z [nm]

disp('CENTERS [nm]:')
lipid_center
water_center
disp('-----')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OXYGEN ADDITION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Conversion of Oxygen Coordinates from Angstroms to nm
P1.x=P1.x/10;
P1.y=P1.y/10;
P1.z=P1.z/10;
P2.x=P2.x/10;
P2.y=P2.y/10;
P2.z=P2.z/10;

Destin.x=lipid_center.x; % Destination - x Component [nm]
Destin.y=lipid_center.y; % Destination - y Component [nm]
Destin.z=(lipid_center.z)+dist_o2; % Destination - z Component [nm]
```

---

```

Translation.x=Destin.x-P1.x; % Translation Vector-x Component [nm]
Translation.y=Destin.y-P1.y; % Translation Vector-y Component [nm]
Translation.z=Destin.z-P1.z; % Translation Vector-z Component [nm]

% Translation of Oxygen Atoms
O1.x=(P1.x+Translation.x);
O1.y=(P1.y+Translation.y);
O1.z=(P1.z+Translation.z);

O2.x=(P2.x+Translation.x);
O2.y=(P2.y+Translation.y);
O2.z=(P2.z+Translation.z);

disp('OXYGEN ATOMS COORDINATES [nm]:')
O1
O2
disp('-----')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OXYGEN TRANSLATION FOR PULLING SIMULATIONS
% IGNORE FOR PHASE 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

PH2_Destin.x=lipid_center.x; % Destination - x Component [nm]
PH2_Destin.y=lipid_center.y; % Destination - y Component [nm]
PH2_Destin.z=abs_dist_o2; % Destination - z Component [nm]

PH2_Translation.x=PH2_Destin.x-P1.x; % Translation Vector-x Component [nm]
PH2_Translation.y=PH2_Destin.y-P1.y; % Translation Vector-y Component [nm]
PH2_Translation.z=PH2_Destin.z-P1.z; % Translation Vector-z Component [nm]

% Translation of Oxygen Atoms
PH2_O1.x=(P1.x+PH2_Translation.x);
PH2_O1.y=(P1.y+PH2_Translation.y);
PH2_O1.z=(P1.z+PH2_Translation.z);

PH2_O2.x=(P2.x+PH2_Translation.x);
PH2_O2.y=(P2.y+PH2_Translation.y);
PH2_O2.z=(P2.z+PH2_Translation.z);

```

---



```
disp('PHASE_2 - INITIAL OXYGEN ATOMS COORDINATES [nm]:')
PH2_01
PH2_02
disp('-----')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

---

## A.5 Calculation of Umbrella Sampling Distances

### - *distances.pl*

---

```
#!/usr/bin/perl -w
use strict;
# loop g_dist command - measure distance in each frame, write to a file
for (my $i=0; $i<=1000; $i++) {
    print "Processing configuration $i...\n";
    system("g_dist_mpi -s md_pull_final.tpr -f frame${i}.gro -o
           dist${i}.xvg < groups.txt &>/dev/null");
}
# write output to single file
open(OUT, ">>summary_distances.dat");
for (my $j=0; $j<=1000; $j++) {
    open(IN, "<dist${j}.xvg");
    my @array = <IN>;
    my $distance;
    foreach $_ (@array) {
        if ($_ =~ /[#@]/) {
            # do nothing, it's a comment or formatting line
        } else {
            my @line = split(" ", $_);
            $distance = $line[1];
        }
    }
    close(IN);
    print OUT "$j\t$distance\n";
}
close(OUT);
# clean up
print "Cleaning up...\n";
for (my $k=0; $k<=1000; $k++) {
    unlink "dist${k}.xvg";
}
exit;
```

---

## A.6 Translation of relative distances - *Translate Distances.m*

---

```
% Translation of relative Distances
%
clc
clear
close all
startpath=cd;
data=importdata('summary_distances.dat');
data(:,3)=data(:,2)+75;
```

---



## GROMACS Parameters Files

### Contents

---

A.1	Creation of Monolayer Topology . . . . .	130
A.2	Creation Symmetry Topology . . . . .	131
A.3	Geometrical Parameters for Monolayer . . . . .	132
A.4	Geometrical Parameters for Symmetry . . . . .	136
A.5	Calculation of Umbrella Sampling Distances . . . . .	141
A.6	Translation of relative distances . . . . .	142

---

## B.1 Energy Minimization

---

```
#####  
; ENERGY MINIMIZATION PARAMETERS FILE  
; FOR USE WITH GROMACS 4.6.1  
;  
; PROGRAMMER: MICHAEL PALAIOKOSTAS AVRAMIDIS  
;  
#####  
;  
#####  
; RUN CONTROL PARAMETERS  
#####  
integrator = steep ; Steepest descent algorithm for energy minimization  
nsteps = 50000 ; Maximum number of steps to integrate [ps]  
comm-mode = Linear ; Remove center of mass translation  
nstcomm = 1 ; Frequency for center of mass motion removal [steps]  
comm-grps = DPPC SOL O2 ; Groups for center of mass removal  
#####  
; OUTPUT CONTROL PARAMETERS  
#####  
nstcalcenergy = 1 ; Frequency for calculating the energies  
#####  
; ENERGY MINIMIZATION  
#####  
emtol = 100.0 ; Convergence Criteria [kJ/mol/nm]  
emstep = 0.01 ; Initial step size [nm]  
#####  
; NEIGHBOUR SEARCHING (pbc = xy for monolayer - xyz for symmetry)  
#####  
cutoff-scheme = Verlet ; Generate a pair list with buffering  
nstlist = 20 ; Frequency to update neighbour list [steps]  
ns_type = grid ; Make a grid and only check atoms in adjacent cells  
;pbc = xyz ; Periodic Boundary Conditions in all directions  
pbc = xy ; Periodic Boundary Conditions in all directions  
verlet-buffer-drift = 0.005 ; Corrects energy drift [kJ/mol/ps]  
#####  
; ELECTROSTATICS  
#####  
coulombtype = PME ; Fast Smooth Particle-Mesh Ewald Electrostatics
```

```
coulomb-modifier = Potential-shift-Verlet ; Shifts Coulomb potential
rcoulomb = 1.0 ; Coulomb cut-off [nm]
;#####
; VAN DER WAALS (DispCorr only for Symmetry)
;#####
vdwtype = Cut-off ; LJ is normal out to rvd-switch and 0 to rvdw
rvdw = 1.0 ; Distance for the LJ or Buckingham cut-off [nm]
vdw-modifier = Potential-shift-Verlet ; Shifts VdW potential
;DispCorr = EnerPres ; Apply long-range dispersion correction
;#####
; EWALD
;#####
fourierspacing = 0.15 ; Lower bound for number of grid nodes [nm]
pme_order = 4 ; Interpolation order of PME
optimize-fft = no ; Calculate optimal FFT plan. Good in long simulations
;#####
; WALLS (ONLY FOR MONOLAYER)
;#####
nwall = 2 ; Wall at z=0 and z=z_max
wall-atomtype = Ar Ar ; Atom type of wall. Defines force field
wall-type = 9-3 ; LJ 9-3 Potential integrated over the wall surface
wall-density = 20 20 ; Number of atoms used as density of wall
wall-ewald-zfac = 3 ; Scaling factor of 3rd dimension when pbc=xy
ewald-geometry = 3dc ; Ewald summation correction for extended z
;#####
```

---

## B.2 NVT Equilibration

---

```
#####  
; NVT EQUILIBRATION PARAMETERS FILE  
; FOR USE WITH GROMACS 4.6.1  
;  
; PROGRAMMER: MICHAIL PALAIOKOSTAS AVRAMIDIS  
;  
#####  
;  
#####  
; RUN CONTROL PARAMETERS  
#####  
integrator = md ; Leap-frog integrator of Newton's Equations of motion  
tinit = 0 ; Run starting time [ps]  
dt = 0.001 ; Integration Time step [ps]  
nsteps = 1000000 ; Maximum number of steps to integrate [ps]  
#####  
; OUTPUT CONTROL PARAMETERS  
#####  
nstxout = 500 ; Frequency to write coordinates to trajectory file [steps]  
nstvout = 500 ; Frequency to write velocities to trajectory file [steps]  
nstlog = 500 ; Frequency to write energies to log [steps]  
nstcalcenergy = 1 ; Frequency for calculating the energies  
nstenergy = 500 ; Frequency to write energies to energy file [steps]  
comm-mode = Linear ; Remove center of mass translation  
nstcomm = 1 ; Frequency for center of mass motion removal [steps]  
comm-grps = DPPC O2 SOL ; Groups for center of mass removal  
#####  
; NEIGHBOUR SEARCHING (pbc = xy for monolayer - xyz for symmetry)  
#####  
cutoff-scheme = Verlet ; Generate a pair list with buffering  
nstlist = 20 ; Frequency to update neighbour list [steps]  
ns_type = grid ; Make a grid and only check atoms in adjacent cells  
pbc = xy ; Periodic Boundary Conditions in all directions  
; pbc = xyz ; Periodic Boundary Conditions in all directions  
verlet-buffer-drift = 0.005 ; Corrects energy drift [kJ/mol/ps]  
#####  
; ELECTROSTATICS  
#####
```

```
coulombtype = PME ; Fast Smooth Particle-Mesh Ewald Electrostatics
coulomb-modifier = Potential-shift-Verlet ; Shifts Coulomb potential
rcoulomb = 1.0 ; Coulomb cut-off [nm]
;#####
; VAN DER WAALS (DispCorr only for Symmetry)
;#####
vdwtype = Cut-off ; LJ is normal out to rvd-switch and 0 to rvdw
rvdw = 1.0 ; Distance for the LJ or Buckingham cut-off [nm]
vdw-modifier = Potential-shift-Verlet ; Shifts VdW potential
;DispCorr = EnerPres ; Apply long-range dispersion correction
;#####
; EWALD
;#####
fourierspacing = 0.15 ; Lower bound for number of grid nodes [nm]
pme_order = 4 ; Interpolation order of PME
optimize-fft = no ; Calculate optimal FFT plan
;#####
; TEMPERATURE COUPLING (Two Thermostat choices)
;#####
tcoupl = nose-hoover ; Nose-Hoover Thermostat
;tcoupl = v-rescale ; Berendsen with velocity rescaling
tc-grps = DPPC O2 SOL ; Groups to couple separately to energy bath
tau_t = 0.5 0.5 0.5 ; Time constant for coupling of each group [ps]
ref_t = 323.15 323.15 323.15 ; Reference coupling temperature [K]
;#####
; PRESSURE COUPLING
;#####
pcoupl = no ; No pressure coupling in NVT
;#####
; VELOCITY GENERATION
;#####
gen_vel = yes ; Assign velocities from Maxwell distribution
gen_temp = 323.15 ; Temperature for Maxwell distribution [K]
gen_seed = -1 ; Generate random seed from Process ID
;#####
; BONDS PARAMETERS
;#####
constraints = all-bonds ; Convert all bonds to constraints
constraint_algorithm = LINCS ; Linear Constraint Solver
continuation = no ; Constrain initial configuration and reset shells
```



```
lincs_order = 4 ; Order in constraint coupling matrix expansion
lincs_iter = 1 ; Iterations number for LINCS correction
;#####
; WALLS (ONLY FOR MONOLAYER)
;#####
nwall = 2 ; Wall at z=0 and z=z_max
wall-atomtype = Ar Ar ; Atom type of wall. Defines force field
wall-type = 9-3 ; LJ 9-3 Potential integrated over the wall surface
wall-density = 20 20 ; Number of atoms used as density of wall
wall-ewald-zfac = 3 ; Scaling factor of 3rd dimension when pbc=xy
ewald-geometry = 3dc ; Ewald summation correction for extended z
;#####
```

---

## B.3 Steered MD

---

```

#####
; MD UMBRELLA PULLING PARAMETERS FILE
; FOR USE WITH GROMACS 4.6.1
;
; PROGRAMMER: MICHAIL PALAIOKOSTAS AVRAMIDIS
; LAST MODIFICATION DATE: 8/7/2013
#####
#####
; RUN CONTROL PARAMETERS
#####
integrator = md ; Leap-frog integrator of Newton's Equations of motion
tinit = 0 ; Run starting time [ps]
dt = 0.001 ; Integration Time step [ps] - 0.001 = 1 fs
nsteps = 500000 ; Maximum number of steps to integrate [ps] - 100000 =
    100 ps
#####
; OUTPUT CONTROL PARAMETERS
#####
nsthout = 500 ; Frequency to write coordinates to output trajectory file
    [steps]
nsthout = 500 ; Frequency to write velocities to output trajectory file
    [steps]
nsthout = 500 ; Frequency to write forces to output trajectory
nsthout = 500 ; Frequency to write energies to log [steps]
nsthout = 1 ; Frequency for calculating the energies
nsthout = 500 ; Frequency to write energies to energy file [steps]
nsthout = 500 ; Frequency to write coordinates to xtc trajectory
    [steps]
comm-mode = Linear ; Remove center of mass translation
nsthout = 1 ; Frequency for center of mass motion removal [steps]
comm-grps = DPPC O2 SOL ; Groups for center of mass removal
#####
; NEIGHBOUR SEARCHING
#####
cutoff-scheme = Verlet ; Generate a pair list with buffering
nsthout = 20 ; Frequency to update neighbour list [steps]
ns_type = grid ; Make a grid and only check atoms in adjacent cells
pbc = xyz ; Periodic Boundary Conditions in all directions

```

```
verlet-buffer-drift = 0.005 ; For use with Verlet-cutoff.Corrects
  energy drift [kJ/mol/ps]
;#####
; ELECTROSTATICS
;#####
coulombtype = PME ; Fast Smooth Particle-Mesh Ewald Electrostatics
coulomb-modifier = Potential-shift-Verlet ; Shifts Coulomb potential to
  0 at the cut-off
rcoulomb = 1.0 ; Coulomb cut-off [nm]
;#####
; VAN DER WAALS
;#####
vdwtype = Cut-off ; LJ is normal out to rvd-switch and 0 to rvdw
rvdw = 1.0 ; Distance for the LJ or Buckingham cut-off [nm]
vdw-modifier = Potential-shift-Verlet ; Shifts VdW potential to 0 at
  cut-off
DispCorr = EnerPres ; Apply long-range dispersion correction for Energy
  and Pres.
;#####
; EWALD
;#####
fourierspacing = 0.15 ; Lower bound for number of grid nodes [nm]
pme_order = 4 ; Interpolation order of PME
optimize-fft = no ; Calculate optimal FFT plan. Good in long simulations
;#####
; TEMPERATURE COUPLING
;#####
tcoupl = nose-hoover ; Nose-Hoover Thermostat
tc-grps = DPPC O2 SOL ; Groups to couple seperately to energy bath
tau_t = 0.5 0.5 0.5 ; Time constant for coupling of each group [ps]
ref_t = 323.15 323.15 323.15 ; Reference coupling temperature of each
  group [K]
;#####
; PRESSURE COUPLING
;#####
pcoupl = no ; No pressure coupling in NVT
;#####
; VELOCITY GENERATION
;#####
gen_vel = no ; Assign velocities from Maxwell distribution
```

```
;gen_temp = 323.15 ; Temperature for Maxwell distribution [K]
;gen_seed = -1 ; Generate random seed from Process ID
;#####
; BONDS PARAMETERS
;#####
constraints = all-bonds ; Convert all bonds to constraints
constraint_algorithm = LINCS ; Linear Constraint Solver
continuation = yes ; Constrain initial configuration and reset shells
lincs_order = 4 ; Order in constraint coupling matrix expansion
lincs_iter = 1 ; Iterations number for LINCS correction
;#####
; COM PULLING
;#####
pull = umbrella
pull_geometry = position ; simple distance increase
pull_start = yes ; define initial COM distance > 0
pull_dim = N N Y
pull_init1= 0.0 0.0 0.0
pull_vec1 = 0.0 0.0 1.0
pull_ngroups = 1
pull_group0 = DPPC
pull_group1 = O2
pull_rate1 = -0.01 ; 0.01 nm per ps = 10 nm per ns
pull_k1 = 1000 ; kJ mol-1 nm-2
```

---

## B.4 Umbrella Simulation

---

```
#####  
; MD UMBRELLA PULLING PARAMETERS FILE  
; FOR USE WITH GROMACS 4.6.1  
;  
; PROGRAMMER: MICHAIL PALAIOKOSTAS AVRAMIDIS  
; LAST MODIFICATION DATE: 8/7/2013  
#####  
#####  
; RUN CONTROL PARAMETERS  
#####  
integrator = md ; Leap-frog integrator of Newton's Equations of motion  
tinit = 0 ; Run starting time [ps]  
dt = 0.001 ; Integration Time step [ps] - 0.001 = 1 fs  
nsteps = 1000000 ; Maximum number of steps to integrate [ps] - 100000 =  
100 ps  
#####  
; OUTPUT CONTROL PARAMETERS  
#####  
nsthout = 1000 ; Frequency to write coordinates to output trajectory  
file [steps]  
nsthout = 1000 ; Frequency to write velocities to output trajectory file  
[steps]  
nsthout = 1000 ; Frequency to write forces to output trajectory  
nsthout = 1000 ; Frequency to write energies to log [steps]  
nsthout = 1 ; Frequency for calculating the energies  
nsthout = 1000 ; Frequency to write energies to energy file [steps]  
nsthout = 1000 ; Frequency to write coordinates to xtc trajectory  
[steps]  
comm-mode = Linear ; Remove center of mass translation  
nsthout = 1 ; Frequency for center of mass motion removal [steps]  
comm-grps = DPPC O2 SOL ; Groups for center of mass removal  
#####  
; NEIGHBOUR SEARCHING  
#####  
cutoff-scheme = Verlet ; Generate a pair list with buffering  
nsthout = 20 ; Frequency to update neighbour list [steps]  
ns_type = grid ; Make a grid and only check atoms in adjacent cells  
pbc = xyz ; Periodic Boundary Conditions in all directions
```

---

```

verlet-buffer-drift = 0.005 ; For use with Verlet-cutoff.Corrects
  energy drift [kJ/mol/ps]
;#####
; ELECTROSTATICS
;#####
coulombtype = PME ; Fast Smooth Particle-Mesh Ewald Electrostatics
coulomb-modifier = Potential-shift-Verlet ; Shifts Coulomb potential to
  0 at the cut-off
rcoulomb = 1.0 ; Coulomb cut-off [nm]
;#####
; VAN DER WAALS
;#####
vdwtype = Cut-off ; LJ is normal out to rvd-switch and 0 to rvdw
rvdw = 1.0 ; Distance for the LJ or Buckingham cut-off [nm]
vdw-modifier = Potential-shift-Verlet ; Shifts VdW potential to 0 at
  cut-off
DispCorr = EnerPres ; Apply long-range dispersion correction for Energy
  and Pres.
;#####
; EWALD
;#####
fourierspacing = 0.15 ; Lower bound for number of grid nodes [nm]
pme_order = 4 ; Interpolation order of PME
optimize-fft = no ; Calculate optimal FFT plan. Good in long simulations
;#####
; TEMPERATURE COUPLING
;#####
tcoupl = nose-hoover ; Nose-Hoover Thermostat
tc-grps = DPPC O2 SOL ; Groups to couple seperately to energy bath
tau_t = 0.5 0.5 0.5 ; Time constant for coupling of each group [ps]
ref_t = 323.15 323.15 323.15 ; Reference coupling temperature of each
  group [K]
;#####
; PRESSURE COUPLING
;#####
pcoupl = no ; No pressure coupling in NVT
;#####
; VELOCITY GENERATION
;#####
gen_vel = no ; Assign velocities from Maxwell distribution

```

```
;gen_temp = 323.15 ; Temperature for Maxwell distribution [K]
;gen_seed = -1 ; Generate random seed from Process ID
;#####
; BONDS PARAMETERS
;#####
constraints = all-bonds ; Convert all bonds to constraints
constraint_algorithm = LINCS ; Linear Constraint Solver
continuation = yes ; Constrain initial configuration and reset shells
lincs_order = 4 ; Order in constraint coupling matrix expansion
lincs_iter = 1 ; Iterations number for LINCS correction
;#####
; COM PULLING
;#####
pull = umbrella
pull_geometry = position ; simple distance increase
pull_start = yes ; define initial COM distance > 0
pull_dim = N N Y
pull_init1= 0.0 0.0 0.0
pull_vec1 = 0.0 0.0 1.0
pull_ngroups = 1
pull_group0 = DPPC
pull_group1 = O2
pull_rate1 = 0 ; 0.01 nm per ps = 10 nm per ns
pull_k1 = 1000 ; kJ mol-1 nm-2
pull_nstxout = 1000 ;
pull_nstfout = 1000 ;
```

---

## B.5 Required Files

The following code is the collection of all the bash commands that will create all the necessary index and electrons.dat files that were used in this Thesis. This work was based on the previous Thesis of Silvia Espinosa [154].

---

```
# Creation of necessary index files
# as well as electrons.dat
# To be able to use the script you have to change permissions:
# In terminal write: chmod 700 Groups.sh
# where 700 give permission only to you (read,write,execute)
#####
# Create electrons.dat
touch electrons.dat
cat > electrons.dat << EOF
133
N    = 7
C13  = 6
H13A = 1
H13B = 1
H13C = 1
C14  = 6
H14A = 1
H14B = 1
H14C = 1
C15  = 6
H15A = 1
H15B = 1
H15C = 1
C12  = 6
H12A = 1
H12B = 1
C11  = 6
H11A = 1
H11B = 1
P    = 15
O13  = 8
O14  = 8
O11  = 8
O12  = 8
```



C1 = 6  
HA = 1  
HB = 1  
C2 = 6  
HS = 1  
O21 = 8  
C21 = 6  
O22 = 8  
C22 = 6  
H2R = 1  
H2S = 1  
C3 = 6  
HX = 1  
HY = 1  
O31 = 8  
C31 = 6  
O32 = 8  
C32 = 6  
H2X = 1  
H2Y = 1  
C23 = 6  
H3R = 1  
H3S = 1  
C24 = 6  
H4R = 1  
H4S = 1  
C25 = 6  
H5R = 1  
H5S = 1  
C26 = 6  
H6R = 1  
H6S = 1  
C27 = 6  
H7R = 1  
H7S = 1  
C28 = 6  
H8R = 1  
H8S = 1  
C29 = 6  
H9R = 1

H9S = 1  
C210 = 6  
H10R = 1  
H10S = 1  
C211 = 6  
H11R = 1  
H11S = 1  
C212 = 6  
H12R = 1  
H12S = 1  
C213 = 6  
H13R = 1  
H13S = 1  
C214 = 6  
H14R = 1  
H14S = 1  
C215 = 6  
H15R = 1  
H15S = 1  
C216 = 6  
H16R = 1  
H16S = 1  
H16T = 1  
C33 = 6  
H3X = 1  
H3Y = 1  
C34 = 6  
H4X = 1  
H4Y = 1  
C35 = 6  
H5X = 1  
H5Y = 1  
C36 = 6  
H6X = 1  
H6Y = 1  
C37 = 6  
H7X = 1  
H7Y = 1  
C38 = 6  
H8X = 1

```
H8Y = 1
C39 = 6
H9X = 1
H9Y = 1
C310 = 6
H10X = 1
H10Y = 1
C311 = 6
H11X = 1
H11Y = 1
C312 = 6
H12X = 1
H12Y = 1
C313 = 6
H13X = 1
H13Y = 1
C314 = 6
H14X = 1
H14Y = 1
C315 = 6
H15X = 1
H15Y = 1
C316 = 6
H16X = 1
H16Y = 1
H16Z = 1
OW = 8
HW1 = 1
HW2 = 1
O2 = 8
EOF
#####
# Create ndx files
# DPPC
make_ndx_mpi -f nvt.tpr -o DPPC.ndx << EOF
2
del 0-6
q
EOF
# Water
```

```
make_ndx_mpi -f nvt.tpr -o H2O.ndx << EOF
4
del 0-6
q
EOF
# Oxygen
make_ndx_mpi -f nvt.tpr -o O2.ndx << EOF
3
del 0-6
q
EOF
# Chain 1 -sn1
make_ndx_mpi -f nvt.tpr -o sn1.ndx << EOF
a C31
a C32
a C33
a C34
a C35
a C36
a C37
a C38
a C39
a C310
a C311
a C312
a C313
a C314
a C315
a C316
del 0-6
q
EOF
# Chain 2 -sn2
make_ndx_mpi -f nvt.tpr -o sn2.ndx << EOF
a C21
a C22
a C23
a C24
a C25
a C26
```

```
a C27
a C28
a C29
a C210
a C211
a C212
a C213
a C214
a C215
a C216
del 0-6
q
EOF
# Phosphorus P
make_ndx_mpi -f nvt.tpr -o P.ndx << EOF
a P
del 0-6
q
EOF
#PC
make_ndx_mpi -f nvt.tpr -o PC.ndx << EOF
a N | a C12 | a H12A | a H12B | a C13 | a H13A | a H13B | a H13C | a C14
  | a H14A | a H14B | a H14C | a C15 | a H15A | a H15B | a H15C | a C11
  | a H11A | a H11B | a P | a O11 | a O12 | a O13 | a O14
del 0-6
name 0 PC
q
EOF
# CG
make_ndx_mpi -f nvt.tpr -o CG.ndx << EOF
a C1 | a HA | a HB | a C2 | a HS | a O21 | a C21 | a O22 | a C3 | a HX
  | a HY | a O31 | a C31 | a O32
del 0-6
name 0 CG
q
EOF
# CH2
make_ndx_mpi -f nvt.tpr -o CH2.ndx << EOF
a C22 | a H2R | a H2S | a C23 | a H3R | a H3S | a C24 | a H4R | a H4S |
  a C25 | a H5R | a H5S | a C26 | a H6R | a H6S | a C27 | a H7R | a H7S
```

```
| a C28 | a H8R | a H8S | a C29 | a H9R | a H9S | a C210 | a H10R | a
H10S | a C211 | a H11R | a H11S | a C212 | a H12R | a H12S | a C213 |
a H13R | a H13S | a C214 | a H14R | a H14S | a C215 | a H15R | a H15S
| a C32 | a H2X | a H2Y | a C33 | a H3X | a H3Y | a C34 | a H4X | a
H4Y | a C35 | a H5X | a H5Y | a C36 | a H6X | a H6Y | a C37 | a H7X |
a H7Y | a C38 | a H8X | a H8Y | a C39 | a H9X | a H9Y | a C310 | a
H10X | a H10Y | a C311 | a H11X | a H11Y | a C312 | a H12X | a H12Y |
a C313 | a H13X | a H13Y | a C314 | a H14X | a H14Y | a C315 | a H15X
| a H15Y
del 0-6
name 0 CH2
q
EOF
# CH3
make_ndx_mpi -f nvt.tpr -o CH3.ndx << EOF
a C216 | a H16R | a H16S | a H16T | a C316 | a H16X | a H16Y | a H16Z
del 0-6
name CH3
q
EOF
#####
mkdir Groups
mv electrons.dat Groups
mv *.ndx Analysis/Groups
echo "#####"
echo "Groups have been created successfully"
echo "#####"
```

---





## Post-Processing Codes

### Contents

---

<b>B.1</b>	<b>Energy Minimization . . . . .</b>	<b>144</b>
<b>B.2</b>	<b>NVT Equilibration . . . . .</b>	<b>146</b>
<b>B.3</b>	<b>Steered MD . . . . .</b>	<b>149</b>
<b>B.4</b>	<b>Umbrella Simulation . . . . .</b>	<b>152</b>
<b>B.5</b>	<b>Required Files . . . . .</b>	<b>155</b>

---



## C.1 Energy Minimization Error Analysis

---

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENERGY MINIMIZATION ERROR ANALYSIS CODE
%
% PROGRAMMER: MICHAEL PALAIOKOSTAS AVRAMIDIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc
clear
close all
start_path = cd;
cd '1.Simulations'
Navogadro= 6.022 * 10^23; %[atoms/mol]
MW_lipids= 0.734039; %[kg/mol]
MW_water= 0.01801528; %[kg/mol]
N_lipids= [4680;9360;4680;9360;4680;9360;];
N_water= [10788;21411;9045;17964;12465;24756];
d=dir;
idir=[d(:).isdir];
name_ref_Files = {d(~idir).name}';
for case_num=1:numel(name_ref_Files)
    case_data=0;
    system_mass=0;
    sp_energy=0;
    nd_max_steps=0;
    nd_max_energy=0;

    filename=char(name_ref_Files(case_num));
    case_data=importdata(filename);
    case_name=filename(3:8);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CALCULATE SPECIFIC ENERGY
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    system_mass=((N_lipids(case_num)*MW_lipids)+...
        (N_water(case_num)*MW_water))./Navogadro;

    sp_energy=case_data(:,2)./system_mass;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% NON-DIMENSIONALISATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

---

```

nd_max_steps=max(case_data(:,1));
nd_max_energy=max(sp_energy);
nd_data.steps=case_data(:,1)./nd_max_steps;
nd_data.energy=(sp_energy-sp_energy(numel(sp_energy),1))./(sp_energy(1,1)
    -sp_energy(numel(sp_energy),1));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PLOTTING DATA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nd_data.steps=downsample(nd_data.steps,50);
nd_data.energy=downsample(nd_data.energy,50);
sp_energy=downsample(sp_energy,50);
case_data=downsample(case_data,50);
colormap= [[220/255 20/255 60/255];[1 0 1];...
           [0 201/255 87/255];[205/255 173/255 0/255];...
           [0 154/255 205/255];[107/255 107/255 107/255];[0.75 0.25 0];
           [0 1 1];[0.25 0.75 0.75] ];
position=[200 55 1440 900];
legend_strings(case_num)={case_name};
f1=figure(1);
hold on
set(gca,'FontName','CMU Serif')
xlabel('Time [ps]')
ylabel('Specific Potential Energy [kJ/kg]')
ylim([-4e25 0.5e25])
xlim([-100 7000])
grid on
box on
h1=plot(case_data(:,1),sp_energy,...
        'Color',[colormap(case_num,1) colormap(case_num,2)
                 colormap(case_num,3)],...
        'LineWidth',2);
legend(legend_strings,'Location','NorthEast');
hold off
f2=figure(2);
hold on
set(gca,'FontName','CMU Serif')
xlabel('Time [ps]')
ylabel('Specific Potential Energy [-]')
ylim([ 0 0.1])
xlim([-100 4000])

```

```
grid on
box on
h2=plot(case_data(:,1),nd_data.energy,...
        'Color',[colormap(case_num,1) colormap(case_num,2)
                 colormap(case_num,3)],...
        'LineWidth',2,'LineStyle','-');
legend(legend_strings,'Location','NorthEast');
hold off
end
cd(start_path)
print(f1,'-dpng','-r0','EM_sp_pe.png');
plot2svg('EM_sp_pe.svg',f1,eps);
print(f2,'-dpng','-r0','EM_sp_pe_norm.png');
plot2svg('EM_sp_pe_norm.svg',f2,eps);
disp('-----')
disp('Completed Successfully!')
disp('-----')
```

---

## C.2 Temperature Convergence Error Analysis

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ERROR ANALYSIS OF TEMPERATURE CONVERGENCE
%
% PROGRAMMER: MICHAEL PALAIOKOSTAS AVRAMIDIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear
close all
start_path = cd;
cd '1.Simulations'
ref_Temp=323.15; %[K]
d=dir;
idir=[d(:).isdir];
name_ref_Files = {d(~idir).name}';
for case_num=1:numel(name_ref_Files)
    case_data_or=0;
    case_data=0;
    filename=char(name_ref_Files(case_num));
    case_data_or=importdata(filename);
    case_name=filename(1:9);
    legend_strings(case_num)={case_name};
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% IMPORTING, DOWNSAMPLING AND SMOOTHING OF DATA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Downsample data of file
    case_data=downsample(case_data_or,1);
    nop(case_num)=numel(case_data(:,1));
    % Calculate Original Fluctuations and Fluct. Magnitude for
    fluct_or=case_data_or(:,2)-ref_Temp;
    fluct=abs(case_data(:,2)-ref_Temp);
    % Smooth data of file and fluct
    temp_yy=smooth(case_data(:,1),case_data(:,2),100,'moving');
    [temp_xx,temp_ind] = sort(case_data(:,1));
    fluct_yy=smooth(case_data(:,1),fluct(:,1),100,'moving');
    [fluct_xx,fluct_ind] = sort(case_data(:,1));
    % Add Results to Data , Sm_Data Fl_Data and sm_Fl_Data Matrices
    data(1:nop(case_num),(2*case_num)-1)=case_data(:,1);
    data(1:nop(case_num),2*case_num)=case_data(:,2);

```

```

sm_data(1:nop(case_num), (2*case_num)-1)=temp_xx(:,1);
sm_data(1:nop(case_num), (2*case_num))=temp_yy(:,1);
fl_data(1:nop(case_num), (2*case_num)-1)=case_data(:,1);
fl_data(1:nop(case_num), (2*case_num))=fluct(:);
sm_fl_data(1:nop(case_num), (2*case_num)-1)=fluct_xx(:,1);
sm_fl_data(1:nop(case_num), (2*case_num))=fluct_yy(:,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ERROR ANALYSIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mean_Temp=mean(case_data_or(:,2));
Temp_var = var(fluct_or(numel(fluct_or)-200:numel(fluct_or)));
RMS=sqrt((sum(fluct_or(numel(fluct_or)-200:numel(fluct_or)).^2))
        /numel(fluct_or)); % same as RMS
L_0=norm(fluct_or(numel(fluct_or)-200:numel(fluct_or)), Inf);
abs_err=abs(ref_Temp-mean_Temp);
rel_err=(100*abs_err)/ref_Temp;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RESULTS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
results(case_num+1,1)={case_name};
results(case_num+1,2)={mean_Temp};
results(case_num+1,3)={Temp_var};
results(case_num+1,4)={L_0};
results(case_num+1,5)={RMS};
results(case_num+1,6)={abs_err};
results(case_num+1,7)={rel_err};
results(1,1)={'Case'};
results(1,2)={'$T_{ave} \; [K]$'};
results(1,3)={'Variance [K]'};
results(1,4)={'$L_{\infty}$ [K]'};
results(1,5)={'$RMSD \; [K]$'};
results(1,6)={'$e_{abs} \; [K]$'};
results(1,7)={'$e_{rel}$'};

end
cd(start_path)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SAVING DATA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
delete('Error Analysis-Temp.xlsx');
xlswrite('Error Analysis-Temp.xlsx',results)

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE PROPERTIES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
colormap_light= [[255/255 182/255 193/255];[176/255 226/255 255/255];...
                [180/255 238/255 180/255];[255/255 246/255 143/255];...
                [1 0 1];[1 0.7 0.5];[0.75 0.25 0];[0 1 1];[0.25 0.75 0.75] ];
colormap= [[220/255 20/255 60/255];[0 154/255 205/255];...
           [0 201/255 87/255];[205/255 173/255 0/255];...
           [1 0 1];[1 0.7 0.5];[0.75 0.25 0];[0 1 1];[0.25 0.75 0.75] ];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Temperature Convergence
f1=figure(1);
hold on
for case_num=1:numel(name_ref_Files)
    h1a=plot(data(1:nop(case_num),(2*case_num)-1),data(1:nop(case_num),2*case_num),...
            'Color',[colormap_light(case_num,1) colormap_light(case_num,2)
                    colormap_light(case_num,3)],...
            'LineStyle','o','LineWidth',0.2,'MarkerSize',3,'MarkerFaceColor',...
            [colormap_light(case_num,1) colormap_light(case_num,2)
            colormap_light(case_num,3)]);
    hasbehavior(h1a,'legend',false);
end
for case_num=1:numel(name_ref_Files)
    h1b=plot(sm_data(1:nop(case_num),(2*case_num)-1),
            sm_data(1:nop(case_num),2*case_num),...
            'Color',[colormap(case_num,1) colormap(case_num,2)
                    colormap(case_num,3)],...
            'LineStyle','-','LineWidth',2);
end
h1c=plot([0 1000],[314
          314],'LineWidth',2,'LineStyle','--','Color','blue');
h1d=plot([0 1000],[323.15
          323.15],'LineWidth',2,'LineStyle','--','Color','black');
hasbehavior(h1c,'legend',false);
hasbehavior(h1d,'legend',false);
legend(legend_strings,'Location','NorthWest');
set(gca,'FontName','CMU Serif')
t1=text(110,314.5,'Phase Transition');

```

```

set(t1,'FontName','CMU Serif')
xlabel('Time [ps]')
ylabel('Temperature [K]')
ylim([312 332])
xlim([0 1000])
grid on
box on
hold off
plot2svg('Temp_conv.svg',f1,eps);
% Fluctuations of Temperature
f2=figure(2);
hold on
for case_num=1:numel(name_ref_Files)
    h2a=plot(f1_data(1:nop(case_num),(2*case_num)-1),
            f1_data(1:nop(case_num),2*case_num),...
            'Color',[colormap_light(case_num,1) colormap_light(case_num,2)
                    colormap_light(case_num,3)],...
            'LineStyle','o','LineWidth',0.2,'MarkerSize',3,'MarkerFaceColor',...
            [colormap_light(case_num,1) colormap_light(case_num,2)
             colormap_light(case_num,3)]);
    hasbehavior(h2a,'legend',false);
end
for case_num=1:numel(name_ref_Files)
    h2b=plot(sm_f1_data(1:nop(case_num),(2*case_num)-1),
            sm_f1_data(1:nop(case_num),2*case_num),...
            'Color',[colormap(case_num,1) colormap(case_num,2)
                    colormap(case_num,3)],...
            'LineStyle','-','LineWidth',2);
end
h2c=plot([0 1000],[0 0],'LineWidth',2,'LineStyle','--','Color','black');
hasbehavior(h2c,'legend',false);
legend(legend_strings,'Location','NorthWest');
set(gca,'FontName','CMU Serif')
xlabel('Time [ps]')
ylabel('Magnitude of Temperature Fluctuations [K]')
ylim([-0.1 10])
xlim([0 1000])
grid on
box on
hold off

```

```
plot2svg('Temp_fluct.svg',f2,eps);  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
disp('-----')  
disp('Error Analysis Completed Successfully!')  
disp('Data Saved For All Cases in:"Error Analysis.xlsx"')  
disp('-----')  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

---



## C.3 Deuterium Order Parameters Error Analysis

---

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ERROR ANALYSIS OF DEUTERIUM ORDER PARAMETERS
%
% PROGRAMMER: MICHAEL PALAIOKOSTAS AVRAMIDIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear
close all
start_path = cd;
output='Deuterium-60';
headerlines=13;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% IMPORTING SIMULATION DATA OF CHAIN SN-1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cd '1.Simulations'
cd 'sn1'
d=dir;
idir=[d(:).isdir];
name_ref_Files = {d(~idir).name}';
for cs_nm=1:numel(name_ref_Files)
    case_data_or=0;
    case_data=0;
    filename=char(name_ref_Files(cs_nm));
    imported=importdata(filename, ' ',headerlines);
    case_data_or=imported.data;
    case_name=filename(1:9);
    legend_strings(cs_nm)={case_name};

    nop(cs_nm)=numel(case_data_or(:,1));

    sn1_data(1:nop(cs_nm),(2*cs_nm)-1)=case_data_or(:,1);
    sn1_data(1:nop(cs_nm),2*cs_nm)=case_data_or(:,2);

end
clear case_data case_data_or filename imported nop legend_strings d idir
name_ref_Files
cd(start_path)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% IMPORTING SIMULATION DATA OF CHAIN SN-2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cd '1.Simulations'
cd 'sn2'
d=dir;
idir=[d(:).isdir];
name_ref_Files = {d(~idir).name}';
for cs_nm=1:numel(name_ref_Files)
    case_data_or=0;
    case_data=0;
    filename=char(name_ref_Files(cs_nm));
    imported=importdata(filename, ' ',headerlines);
    case_data_or=imported.data;
    case_name=filename(1:9);
    legend_strings(cs_nm)={case_name};

    nop(cs_nm)=numel(case_data_or(:,1));

    sn2_data(1:nop(cs_nm),(2*cs_nm)-1)=case_data_or(:,1);
    sn2_data(1:nop(cs_nm),2*cs_nm)=case_data_or(:,2);
end
cd(start_path)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% AVERAGING SIMULATION DATA OF CHAINS SN-1 SN-2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:numel(sn1_data(:,1))
    for j=2:2:8
        sn_ave(i,j)=(sn1_data(i,j)+sn2_data(i,j))/2;
    end
    for j=1:2:7
        sn_ave(i,j)=sn1_data(i,j) ;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% IMPORTING REFERENCE DATA OF AVERAGED VALUES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cd 2.References
cd 'ave'
d=dir;
idir=[d(:).isdir];

```

```

ref_name_ref_Files = {d(~idir).name}';
for ref_cs_nm=1:numel(ref_name_ref_Files)
    ref_case_data_or=0;
    ref_case_data=0;
    ref_filename=char(ref_name_ref_Files(ref_cs_nm));
    ref_imported=importdata(ref_filename);
    ref_case_data_or=ref_imported;
    ref_case_name=ref_filename(3:15);
    ref_legend_strings(ref_cs_nm)={ref_case_name};

    ref_nop(ref_cs_nm)=numel(ref_case_data_or(:,1));

    ref_data(1:ref_nop(ref_cs_nm), (2*ref_cs_nm)-1)=ref_case_data_or(:,1);
    ref_data(1:ref_nop(ref_cs_nm), 2*ref_cs_nm)=ref_case_data_or(:,2);
end
cd(start_path)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ERROR ANALYSIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for case_num=1:numel(name_ref_Files)
    filename=char(name_ref_Files(case_num));
    case_name=filename(1:9);
    x=abs(ref_data(:,2)-sn_ave(:,2*case_num));
    SCD_ave=mean(sn_ave(:,2*case_num));
    L_0=norm(x,inf);
    L_1=norm(x,1);
    RMS=rms(x);
    ave_abs_err=mean(x);
    ave_rel_err=mean(x(:)./abs(ref_data(:,2)))*100;
    results(case_num+1,1)={case_name};
    results(case_num+1,2)={SCD_ave};
    results(case_num+1,3)={L_0};
    results(case_num+1,4)={RMS};
    results(case_num+1,5)={ave_abs_err};
    results(case_num+1,6)={ave_rel_err};
    results(1,1)={'Case'};
    results(1,2)={'S_{CD_{ave}}'};
    results(1,3)={'$L_{\infty}$'};
    results(1,4)={'RMS'};
    results(1,5)={'Av. Abs'};

```

```

    results(1,6)={'Av.Rel %'};
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SAVING ERROR ANALYSIS RESULTS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cd(start_path)
cd 3.Results
output_name=strcat('Error_Analysis_',output,'.xlsx');
delete(output_name);
xlswrite(output_name,results)
disp('-----')
disp('Error Analysis Completed Successfully!')
disp('Data Saved For All Cases in:"Error Analysis.xlsx"')
disp('-----')
cd (start_path)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE PROPERTIES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
colormap_light= [[255/255 182/255 193/255];[1 187/255 1];...
                [180/255 238/255 180/255];[255/255 246/255 143/255];...
                [176/255 226/255 255/255];[186/255 186/255 186/255];[0.75 0.25 0];[0
                1 1];[0.25 0.75 0.75] ];

colormap= [[220/255 20/255 60/255];[1 0 1];...
           [0 201/255 87/255];[205/255 173/255 0/255];...
           [0 154/255 205/255];[107/255 107/255 107/255];[0.75 0.25
           0];[0 1 1];[0.25 0.75 0.75] ];
markermap=['^' 'o' 'v' 'x'];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot Sn-ave
f1=figure(1);
hold on
for cs_nm=1:2:3
    h1a=plot(sn_ave(1:nop(cs_nm),(2*cs_nm)-1),sn_ave(1:nop(cs_nm),2*cs_nm),...
            'Color',[colormap(cs_nm,1) colormap(cs_nm,2) colormap(cs_nm,3)],...
            'LineStyle','-','LineWidth',2,'Marker','s','MarkerSize',10);

    h1b=plot(sn_ave(1:nop(cs_nm),(2*cs_nm)-1+2),sn_ave(1:nop(cs_nm),2*cs_nm+2),...

```

```

'Color',[colormap_light(cs_nm,1) colormap_light(cs_nm,2)
colormap_light(cs_nm,3)],...
'LineStyle','--','LineWidth',2,'Marker','d','MarkerSize',10);
end
for ref_cs_nm=1:numel(ref_name_ref_Files)
h1c=plot(ref_data(1:ref_nop(ref_cs_nm),(2*ref_cs_nm)-1),...
ref_data(1:ref_nop(ref_cs_nm),2*ref_cs_nm),...
'Color','k','LineStyle',markermap(ref_cs_nm),'LineWidth',1,...
'MarkerSize',10,'MarkerFaceColor','k');
end
txtstring={' - Nose-Hoover' '-- V-Rescale'};
a1=annotation('textbox',[0.65 0.81 0.25 0.1],'String',txtstring);
set(a1,'BackgroundColor','white',...
'FontName','CMU Serif','FontSize',12,'FontWeight','bold',...
'HorizontalAlignment','left','Margin',5,'EdgeColor','white')

legend_strings_final=cat(2,legend_strings,ref_legend_strings);
[legend_h, object_h, plot_h,
text_strings]=legend(legend_strings_final,'Location','Best');
set(gca,'FontName','CMU Serif')
xlabel('Atoms')
ylabel('Deuterium Order Parameters S_{CD}')
xlim([1 15])
grid on
box on
hold off
cd 3.Results
plot2svg(strcat(output,'.svg'),f1,eps);
cd(start_path)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

---

## C.4 MSD Post-Processing

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% POST-PROCESSING OF PHOSPHORUS MSD
%
% MICHAEL PALAIOKOSTAS AVRAMIDIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear
close all
start_path = cd;
output='MSD-70';
cd '1.Simulations'
headerlines=0;
d=dir;
idir=[d(:).isdir];
name_ref_Files = {d(~idir).name}';
for case_num=1:numel(name_ref_Files)
    case_data_or=0;
    case_data=0;
    filename=char(name_ref_Files(case_num));
    imported=importdata(filename);
    case_data_or=imported;
    case_name=filename(1:9);
    legend_strings(case_num)={case_name};
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% IMPORTING DATA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    nop(case_num)=numel(case_data_or(:,1));
    data(1:nop(case_num),(2*case_num)-1)=case_data_or(:,1);
    data(1:nop(case_num),2*case_num)=case_data_or(:,2);
end
cd(start_path)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE PROPERTIES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
colormap_light= [[255/255 182/255 193/255];[176/255 226/255 255/255];...
                [180/255 238/255 180/255];[255/255 246/255 143/255];...
                [1 0 1];[1 0.7 0.5];[0.75 0.25 0];[0 1 1];[0.25 0.75 0.75] ];

```

```

colormap= [[220/255 20/255 60/255];[0 154/255 205/255];...
          [0 201/255 87/255];[205/255 173/255 0/255];...
          [1 0 1];[1 0.7 0.5];[0.75 0.25 0];[0 1 1];[0.25 0.75 0.75] ];
linestyle_m={'-.', '--', '-.', '--'};
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f1=figure(1);
for case_num=1:numel(name_ref_Files)
    h1b=loglog(data(1:nop(case_num),(2*case_num)-1),data(1:nop(case_num),2*case_num),...
        'Color',[colormap(case_num,1) colormap(case_num,2)
            colormap(case_num,3)],...
        'LineStyle',linestyle_m{case_num},'LineWidth',2);hold on
end
% Single-File Diffusion
%single_file.x=0.5:0.5:1000;
single_file.x=data(1:nop(1),1);
single_file.y=sqrt(single_file.x);
plot(single_file.x,single_file.y,'LineWidth',2,'LineStyle','-');
% Fickian Diffusion
Fickian.x=data(1:nop(1),1);
Fickian.y=Fickian.x;
plot(Fickian.x,Fickian.y,'r','LineWidth',2,'LineStyle','-');
% Ballistic
Ballistic.x=data(1:nop(1),1);
Ballistic.y=Ballistic.x.^2;
plot(Ballistic.x,Ballistic.y,'green','LineWidth',2,'LineStyle','-');
% Other Properties
legend(legend_strings,'Location','NorthWest');
set(gca,'FontName','CMU Serif')
xlabel('Time [ps]')
ylabel('MSD of Phosphorus Atoms [nm^2]')
xlim([0.5 1000])
box on
hold off
plot2svg(strcat(output,'.svg'),f1,eps);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

---

## C.5 PMF and Diffusion Coefficient Calculation

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CALCULATION OF THE DIFFUSION COEFFICIENT
%
% PROGRAMMER: MICHAEL PALAIOKOSTAS AVRAMIDIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear
close all
start_path = cd;
output='PMF';
import=importdata('pmf_0025_kb_1000.xvg');
PMF_in=import.data;
DeltaG=PMF_in(1,2)-PMF_in(numel(PMF_in(:,1)),2)
t=1e-9 ; %[s]
T=323.15 ; %[K]
d=(PMF_in(numel(PMF_in(:,1)),1)-PMF_in(1,1))*1e-9; %[m]
D=((d^2)*(1e+04))/(DeltaG*t)%[cm^2/s]
import_04=importdata('pmf_04_100.xvg');
data_04=import_04.data;
import_02=importdata('pmf_02_100.xvg');
data_02=import_02.data;
import_01=importdata('pmf_01_100.xvg');
data_01=import_01.data;
import_005=importdata('pmf_005_100.xvg');
data_005=import_005.data;
import_0025=importdata('pmf_0025_100.xvg');
data_0025=import_0025.data;
import_hist_04=importdata('hist_04_100.xvg');
hist_04=import_hist_04.data;
import_hist_02=importdata('hist_02_100.xvg');
hist_02=import_hist_02.data;
import_hist_01=importdata('hist_01_100.xvg');
hist_01=import_hist_01.data;
import_hist_005=importdata('hist_005_100.xvg');
hist_005=import_hist_005.data;
import_hist_0025=importdata('hist_0025_100.xvg');
hist_0025=import_hist_0025.data;
import_hist_01_3000=importdata('histo_k3000.xvg');

```



```

hist_01_3000=import_hist_01_3000.data;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE PROPERTIES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
colormap_light= [[255/255 182/255 193/255];[176/255 226/255 255/255];...
                [180/255 238/255 180/255];[255/255 246/255 143/255];...
                [1 0 1];[1 0.7 0.5];[0.75 0.25 0];[0 1 1];[0.25 0.75 0.75] ];
colormap= [[220/255 20/255 60/255];[0 154/255 205/255];...
           [0 201/255 87/255];[205/255 173/255 0/255];...
           [1 0 1];[1 0.7 0.5];[0.75 0.25 0];[0 1 1];[0.25 0.75 0.75] ];
linestyle_m={'-.', '--', '-.', '--'};
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f1=figure(1);
hold on
plot(data_04(:,1)+75,data_04(:,2),...
      'Color',[colormap(5,1) colormap(5,2) colormap(5,3)],...
      'LineStyle',linestyle_m{1},...
      'LineWidth',1);
plot(data_02(:,1)+75,data_02(:,2),...
      'Color',[colormap(4,1) colormap(4,2) colormap(4,3)],...
      'LineStyle',linestyle_m{1},...
      'LineWidth',1);
plot(data_01(:,1)+75,data_01(:,2),...
      'Color',[colormap(3,1) colormap(3,2) colormap(3,3)],...
      'LineStyle',linestyle_m{1},...
      'LineWidth',1);
plot(data_005(:,1)+75,data_005(:,2),...
      'Color',[colormap(2,1) colormap(2,2) colormap(2,3)],...
      'LineStyle',linestyle_m{1},...
      'LineWidth',1);
plot(data_0025(:,1)+75,data_0025(:,2),...
      'Color',[colormap(1,1) colormap(1,2) colormap(1,3)],...
      'LineStyle',linestyle_m{1},...
      'LineWidth',2);
% Other Properties
legend('d=0.4[nm]', 'd=0.2[nm]', 'd=0.1[nm]', 'd=0.05[nm]', ...
      'd=0.025[nm]', 'Location', 'NorthEast');
set(gca, 'FontName', 'CMU Serif')

```

```

xlabel('Height z [nm]')
ylabel('Mean Force Potential [kJ/mol]')
%xlim([0.5 1000])
grid on
box on
hold off
plot2svg(strcat(output, '_d.svg'),f1,eps);

f2=figure(2);
hold on
plot(data_0025(:,1)+75,data_0025(:,2),...
      'Color',[colormap(1,1) colormap(1,2) colormap(1,3)],...
      'LineStyle',linestyle_m{1},...
      'LineWidth',2);

% Other Properties
%legend('d=0.4[nm]', 'd=0.2[nm]', 'd=0.1[nm]', 'd=0.05[nm]',...
%      'd=0.025[nm]', 'Location', 'NorthEast');
set(gca, 'FontName', 'CMU Serif')
xlabel('Height z [nm]')
ylabel('Mean Force Potential [kJ/mol]')
%xlim([0.5 1000])
grid on
box on
hold off
plot2svg(strcat(output, '_final.svg'),f2,eps);

f3=figure(3);
hold on
plot(hist_01(:,1)+75,hist_01(:,2:33),...
      'Color',[colormap(3,1) colormap(3,2) colormap(3,3)],...
      'LineStyle',linestyle_m{2},...
      'LineWidth',2);
plot(hist_02(:,1)+75,hist_02(:,2:17),...
      'Color',[colormap(4,1) colormap(4,2) colormap(4,3)],...
      'LineStyle',linestyle_m{3},...
      'LineWidth',2);
plot(hist_04(:,1)+75,hist_04(:,2:9),...
      'Color',[colormap(5,1) colormap(5,2) colormap(5,3)],...

```

```

    'LineStyle',linestyle_m{1},...
    'LineWidth',2);

% Other Properties
legend('d=0.1[nm]', 'd=0.2[nm]', 'd=0.4[nm]', ...
    'Location', 'North');
set(gca, 'FontName', 'CMU Serif')
xlabel('Height z [nm]')
ylabel('Samples number [-]')
%xlim([0.5 1000])
grid on
box on
hold off
plot2svg(strcat(output, '_hist_1_d.svg'),f3,eps);

f4=figure(4);
hold on
plot(hist_005(:,1)+75,hist_005(:,2:65),...
    'Color',[colormap(2,1) colormap(2,2) colormap(2,3)],...
    'LineStyle',linestyle_m{1},...
    'LineWidth',1);
% Other Properties
legend('d=0.05[nm]', ...
    'Location', 'North');
set(gca, 'FontName', 'CMU Serif')
xlabel('Height z [nm]')
ylabel('Samples number [-]')
%xlim([0.5 1000])
grid on
box on
hold off
plot2svg(strcat(output, '_hist_2_d.svg'),f4,eps);

f5=figure(5);
hold on
plot(hist_0025(:,1)+75,hist_0025(:,2:127),...
    'Color',[colormap(1,1) colormap(1,2) colormap(1,3)],...
    'LineStyle',linestyle_m{1},...
    'LineWidth',0.5);
% Other Properties

```

```

legend('d=0.025[nm]',...
      'Location','North');
set(gca,'FontName','CMU Serif')
xlabel('Height z [nm]')
ylabel('Samples number [-]')
%xlim([0.5 1000])
grid on
box on
hold off
plot2svg(strcat(output,'_hist_3_d.svg'),f5,eps);

f6=figure(6);
hold on
plot(hist_01(:,1)+75,hist_01(:,2:33),...
     'Color',[colormap(3,1) colormap(3,2) colormap(3,3)],...
     'LineStyle',linestyle_m{2},...
     'LineWidth',1);
plot(hist_01_3000(:,1)+75,hist_01_3000(:,2:24),...
     'Color',[colormap(7,1) colormap(7,2) colormap(7,3)],...
     'LineStyle',linestyle_m{1},...
     'LineWidth',1);
% Other Properties
legend('d=0.01[nm] and k=1000','d=0.01[nm] and k=3000',...
      'Location','North');
set(gca,'FontName','CMU Serif')
xlabel('Height z [nm]')
ylabel('Samples number [-]')
%xlim([0.5 1000])
grid on
box on
hold off
plot2svg(strcat(output,'_hist_3000.svg'),f6,eps);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('-----')
disp('Completed Successfully!')
disp('-----')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```