

MOLECULAR DYNAMICS SIMULATIONS OF LUNG  
SURFACTANT PHOSPHOLIPID MONOLAYERS

DOYLE ROSE







# Molecular Dynamics Simulations of Lung Surfactant Phospholipid Monolayers

By

© Doyle Rose

A thesis submitted to the School of Graduate Studies  
in partial fulfillment of the requirements for  
the Degree of Master of Science

Computational Science  
Memorial University of Newfoundland

April 2007

St. John's

Newfoundland



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
*ISBN: 978-0-494-31282-7*  
*Our file* *Notre référence*  
*ISBN: 978-0-494-31282-7*

**NOTICE:**

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

**AVIS:**

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

# Abstract

Lung surfactant is a thin layer of lipids and proteins that lines the alveoli of the lungs. It enables breathing by lowering the surface tension at the air/water interface, thus preventing lung collapse during expiration and reducing the work necessary to re-expand the lung. The lipids are believed to be responsible for the reduction in surface tension, but the molecular mechanisms that bring about this reduction are not well understood.

Molecular dynamics simulations are a means of studying the molecular interactions of the lung surfactant lipids. Molecular dynamics models a complex chemical system using detailed knowledge on the atomic scale. The GROMACS software package was used to model the interactions between the lung surfactant lipids.

Simulations were performed on monolayers with different lipid compositions and at different packing densities. In tightly packed monolayers some lipids were observed to rise up out of the monolayer. In mixed monolayers there was no preference as to which type of lipid would rise. Tightly packed monolayers had large expansive lateral pressures that led to the calculation of negative surface tensions. Loosely packed monolayers had positive surface tensions that were similar to experimental measurements. Comparison to experiments might not be appropriate for tightly packed monolayers, since the interface between lipids and water was not flat.

# Acknowledgments

I would like to express my deepest appreciation to my co-supervisors, Drs. Valerie Booth and Kaushik Nag, for all of their help and assistance throughout my program of study. Their knowledge and support made this research possible, and it has been a pleasure to work with them.

I wish to thank Jennifer Rendell and Derrick Lee for their support and assistance in my research. In addition to their valuable knowledge and insight into GROMACS, they prepared the initial monolayer systems I used in my research.

I would also like to thank the other members of the Booth lab, Sarah Bourbigot, Erin Dodd, Chrystal Horwood, Sheila Kennedy, Muzaddid Sarker, June Stewart and Tran-Chin Yang, for the support, assistance and useful suggestions they have given me over the past two years.

Thanks must also go to Dr. Kevin Keough for his generous support of my research.

I greatly acknowledge the Atlantic Computational Excellence Network for providing the high performance computing resources that made this research possible. I give special thanks to Mark Staveley, Sili (Wesley) Huang and Phil Romkey for their assistance in installing and operating the GROMACS software package.

I also wish to acknowledge and thank Mauricia Garcia, who performed the experiments on monolayers that I have used as a comparison to my simulations.

Finally, I wish to thank my family for the support and assistance they have given me throughout both this program and my entire life.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Acknowledgments</b>	<b>iii</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Abbreviations</b>	<b>xi</b>
<b>1 Introduction - Lung Surfactant</b>	<b>1</b>
1.1 Lung Surfactant Composition . . . . .	3
1.1.1 Lung Surfactant Phospholipids . . . . .	4
1.1.2 Lung Surfactant Proteins . . . . .	6
1.2 Surface Activity . . . . .	8
1.3 Motivation . . . . .	11
<b>2 Introduction - Molecular Dynamics</b>	<b>13</b>
2.1 Molecular Dynamics in General . . . . .	13
2.1.1 Molecular Dynamics Algorithms . . . . .	14
2.1.2 Force Fields . . . . .	15

2.1.3	Periodic Boundary Conditions and the Minimum Image Criterion	17
2.1.4	Limitations in Molecular Dynamics . . . . .	18
2.2	GROMACS . . . . .	20
2.2.1	GROMACS Force Fields . . . . .	20
2.2.2	GROMACS Algorithm . . . . .	23
2.2.3	Temperature and Pressure Coupling . . . . .	25
2.2.4	Constraints . . . . .	27
2.2.5	Neighbour Searching . . . . .	28
2.2.6	Calculation of Long Range Electrostatic Interactions . . . . .	29
2.2.7	Parallel Algorithm . . . . .	30
2.2.8	Energy Minimization . . . . .	32
2.2.9	Other Features . . . . .	33
2.3	Previous MD Simulation . . . . .	35
2.3.1	Monolayer and Bilayer Simulation Results . . . . .	35
2.3.2	Force Field Development . . . . .	37
<b>3</b>	<b>Computational Procedure</b>	<b>39</b>
3.1	Monolayer Creation . . . . .	39
3.2	Monolayer Preparation . . . . .	40
3.3	MD Simulation . . . . .	46
3.3.1	Force Field . . . . .	46
3.3.2	Parameters . . . . .	46
3.3.3	Computer Resources . . . . .	48
3.3.4	Simulations . . . . .	49
<b>4</b>	<b>Optimization of the MD Procedure</b>	<b>51</b>
4.1	Initial Results . . . . .	52

4.2	Coupling Temperature . . . . .	53
4.3	Pressure Coupling . . . . .	54
4.4	Electrostatic Parameters . . . . .	57
4.5	Force Fields . . . . .	59
4.6	Simulation with a Bilayer . . . . .	62
4.7	Serial vs. Parallel . . . . .	63
<b>5</b>	<b>Computational Results</b>	<b>69</b>
5.1	Initial Configuration Observations . . . . .	69
5.2	Simulation Observations . . . . .	71
5.2.1	Energy Analysis . . . . .	71
5.2.2	Pure DPPC Monolayer . . . . .	73
5.2.3	Mixed DPPC/POPA Monolayer . . . . .	77
5.2.4	Mixed DPPC/POPC Monolayer . . . . .	79
5.2.5	Mixed DPPC/POPG Monolayer . . . . .	81
5.2.6	Pure POPG Monolayer . . . . .	82
5.3	Water Layer Simulation . . . . .	82
5.4	Surface Tension . . . . .	85
5.5	Comparison to Experiment . . . . .	92
5.6	Comparison to Previous Simulations . . . . .	97
<b>6</b>	<b>Conclusion and Future Work</b>	<b>100</b>
6.1	Conclusion . . . . .	100
6.2	Future Work . . . . .	103
	<b>Bibliography</b>	<b>105</b>
	<b>Appendix 1 - LINCS Algorithm</b>	<b>114</b>

# List of Tables

1.1	Components of Human Lung Surfactant. Percentages by Dry Weight	4
5.1	Surface Tensions on a Water Layer without Lipids . . . . .	84
5.2	Surface Tensions on a Water Layer from Different Water Force Fields	85

# List of Figures

1.1	Diagram of the Lungs and Lung Alveoli [1] . . . . .	2
1.2	Molecular Explanation of Surface Tension in a Body of Water . . . . .	2
1.3	Phospholipids found in LS. Tails are composed of aliphatic carbon (CH <sub>2</sub> ) chains. . . . .	5
1.4	DPPC Monolayer on Water's Surface. Colour Scheme: Carbon - Black, Oxygen - Red, Phosphorous - Tan, Nitrogen - Blue . . . . .	7
1.5	Cross Sectional Area of PC with Different Aliphatic Carbon Tails. Left: No double bonds. Right: 1 double bond in right tail. . . . .	8
2.1	Periodic Boundary Conditions and the Minimum Image Criterion . . . . .	17
2.2	LINCS Algorithm. Dashed line is the old bond length. Thick lines are the new bonds. . . . .	28
2.3	Ring Processor Topology Used by GROMACS . . . . .	31
3.1	DPPC Lipids Arranged in a Monolayer . . . . .	40
3.2	Bond Length During Energy Minimization Sequence . . . . .	42
3.3	Water Added to a Scaled DPPC Monolayer by Genbox . . . . .	43
3.4	DPPC and Water Monolayer System with Displaced Water Molecules . . . . .	44
3.5	Final Monolayer with Water Configuration Used in MD . . . . .	45
4.1	Initial Surface Tension Results . . . . .	52

4.2	Surface Tension of a Mixed DPPC/POPC Monolayer at Different Coupling Temperatures . . . . .	54
4.3	Simulation Box Height During MD Simulation with Pressure Coupling and Lipid Density 41.4 Å <sup>2</sup> /lipid . . . . .	56
4.4	Simulation Box Height During MD Simulation with Pressure Coupling and Lipid Density 59.2 Å <sup>2</sup> /lipid . . . . .	57
4.5	Surface Tension Calculations for Two Different Sets of Particle-Mesh Ewald Electrostatic Parameters . . . . .	59
4.6	Calculated Surface Tensions for Different Force Fields . . . . .	61
4.7	Time Needed to Run a 10 ps Benchmarking Simulation . . . . .	65
4.8	Speedup and Efficiency of a 10 ps Benchmarking Simulation . . . . .	66
4.9	Time to Perform 10ns Simulation on a DPPC Monolayer . . . . .	67
5.1	DPPC Monolayer Initial Configurations with Lipid Densities 41.4 Å <sup>2</sup> /lipid (left) and 77.0 Å <sup>2</sup> /lipid (right) . . . . .	70
5.2	Mixed DPPC/POPA (Green/Blue) Monolayer Initial Configurations with Lipid Densities 40 Å <sup>2</sup> /lipid (left) and 80 Å <sup>2</sup> /lipid (right). Sodium Atoms are Cyan Spheres . . . . .	70
5.3	Total Energy of a DPPC Monolayer in Water at a Lipid Density of 60 Å <sup>2</sup> /lipid . . . . .	72
5.4	Total Energy as a Function of Lipid Density . . . . .	73
5.5	Configuration of a DPPC Monolayer with Lipid Density 40 Å <sup>2</sup> /lipid After 10 ns Simulation . . . . .	75
5.6	Phosphorous Atoms of DPPC Lipids in a Monolayer with Lipid Density 80 Å <sup>2</sup> /lipid Prior to (Above) and After (Below) the 10 ns Simulation . . . . .	76

5.7	Phosphorous Atoms of DPPC Lipids in a Monolayer with Lipid Density 40 Å <sup>2</sup> /lipid Prior to (Above) and After (Below) the 10 ns Simulation	77
5.8	Mixed DPPC/POPA (Green/Blue) Monolayer with Lipid Density 45 Å <sup>2</sup> /lipid during 10 ns Simulation. Sodium Atoms are Cyan Spheres. .	78
5.9	Mixed DPPC/POPA (Green/Blue) Monolayer with Lipid Density 40 Å <sup>2</sup> /lipid during 10 ns Simulation. Headgroup phosphorous atoms are shown as tan spheres, sodium atoms as cyan spheres. . . . .	80
5.10	Mixed DPPC/POPC (Green/Blue) Monolayer with Lipid Density 40 Å <sup>2</sup> /lipid during 10 ns Simulation . . . . .	81
5.11	POPG Monolayer with Lipid Density 40 Å <sup>2</sup> /lipid after 10 ns Simulation	83
5.12	Surface Tension in a DPPC Monolayer . . . . .	86
5.13	Time Averaged Pressure Components from a DPPC Monolayer as a Function of Lipid Density . . . . .	88
5.14	Surface Tension in a Mixed DPPC/POPA Monolayer . . . . .	89
5.15	Surface Tension in a Mixed DPPC/POPC Monolayer . . . . .	90
5.16	Surface Tension in a Mixed DPPC/POPG Monolayer . . . . .	91
5.17	Surface Tension in a POPG Monolayer . . . . .	92
5.18	Surface Tension in All Monolayers . . . . .	93
5.19	Surface Tension in All Monolayers between 50 and 80 Å <sup>2</sup> /lipid . . . .	94
5.20	Computational and Experimental Measurements of a DPPC Monolayer in Water . . . . .	95
5.21	Computational and Experimental Measurements of a Mixed DPPC/POPG Monolayer in Water . . . . .	97
5.22	Surface Pressures . . . . .	98

# List of Abbreviations

ACEnet	Atlantic Computational Excellence Network
ARDS	Acute Respiratory Distress Syndrome
DMPC	Dimyristoylphosphatidylcholine
DPPC	Dipalmitoylphosphatidylcholine
DPPG	Dipalmitoylphosphatidylglycerol
EM	Energy Minimization
GB	Gigabyte
GHz	Gigahertz
LINCS	LINear Constraint Solver
LS	Lung Surfactant
MD	Molecular Dynamics
PA	Phosphatidic Acid
PBC	Periodic Boundary Conditions
PC	Phosphatidylcholine
PE	Phosphatidylethanolamine
PG	Phosphatidylglycerol
PI	Phosphatidylinositol
PME	Particle-Mesh Ewald
POPA	1-palmitoyl-2-oleoyl-phosphatidic Acid

POPC	1-palmitoyl-2-oleoyl-phosphatidylcholine
POPG	1-palmitoyl-2-oleoyl-phosphatidylglycerol
PR	Position Restraint
PS	Phosphatidylserine
RAM	Random Access Memory
RDS	Respiratory Distress Syndrome
SP-A	Surfactant Protein A
SP-B	Surfactant Protein B
SP-C	Surfactant Protein C
SP-D	Surfactant Protein D
SPC	Single Point Charge
SPC/E	Extended Single Point Charge
TM	Tubular Myelin

# Chapter 1

## Introduction - Lung Surfactant

Living things need a means to bring in oxygen for metabolic reactions. At the same time they need a means of getting rid of carbon dioxide that is produced in metabolic reactions. In mammals such as humans this exchange is done in the lungs (Figure 1.1).

Air exchange in the lungs occurs in the alveoli. The alveoli are small (50 - 100  $\mu m$ ), approximately spherical sacs [2] located at the ends of the lung airways. The alveoli are surrounded by capillaries, and are extremely thin, which allows for rapid transfer of oxygen into and carbon dioxide out of the blood. The surface of an alveolus is lined with a thin layer of water to prevent it from drying out due to its constant contact with air.

The presence of the water layer gives rise to a surface tension at the air/water interface of the alveoli. This surface tension results because the water molecules at the surface of the water layer experience a net attraction into the interior of the layer, whereas the molecules in the layer are attracted by all surrounding molecules and thus experience zero net attraction (Figure 1.2). This surface tension places a pressure on

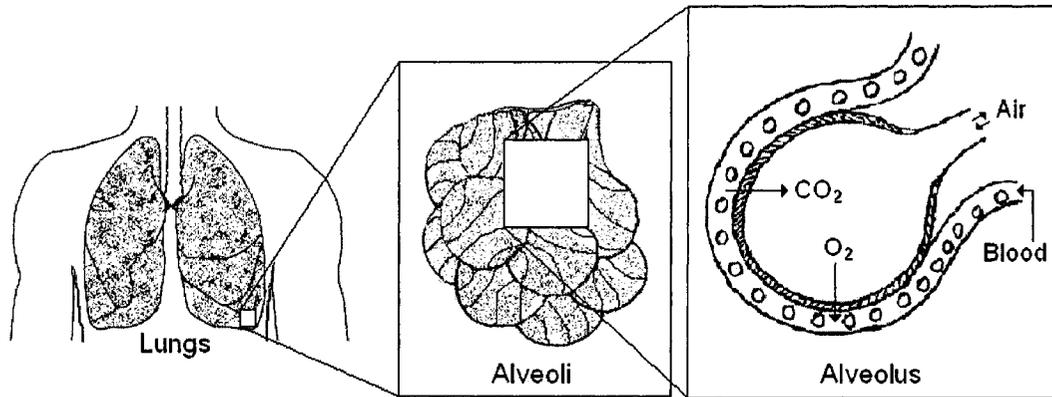


Figure 1.1: Diagram of the Lungs and Lung Alveoli [1]

the alveoli that is governed by the Laplace Law

$$\gamma = \frac{PR}{2} \quad (1.1)$$

where  $\gamma$  is the surface tension,  $P$  is the pressure and  $R$  is the alveolar radius [2]. This translates to a minimum radius of curvature of approximately  $70 \mu\text{m}$ , assuming a surface tension of water of  $72 \text{ mN/m}$  [3]. Should the radius of curvature decrease further, as it does during expiration, the alveolus will collapse due to the pressure generated by the surface tension.

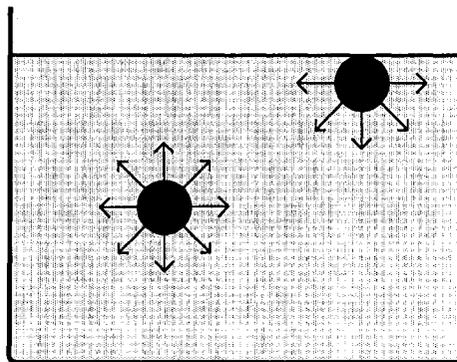


Figure 1.2: Molecular Explanation of Surface Tension in a Body of Water

Lung surfactant (LS) prevents this collapse. LS is a monomolecular layer of lipids and proteins that lines the air/water interface of the alveoli. The primary function of LS is to greatly reduce the surface tension at the air/water interface. This reduction prevents the alveoli from collapsing during expiration and reduces the work needed to re-expand the lung during inspiration. A lack of or deficiency in lung surfactant leads to serious respiratory ailments such as Respiratory Distress Syndrome (RDS) in premature infants and Acute RDS (ARDS) in adults [4].

## 1.1 Lung Surfactant Composition

LS is synthesized on the endoplasmic reticulum of type II pneumocytes (alveoli cells). The surfactant is assembled in the form of bilayers and is stored in specialized organelles called lamellar bodies. The lamellar bodies are secreted into the aqueous layer that covers the alveolar surface upon the reception of a proper regulatory signal, such as the stretching of the alveolar surface during inspiration. In the aqueous layer the lamellar bodies become hydrated and are converted into tubular myelin (TM). TM consists of ordered arrays of crossing bilayers that form stacked, rectangular tubes. LS in the form of TM absorbs readily into the monolayer at the air/water interface [2] [4].

LS is a complex mixture of lipids and proteins. It is composed primarily of phospholipids, which make up approximately 80% by weight. The most abundant phospholipid species is phosphatidylcholine (PC), with dipalmitoylphosphatidylcholine (DPPC) making up about half of the PC content. Neutral lipids make up another 10% by weight. Proteins make up the final 10%. A full list of the components of LS is given in Table 1.1 [4].

The phospholipids are believed to be the main surface active (surface tension

Table 1.1: Components of Human Lung Surfactant. Percentages by Dry Weight

Phospholipids (80%)	<ol style="list-style-type: none"> <li>1. Phosphatidylcholine (PC) 70% 35% dipalmitoylphosphatidylcholine (DPPC) 15% other disaturated species 20% unsaturated species</li> <li>2. Phosphatidylglycerol (PG) 8-9%</li> <li>3. minor amount of Phosphatidylinositol (PI)</li> <li>4. minor amount of Phosphatidylethanolamine (PE)</li> <li>5. minor amount of Phosphatidylserine (PS)</li> <li>6. minor amount of Sphingomyelin (SM)</li> </ol>
Neutral Lipids (10%)	<ol style="list-style-type: none"> <li>1. Cholesterol 10%</li> <li>2. minor amounts of monoglycerides, diglycerides and triglycerides</li> <li>3. minor amount of free fatty acids</li> </ol>
Proteins (10%)	<ol style="list-style-type: none"> <li>1. Surfactant Protein A (SP-A) 6%</li> <li>2. Surfactant Protein B (SP-B) 1.5%</li> <li>3. Surfactant Protein C (SP-C) 1.5%</li> <li>4. Surfactant Protein D (SP-D) 1%</li> </ol>

reducing) component. Experiments with compressed phospholipid films have shown that some films can lower surface tension to near zero values. The proteins are believed to regulate and assist in the activity of the phospholipids. The neutral lipids are not believed to play a significant role in surface activity [4] [5].

### 1.1.1 Lung Surfactant Phospholipids

Phospholipids are the primary component of LS. The phospholipids are composed of a head group and a pair of aliphatic carbon chains or tails. The structures of a number of phospholipids are shown in Figure 1.3 [6] [7].

A phospholipid is characterized by its headgroup. The most common phospholipid in LS is phosphatidylcholine (PC). Other phospholipid species in LS include phosphatidylglycerol (PG), phosphatidylethanolamine (PE) and phosphatidylserine (PS).

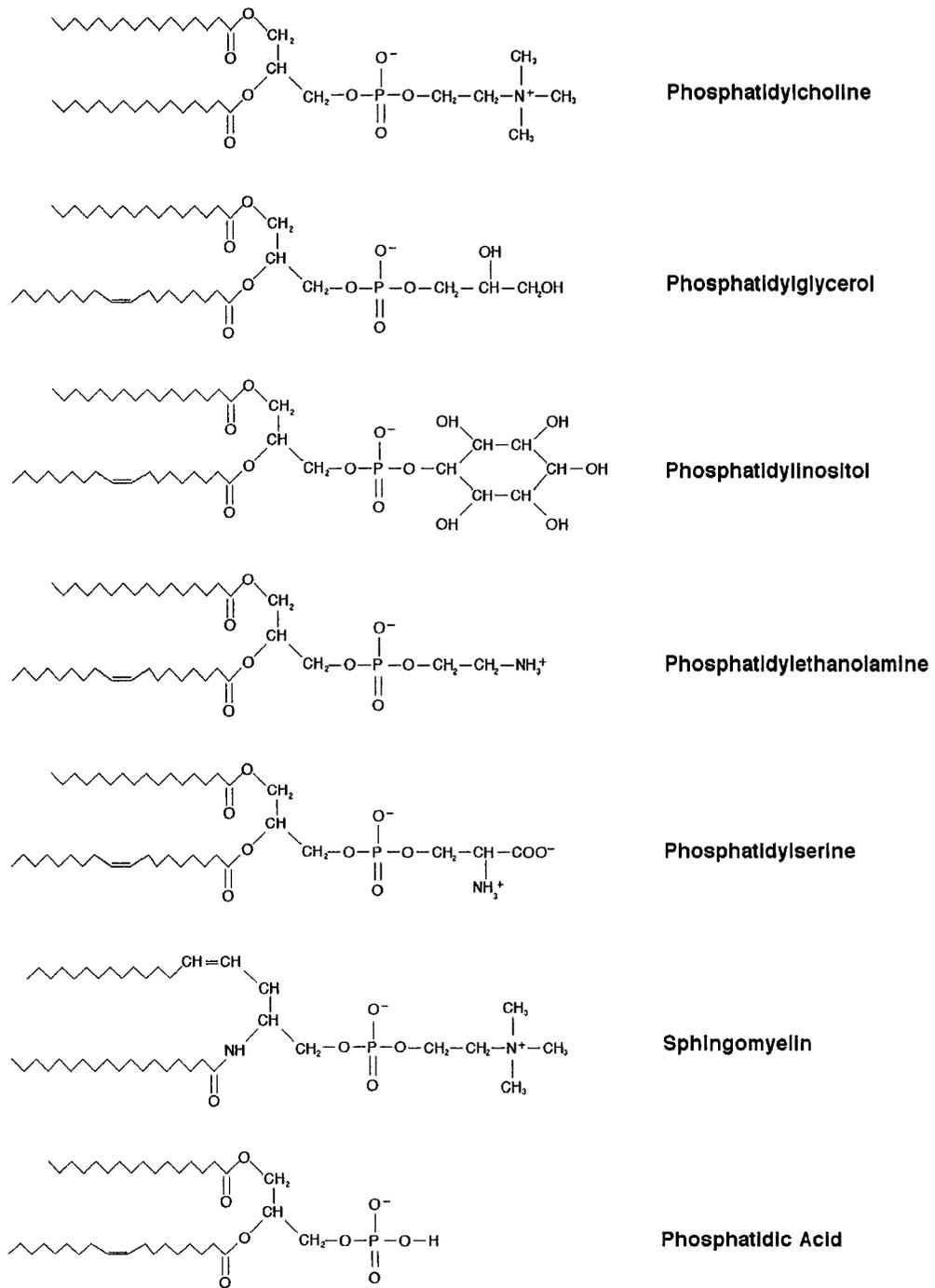


Figure 1.3: Phospholipids found in LS. Tails are composed of aliphatic carbon ( $\text{CH}_2$ ) chains.

These headgroups carry a charge, which allows them to bind with water, making them soluble in water [2].

The most common PC is DPPC (Table 1.1). DPPC has two tails of length 16 carbon atoms, and is saturated. That is, there are no double bonds in its aliphatic carbon tails. This is unusual for a phospholipid. Most phospholipids are unsaturated, with a double bond in one of their aliphatic carbon tails. PC's are zwitterionic, which means that the headgroup has both a positive and a negative charge. Other phospholipids, such as PG's, are anionic with a negative charge (Figure 1.3).

While the headgroups are hydrophilic the tails are hydrophobic and not soluble in water. Thus when placed in water phospholipids will orient themselves so that the headgroup lies on the water's surface while the tails stick out into the air. In this way the phospholipids form a monolayer at the water's surface, as shown in Figure 1.4 [2].

The presence of a double bond in a aliphatic carbon chain produces a kink or bend in the chain (Figure 1.5). The presence of this kink increases the cross sectional area of the lipid in the monolayer. As a result, unsaturated lipids with kinks cannot be compressed to the extent that saturated lipids can be compressed. To do so would result in a collapse of the monolayer [2].

### **1.1.2 Lung Surfactant Proteins**

There have been four different proteins found in LS. These have been named surfactant protein A (SP-A), SP-B, SP-C and SP-D, in chronological order of their discovery. The proportions of these proteins in LS is given in Table 1.1. The proteins in lung surfactant are believed to be modulating agents, whose role is to assist in the transfer of LS to the air/water interface, and then to optimize the function of the phospholipids during the respiratory cycle. They are not believed to play a direct

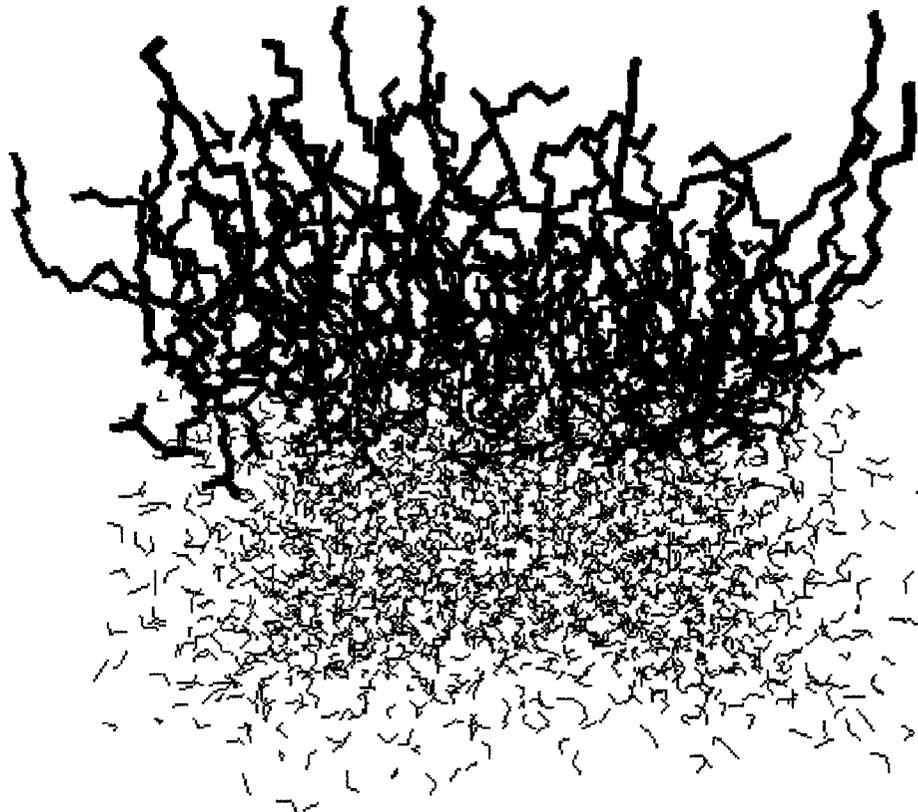


Figure 1.4: DPPC Monolayer on Water's Surface. Colour Scheme: Carbon - Black, Oxygen - Red, Phosphorous - Tan, Nitrogen - Blue

role in the reduction of surface tension [8].

SP-B and SP-C are considered to be the most important proteins in the functioning of the phospholipid monolayer. These proteins are hydrophobic and are believed to bind with anionic phospholipids. They are believed to promote the absorption of phospholipids from TM into the monolayer [5], and to optimize phospholipid function in the monolayer [8].

SP-A is a large, water soluble protein [8]. It is believed to have only a minor role in surface activity. Experiments with knock-out mice, engineered to deactivate the expression of SP-A, showed that the mice could breath normally without SP-A. Those same experiments showed that the mice were extremely prone to infection, indicating

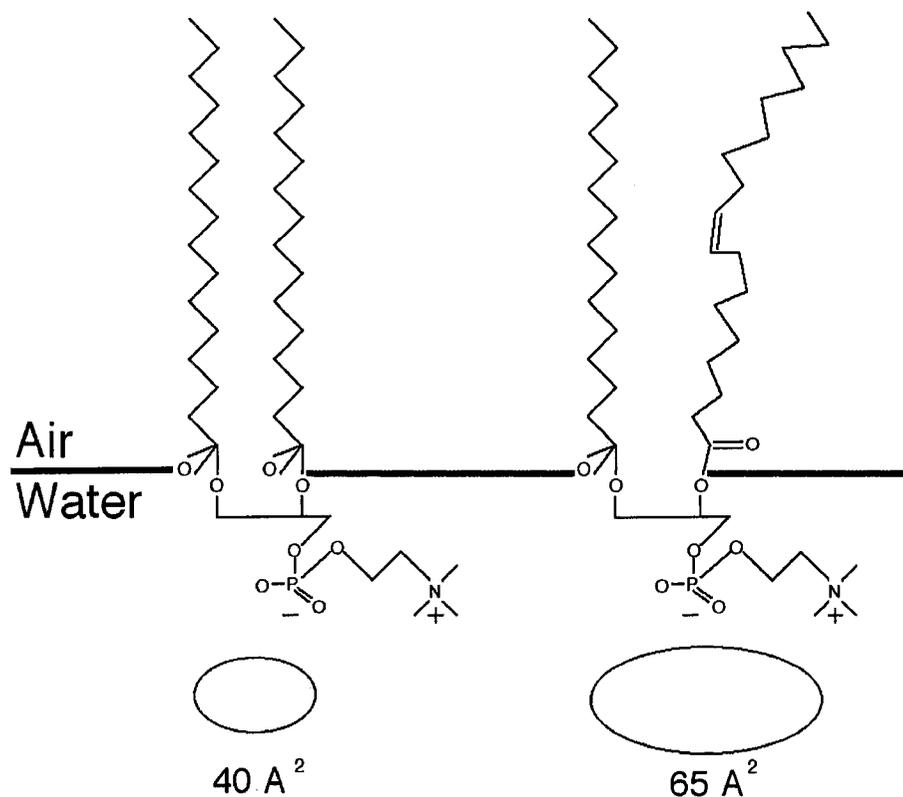


Figure 1.5: Cross Sectional Area of PC with Different Aliphatic Carbon Tails. Left: No double bonds. Right: 1 double bond in right tail.

that SP-A has a role in protecting the lung from pathogens [4]. SP-A is also believed to play a role in the formation of TM [8].

SP-D, like SP-A, is also large and water soluble. It is not believed to play a role in surface activity, since it is not associated with the surface active lipid-protein complexes [4].

## 1.2 Surface Activity

Surface activity in LS is the ability of LS to lower the surface tension at the air/water interface. Phospholipids are the main surface active component in LS, while the

proteins are believed to be modulating agents which optimize the functioning of the phospholipids [4]. The charged phospholipids reduce surface tension by binding water molecules at the surface, counteracting the attraction of water molecules within the layer [2].

LS was discovered by von Neergaard in the 1920's [9]. The significance of the discovery was not recognized by the scientific and medical communities at the time [5], and LS would not be rediscovered until the 1950's [10] [11]. These experiments revealed three important properties of the lungs and LS. First, the work done against surface tension forces constitutes a major part of the effort required to inflate the lung. Second, observed differences in the pressure-volume relationship of lungs being inflated or deflated indicated that the surface tension of the alveoli changed during the respiratory cycle. Third, the surface tension was low compared to saline [2].

LS participates in two different physiochemical processes. The first occurs at birth, when the newborn baby takes its first breath. The air/water interface is expanded from 2 cm<sup>2</sup> (surface area of the baby's larynx) to 2 - 3 m<sup>2</sup> (surface area of baby's expanded lung). A considerable amount of work is needed to stretch the air/water interface, and a reduction in surface tension greatly reduces this work. The second process is to develop and keep stable the alveoli throughout the subsequent respiratory cycles, when repeated expirations will reduce the size of the alveoli to the point of collapse [3].

To accomplish these tasks LS must possess two essential properties. First, it must show very rapid adsorption kinetics to the air/water interface, and lower surface tension to around 25 mN/m, the lowest surface tension possible in an uncompressed monolayer, within seconds. It is critical that the surfactant reaches the interface before the newborn completes its first inspiration. In later respiratory cycles good adsorption rates allow LS to replenish the interface. Surface active molecules lost

during expiration are replaced during inspiration [4].

The second essential property LS must possess is the ability to lower surface tension to near 0 mN/m upon compression. These very low surface tensions are needed to prevent alveoli collapse during expiration, and to reduce the work needed to reinflate the lungs during inspiration [4].

The most important surface active phospholipid is DPPC. Experiments with monolayers of DPPC have shown that when these monolayers are compressed the surface tension at the air/water interface drops to almost 0 mN/m. Furthermore, these compressed monolayers are stable and the near zero surface tension is maintained for hours [5]. A DPPC enriched monolayer is believed to be responsible for the reduction of surface tension to almost zero.

While DPPC meets the second essential property of LS it does not meet the first. DPPC has very poor adsorption properties at temperatures below its liquid-gel transition of 41°C [3]. Thus on its own it cannot fill the air/water interface of a newborn, nor can it replenish itself in a monolayer during inspiration.

One way to facilitate DPPC adsorption is to mix it with another phospholipid, such as PG or PI [5]. Experiments with such mixtures have shown substantial increases in DPPC adsorption. Bangham and his associates have gone so far as to develop an artificial LS using only DPPC and 1-palmitoyl-2-oleoyl-PG (POPG) in a 7:3 ratio [12]. While these fluid lipids promote adsorption, they are unsaturated and have a higher cross sectional area (Figure 1.5) than DPPC. Thus if these mixed monolayers are compressed to the extent needed to obtain near zero surface tensions, they will collapse. Of all the LS phospholipids, only DPPC can be compressed to the necessary degree [4].

It is believed that the proteins resolve this conundrum through their interactions with the lipids. One theory regarding these interactions is the “Squeeze-Out” theory.

In this theory the proteins unpack the lipids from the TM at the air/water interface and insert them into the monolayer during inspiration, as the monolayer expands. The proteins and non-DPPC lipids aid in DPPC adsorption. During expiration the proteins remove the non-DPPC lipids, leaving a DPPC rich monolayer capable of reducing surface tension to near zero values. The removed lipids are stored in bilayer structures above or below the monolayer until they are reinserted during inspiration [4].

This is one of several theories. An alternate theory has been proposed by Bangham and his associates. They reject the idea that DPPC lowers surface tension to near zero, since this would imply that there was no longer an interface [3]. They agree that during inspiration fluid lipids such as POPS help DPPC adsorb to the air/water interface. However, during expiration they propose that DPPC crystallizes out of the air/water interface to form a solid phase. Thus there would be a liquid/solid interface and a solid/air interface. The solid and rigid DPPC would then resist the high pressures brought on by compression, and in so doing prevent the alveoli from collapsing. During inspiration DPPC adsorbed into the monolayer from TM would reduce the work needed to reinflate the lung [3] [12] [13].

### 1.3 Motivation

The theories presented in the previous section, along with others, have been proposed and are supported to varying degrees by experiment. Experiments do not provide a complete picture of LS function, since they measure macroscopic properties. The interactions at the molecular level are inferred, and to date are not fully understood. It is hoped that MD simulations, such as the simulations performed for this work, will provide insight these molecular interactions. This will lead to a better understanding

of LS function, and ultimately to better treatments for LS dysfunction.

An introduction to MD is given in the next chapter. The general theory of MD and how it is implemented in GROMACS is described. Chapter 3 discusses the methods and procedure used in the monolayer simulations. Chapter 4 describes some specific modifications to the procedure that were considered and why they were or were not implemented. The results of the simulations are given and discussed in Chapter 5. The conclusions drawn from the simulations are summarized in Chapter 6, along with some suggestions for future work.

## Chapter 2

# Introduction - Molecular Dynamics

### 2.1 Molecular Dynamics in General

Molecular dynamics is a computer simulation technique where the time evolution of a set of interacting atoms is followed by integrating their equations of motion [14]. MD is used to mimic the dynamic behaviour of complex chemical or biochemical systems. The aim of MD is to understand and predict macroscopic behaviour based on detailed knowledge on an atomic scale [15].

MD is in principle a deterministic method [14]. Given an initial set of positions and velocities, the time evolution of the system is completely determined.

Molecular dynamics has become an important tool in the study of biochemical systems. Experiments cannot provide a complete picture of the behaviour of the system being studied. Experiments can provide information about macroscopic properties, but the interactions on the molecular level remain elusive. For example, the composition of the lipids and proteins that make up LS is well known. However, the interactions between the lipids and proteins at the molecular level are not well understood. MD simulations can be used to model the interactions of the LS components.

These models provide insight into the behaviour of LS [16].

### 2.1.1 Molecular Dynamics Algorithms

Molecular dynamics simulations solve Newton's equations of motion for a system of  $N$  interacting atoms

$$m_i \frac{\partial^2 \mathbf{r}_i}{\partial t^2} = \mathbf{F}_i, i = 1 \dots N \quad (2.1)$$

where  $\mathbf{r}_i$  is the position of atom  $i$ ,  $m_i$  is its mass and  $\mathbf{F}_i$  is the force acting on it. The force is calculated from the negative derivatives of a potential function  $V(\mathbf{r}_1, \dots, \mathbf{r}_N)$ :

$$\mathbf{F}_i = -\frac{\partial V}{\partial \mathbf{r}_i} \quad (2.2)$$

The equations are solved simultaneously in small time steps. The system is followed for some time, taking care that the temperature and pressure remain at the required values, and the atomic coordinates are written to an output file at regular intervals. The coordinates as a function of time represent a trajectory of the system [17, Pages 2 – 3].

The most widely used algorithm for integrating the equations of motion is the Verlet algorithm [18, Pages 78 – 82]. This algorithm was derived from the Taylor series expansion about  $\mathbf{r}(t)$

$$\begin{aligned} \mathbf{r}(t + \Delta t) &= \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2 + \dots \\ \mathbf{r}(t - \Delta t) &= \mathbf{r}(t) - \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2 + \dots \end{aligned} \quad (2.3)$$

where  $\mathbf{v}(t) = \frac{\partial \mathbf{r}}{\partial t}$  is the velocity and  $\mathbf{a}(t) = \frac{\partial^2 \mathbf{r}}{\partial t^2}$  is the acceleration. Adding the two

expansions gives the equation for updating the positions

$$\mathbf{r}(t + \Delta t) = 2\mathbf{r}(t) - \mathbf{r}(t - \Delta t) + \mathbf{a}(t)\Delta t^2 \quad (2.4)$$

which has an error on the order of  $\Delta t^4$ .

The Verlet algorithm is time-reversible and, given conservative forces, will conserve linear momentum. The main drawback to the algorithm is that velocities are not known. There are modifications to the Verlet algorithm that do explicitly calculate velocities [18, Pages 78 – 82].

### 2.1.2 Force Fields

In MD, atoms interact with each other. These interactions create forces which influence the motion of the atoms. Hence the physics in an MD simulation is contained in the forces. A simulation mimics the behaviour of a real system of atoms only to the extent that the inter-atomic forces in the simulation are similar to the forces that real atoms would experience [14].

Force fields usually consist of two major components, each of which describes a different type of interaction. The first component describes interactions between atoms connected via covalent bonds. This can include such things as bond length, bond angle and dihedrals. The second component describes non-bonded interactions. Usually this includes van der Waals interactions and electrostatic interactions between charged atoms or molecules [19].

Different force fields use different levels of detail. All-atom force fields treat every atom, including hydrogen, explicitly. United-atom force fields combine each aliphatic carbon and its associated hydrogens into a single particle. Coarse-grained force fields treat larger molecular units, such as amino acid side-chains and water molecules, as

single particles [19].

A lot of work goes into creating a force field. Force fields are obtained by a process of refinement against experimental or quantum mechanical data [20]. The information obtained from experiments is expressed as potential energy functions [14]. These potential energy functions are then used to drive MD simulations. These simulations calculate macroscopic properties which can be compared to experiment [21]. Some of the specific potential functions used in MD are discussed in Section 2.2.1.

There are four main force fields in common use for simulating biological systems [19]. These are

- AMBER [22] [23]
- CHARMM [24] [25]
- GROMOS [26]
- OPLS [27]

There are two phospholipid force fields in common use today. One is the all-atom CHARMM force field. The second was created by Berger and his coworkers [28], and was developed with parameters taken from united-atom versions of AMBER and OPLS, as well as parameters derived from experiment (Section 2.3.2).

One final note about forces is that they are usually subjected to a cutoff. The potential functions can have an infinite range. In practice, a cutoff radius  $R_c$  is established, and interactions between atoms separated by more than  $R_c$  are ignored. This results in enormous savings in computer resources, because the number of atomic pairs separated by a distance  $r$  grows by  $r^2$  [14]. Ignored interactions can introduce errors in the force calculation. Since long range forces fall off as  $1/r^2$  or quicker (Section 2.2.1), a sufficiently large cutoff will result in a negligible error.

### 2.1.3 Periodic Boundary Conditions and the Minimum Image Criterion

Periodic boundary conditions (PBC) are employed in MD simulations to mitigate boundary effects [29]. The simulated system, regardless of its size, would be negligible to a macroscopic piece of matter. The system should see a continuation of the system in all directions, not empty space. Simply having the system end at its boundaries would greatly magnify surface effects [14].

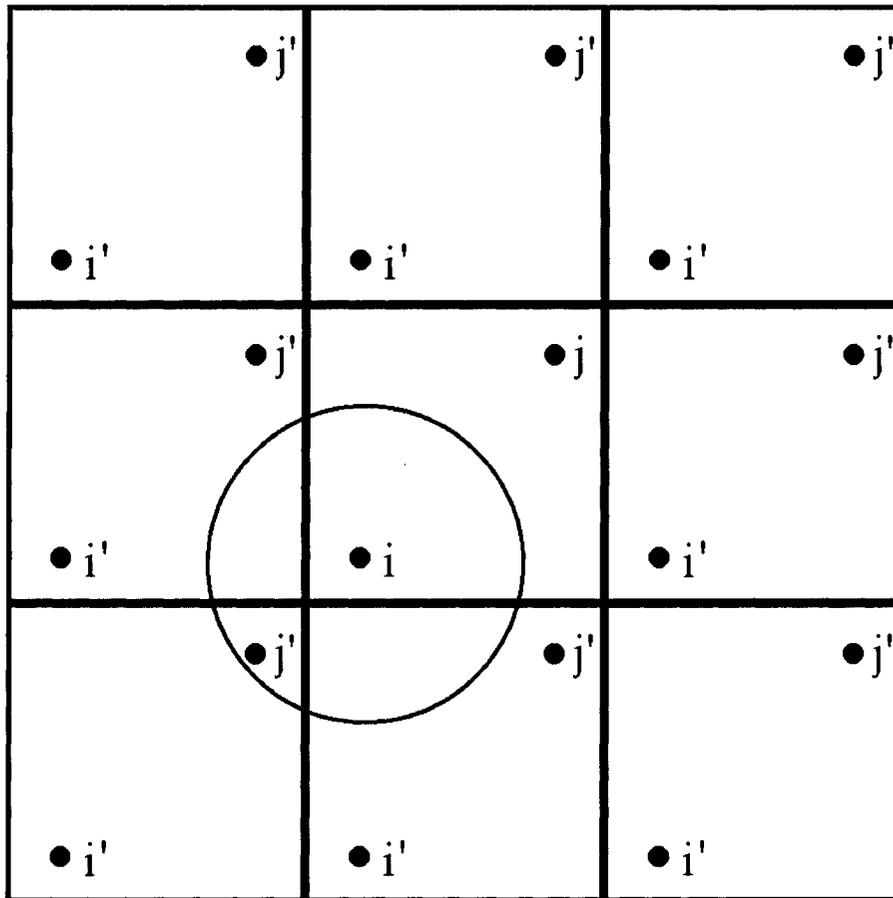


Figure 2.1: Periodic Boundary Conditions and the Minimum Image Criterion

When PBC are used, the particles in the simulated system are enclosed in a box.

The box is surrounded by an infinite number of identical image boxes, which are stacked in a space-filling way [29]. This is illustrated in two dimensions in Figure 2.1. The simulation box, containing particles  $i$  and  $j$ , is surrounded by identical image boxes containing image particles  $i'$  and  $j'$ .

The use of PBC introduces an infinite number of image particles  $j'$  to interact with particle  $i$  [14]. The number of interactions is minimized by employing the minimum image criterion. If  $L_{min}$  is the smallest box dimension, the force cutoff should be chosen such that  $R_c < \frac{1}{2}L_{min}$ . This is the minimum image criterion [29]. When this criterion is met, at most only one of  $j$  and its images  $j'$  will be within the cutoff radius of  $i$  (circle in Figure 2.1). Thus despite the introduction of an infinite number of images there will only be one interaction for the simulation to calculate.

#### 2.1.4 Limitations in Molecular Dynamics

There are a number of limitations inherent in MD [17, Pages 3 – 5].

- The simulations use classical mechanics. Quantum mechanical effects are ignored. A test of the validity of the classical assumption is based on the de Broglie thermal wavelength, defined as

$$\Lambda = \sqrt{\frac{2\pi\hbar}{Mk_bT}} \quad (2.5)$$

where  $\hbar$  is Planck's constant divided by  $2\pi$ ,  $M$  is the atomic mass,  $k_b$  is Boltzmann's constant and  $T$  is the temperature. The classical approximation is valid if  $\Lambda \ll a$ , where  $a$  is the mean nearest neighbour separation [14].  $\Lambda$  is on the order of 0.1 for light elements such as lithium, and decreases for heavier elements. Thus for systems containing heavy phospholipids (about 50 carbon, nitrogen, oxygen and phosphorous atoms) the approximation is good.

- Electrons are in their ground state. MD uses a conservative force field that is a function of the positions of the atoms only. The electron motions are not considered, and the electrons are expected to adjust their dynamics instantly when the atomic positions change (the Born-Oppenheimer approximation). Problems arise for electron transfer processes, electronically excited states and chemical reactions. None of these occur in the phospholipid simulations.
- The force fields are approximate. They are not really a part of the simulation method and their parameters can be user-modified as the need arises or knowledge improves. How good the simulation is depends on how good the force fields are. The force fields used in the simulations are discussed in Sections 2.2.1, 3.3.1 and 4.5.
- The force field is pair-additive. All non-bonded forces result from the sum of non-bonded pair interactions. Non pair-additive interactions, such as interaction through atomic polarizability, are represented by effective pair potentials. This also means that the pair interactions are not valid for isolated pairs or for situations that differ appreciably from the test systems on which the models were parameterized.
- Long-range interactions are cutoff. This can lead to errors in the force calculations. Since forces fall off as  $1/r^2$  or greater, a sufficiently large cutoff should not have too serious an effect. The effects due to cutoffs are discussed in Section 4.4.
- Boundary conditions are unnatural. This is the least important limitation, and any errors introduced are small.

## 2.2 GROMACS

GROMACS [30] [31] [32] [33] is a software package developed to perform molecular dynamics simulations and energy minimizations. GROMACS is a versatile, optimized and parallelized collection of programs and libraries for the simulation of MD and the analysis of the resulting data [32]. GROMACS consists of a serial preprocessor that takes care of system decomposition and molecular topology definition, the parallel MD program kernel, and a number of post-processing analysis programs [31].

A description of how GROMACS force fields are set up is given in the next section. This is followed by a description of the GROMACS algorithm, which shows how GROMACS uses the force fields to perform the simulations. Some important modifications to the basic algorithm are then described. These modifications, such as temperature coupling, were implemented in the monolayer simulations.

### 2.2.1 GROMACS Force Fields

The GROMACS force field is built up from two distinct components [17, Pages 41 – 42]:

- The set of equations (potential functions) used to generate the potential energies and their derivatives, the forces.
- The parameters used in this set of equations.

The potential functions can be subdivided into three parts

1. Non-bonded: Lennard-Jones or Buckingham, and Coulomb or modified Coulomb. These are distance interactions. They contain a repulsion term and a dispersion term (Lennard-Jones or Buckingham), and a Coulomb term.

2. Bonded: covalent bond stretching, angle-bending, improper dihedrals, and proper dihedrals. These are interactions between connected atoms. The connections have a structure that must be preserved. The range of these interactions is one atom for bonds, two atoms for angles and three atoms for dihedrals.
3. Restraints: position, angle, distance, orientation and dihedral restraints.

In GROMACS the force fields are used with what is known as a topology file. The topology of the system is a description of the structure of the system on the atomic level. That is, the types of atoms in molecules, which atoms are bonded within molecules, the angles between groups of bonded atoms, etc. The topology file tells GROMACS on which atoms and combinations of atoms the various potential functions must act. It also tells GROMACS what parameters must be applied to the various potential functions. The bonded and restraint interactions are determined by fixed lists that are included in the topology file [17, Page 83].

### Non-Bonded Potentials

GROMACS uses three non-bonded potentials. These are the Lennard-Jones, Buckingham and Coulomb potentials [31].

The Lennard-Jones potential for two atoms separated by a distance  $r$  is given by [31]

$$V(r) = \frac{C_{12}}{r^{12}} - \frac{C_6}{r^6} \quad (2.6)$$

where  $C_6$  and  $C_{12}$  are experimentally determined parameters for the types of atoms involved.

The Lennard-Jones potential provides a short-range repulsion term and a dispersion term. The Buckingham potential also provides these terms. It has a more flexible and realistic repulsion term, but is more expensive to compute. The form of

the Buckingham potential is [31]

$$V(r) = A \exp(-Br) - \frac{C_6}{r^6} \quad (2.7)$$

where  $A$ ,  $B$  and  $C_6$  are experimentally determined parameters.

Finally, the Coulomb potential is [31]

$$V(r) = \frac{1}{4\pi\epsilon_0} \frac{q_i q_j}{r} \quad (2.8)$$

where  $q_i$  and  $q_j$  are the charges on the atoms.

### Bonded Potentials

Bonded forces are based on a fixed list of atoms per interaction. There are four types [31].

Covalent bonds are interactions between two atoms. They use a two-body harmonic potential [31].

$$V(r) = \frac{1}{2} k_b (r - b)^2 \quad (2.9)$$

Covalent bond angles use a three-body harmonic potential [31]

$$V(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3) = \frac{1}{2} k_\theta (\theta - \theta_0)^2 \quad (2.10)$$

where  $\theta$  is the angle between  $\mathbf{r}_{21}$  and  $\mathbf{r}_{23}$ . Dihedral angles use a four-body potential given by [31]

$$V(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4) = \frac{1}{2} V_o [1 + \cos(n\phi - \phi_0)] \quad (2.11)$$

where  $\phi$  is the angle between the planes defined by  $\{\mathbf{r}_{21}, \mathbf{r}_{31}\}$  and  $\{\mathbf{r}_{32}, \mathbf{r}_{43}\}$ . Improper dihedrals also use a four-body potential. Finally, an alternative form known as the

Ryckaert-Bellmans potential is allowed.

$$V(\phi) = \frac{1}{2} \sum_{n=0}^5 a_n \cos^n \phi \quad (2.12)$$

## 2.2.2 GROMACS Algorithm

A preprocessor is run before the MD simulation. The preprocessor reads in a description of the molecules in the system (the topology and force fields) and creates an atomic description that can be used in the MD. The preprocessor also reads in a coordinate file listing the initial positions of the atoms, and a file of parameters for the simulation. The preprocessor outputs a single binary file that has all of the information needed for the MD run [31]. The preprocessor can be run on any computer, not necessarily the one that will perform the MD.

The MD program performs the MD simulation. The MD program uses a simple four step algorithm to perform the simulation [17, Pages 16 – 22]. A review of this algorithm shows how GROMACS calculates system properties such as energy and temperature as it computes the movement of the atoms.

In the first step the initial conditions of the simulation are loaded from the preprocessor output. These consist of the system topology and force field, and the initial positions of the atoms. The initial velocities of the atoms can also be input at this time. If the initial velocities are not input, then they are generated with a Maxwellian distribution using random numbers.

In the second step the forces acting on the individual atoms are calculated. The force on an atom is computed by calculating the forces from non-bonded atoms plus the forces due to bonded interactions, along with any restraining or external forces. The potential energy of each interaction is computed at this time, and summed to

give the total potential energy of the system. A kinetic energy tensor is calculated from the atomic velocities  $\mathbf{v}_i$  and masses  $m_i$

$$\mathbf{E}_{kin} = \frac{1}{2} \sum_{i=1}^N m_i \mathbf{v}_i \otimes \mathbf{v}_i. \quad (2.13)$$

The system temperature  $T$  can be obtained from

$$\frac{1}{2} N_{df} k_b T = E_{kin} \quad (2.14)$$

where  $N_{df}$  is the number of degrees of freedom of the system and  $k_b$  is Boltzmann's constant. A virial tensor is also calculated.

$$\mathbf{\Xi} = -\frac{1}{2} \sum_{i < j} \mathbf{r}_{ij} \otimes \mathbf{F}_{ij} \quad (2.15)$$

The pressure tensor is calculated from the kinetic energy and the virial, and the box volume  $V$ .

$$\mathbf{P} = \frac{2}{V} (\mathbf{E}_{kin} - \mathbf{\Xi}) \quad (2.16)$$

Finally the surface tension is calculated as the difference between the normal and lateral pressures on the system, multiplied by the inverse of the box height.

$$\gamma_s = \frac{1}{h_z} \left( P_{zz} - \frac{P_{xx} + P_{yy}}{2} \right) \quad (2.17)$$

The positions of the atoms are updated in the third step. GROMACS uses the positions  $\mathbf{r}$  and forces  $\mathbf{F}$  at time  $t$ , and the velocities  $\mathbf{v}$  at time  $t - \frac{\Delta t}{2}$ , where  $\Delta t$  is the size of the time step. The update is performed with a leap-frog algorithm, which is equivalent to the commonly used Verlet algorithm from Section 2.1.1 [18, Pages 78

– 82]. The position at time  $t + \Delta t$  is calculated from

$$\begin{aligned}\mathbf{v}\left(t + \frac{\Delta t}{2}\right) &= \mathbf{v}\left(t - \frac{\Delta t}{2}\right) + \frac{\mathbf{F}(t)}{m}\Delta t \\ \mathbf{r}(t + \Delta t) &= \mathbf{r}(t) + \mathbf{v}\left(t + \frac{\Delta t}{2}\right)\Delta t\end{aligned}\tag{2.18}$$

The algorithm is of third order in  $\mathbf{r}$  and is time-reversible. The advantage of the leap-frog algorithm is that it explicitly calculates velocities. The velocity at time  $t$ , given by

$$\mathbf{v}(t) = \frac{\mathbf{v}(t - \Delta t) + \mathbf{v}(t + \Delta t)}{2}\tag{2.19}$$

can then be used to calculate the kinetic energy.

The final step is to write output to files. This output consists of atomic positions and velocities, along with the energies, temperature, pressure, etc. The output is written to files at regular intervals. Steps two through four are then repeated for the required number of time steps.

### 2.2.3 Temperature and Pressure Coupling

In GROMACS it is possible to control the temperature of the simulated system. The temperature is controlled by coupling the system to a temperature bath at reference temperature  $T_0$ . This is done with the weak coupling scheme of Berendsen *et al* [34].

The effect of the Berendsen algorithm is that a deviation of the system temperature from  $T_0$  is slowly corrected according to

$$\frac{dT}{dt} = \frac{T_0 - T}{\tau}.\tag{2.20}$$

Thus the deviation decays exponentially with time constant  $\tau$ . The algorithm is im-

plemented by scaling the velocities at each time step with a time-dependent constant  $\lambda$  [31]

$$\lambda = \left[ 1 + \frac{\Delta t}{\tau_T} \left( \frac{T_0}{T(t - \frac{1}{2}\Delta t)} - 1 \right) \right]^{\frac{1}{2}} \quad (2.21)$$

where  $\tau_T$  is related to  $\tau$  by the total heat capacity of the system  $C_V$  [17, Pages 22 – 23]

$$\tau = \frac{2 C_V \tau_T}{N_{df} k_b}. \quad (2.22)$$

In practice  $\lambda$  is very close to one.

Temperature coupling is applied during the leap-frog algorithm. The updated algorithm is

$$\begin{aligned} \mathbf{v} \left( t + \frac{\Delta t}{2} \right) &= \lambda \left[ \mathbf{v} \left( t - \frac{\Delta t}{2} \right) + \frac{\mathbf{F}(t)}{m} \Delta t \right] \\ \mathbf{r}(t + \Delta t) &= \mathbf{r}(t) + \mathbf{v} \left( t + \frac{\Delta t}{2} \right) \Delta t \end{aligned} \quad (2.23)$$

Pressure coupling is also accomplished with the Berendsen algorithm [34]. The coordinates and box vectors are scaled at every step with matrix  $\mu$ , defined by [31]

$$\mu = \left[ 1 + \frac{\Delta t}{\tau_P} \beta (P(t) - P_0) \right]^{\frac{1}{3}} \quad (2.24)$$

where  $P_0$  is the reference temperature,  $\tau_P$  is the time constant and  $\beta$  is the isothermal compressibility of the system.

Pressure coupling is applied after the positions have been updated and constrained (Section 2.2.4). The coordinates and vector containing the box size ( $\mathbf{b}$ ) are scaled.

$$\begin{aligned} \mathbf{r}(t + \Delta t) &= \mu \mathbf{r}(t + \Delta t) \\ \mathbf{b} &= \mu \mathbf{b} \end{aligned} \quad (2.25)$$

## 2.2.4 Constraints

The motions of atoms in a molecule are constrained by their bonds to other atoms. The atoms are free to move provided that their distance and orientation with their bonded counterparts do not change. In GROMACS the update of atom positions is unconstrained. Thus after an update the bonds need to be reset to their correct length and orientation.

Bond lengths in GROMACS are modeled as harmonic potentials. It is possible instead to constrain the bond length to a fixed distance. Constraints are considered a more faithful representation of the physical behaviour of bond vibrations, and allow for an increase in the time step by a factor of four [35]. There are several algorithms that perform this task, such as SETTLE [36] for small molecules or SHAKE [37] for large molecules.

A constraint algorithm was developed for GROMACS, called LINCS [35], or Linear Constraint Solver. LINCS was developed due to shortcomings in the other algorithms. SETTLE is an analytical method that works well for small molecules such as water, but cannot be applied to larger molecules. SHAKE is an iterative method that does not guarantee convergence, and which cannot be parallelized. LINCS is a two-step, parallelizable algorithm.

The LINCS algorithm is illustrated in Figure 2.2. In the first step the projections of the new bonds on the old bonds are set to zero. In the second step a correction is applied for the lengthening of the bonds due to rotation.

The derivation of LINCS is an illustrative example of the optimized numerical algorithms used in GROMACS. LINCS is based on matrices but avoids matrix-matrix multiplication and matrix inversion. The derivation of the algorithm [35] is given as an appendix.

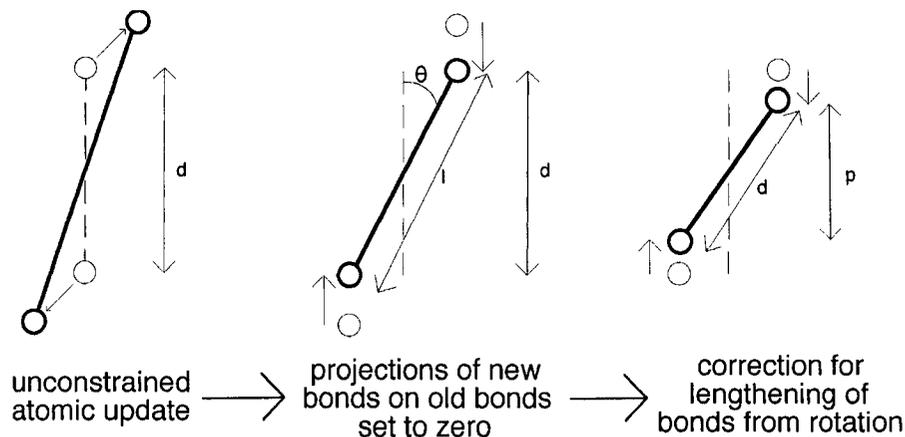


Figure 2.2: LINCS Algorithm. Dashed line is the old bond length. Thick lines are the new bonds.

## 2.2.5 Neighbour Searching

GROMACS keeps track of interacting atoms in lists. For bonded forces, these lists are fixed. For non-bonded forces, a list is made for each particular atom  $i$  of the atoms that are within the cutoff distance. This list is updated periodically, usually every five to ten time steps, to keep track of atoms that enter or leave the cutoff distance. Atomic motion is usually slow enough that only a few atoms will enter or leave the cutoff distance over the course of ten time steps.

The advantage in using these lists is that GROMACS does not need to search for interacting atoms at each time step. Less time and effort is spent determining which atoms interact with each other.

GROMACS uses a grid search method to find atoms with the cutoff distance. All of the atoms are put on the grid, which has a smallest spacing greater than  $R_{co}/2$  in each direction. An atom  $i$  has three images in each direction, corresponding to translations of  $-1$ ,  $0$  and  $+1$  box vector. First the images of  $i$  are constructed and then the neighbouring cells are searched. For every atom, fewer than 125 ( $5^3$ ) neighbouring cells are searched. As a result the grid search algorithm scales linearly

with the number of atoms. [17, Pages 18 – 20]

## 2.2.6 Calculation of Long Range Electrostatic Interactions

The total electrostatic energy of  $N$  particles in a box with index vector  $(n_x, n_y, n_z)$  is given by

$$V = \frac{f}{2} \sum_{n_x} \sum_{n_y} \sum_{n_z^*} \sum_i^N \sum_j^N \frac{q_i q_j}{r_{ij}} \quad (2.26)$$

where  $f = \frac{1}{4\pi\epsilon_0}$  and  $r_{ij}$  is the real distance between charges and not the minimum image. The star indicates that terms with  $i = j$  should be omitted when  $\mathbf{n} = (0, 0, 0)$ . This sum is conditionally convergent, but very slow [17, Page 77].

There are several options for calculating  $V$  in an MD simulation. The simplest method would be to ignore charges beyond a cutoff distance. P.P. Ewald suggested a different method [38]. Ewald converted Equation 2.26 into two quickly converging sums, one in direct space and one in reciprocal space, and a constant term.

$$V = V_{dir} + V_{rec} + V_0 \quad (2.27)$$

$$V_{dir} = \frac{f}{2} \frac{1}{4\pi\epsilon_0} \sum_{n_x} \sum_{n_y} \sum_{n_z^*} q_i q_j \frac{\text{erfc}(\beta r_{ij})}{r_{ij}} \quad (2.28)$$

$$V_{rec} = \frac{f}{2\pi V_{box}} q_i q_j \sum_{m_x} \sum_{m_y} \sum_{m_z^*} \frac{\exp\left(-\left(\frac{\pi \mathbf{m}}{\beta}\right)^2 + 2\pi i \mathbf{m} \cdot (\mathbf{r}_i - \mathbf{r}_j)\right)}{\mathbf{m}^2} \quad (2.29)$$

$$V_0 = -\frac{f\beta}{\sqrt{\pi}} \sum_i^N q_i^2 \quad (2.30)$$

where  $\beta$  is a parameter that determines the relative weight of the direct and reciprocal sums,  $V_{box}$  is the the volume of simulation box and  $\mathbf{m} = (m_x, m_y, m_z)$ . The reciprocal sum is a sum over wave vectors. A typical cutoff would be ten wave vectors in each direction. The reciprocal sum is computationally expensive, increasing as  $O(N^{3/2})$ ,

and is good only for small systems [17, Page 77].

A better method for dealing with the reciprocal space sum is the Particle-Mesh Ewald (PME) method proposed by Tom Darden [39] [40]. In PME the wave vectors are not summed directly. Instead, the charges are assigned to a grid using cardinal  $B$ -spline interpolation. This grid is then Fourier transformed with a 3D FFT algorithm and the reciprocal energy term obtained by a single sum over the grid in  $k$ -space. The potential at the grid points is calculated by inverse transformation, and interpolation factors are used to calculate the forces on each atom. The PME algorithm scales as  $N \log(N)$ , and is substantially faster than ordinary Ewald summation on medium to large systems [17, Page 78].

### 2.2.7 Parallel Algorithm

The GROMACS software package was written to make use of parallel computing [32]. The most computationally intensive and time consuming part of an MD simulation is the evaluation of the non-bonded forces between pairs of atoms [15]. GROMACS divides this work between a number of different processors, which do the calculations simultaneously. In this way the time needed to perform the calculations is reduced.

GROMACS uses the message-passing parallel method, in which all parallelism has been explicitly programmed into the source code. A minimalist approach to parallelization was taken. GROMACS uses as many existing routines as possible. The code was written in ANSI C and uses only five communication routines that were not already available [15].

In keeping with this minimalist approach the simplest processor topology, or connection scheme, was used. This was a ring, as shown in Figure 2.3 [15]. Each processor is connected to and communicates with the processors immediately in front of and

behind it in the ring.

In a message passing scheme the atoms must be divided between the processors. GROMACS allocates atoms to processors using the particle decomposition method. Each processor is given an equal number of atoms. If there are  $N$  atoms and  $P$  processors then each processor gets  $N/P$  atoms. Each processor calculates the interactions for its “home” atoms [15]. This simplifies the calculation of bonded interactions, since bonded atoms can be placed on the same processor [31].

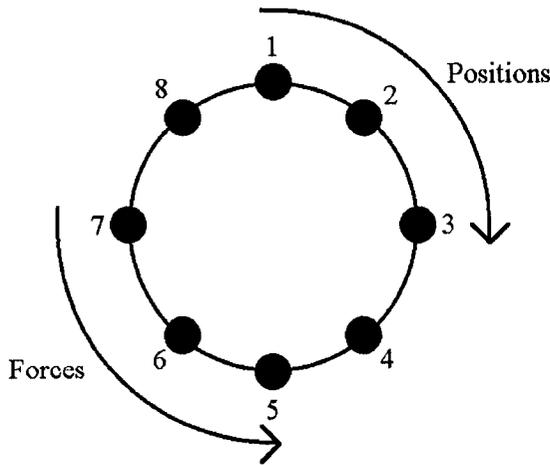


Figure 2.3: Ring Processor Topology Used by GROMACS

The non-bonded forces are forces between pairs of atoms, and depend on the distance  $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ . Let  $\mathbf{F}_{ij}$  be the force on atom  $i$  from atom  $j$ . From Newton’s third law

$$\mathbf{F}_{ij} = -\mathbf{F}_{ji}. \quad (2.31)$$

Thus it is only necessary to calculate one of  $\mathbf{F}_{ij}$  or  $\mathbf{F}_{ji}$ . GROMACS makes use of this fact in its communication scheme. Prior to the force calculation the positions of the particles are passed along half of the ring in one direction. This implies that for every atom pair  $i$  and  $j$ , either  $\mathbf{r}_i$  is passed to the home processor of atom  $j$  or  $\mathbf{r}_j$  is passed to the home processor of atom  $i$ . The processor with both  $\mathbf{r}_i$  and  $\mathbf{r}_j$  calculates the

force. Then the forces are passed back over half of the ring in the opposite direction. At the conclusion of this second communication each processor has all of the forces acting on its atoms. These forces are summed to give the net force acting on each atom, which is used in the leap-frog position update algorithm [31].

### 2.2.8 Energy Minimization

A function related to MD is energy minimization (EM). In an EM the potential energy of the system is reduced to a (local) minimum. This technique is used to reduce excessive forces when the system is far from equilibrium. It can also be used to remove all the kinetic energy from the system, which reduces thermal noise [17, Page 2].

The potential energy function of a molecular system has an extremely complex landscape in a large number of dimensions. It has one deepest point, the global minimum, and a very large number of local minima and saddle points. The dimensionality of the configurational space and the number of local minima is so high that it is impossible to sample the space at a sufficient number of points to obtain a complete picture of all the minima. No minimization method exists that guarantees the determination of the global minimum in any practical amount of time. However, given a starting configuration, it is possible to find the nearest local minimum. Nearest in this context means the minimum that can be reached by systematically moving down the steepest local gradient [17, Pages 5 – 6].

The steepest descent method uses derivative information to find the nearest local minimum in the potential energy. It takes a step in the direction of the negative gradient (the direction of the force), without any consideration of the history built up in previous steps. The step size is adjusted such that the search is fast but the

motion is always downhill. This is a simple and sturdy method, but its convergence can be quite slow, especially in the vicinity of the local minimum [17, Pages 5 – 6].

The algorithm begins with the coordinate vector  $\mathbf{r}$  and an initial maximum displacement  $h_0$ . The forces and potential energy are calculated. New positions are calculated by

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \frac{\mathbf{F}_n}{\max(|\mathbf{F}_n|)} h_n \quad (2.32)$$

where  $h_n$  is the maximum displacement and  $\mathbf{F}_n$  is the force, or the negative gradient of the potential  $V$ . The notation  $\max(|\mathbf{F}_n|)$  means the largest of the absolute values of the force components. The forces and energy are again computed for the new positions.

- If ( $V_{n+1} < V_n$ ) the new positions are accepted and  $h_{n+1} = 1.2h_n$ .
- If ( $V_{n+1} \geq V_n$ ) the new positions are rejected and  $h_{n+1} = 0.2h_n$

The algorithm stops when either a user specified number of force evaluations has been performed, or when the maximum of the absolute values of the force (gradient) components is smaller than a specified value  $\epsilon$  [17, Pages 33 – 34].

### 2.2.9 Other Features

GROMACS was developed to make use of advances in high performance computing. In addition to the features already discussed, GROMACS has several other interesting features.

- GROMACS was written in the ANSI C programming language. It can be built with only a C compiler. Since essentially all operating system kernels are written in C, compilers are available on just about all computer hardware. Thus GROMACS is extremely portable [33].

- The calculation and summation of forces takes place in loops within the GROMACS program. These “inner loops” are the most computationally intense part of MD. The inner loops of the GROMACS code were rewritten in FORTRAN to take advantage of better optimization capabilities in FORTRAN compilers over C compilers. On personal computers the inner loops have been supplanted with assembly loops for common processors such as Intel or AMD [32].
- The virial (Equation 2.15) is normally calculated in the inner loop of the non-bonded force calculation. This results in a large amount of time being spent on the calculation. For GROMACS the virial calculation was reformulated and moved out of the inner loop. This resulted in considerable savings in calculation and time [29] [41].
- The most expensive floating-point operation is the calculation of  $1/r$  from  $r^2$ . A special inverse square root function was designed for GROMACS to calculate  $1/r$  directly from  $r^2$ . An initial table lookup with  $r^2$  is used to get an approximate result  $a$  with twelve bits of accuracy. A single Newton-Raphson iteration defined by

$$\frac{1}{r} = \frac{a(3 - ar^2)}{2} \quad (2.33)$$

yields the full single precision value of  $1/r$ . A second iteration yields the double precision value [33]. Some interactions require  $1/r^2$  from  $r^2$  instead of  $1/r$ . This is calculated using a table lookup and the iteration [32]

$$\frac{1}{r^2} = a(2 - ar^2). \quad (2.34)$$

## 2.3 Previous MD Simulation

### 2.3.1 Monolayer and Bilayer Simulation Results

Most MD studies on membranes have looked at bilayers, since most biological membranes are in the form of bilayers. LS is a notable exception. Experiment on membranes, however, are usually performed on monolayers. Experimental apparatus, such as the Langmuir-Wilhelmy surface balance, are more suited for studying a single layer monolayer than a double layer bilayer. MD and experiment are bridged by the assumption that a bilayer can be thought of as two weakly coupled monolayers [16].

Justification for this link between monolayers and bilayers was provided by Chiu *et. al.* [42]. They studied a bilayer composed of dimyristoylphosphatidylcholine (DMPC). Chiu's group wanted to simulate a fluid phase hydrated lipid bilayer membrane. Previous simulations using constant volume or constant pressure boundary conditions could only achieve the fluid phase with artificial or arbitrary biases towards fluidity, such as unrealistic areas per lipid or reduced charges on the lipid. Chiu applied a constant surface tension boundary condition. A pressure of 1 atmosphere was applied in the normal direction to simulate laboratory conditions. The lateral pressure was chosen based on Equation 2.17 to give the experimentally observed surface tension. This surface tension was calculated for a monolayer. The results of the simulation showed that boundary conditions derived from a monolayer yielded a realistic simulation of a bilayer. The fluid phase was simulated successfully without any need for bias towards fluidity in the simulation. The constant surface tension boundary condition developed by Chiu was verified.

One of the most notable MD studies on bilayers was performed by Tieleman and Berendsen in 1996 [43]. They performed simulations on a bilayer of DPPC to study the effects of different force fields and boundary conditions. They compared

two different parameters sets for the Lennard-Jones interaction (Section 2.2.1), two different water models (SPC and SPC/E) and three different boundary conditions (constant pressure, constant surface tension and constant volume). They found that constant pressure and constant surface tension worked well with bilayers, but not constant volume. Imposing a constant volume led to poor results since it was difficult to obtain accurate experimental values for the dimensions of the system. The way water was modelled had different effects on the results, but neither model was found to be better than the other. There were trade-offs with each. SPC/E gave a better density, radial distribution function, self-diffusion constant and dielectric constant than SPC. However, the effective free energy of SPC/E was too low and the liquid state was thermodynamically too much favoured. Since solubility at the water/lipid interface was important, they recommended that SPC be used in simulations involving lipids. They also found a preferred set of Lennard-Jones constants. However, these constants predated the work by Berger [28], which they use today [19].

A lot of work has been recently been done on phospholipid monolayers. Different types of phospholipid monolayers were studied by Kaznessis *et. al.* [16]. They performed MD simulations on zwitterionic DPPC, on anionic dipalmitoylphosphatidylglycerol (DPPG) and on a mixture of the two. Their results with all three systems showed very good agreement with experimental results. These simulations were done for lipid densities in the loosely packed 55 to 80 Å<sup>2</sup>/lipid range.

Phase transitions in DPPC monolayers were studied by Knecht *et. al.* [44]. They used MD simulations to look at a DPPC monolayer at varying area densities. Their work was motivated by the recent discovery of a sharp transition in the order of the lipid chains at 1.10 nm<sup>2</sup>/molecule. Based on their simulations they attributed this transition to the onset of the gas-liquid coexistence region. This onset took the form of a rupture in the monolayer and the beginning of the formation of pores.

More recent studies have looked at one of the surfactant proteins in a monolayer. The peptide SP-B<sub>1-25</sub> was studied in a palmitic acid monolayer by Lee, Kandasamy and Larson [45]. The MD simulations determined that the residues Arginine 12 and 17 anchored the peptide in the monolayer by interacting with ionized palmitic acid molecules, while Tyrosine 7 and Glutamine 19 controlled the tilt of the peptide by interacting with the palmitic acid headgroups. A similar study was performed in a DPPC monolayer by Kandasamy and Larson [46], which found similar results. It was concluded that the polar and cationic amino acids of the peptides formed hydrogen bonds with the zwitterionic DPPC. Similar work has been done by Kaznessis *et. al.* [47] and Freites *et. al.* [48].

### 2.3.2 Force Field Development

The development and refinement of force fields is an ongoing process. Considerable work has gone into creating force fields and updating them with new experimental data.

Water has been the subject of MD since the early days of computer simulations, and new water models are still appearing. Force fields have been developed that give good results for bulk properties. However, it has proven difficult to successfully model the surface properties of water. In particular the most common force fields for water do not yield good results for surface tension [49].

Berger *et. al.* studied the Lennard-Jones parameters [28] used in the force fields. They noted that the existing force fields did not give the experimentally measured lipid density or heat of vaporization for lipid bilayers. An incorrect lipid density implied that other membrane properties calculated by the simulation, such as the area per lipid and order parameter, were not correct. They simulated a system of

pentadecane molecules, which represented the hydrophobic tails of the lipids. The Lennard-Jones parameters were adjusted to give the correct volume per lipid and heat of vaporization for the pentadecane. These new parameters gave the correct density and area per lipid in a bilayer simulation.

The Gromos force field [26] has been evolving since its creation. The latest version of Gromos is Gromos96. GROMACS incorporates a version of this force field, called ffG45a3. The ffG45a3 force field was created in an attempt to add lipid parameters to the Gromos96 force field. The 45a3 parameter set was an update to the 45a2 parameter set. Simulations with the 45a2 parameter set found that it was extremely successful for a variety of biological systems, but it yielded incorrect densities for simple aliphatic chains that contained more than six CH<sub>2</sub> groups [20]. The 45a3 force field has been tested and found to work well with lipids as well as proteins [20], [21].

Tieleman *et. al.* have another approach to force field development. Instead of creating new force fields they have taken different force fields for different systems and used them in the same simulation [19]. They took the united-atom lipid force field created with the Berger parameters [28] and used it with the all-atom OPLS-AA force field [27] to simulate peptides in a bilayer. Their results showed that this was a reasonable approach.

# Chapter 3

## Computational Procedure

### 3.1 Monolayer Creation

The lipid monolayers used in the simulations were obtained from several sources. DPPC and POPC bilayer structures were obtained from Peter Tieleman, along with the related topologies (molecular descriptions). A POPG bilayer was obtained from Mikko Karttunen.

The DPPC monolayer was taken from a DPPC bilayer [50] with 128 DPPC lipids. One leaflet was removed, leaving a monolayer with 64 DPPC. During the initial test simulation 19 lipids were observed to cross the periodic boundaries. These 19 DPPC were removed to create a monolayer with 45 DPPC.

A POPC bilayer was also obtained from the Tieleman website. Twelve POPC lipids were obtained from this bilayer, and were used to form a monolayer with 28 DPPC lipids. The 28 DPPC to 12 POPC ratio (7:3) was chosen since it gives similar results to lung surfactant [12].

At the start of the research a POPG lipid structure was not available. POPA was chosen as an alternative, since it has the same anionic charge as POPG. POPA lipids

were formed by modifying the heads of the POPC lipids. The topology for POPA was made by modifying the topology for POPC.

POPG lipids were later obtained. Twelve of these were used to form a monolayer with 28 DPPC. Later, a POPG bilayer and topology were obtained from the website of Mikko Karttunen [51]. A 64 lipid POPG monolayer was taken from this bilayer. The naming of the some of the POPG atoms in the mixed DPPC/POPG monolayer were changed to match the Karttunen topology.

## 3.2 Monolayer Preparation

The bilayers obtained from Tieleman and Kafttunen came with water molecules. These water molecules were removed, leaving a “dry” monolayer. An example of such a dry monolayer is shown in Figure 3.1. Note that some DPPC cross the periodic box boundaries, resulting in some carbon tails appearing as fragments.

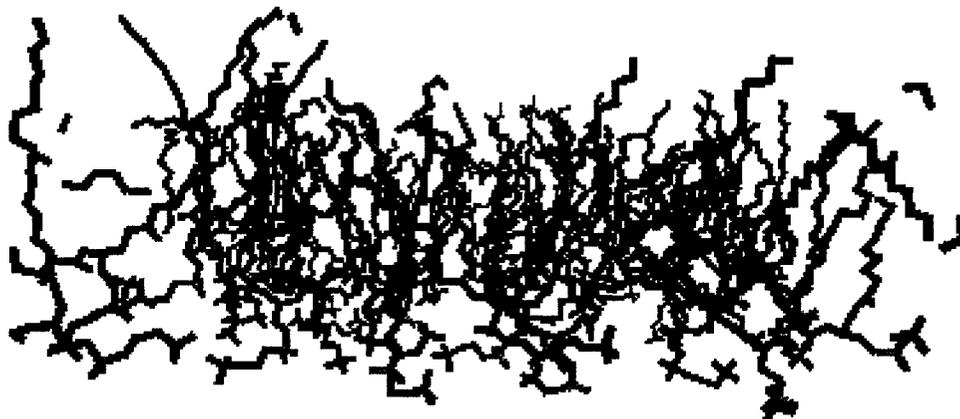


Figure 3.1: DPPC Lipids Arranged in a Monolayer

The monolayers were scaled to give a desired lipid density. For a monolayer in a

square simulation box the length of a side was

$$\text{Desired Length} = \sqrt{\text{Lipid Density} \left( \text{\AA}^2/\text{lipid} \right) \times \text{Number of Lipids} \text{\AA}}.$$

The actual x- and y-dimensions of the simulation box were included in the atomic position files. The necessary scale factors were

$$\text{Scale Factor} = \frac{\text{Desired Length}}{\text{Actual Length}}.$$

These scale factors were used in the GROMACS program editconf [52], which scaled the simulation box and the monolayer within it.

For the initial simulations scale factors were chosen to give lipid densities of 40, 60 and 80  $\text{\AA}^2/\text{lipid}$ . These scale factors were chosen to because they correspond to different stages in the respiratory cycle, from full expiration at 40  $\text{\AA}^2/\text{lipid}$  to full inspiration at 80  $\text{\AA}^2/\text{lipid}$ . Later, simulations with lipid densities of 45 and 50  $\text{\AA}^2/\text{lipid}$  were added. In the case of the DPPC monolayer the scale factors were calculated with the assumption that 5 DPPC would be replaced with a peptide with area 265  $\text{\AA}^2$ . Consequently, the actual lipid densities were 41.4, 45.9 50.3, 59.2 and 77.0  $\text{\AA}^2/\text{lipid}$ . Simulations with lipid densities of 55 and 70  $\text{\AA}^2/\text{lipid}$  were added after these to provide more data points on the surface tension isotherms.

The act of scaling the box stretches or contracts everything inside the box. The distances (bond lengths) between the atoms comprising the lipid molecules have a fixed length. When the lipids were stretched in the scaling the bond lengths were stretched. To correct for this a series of EM were run to allow the lipids to shrink or expand back to their equilibrium size.

The effect the scaling procedure had on the bond lengths is shown in Figure 3.2.

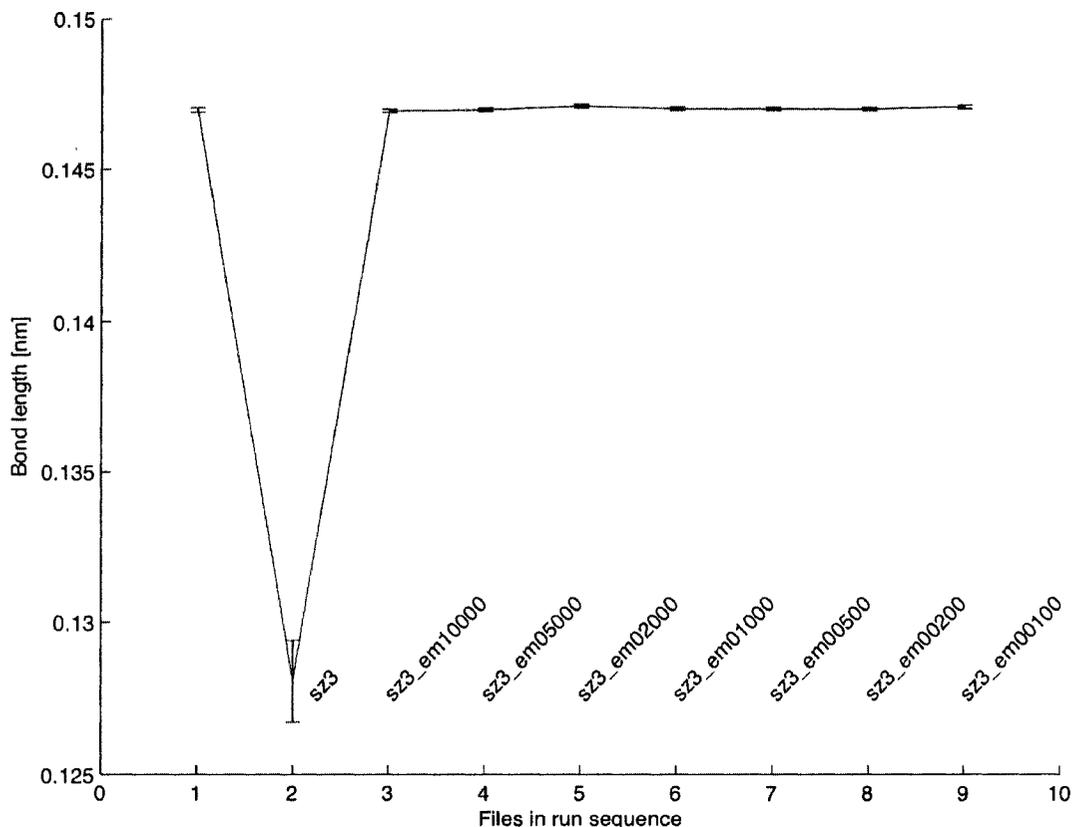


Figure 3.2: Bond Length During Energy Minimization Sequence

This figure comes from the scaling of the pure DPPC monolayer, which was scaled to  $59.2 \text{ \AA}^2/\text{lipid}$ . The figure looks at the average bond length between the nitrogen atom (N4) in the head group and one of the carbon atoms (C1) attached to it. The initial scaling compressed the bond lengths by a considerable amount. The EM allowed the bonds to relax back to their original lengths.

EM was performed using the steepest descent method. A step size of 0.02 nm was used. The series of tolerances was 10000, 5000, 2000, 1000, 500, 200 and 100  $\text{kJ mol}^{-1} \text{ nm}^{-1}$ .

Water molecules were added to the monolayer once the scaling was complete. Sin-

gle point charge (SPC) water molecules were added with the GROMACS program genbox [52]. The water molecules were added so as to fill the simulation box, as in Figure 3.3. Enough water molecules were added to give a water density of approximately  $1000\text{g}/\text{cm}^3$ . Since each lipid density required a different box size, each received a different number of water molecules. The water molecules on the tail side of the monolayer were removed, and a series of EM was performed to equilibrate the monolayer/water system. This series consisted of tolerances of 5000, 2000, 1000, 750, 500, 200 and  $100\text{ kJ mol}^{-1}\text{ nm}^{-1}$ .



Figure 3.3: Water Added to a Scaled DPPC Monolayer by Genbox

POPA and POPG are charged lipids. Each of their headgroups has a charge of -1

electronic charge ( $1.602 \times 10^{-19}$  C). These charges were neutralized by changing water molecules into sodium atoms. For each charged lipid a water molecule was chosen at random and changed. The positive and negative charges on the zwitterionic DPPC neutralized each other, and so sodium atoms were not needed for the PC lipids.

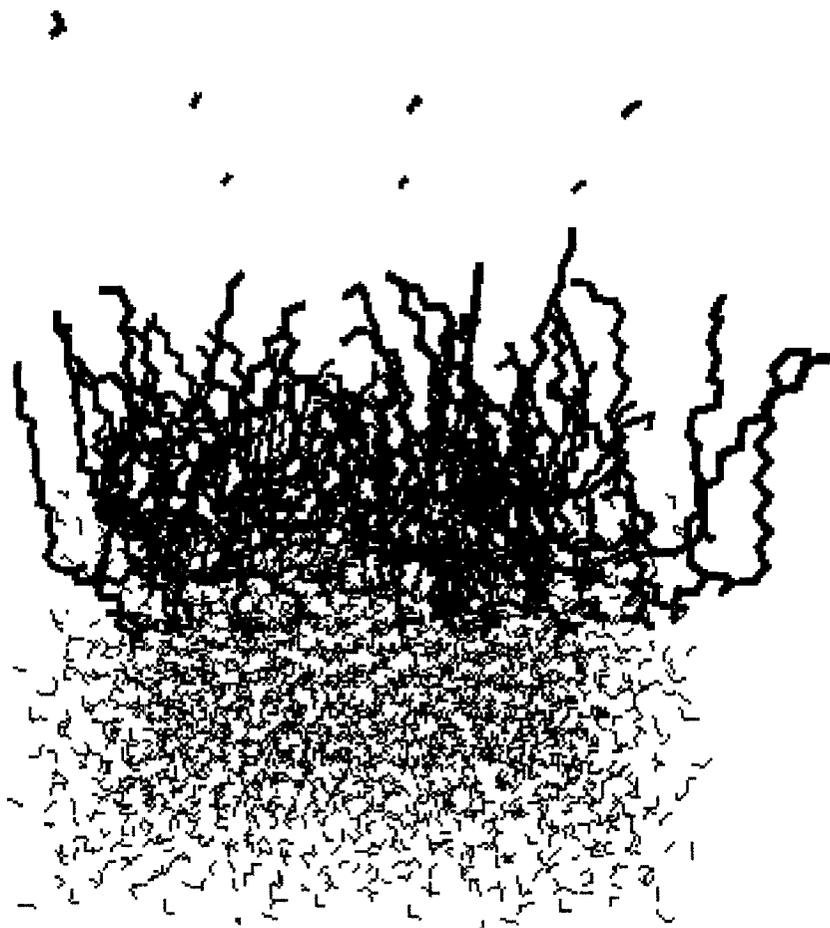


Figure 3.4: DPPC and Water Monolayer System with Displaced Water Molecules

The water molecules occupied the space from the monolayer headgroups to the simulation box boundary. Due to the periodic boundary conditions, it was possible for water molecules to cross over to the top of the box and interact with the tails, as shown in Figure 3.4. To prevent this, the box boundary was moved. The lower boundary was extended 100 Å, leaving a gap between the water molecules and the

lower boundary. While it was possible for water molecules to traverse the gap and cross the boundary, such occurrences were very rare.

The final step in preparing the monolayer was to run a short, 10 ps MD simulation using position restraints (PR). The lipids were held in place to allow water molecules to move into the area around the headgroups. PR MD was performed using the same run parameters as the full, unrestrained MD simulation.

The final monolayer configuration appears in Figure 3.5. The lipids form a monolayer that lies in a layer of water. There is vacuum around the lipid tails in place of air, which would be present in lung surfactant. There is also vacuum below the water layer.

## Monolayer Configuration

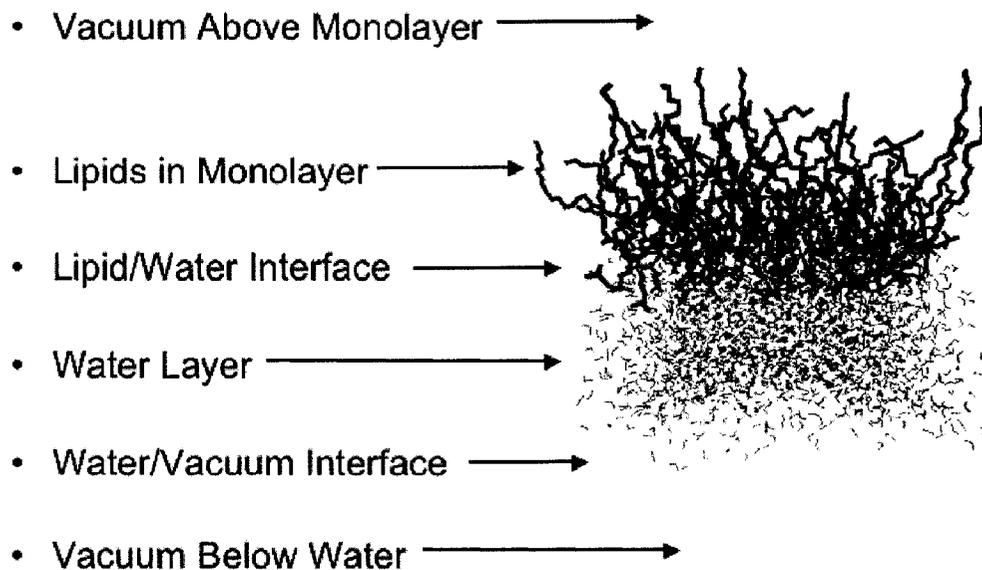


Figure 3.5: Final Monolayer with Water Configuration Used in MD

## 3.3 MD Simulation

### 3.3.1 Force Field

The simulations used the modified ffgmx GROMACS force field [53]. The main components of this force field were the Berger lipid force field [28] incorporated into the standard ffgmx force field, which contained parameters taken from the Gromos87 force field [26]. This force field was set up by Peter Tieleman [19].

This force field was chosen since it was optimized for lipids. The GROMOS parameters would allow for interactions with proteins at a later date.

The SPC water molecules used the standard SPC force field that came with the GROMACS package .

### 3.3.2 Parameters

MD Simulations were performed on five different monolayers:

- Pure DPPC with 45 lipids
- Pure POPG with 64 lipids
- Mixed DPPC/POPA with 28 DPPC lipids and 12 POPA Lipids
- Mixed DPPC/POPC with 28 DPPC lipids and 12 POPC Lipids
- Mixed DPPC/POPG with 28 DPPC lipids and 12 POPG Lipids

The simulations used a time step of 0.002 ps. The simulations were run for 5,000,000 time steps, giving a simulation time of 10 ns. Position, velocity and energy information was written to files every 500 time steps, or 1 ps. Checkpointing was done

every 1000 time steps. In case of system failure the checkpointed data could be used to restart the simulation, instead of rerunning the simulation from the beginning.

The bonds between atoms were converted to constraints. Otherwise the systems were unconstrained. The LINCS constraint algorithm [35] was used with fourth order expansion and one iteration to correct for rotational lengthening. Water molecules were constrained with the SETTLE [36] algorithm.

At startup the initial velocities were generated from a Maxwell distribution at temperature 300K, which was the default setting for GROMACS. The bond constraints were applied at startup.

The grid neighbour search method was used, with a cutoff distance for the short-range neighbor list of 0.9 nm. The neighbour list was updated every 10 time steps. Periodic boundary conditions were used.

For electrostatic interactions the Fast Particle-Mesh Ewald [40] method with a real space cutoff of 0.9 nm was used. PME used fourth order (cubic) interpolation with a grid spacing of 0.15 nm and tolerance of  $10^{-5}$ . The 3dc geometry was used [54]. This geometry gave a pseudo-2d summation for a system with slab geometry in the x-y plane [17, Page 143]. The van der Waals interactions used the cutoff method with distance 1.4 nm. These parameters were recommended in an email to the GROMACS mailing list from Berk Hess [55], who is one of the GROMACS developers.

The system was coupled to a temperature bath with the Berendsen algorithm [34]. Each molecule type (lipid and water) was coupled separately. The initial simulations of DPPC and DPPC/POPA and DPPC/POPC used a coupling temperature of 300K. The later DPPC, DPPC/POPC, DPPC/POPG and POPG simulations used a coupling temperature of 310K, or 37°C. Pressure coupling was not used.

### 3.3.3 Computer Resources

Computing resources were supplied by ACEnet, the Atlantic Computational Excellence Network. Most of the simulations were performed on a 164-processor Sun V60 cluster known as Chorus [56]. The Chorus cluster consists of two dual processor master nodes (each composed of two 3.0 GHz Intel Xeon processors with 3 GB RAM and hyperthreading enabled) and 80 dual processor slave nodes (each composed of two 2.8 GHz Intel Xeon processors with 2 to 3 GB RAM and hyperthreading disabled). The processors communicated over 1000 Mbps Base-T Ethernet.

Chorus had installed on it GROMACS version 3.2.1. This was the version in use at the start of this research project. This version was used for most of the simulations, since GROMACS version 3.3 was released towards the end of this research project. Chorus used MPICH 1.2.5.2 [57] software for communication between processors. Due to a problem between GROMACS, MPICH version 1 and Ethernet, parallel (multiple processor) computing could not be used [58], and all simulations were performed in serial on one processor. More will be said on this in Section 4.7.

Towards the end of this research project ACEnet brought online new computing resources. One of these resources was Mahone, a 128 core Myrinet/4-way cluster composed of 32 dual socket SunFire x4100 nodes populated with 2.6 GHz dual-core Opteron 285 SE processors and 4 GB RAM per core [59]. The latest version of GROMACS, 3.3.1, was installed on Mahone. Mahone used MPICH-GM for processor communication. Since the processors were connected with Myrinet instead of Ethernet, it was possible to use parallel GROMACS on Mahone.

### 3.3.4 Simulations

The initial simulation run was of a DPPC monolayer. The initial plan called for five lipids to be replaced with a SP-B peptide with cross-sectional area of  $265 \text{ \AA}^2$ . This lipid plus peptide system was to be studied at lipid densities of 40, 45, 50, 60 and  $80 \text{ \AA}^2/\text{lipid}$ . The DPPC without peptide monolayer was scaled and simulated with the box dimensions that would be used with the peptide. The actual lipid densities were 41.4, 45.9, 50.3, 59.2 and  $77.0 \text{ \AA}^2/\text{lipid}$ . The simulations were performed with temperature coupling to 300K, which was the GROMACS default coupling temperature.

The second and third sets of simulations were with the DPPC/POPA monolayer and the DPPC/POPC monolayer. These monolayers, like DPPC, were simulated with temperature coupling of 300K. The monolayers were simulated at lipid densities of 40, 45, 50, 60 and  $80 \text{ \AA}^2/\text{lipid}$ . The presence of a peptide was not taken into account.

The main role of lung surfactant is to lower surface tension at the air/water interface. Hence the surface tensions of the simulated monolayers was an important calculation. The surface tensions calculated from the initial simulations did not agree with initial expectations, and a number of modifications to the simulation set up were made. These modifications will be discussed in the Chapter 4.

The only change to the procedure was to use a temperature coupling of 310 K (Section 4.2). Monolayers of DPPC/POPC, DPPC/POPG and POPG were simulated at this temperature. The POPG replaced the POPA, which while similar to POPG is not found in LS. Simulations were performed at lipid densities of 40, 45, 50, 55, 60, 70 and  $80 \text{ \AA}^2/\text{lipid}$ . The 55 and  $70 \text{ \AA}^2/\text{lipid}$  simulations were added to provide more data points on the resulting surface tension isotherms, which were compared to experimental results.

All of the simulations up to this point had been performed on Chorus. The DPPC monolayer was then revisited, and the simulations were performed on Mahone. Simulations were performed for actual lipid densities of 40, 45, 50, 55, 60, 70 and 80 Å<sup>2</sup>/lipid. The monolayer systems were coupled to a temperature bath at 310K. Simulations were performed in parallel with 16 processors.

The output from the simulations was analyzed in two ways. GROMACS routines were used to extract and analyze calculated quantities such as energy, pressure, temperature and surface tension. The data was also analyzed and plotted with the Matlab software package. The trajectory of the simulation (positions of the atoms over time) was viewed with the VMD visualization package [60].

## Chapter 4

# Optimization of the MD Procedure

The surface tensions calculated from the initial MD simulations gave some unexpected results. It was expected that the surface tensions would be similar to what was seen from experiments [61]. This was true to a large extent for loose packing densities (55 to 80 Å<sup>2</sup>/lipid). For tightly packed monolayers (40 to 50 Å<sup>2</sup>/lipid) the calculated surface tensions were considerably different. In particular, negative surface tension values were calculated. At the time this was thought to indicate that something was wrong with the simulations.

A number of modifications to the basic MD procedure were considered to see if the negative surface tensions could be eliminated. These modifications involved monolayer size, temperature and pressure coupling, simulation parameters and force fields. The modifications yielded important information as to how changes in the simulation parameters affected the simulations.

No modification was found to eliminate the negative surface tensions. After all of the simulations were completed the negative surface tension issue was reconsidered. The tightly packed monolayers produced large lateral pressures, which mathematically accounted for the negative surface tensions. Also, these monolayers had a

non-flat lipid/water interface. Surface tension calculations usually assume a flat interface. The discrepancy between simulation and experiment was therefore no longer considered to be a serious concern. This is discussed further in Chapters 5 and 6.

## 4.1 Initial Results

The initial surface tension measurements are shown in Figure 4.1. The details regarding how these surface tensions were calculated are discussed in the Section 5.4. The monolayers simulated in these initial runs were DPPC, DPPC/POPA and DPPC/POPC.

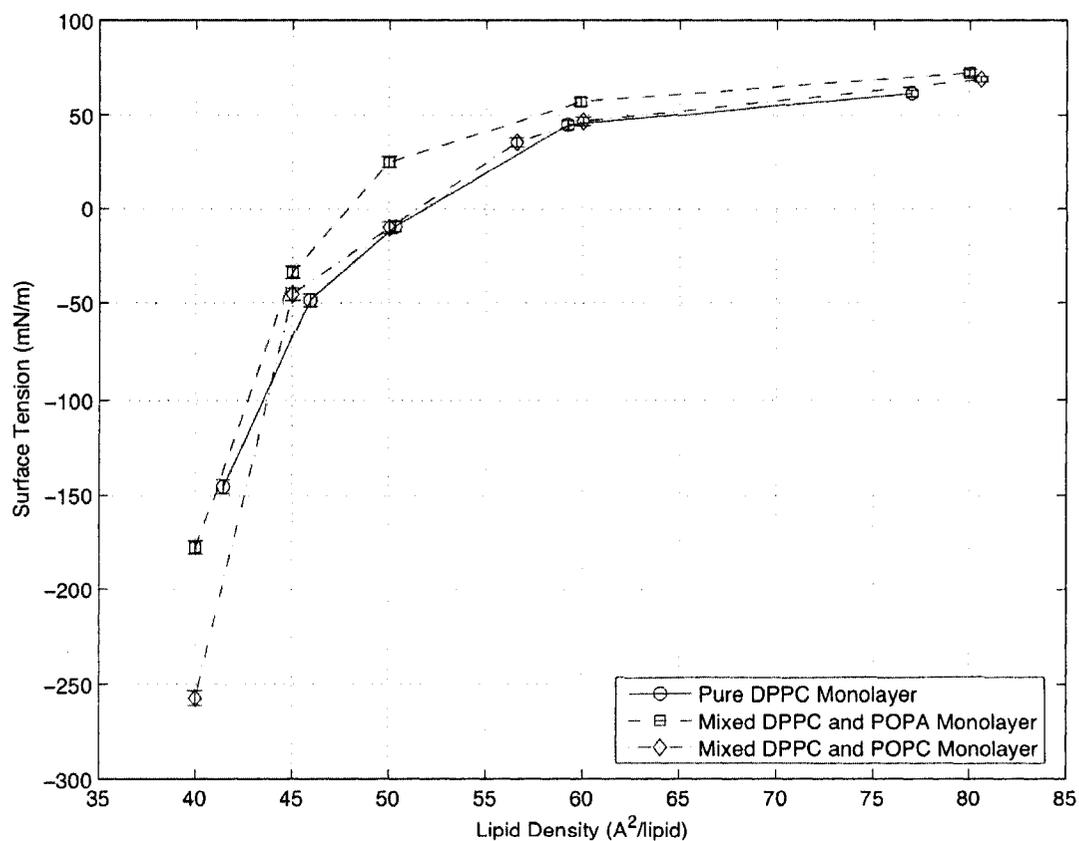


Figure 4.1: Initial Surface Tension Results

In the loosely packed monolayer range (55 to 80  $\text{\AA}^2/\text{lipid}$ ) the surface tensions

were between 0 and 72 mN/m, which was the expected range. The surface tension decreased gradually as packing was increased. The fall in surface tension increased in the tightly packed region (40 to 50 Å<sup>2</sup>/lipid). The surface tensions decreased at a rapid rate, and very quickly took on negative values.

## 4.2 Coupling Temperature

The initial simulations were coupled to a temperature bath at 300 K. This was the default temperature that came with the GROMACS program. Lung surfactant in the body is kept at a temperature of 37°C, or approximately 310 K. The simulations with a mixed DPPC/POPC monolayer were repeated with temperature coupled to 310 K to see if this would have any effect on the surface tensions.

The surface tensions calculated for the two temperatures are shown in Figure 4.2. The temperature difference did have a small effect. Surface tensions at 300 K were greater than surface tensions at 310 K, but only by several mN/m. The curves, or isotherms, of surface tension as a function of lipid density were very similar. Both isotherms fell into the negative surface tension range.

The behaviour of the lipids in the monolayers over time for both temperatures was observed. There were no noticeable differences in lipid behaviour between the two temperatures. Lipid behaviour will be discussed in more detail in the next chapter.

The two coupling temperatures show that temperature has only a small effect on surface tension calculations. Changes in the coupling temperature changed the height of the surface tension isotherm, but did not change the shape.

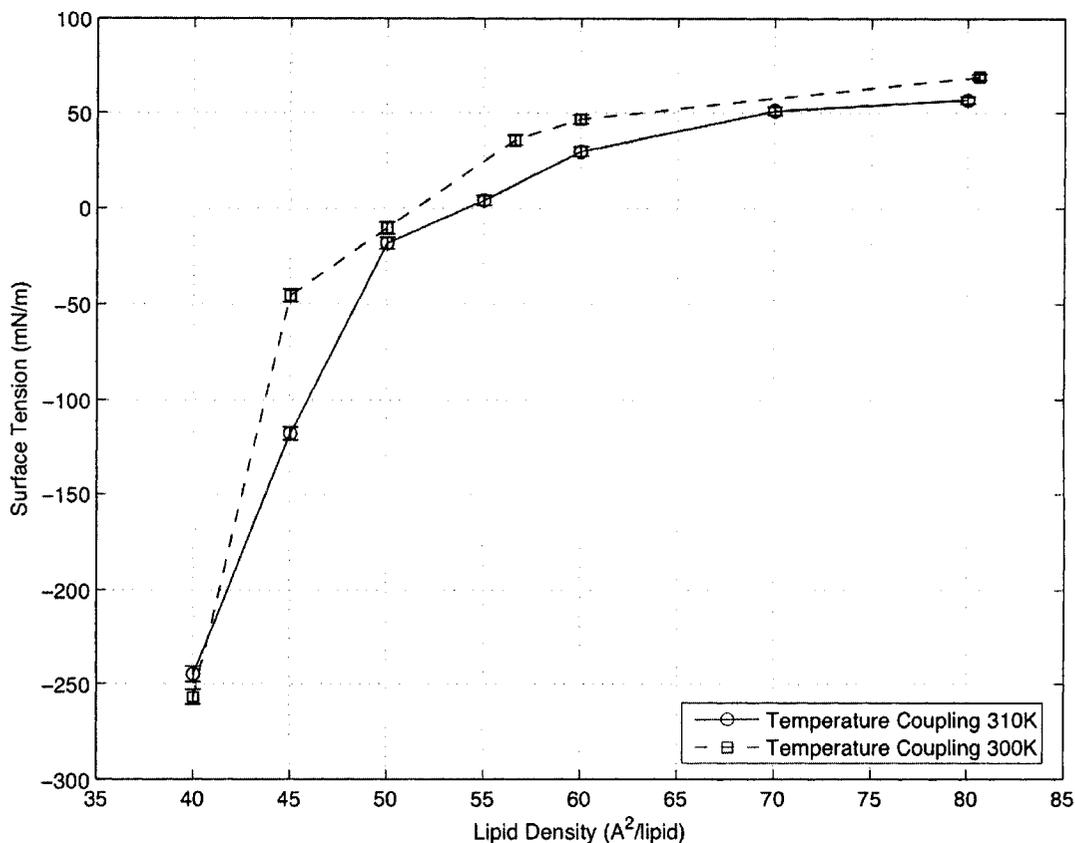


Figure 4.2: Surface Tension of a Mixed DPPC/POPC Monolayer at Different Coupling Temperatures

### 4.3 Pressure Coupling

The MD simulations were performed using what is known as an NVT ensemble. In an NVT ensemble the number of atoms ( $N$ ), volume ( $V$ ) and temperature ( $T$ ) are kept constant throughout the simulation. In such an ensemble pressure is allowed to fluctuate. Studies by Tieleman and Berendsen [43] suggest that a better method would be to keep pressure constant and allow volume to fluctuate. This was not appropriate for this study, since we must restrict volume to maintain a constant lipid density. However, pressure coupling can be applied anisotropically. A test simulation

was performed, with pressure coupling only in the  $z$  direction to keep the lipid density constant.

The Berendsen pressure coupling algorithm was used. It was applied using the semiisotropic type. This type applied the coupling isotropically in the  $x$  and  $y$  directions but differently in the  $z$  direction. Compressibility was set to zero in the  $x$  and  $y$  directions, and to the compressibility of water ( $4.5 \times 10^{-5} \text{ bar}^{-1}$ ) in the  $z$  direction. A reference pressure of 1.0 bar in all directions and time constant of 0.5 ps were used. The temperature coupling was to 300 K.

The test simulation was performed on the DPPC monolayer with lipid density of  $41.4 \text{ \AA}^2/\text{lipid}$ . The simulation was to run for 10 ns (10 000 ps), but crashed after 2874 ps. The simulation was restarted, but crashed again at 3546 ps. The run was examined with VMD. At around 1400 ps the monolayer and water started to drift upwards in the positive  $z$  direction. This was due to the upper  $z$  boundary of the box increasing. A plot of the box height versus simulation time is given in Figure 4.3. The box height appeared to increase at an exponential rate.

An examination of the pressure tensor showed that the diagonal component  $ZZ$  was  $16 \pm 3 \text{ bar}$ . This large pressure was what caused the box expansion. In simulations without pressure coupling the  $ZZ$  component was usually around zero (Section 5.4).

A second simulation was attempted. In this simulation the reference temperature in the  $z$  direction was set to 0.0 bar. This simulation crashed after 1000 ps, and a change in box height was observed from the beginning of the simulation.

The original pressure coupling parameters were applied to a DPPC monolayer with lipid density  $59.2 \text{ \AA}^2/\text{lipid}$ . This simulation ran for 5905 ps before crashing. There was some change in box height, shown in Figure 4.4. This change was much less than for the tightly packed monolayer.

The average  $ZZ$  pressure tensor component was  $1 \pm 1 \text{ bar}$ . The slightly greater

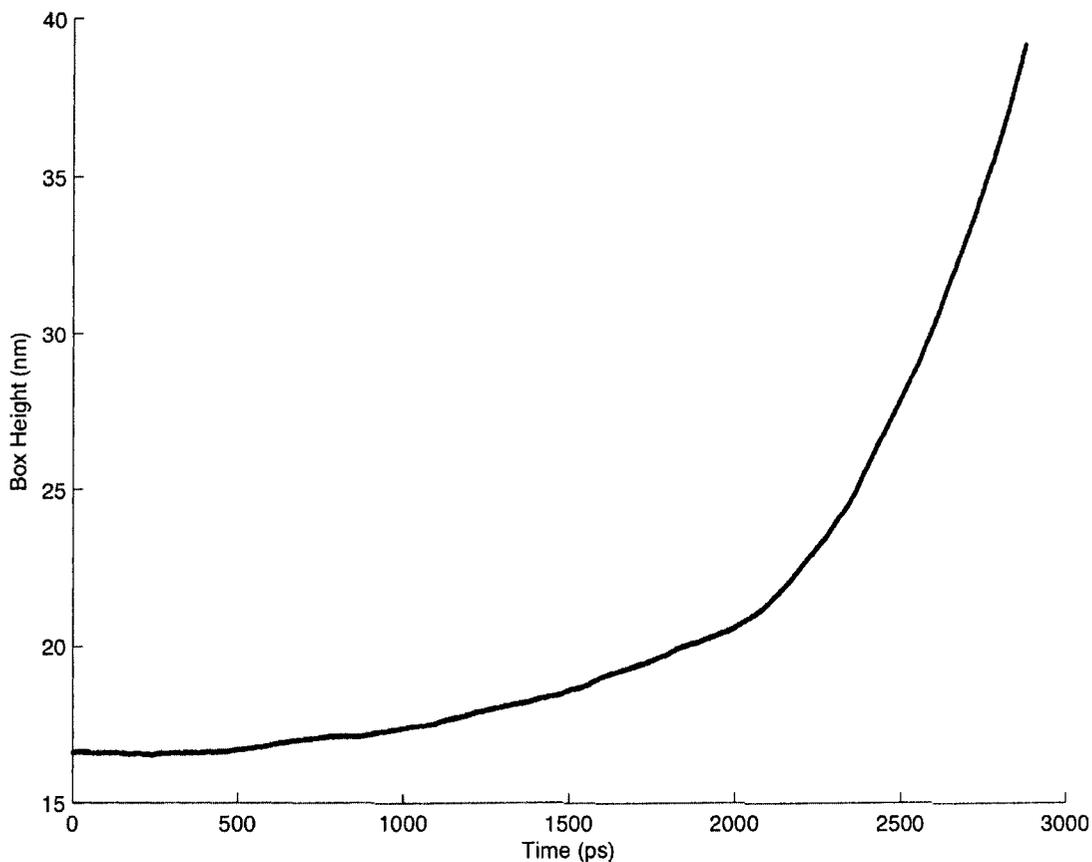


Figure 4.3: Simulation Box Height During MD Simulation with Pressure Coupling and Lipid Density  $41.4 \text{ \AA}^2/\text{lipid}$

than zero pressure in the z direction resulted in the small positive change in box height.

It was decided not to use pressure coupling in the simulations. Limiting the coupling to one direction resulted in an increase in box size that did not correspond to any activity that would be seen in LS. Also, since the surface tension calculation is based on pressures (Equation 2.17), allowing the ZZ pressure component to fluctuate while holding XX and YY constant might not lead to an accurate surface tension for the system.

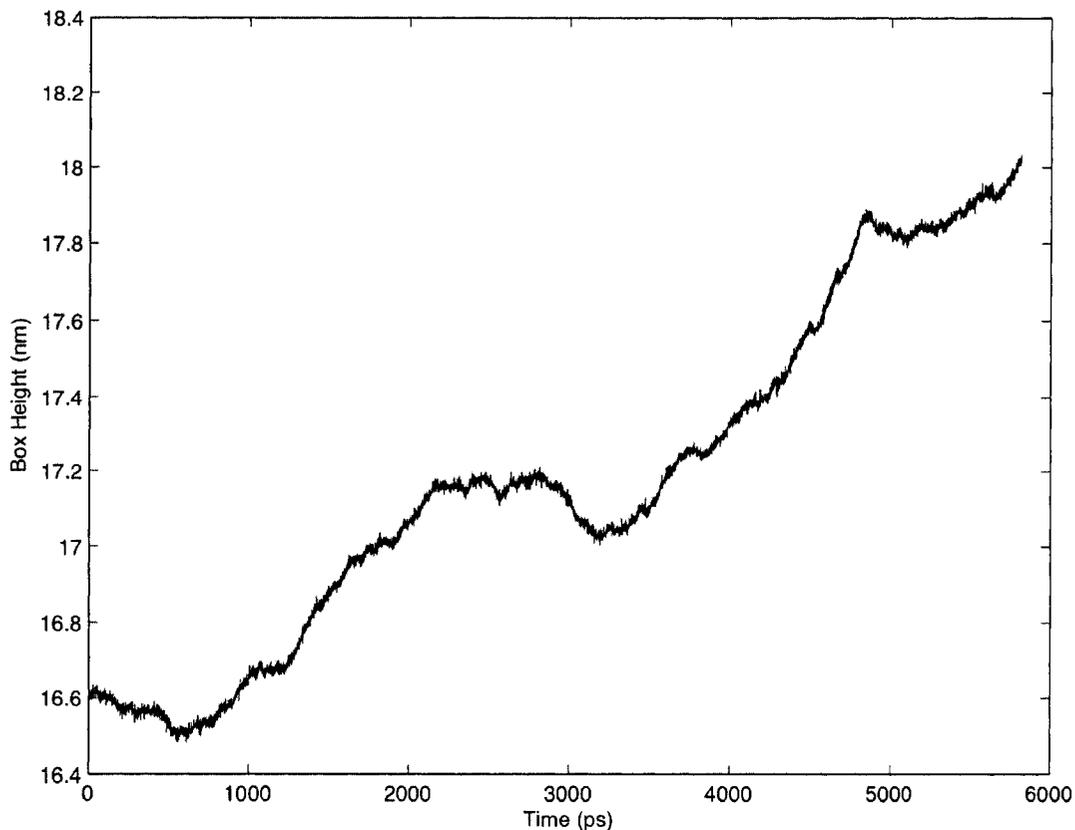


Figure 4.4: Simulation Box Height During MD Simulation with Pressure Coupling and Lipid Density  $59.2 \text{ \AA}^2/\text{lipid}$

## 4.4 Electrostatic Parameters

The particle-mesh Ewald method was used to calculate long range electrostatics. PME used fourth order (cubic) spline interpolation with a grid spacing of 0.15 nm. The cut-off for Coulomb interactions was set equal to the cut-off for the short range neighbour list, 0.9 nm. Van der Waals interactions were cut-off at 1.4 nm.

The choice of PME parameters had an effect on the simulation. The GROMACS manual recommends that the Coulomb cut-off should be the same as the neighbour list cut-off when using Ewald [17, Page 139]. For Coulomb, neighbour list and van

der Waals interactions the manual recommends a cut-off distance of 0.9 nm. It also recommends a grid spacing of 0.12 nm, which with the other parameters should give electrostatic energies accurate to about  $5 \times 10^{-3}$ . However, increasing the grid spacing to 0.15 nm should not increase the error too greatly [17, Page 140].

To test the effect of the PME parameters a series of simulations were performed on the DPPC monolayer with the GROMACS manual PME parameters. The new parameters were a van der Waals cut-off of 0.9 nm (instead of 1.4 nm) and a fourier spacing of 0.12 nm (instead of 0.15 nm). The simulation was temperature coupled to 300 K. A simulation was also performed on the water layer (lipids removed) from the  $41.4 \text{ \AA}^2/\text{lipid}$  monolayer, to determine the surface tension at the water/vacuum interface (see Section 5.4). The results of the new parameters are compared to the results of old parameters in Figure 4.5.

The new parameters gave a surface tension isotherm that was substantially different from the original. The new parameters gave a steeper decline in the loosely packed region ( $60 - 80 \text{ \AA}^2/\text{lipid}$ ) and a more gradual decline in the tightly packed region ( $40 - 60 \text{ \AA}^2/\text{lipid}$ ). The new isotherm had a shape that was substantially different from experimental curves [61]. The calculated surface tensions were lower, and the isotherm crossed into the negative region at a larger area per lipid, around  $61 \text{ \AA}^2/\text{lipid}$ .

Changing the parameters affected the surface tension calculation significantly. The new parameters decreased the van der Waals cut-off, leading to larger errors. It is possible that increasing the cut-offs could give better results. This is something that should be considered in future work.

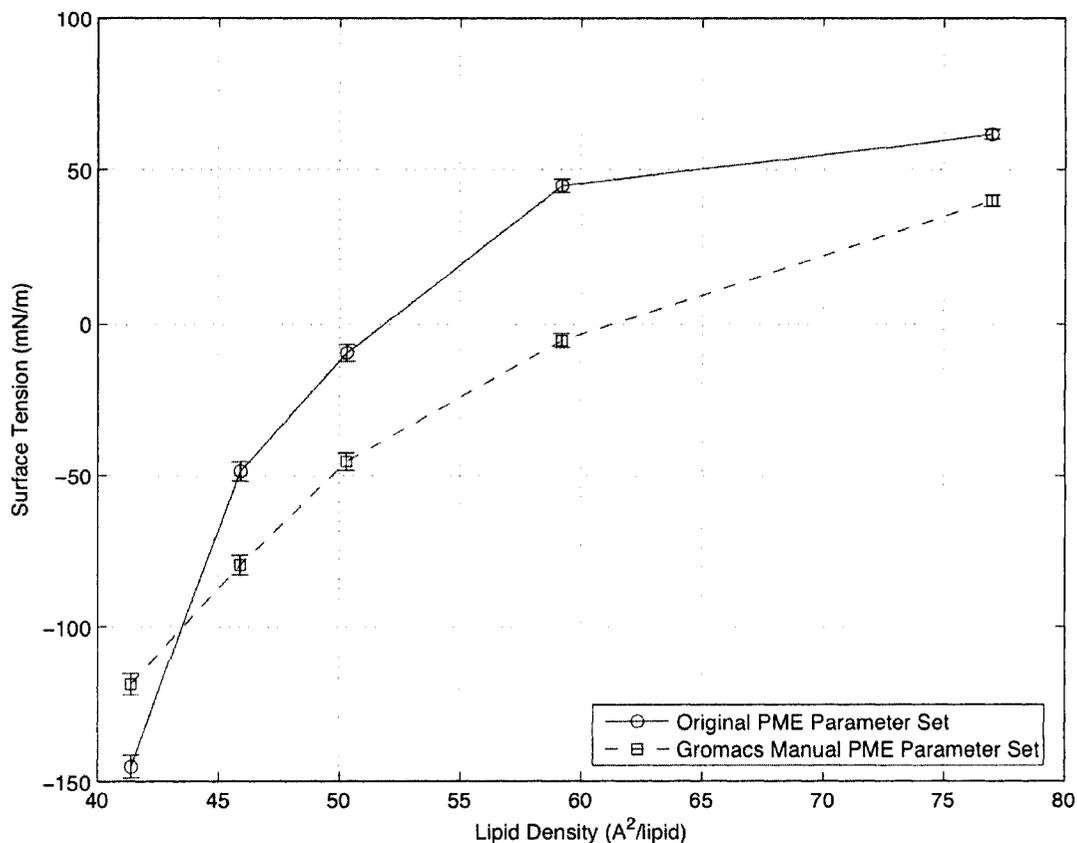


Figure 4.5: Surface Tension Calculations for Two Different Sets of Particle-Mesh Ewald Electrostatic Parameters

## 4.5 Force Fields

The selection of a force field may be the most important consideration in an MD simulation. For the simulations a mix of force fields was used. The water molecules used the SPC force field. Sodium ions, when needed to neutralize charged lipids, used GROMACS' ions force field. The lipids were simulated with the modified ffgmx force field (Section 3.3.1). There were other choices available, and some of these were considered.

The GROMACS developers have made available a set of benchmarking examples.

One of these benchmarks is a DPPC bilayer with its own force field. This force field uses the Berger parameters [28], but is missing one of the dihedral structures [62]. This problem was not corrected since the force field was only used for benchmarking.

The ffG43a2x force field was a modification of the Gromos96 43a2 force field [63]. It was created by Jakob Wohlert as a diploma project [64] to add lipid parameters to the Gromos96 force field. It is an informal work that is not officially endorsed [64].

The ffG45a3 force field was another attempt to add lipid parameters to the Gromos96 force field. The 45a3 parameter set was an update to the 45a2 parameter set. It has been tested and found to work well with lipids as well as proteins [20], [21].

These force fields were tested with the DPPC monolayer, using the lipid densities of 41.4, 45.9 50.3, 59.2 and 77.0 Å<sup>2</sup>/lipid and original temperature coupling of 300 K. The benchmark force field had its own topology for DPPC. The Gromos96 force fields did not have the necessary topology, so the Tieleman topology files were used.

The behaviour of the lipids in the monolayers under the new force fields was observed to be identical to the behaviour under the original force field. The surface tensions calculated for these force fields are shown in Figure 4.6. The benchmark force field gave similar results to the original fgm<sub>x</sub> force field. The ffG43a2x and ffG45a3 force fields gave results that were absolutely identical, and only ffG43a2x was plotted. In the loosely packed lipid region these Gromos96 force fields were the same as the other force fields. In the tightly packed lipid region they calculated a slower decline in the surface tension, but still took on negative values.

There are several possible reasons why ffG43a2x and ffG45a3 gave identical results. It may have been that the Tieleman lipid parameters in the topology were used for lipid-lipid interactions, and the Gromos96 parameters only came into play in interactions between lipids and water. It is also possible that the Gromos96 lipid parameters were used, and that both force fields had been parameterized identically.

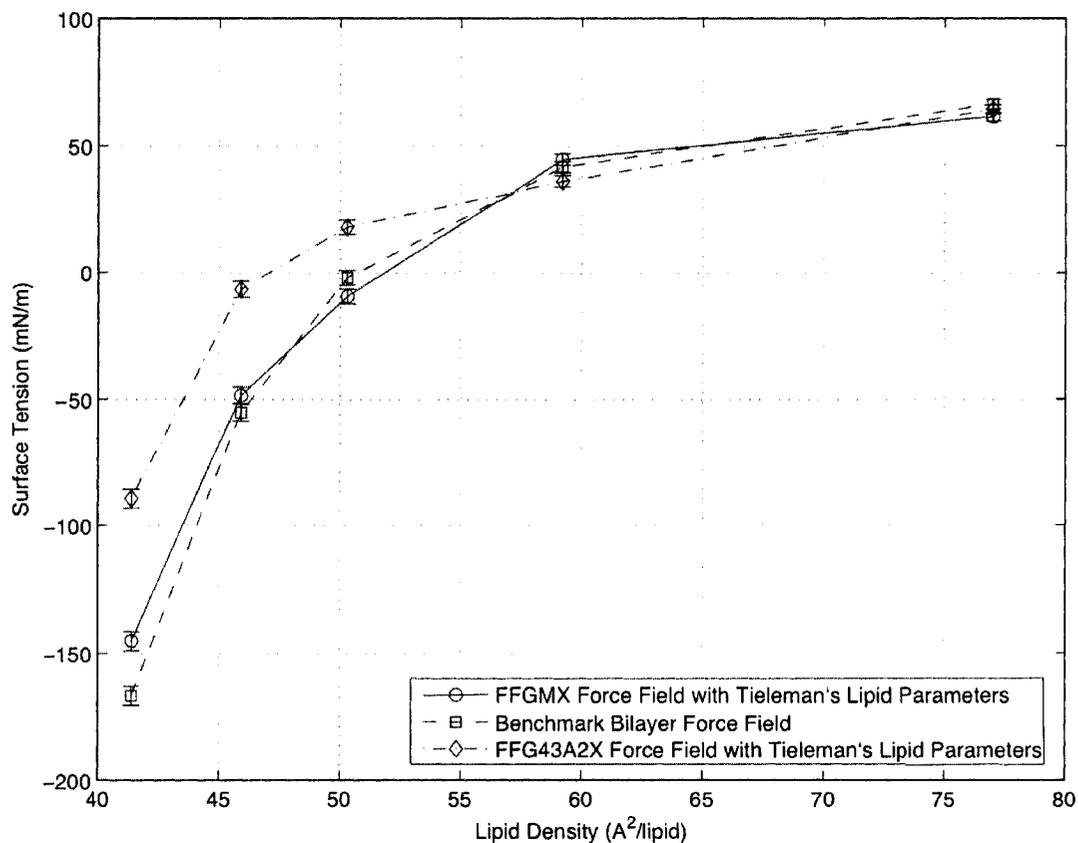


Figure 4.6: Calculated Surface Tensions for Different Force Fields

The next step at this point was to perform the simulations using only the ffG45a3 or ffG43a2x force fields. To this end the Tieleman force field file that contained the structural data on the DPPC molecule was modified for use with ffG43a2x. This file was needed since the ffG43a2x force field did not have the structural data. A test simulation was performed on a DPPC monolayer with lipid density  $59.2 \text{ \AA}^2/\text{lipid}$ . The lipids behaved as they did under the other force fields. The calculated surface tension was  $107 \pm 2 \text{ mN/m}$ , which was much higher than from the other simulations.

It was decided to remain with the Tieleman force field. The Tieleman force field has been used extensively and has been found to give good results. The necessary

topologies for all lipid types are available, and the monolayer systems were initially developed for use with this force field.

More work on force fields is required, and not just for lipids. Future plans call for the inclusion of proteins in the simulations. The original ffgmx force field has deprecated. Its protein parameters are based on the old Gromos87 force field, and it is recommended that they not be used. Since these same protein parameters are used in the modified ffgmx, a new force field will be needed for protein simulations. Tieleman has already started work on incorporating the Berger lipid parameters with a more recent protein force field [19].

## 4.6 Simulation with a Bilayer

The monolayers used in the simulations were small, containing 40 to 64 lipids. The periodic boundary conditions were used to give an approximation of a larger monolayer. However, the periodic systems were only mirrors of the original system. There may not have been enough lipids in the original system to simulate all of the actions, such as squeeze-out, that take place in LS.

The GROMACS benchmarking bilayer system consisted of 1024 DPPC lipids with 23 water molecules per lipid. A bilayer is composed of two monolayers. Thus a large monolayer consisting of 512 DPPC lipids was available. The lipid density of the monolayer/bilayer was  $63.3 \text{ \AA}^2/\text{lipid}$ .

The main obstacle to using such a large monolayer was the additional time that would be needed to run the simulations. A test simulation was performed with the original bilayer. This simulation used the same force field as the small monolayers. It was run for only half the time since the bilayer had twice as many lipids as a monolayer. Also, the long range electrostatic interactions were calculated with a

simple and less accurate cut-off instead of the more computationally intensive PME.

The 5 ns simulation with the large bilayer required over 33 days to finish. This was with a simplified calculation method for electrostatic interactions. It was decided that the long simulation times were too impractical, and simulations with the large monolayer were not performed. However, ACEnet has recently brought online a new computer capable of running GROMACS in parallel. The time savings from parallel computing would make the large monolayer practical. Unfortunately, there was no time to implement this.

## 4.7 Serial vs. Parallel

The GROMACS software can be run in parallel on computer systems which contain multiple processors. When run in parallel the calculations for force, position, etc. are split between a number of processors. For this to work, GROMACS must be run within a “wrapper” program which handles the communications between processors. GROMACS, the wrapper program and the system hardware work together to spread the work between the processors.

Running simulations in parallel offers several advantages. Simulations complete more quickly, allowing more to be performed. Very large systems can be simulated in a feasible amount of time. Cut-off distances can be increased, leading to increased accuracy in a shorter simulation time.

Most of the work with GROMACS was done on a 164-processor Sun V60 cluster called Chorus. The processors on Chorus were connected with 1000 Mbps Base-T Ethernet [56]. The wrapper program Chorus used for processor communication was MPICH v1.2.5.2 [57]. There is a known bug with GROMACS when MPICH v1 is used to communicate over ethernet [58]. This bug causes a simulation to crash if

more than five processors are used.

Simulations were attempted with the DPPC monolayer using the run parameters used for the serial simulations. The test simulations used 1, 2, 4, 6, 8 and 16 (maximum allowed) processors. The simulations with 6 to 16 processors very quickly crashed. This was consistent with the bug.

In the case of 2 and 4 processors another problem arose. It was found that the PME calculations did not parallelize well. A 1 ns simulation with PME on one processor took 43671 seconds to complete. When the simulation was run on four processors it took 134664 seconds, or about three times as long. This problem with parallelizing PME is also known [65].

A solution for the problems with parallel GROMACS on Chorus was not found. However, in the fall of 2006 ACEnet brought online three new multiple processor machines. GROMACS was installed on one of these machines, Mahone. Mahone used Myrinet-2000 instead of Ethernet for processor communications. The use of Myrinet meant that the bug would not be an issue. Also, Mahone used the latest version of GROMACS, 3.3.1, which had an improved version of PME that was designed to be used in parallel [66]. Thus it was possible to run GROMACS simulations in parallel on Mahone.

A benchmarking test was run on Mahone to determine an optimum number of processors to use with GROMACS. The test used the GROMACS benchmark bilayer and force field. The benchmark simulations were run for 10 ps using the run parameters supplied with the force field. The run times are shown in Figure 4.7.

The run times initially fell very rapidly as more processors were used. This fall leveled off and the run times reached a minimum with 48 processors. As more processors were added the run times increased at a steady rate.

Parallel performance is usually measured in terms of its speedup and efficiency.

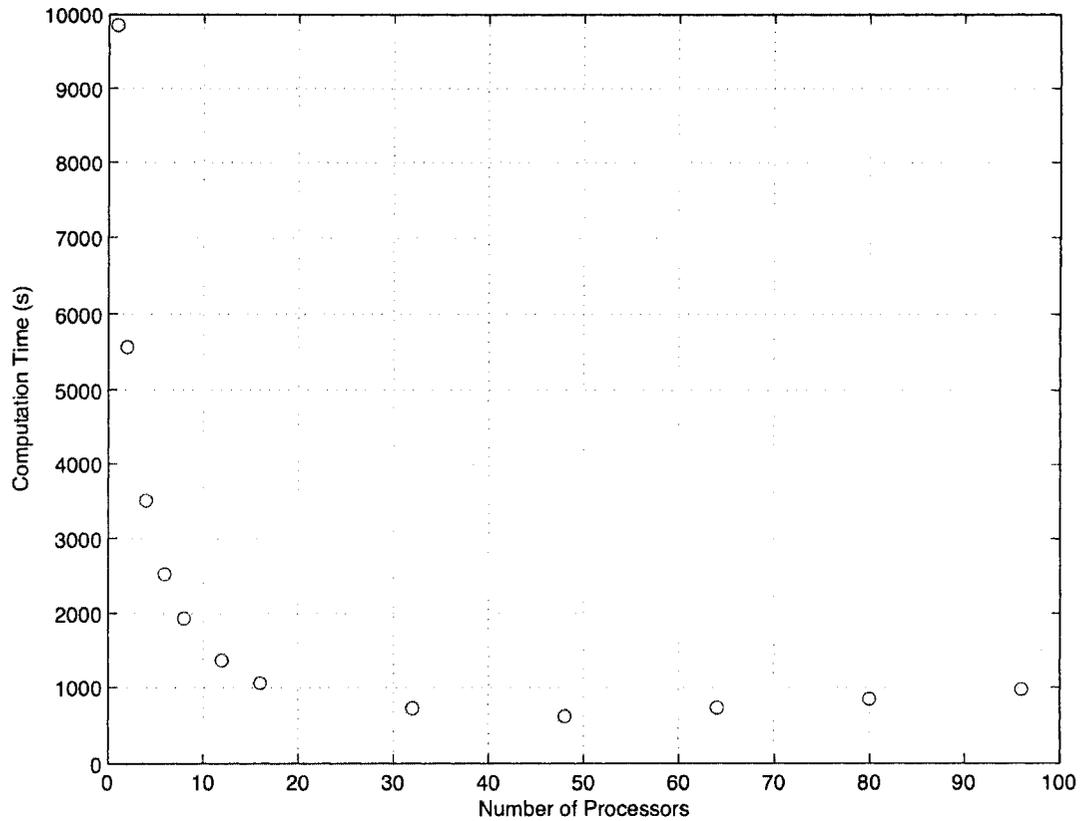


Figure 4.7: Time Needed to Run a 10 ps Benchmarking Simulation

The speedup for  $p$  processors is simply the time taken for one processor divided by the time for  $p$  processors [67, Page 61].

$$S_P = \frac{t_1}{t_p}$$

The efficiency is the speedup divided by the number of processors.

$$E_P = \frac{t_1}{pt_p}$$

The speedup and efficiency of the benchmarking tests are shown in Figure 4.8.

The speedup increases linearly up to 16 processors, after which it levels off. Maximum speedup is reached at 48 processors, where a speedup factor of 16 is reached. The efficiency is not good. Efficiency falls to about 60% with 16 processors and then to 33% with 48 processors.

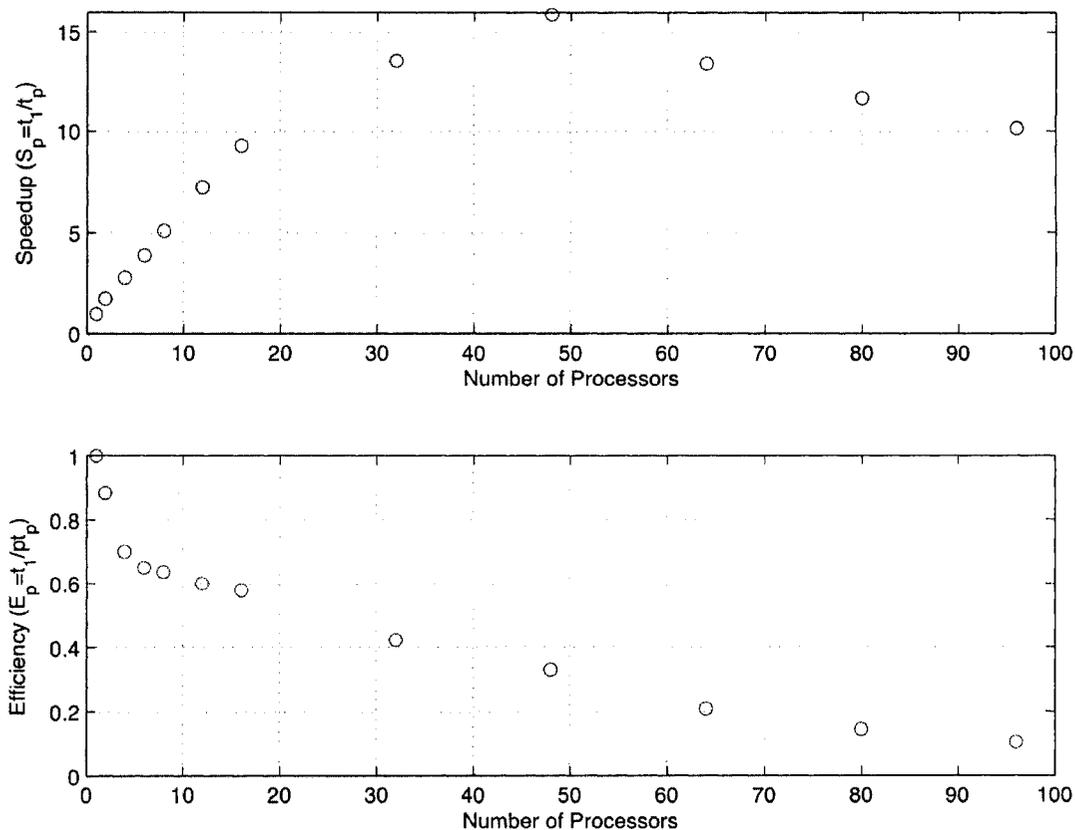


Figure 4.8: Speedup and Efficiency of a 10 ps Benchmarking Simulation

An efficiency of 100% with more than one processor is not a realistic expectation. Not all of the GROMACS algorithm can be parallelized. Some calculations must still be done in serial. This places a limit on the speedup and efficiency. Two processors will not halve the simulation time. Also, communication between processors takes time. If too many processors are used the communication could take more time than would be saved through the use of parallel computations [67, Pages 47 - 64].

Based on the above results it was decided that 16 processors would be used in the parallel simulations. This number of processors gave a speedup of 9-10 times while not hogging system resources that others were using.

The actual time needed to perform MD simulations on DPPC monolayers in serial and in parallel are shown in Figure 4.9. The plots show time in days as a function of number of atoms (DPPC and water) in the system. Simulations performed in serial on Chorus required between  $3\frac{1}{2}$  and  $6\frac{1}{2}$  days. These same simulations when performed in parallel on Mahone required between 1 and  $1\frac{1}{2}$  days. This was a speedup of about 4, which was not as good as the benchmarking indicated.

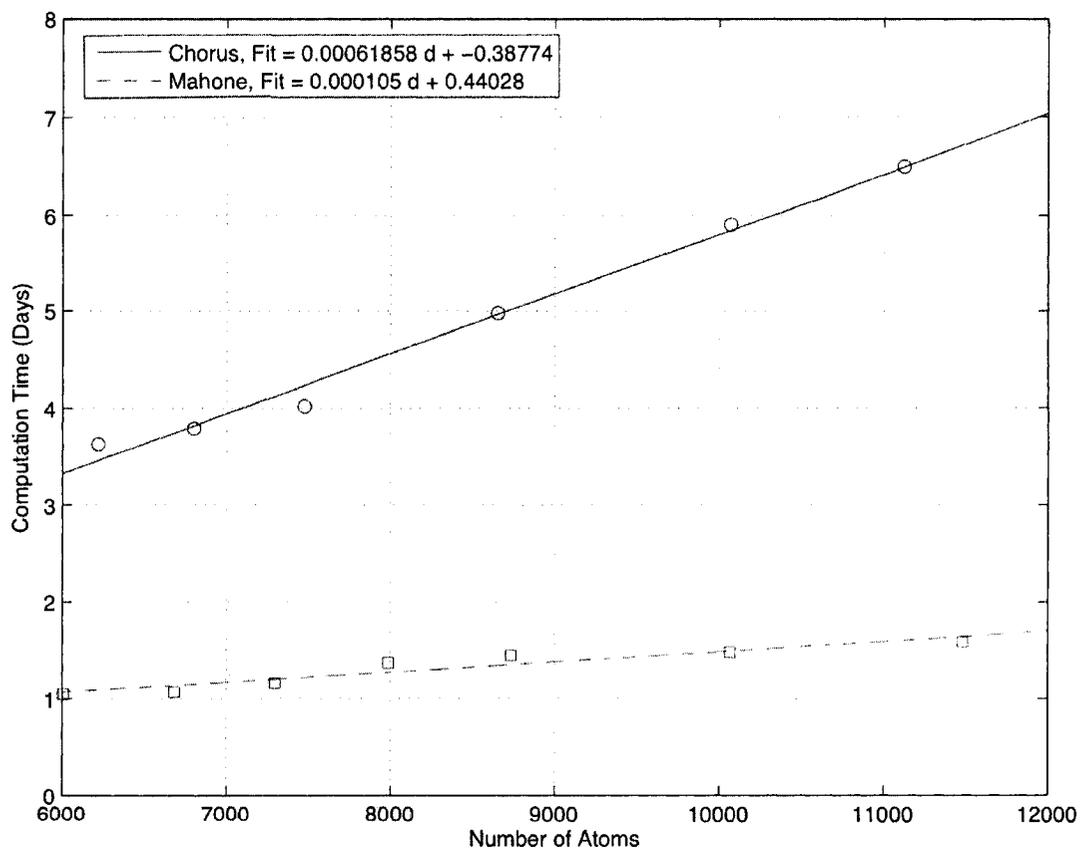


Figure 4.9: Time to Perform 10ns Simulation on a DPPC Monolayer

There are several explanations for why the speedup was not as good as expected. The monolayer simulations used PME, whereas the benchmark simulations used the simpler cut-off method for calculating electrostatic interactions. The longer monolayer simulations required much more writing output to the disk. Also, the parallel simulations were not set up properly with respect to load balancing. When the GRO-MACS preprocessor was run the options for shuffling and sorting the atoms between processors were not used. These options optimize the assignment of atoms to processors to get the best possible simulation speeds.

# Chapter 5

## Computational Results

### 5.1 Initial Configuration Observations

The monolayer configurations were subjected to energy minimization and a short MD simulation with position restraint to prepare them for the full MD simulation. The state of the configurations at the end of this preparation yielded some interesting results.

Figure 5.1 shows the starting configuration of DPPC monolayers with lipid densities of  $41.4 \text{ \AA}^2/\text{lipid}$  and  $77.0 \text{ \AA}^2/\text{lipid}$ . At  $41.4 \text{ \AA}^2/\text{lipid}$  the carbon tails were observed to stick up perpendicular to the water layer. The headgroups were close to the water surface. At  $77.0 \text{ \AA}^2/\text{lipid}$  the tails angled away from vertical and the headgroups sunk into the water layer.

Figure 5.2 shows the starting configuration of mixed DPPC/POPA monolayers with lipid densities of  $40 \text{ \AA}^2/\text{lipid}$  and  $80 \text{ \AA}^2/\text{lipid}$ . At  $40 \text{ \AA}^2/\text{lipid}$  the headgroups and carbon tails were positioned and aligned in a similar manner to pure DPPC. At  $80 \text{ \AA}^2/\text{lipid}$  the tails were still aligned largely to the vertical, indicating a tighter packing due to the larger cross sectional area of the POPA aliphatic carbon tails.

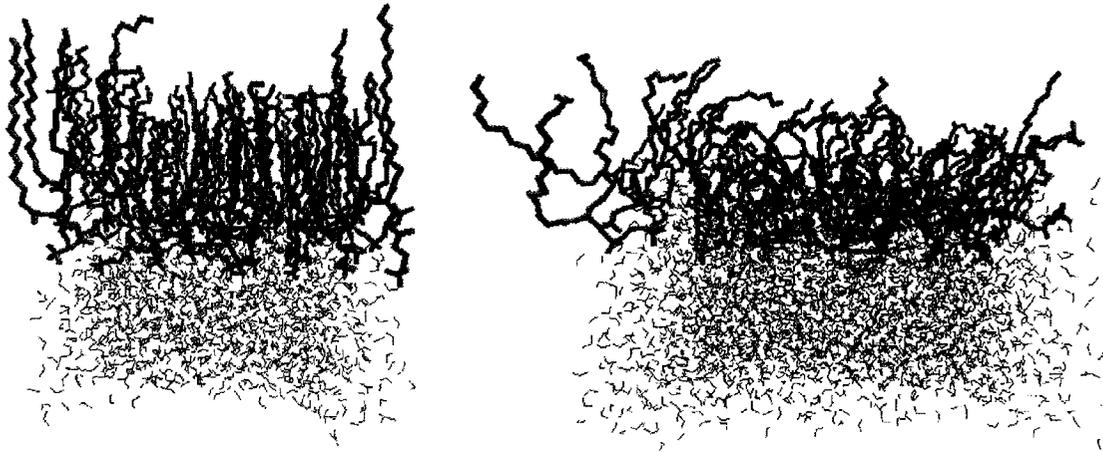


Figure 5.1: DPPC Monolayer Initial Configurations with Lipid Densities  $41.4 \text{ \AA}^2/\text{lipid}$  (left) and  $77.0 \text{ \AA}^2/\text{lipid}$  (right)

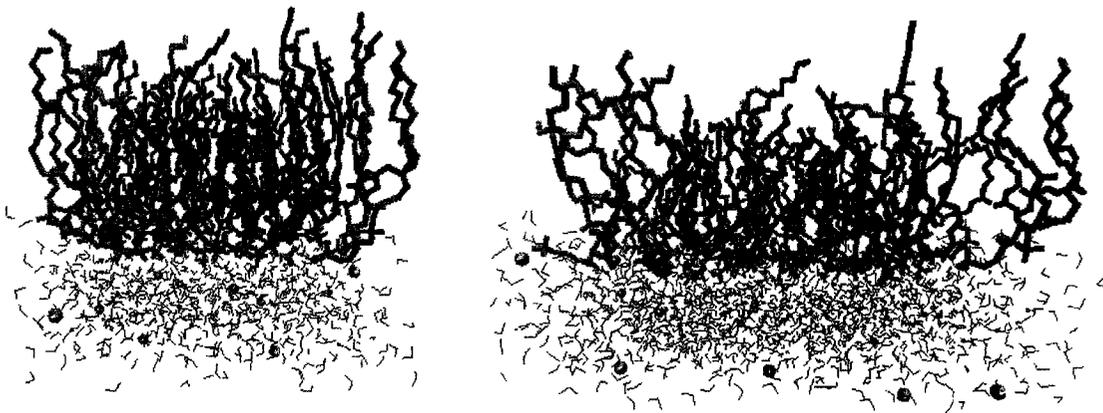


Figure 5.2: Mixed DPPC/POPA (Green/Blue) Monolayer Initial Configurations with Lipid Densities  $40 \text{ \AA}^2/\text{lipid}$  (left) and  $80 \text{ \AA}^2/\text{lipid}$  (right). Sodium Atoms are Cyan Spheres

## 5.2 Simulation Observations

GROMACS provided a record of the energies of the monolayer systems over time. These energies, which included temperature and pressure, were analyzed with the GROMACS program `g_energy` and Matlab. The energy data was converted from binary format to ascii format with `g_energy`. The ascii output, in the form of an `xmgr` file, was converted to a Matlab `m`-file via a Perl script we wrote for this purpose. The `m`-file was then modified to perform the required analysis when it was called by Matlab.

GROMACS also provided a record of the system trajectories. These trajectories consisted of the atomic positions and velocities at each output time step. The trajectories were examined with VMD [60], which was used to view each output frame (atom positions) in sequence to observe the behaviour of the monolayers over time. This visual inspection yielded some interesting observations.

### 5.2.1 Energy Analysis

An example of the total energy of a monolayer simulation is shown in Figure 5.3. This simulation began after the energy minimizations and short position restraint simulation. Initially there was a rapid decrease in energy. This decrease started to level off within 500 ps and an equilibrium energy level was reached after 1000 to 2000 ps. For the rest of the simulation the energy oscillated around this equilibrium value. The oscillations did not exceed 0.5% of the equilibrium value. This behaviour in the total energy was observed in all of the monolayers at all of the lipid densities.

The equilibrium energies of the systems were all negative, as shown in Figure 5.4. The loosely packed monolayers ( $80 \text{ \AA}^2/\text{lipid}$ ) had energies that were more negative than the tightly packed monolayers ( $40 \text{ \AA}^2/\text{lipid}$ ). Total energy seemed to decrease

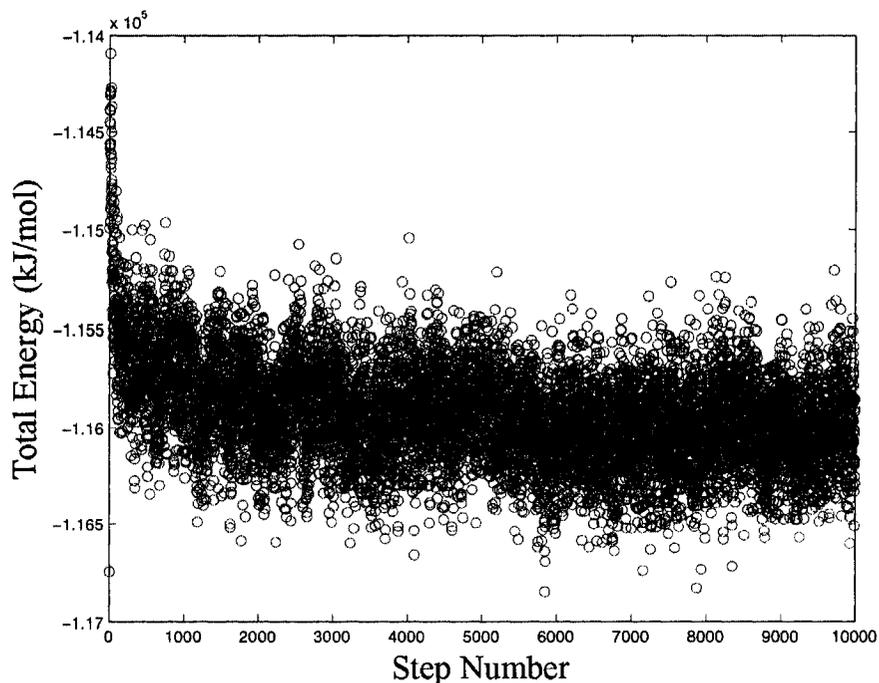


Figure 5.3: Total Energy of a DPPC Monolayer in Water at a Lipid Density of  $60 \text{ \AA}^2/\text{lipid}$

linearly with increasing lipid density (decreasing area per lipid) in four of the five monolayers. In the mixed DPPC/POPA monolayer the decrease was mostly linear with one deviation at  $50 \text{ \AA}^2/\text{lipid}$ .

Total energy was used as an indicator as to whether or not the system was in a stable equilibrium. For the first 2 ns the systems appeared to be in a non-equilibrium state. It is possible that the short position restraint run did not give the system enough time to reach equilibrium. It was decided that for calculating quantities such as surface tension, data from the first 2 ns would be discarded.

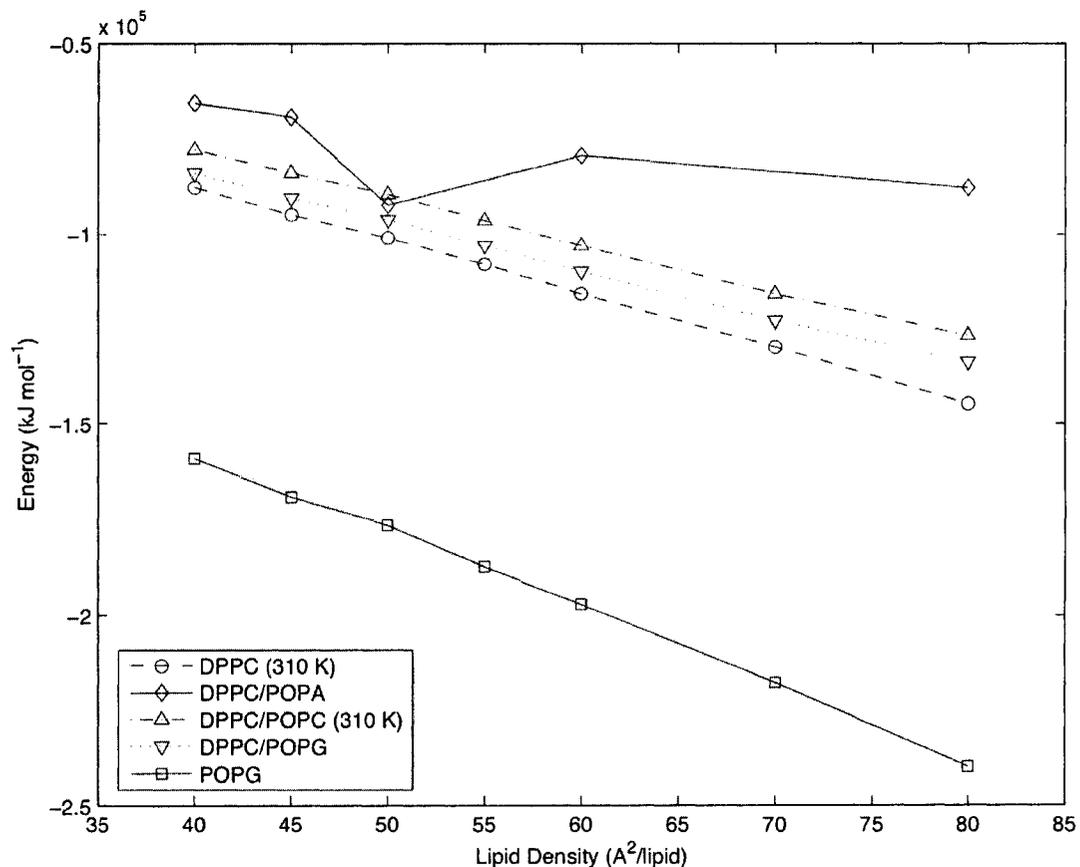


Figure 5.4: Total Energy as a Function of Lipid Density

## 5.2.2 Pure DPPC Monolayer

Monolayers with lipid densities from 55 to 80 Å<sup>2</sup>/lipid showed similar behaviour. The DPPC lipids rested on the water layer. Lipids were observed to move around horizontally, and the tails were free to flop around. The lipids did not move vertically. Lipids did not descend into nor rise up out of the water layer. Instead the lipid headgroups all remained at approximately the same vertical position. The interface between the lipids and water remained flat.

Throughout the simulations the water layer remained intact. Water molecules mingled with the headgroups but did not rise up to the carbon tails. A few water

molecules did separate from the layer and travel downwards towards the periodic boundary, but the number of molecules that crossed the boundary to interact with the tails was insignificant.

The main effect of the lipid density was on the orientation of the lipid tails with respect to the vertical axis. At  $80 \text{ \AA}^2/\text{lipid}$  the lipid tails tilted away from the vertical. As the area per lipid decreased the tails took on a more vertical orientation, and were vertically aligned by  $60 \text{ \AA}^2/\text{lipid}$ .

Different behaviour was observed in the monolayers with lipid densities from 40 to  $50 \text{ \AA}^2/\text{lipid}$ . At  $50 \text{ \AA}^2/\text{lipid}$  there appeared to be some vertical movement with the lipids. Individual lipids were observed to start to rise up relative to the other lipids. This became more pronounced at  $45 \text{ \AA}^2/\text{lipid}$ , in which the lipids in one corner of the monolayer started to rise up, forming a sort of hill. It was observed that as the lipids rose the surrounding water molecules rose with them, deforming the flat lipid/water interface.

The starting configurations of these tightly packed lipid monolayers were not totally flat. That is, the lipid headgroups were not all at the same vertical height. The height difference was minor for 50 and  $45 \text{ \AA}^2/\text{lipid}$ , but was pronounced at  $40 \text{ \AA}^2/\text{lipid}$ .

The simulation at  $40 \text{ \AA}^2/\text{lipid}$  showed wavelike behaviours. The configuration from the end of the simulation is shown in Figure 5.5. Along one lateral axis the lipids were arranged such that the central lipids rose relative to the side lipids. With the periodic boundary conditions this would be an undulation in the lipid/water interface. During the simulation the peak of the undulation was observed to move around.

The amplitude of the undulation was large. The headgroups of the peak lipids came close to the tips of the tails of the trough lipids. However, no lipid actually left the monolayer. Water molecules were drawn up with the lipids. Thus while the

monolayer was deformed it remained intact.

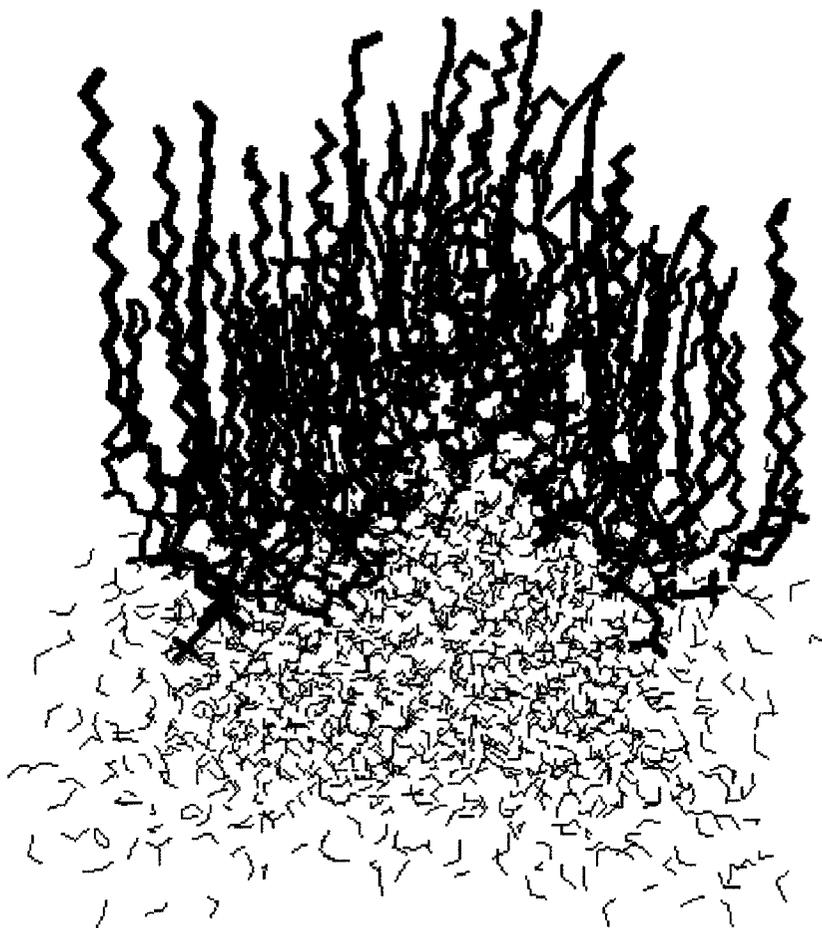


Figure 5.5: Configuration of a DPPC Monolayer with Lipid Density  $40 \text{ \AA}^2/\text{lipid}$  After 10 ns Simulation

To get a better understanding of what is happening at small lipid densities consider the phosphorus atoms in the DPPC lipid headgroups. In a monolayer the phosphorous atoms normally reside at the lipid/water interface, and should all be at approximately the same vertical height. When they are at the same height the lipid/water interface is flat.

Figure 5.6 shows side on views of the DPPC phosphorous atoms in the  $80 \text{ \AA}^2/\text{lipid}$  monolayer before and after the 10ns simulation. The phosphorous atoms were posi-

tioned in a planar formation parallel to the x-y axis, with some minor height variation between the atoms. The planar formation was maintained throughout the simulation. Note that the size of the phosphorous atoms was exaggerated for illustrative purposes.

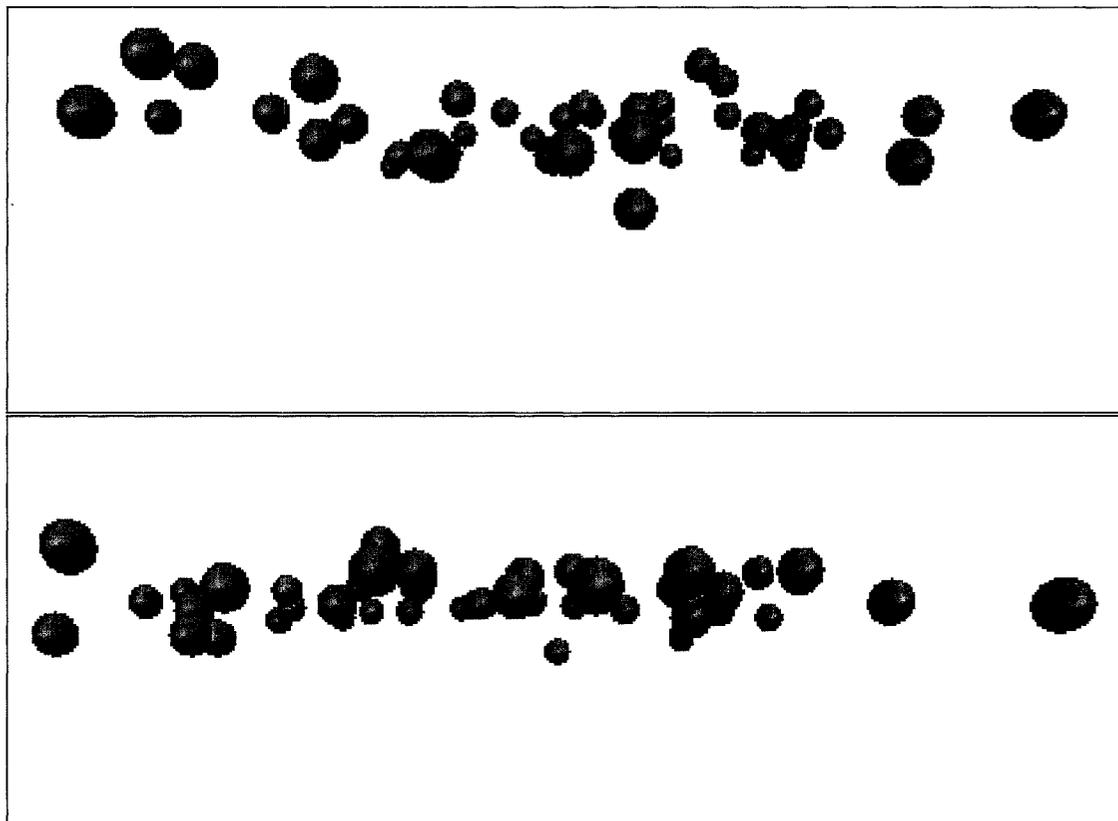


Figure 5.6: Phosphorous Atoms of DPPC Lipids in a Monolayer with Lipid Density  $80 \text{ \AA}^2/\text{lipid}$  Prior to (Above) and After (Below) the 10 ns Simulation

The monolayer with a lipid density of  $40 \text{ \AA}^2/\text{lipid}$  shows much different behaviour (Figure 5.7). At the start of the simulation the phosphorous atoms are not planar. There is a great deal of variation in height between the atoms. This variation is unstructured. During the simulation undulations in the vertical positions of the lipids formed.

The vertical movement of the lipids is likely due to the tight packing. The lipids are squeezed together tightly in the monolayer, close to their van der Waals radii.

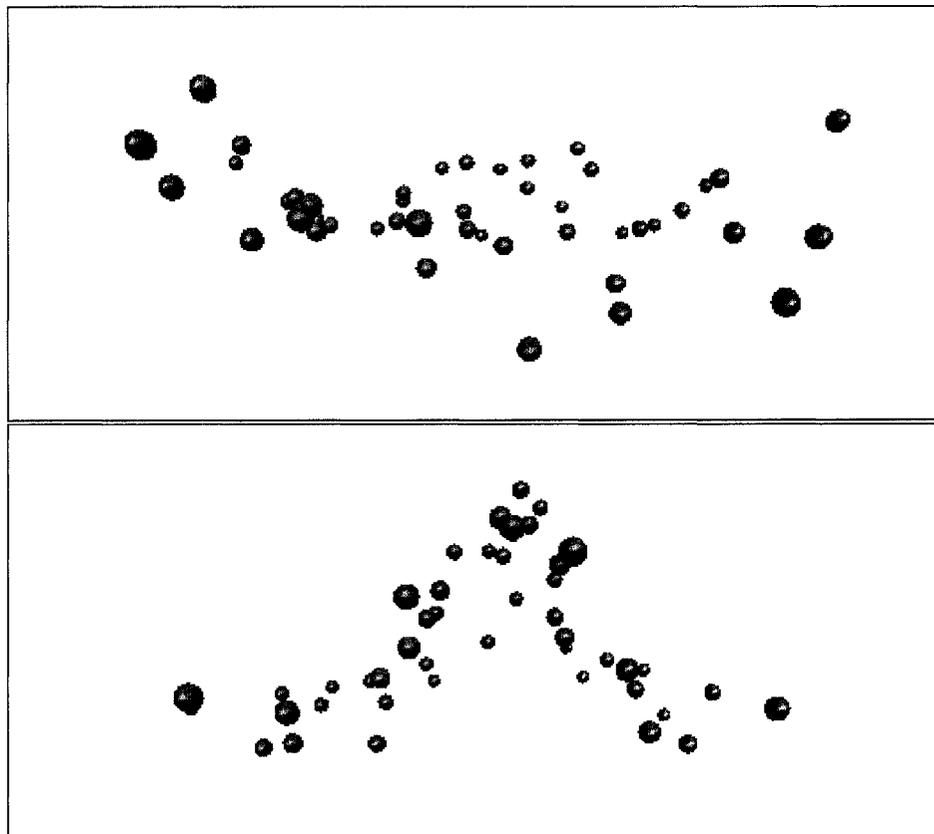


Figure 5.7: Phosphorous Atoms of DPPC Lipids in a Monolayer with Lipid Density  $40 \text{ \AA}^2/\text{lipid}$  Prior to (Above) and After (Below) the 10 ns Simulation

Vertical movement is one way for the lipids to gain more space.

It is important to note that while it appeared that some lipids were rising up to a considerable degree, no lipids were observed to completely leave the monolayer. This held true in all of the simulations, regardless of the type of lipids in the monolayer or the lipid density.

### 5.2.3 Mixed DPPC/POPA Monolayer

Loosely packed mixed monolayers of DPPC and POPA behaved similarly to loosely packed pure DPPC monolayers. At lipid densities of 60 and  $80 \text{ \AA}^2/\text{lipid}$  the lipid/water

interfaces were observed to be flat, with the lipid headgroups at approximately the same vertical height. Lipid movement was horizontal only. At  $60 \text{ \AA}^2/\text{lipid}$  the lipid chains were orientated vertically. At  $80 \text{ \AA}^2/\text{lipid}$  the tails started vertical but angled away from vertical as the simulation progressed.

Behaviour similar to that seen in the pure DPPC monolayer started to occur at a lipid density of  $50 \text{ \AA}^2/\text{lipid}$ . At this density individual lipids were observed to start to rise up higher than the other lipids, only to settle back down a short time later. Both DPPC and POPA were seen to rise. There did not seem to be a preference as to which type of lipid would rise.

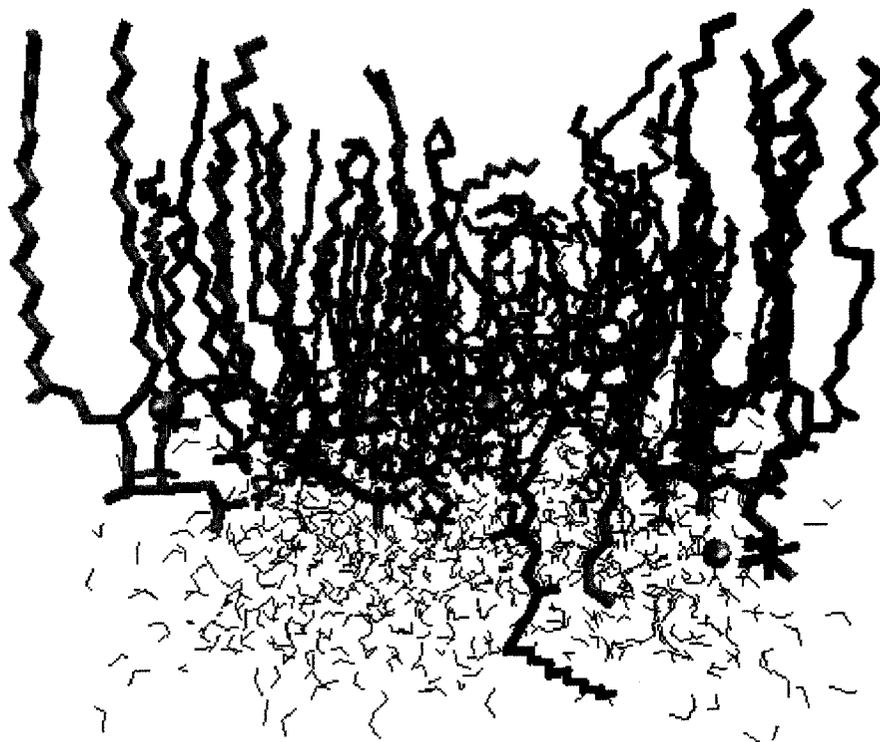


Figure 5.8: Mixed DPPC/POPA (Green/Blue) Monolayer with Lipid Density  $45 \text{ \AA}^2/\text{lipid}$  during 10 ns Simulation. Sodium Atoms are Cyan Spheres.

Groups of rising lipids were observed at a lipid density of  $45 \text{ \AA}^2/\text{lipid}$ . A more significant observation was that one POPA lipid was seen to go down into the water

layer (Figure 5.8). The POPA twisted in such a manner that one of its tails was pointing downwards, into and through the water layer. This was unexpected, given the hydrophobic nature of the lipid tails. The lipid and tail remained submerged for most of the 10 ns simulation. Note also that most of the sodium atoms drifted among the lipid headgroups.

The lipid/water interface lost all flatness at  $40 \text{ \AA}^2/\text{lipid}$  lipid density (Figure 5.9). Lipids, both DPPC and POPA, along opposite sides of the monolayer were seen to rise up out of the lipid/water plane. This is the same undulating behaviour that was seen in the pure DPPC monolayer. The rising lipids on opposite sides of the monolayer actually form a peak with each other, due to the periodic boundary conditions. Headgroup position in Figure 5.9 is illustrated with tan spheres for the phosphorous atoms.

One important observation from the tightly packed monolayers was that rising lipids were not limited to one type. There did not appear to be any preference for DPPC or POPA. When lipids rose they did so in groups based on position, not type.

#### 5.2.4 Mixed DPPC/POPC Monolayer

The mixed DPPC/POPC monolayer behaved the same as the previous monolayers for lipid densities from  $55$  to  $80 \text{ \AA}^2/\text{lipid}$ . The interface between lipid and water remained flat. Lipids moved around horizontally but not vertically.

Vertical lipid movement started to occur at a lipid density of  $50 \text{ \AA}^2/\text{lipid}$ . Some lipids, both DPPC and POPC, were observed to start to rise up and later settle back down during the simulation. At  $45 \text{ \AA}^2/\text{lipid}$  the difference in lipid height lasted throughout the simulation.

As with the DPPC/POPA monolayer, flatness in the lipid/water interface was

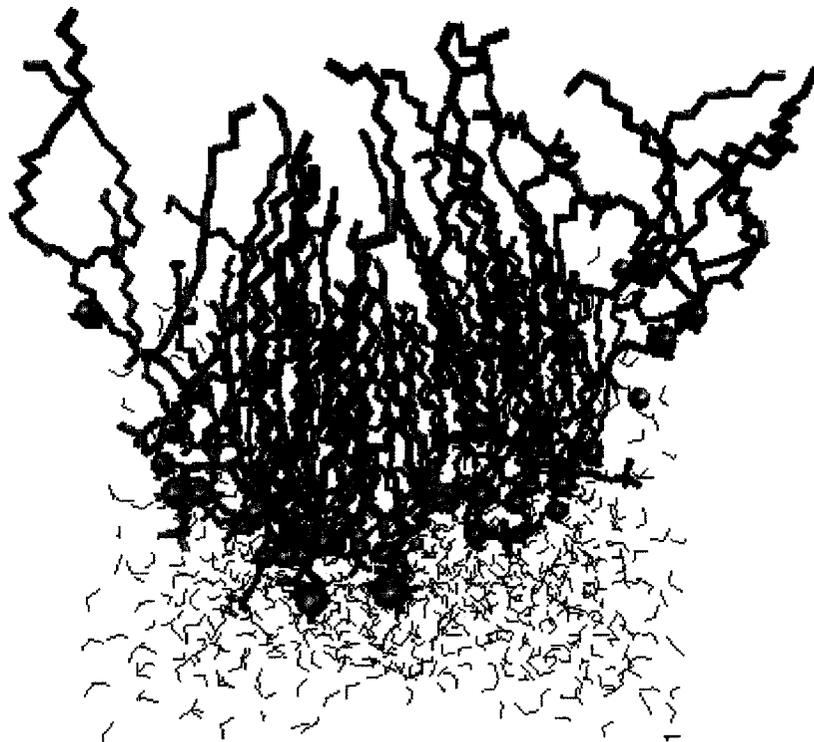


Figure 5.9: Mixed DPPC/POPA (Green/Blue) Monolayer with Lipid Density  $40 \text{ \AA}^2/\text{lipid}$  during 10 ns Simulation. Headgroup phosphorous atoms are shown as tan spheres, sodium atoms as cyan spheres.

lost at a lipid density of  $40 \text{ \AA}^2/\text{lipid}$ . More significantly, more submerged tails were observed (Figure 5.10). A DPPC lipid was observed to descend into the water layer. Several DPPC and POPC lipid tails were also submerged, even though the connected headgroups remained in the lipid/water interface. These tails remained submerged throughout the simulation.

Again, we observe that at tight lipid densities some lipids will start to rise up and, very rarely, descend into the water. There is no preference with regards to lipid type as to which lipids will rise or descend.

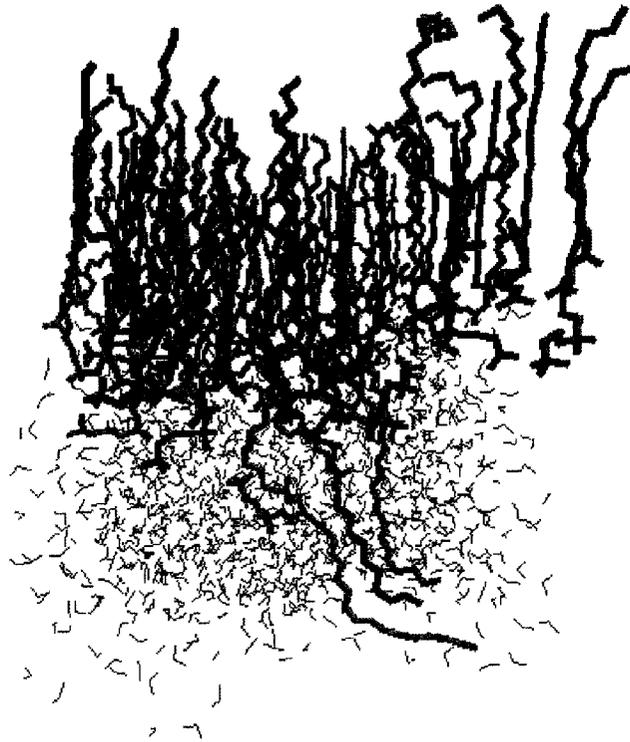


Figure 5.10: Mixed DPPC/POPC (Green/Blue) Monolayer with Lipid Density  $40 \text{ \AA}^2/\text{lipid}$  during 10 ns Simulation

### 5.2.5 Mixed DPPC/POPG Monolayer

The mixed DPPC/POPG monolayer showed the same behaviour as the other monolayers. For loose lipid packing ( $55$  to  $80 \text{ \AA}^2/\text{lipid}$ ) the lipids formed a flat interface with the water. Lipids moved horizontally but not vertically. Lipids started to rise up out of the interface plane for a lipid density of  $50 \text{ \AA}^2/\text{lipid}$ . This rising became more pronounced at  $45 \text{ \AA}^2/\text{lipid}$ . At  $40 \text{ \AA}^2/\text{lipid}$  any concept of a flat interface was lost.

Both DPPC and POPG lipids rose. There was no preference for one or the other.

### 5.2.6 Pure POPG Monolayer

The pure POPG monolayer exhibited the same behaviour as the other monolayers for loosely packed monolayers. The lipids formed a flat interface with the water layer. Lipid movement was horizontal along the interface. Lipid tails angled away from vertical at very loose packing (70 to 80 Å<sup>2</sup>/lipid) and took on vertical alignment at tighter packing. The sodium atoms were observed to mostly occupy the area around the headgroups, although a few did move around in the water layer.

One major difference from the other monolayers was that the flat interface persisted to lipid densities of 45 Å<sup>2</sup>/lipid. There was no indication of lipids rising at 45 or 50 Å<sup>2</sup>/lipid. At 50 Å<sup>2</sup>/lipid one lipid was observed to descend into the water layer by a small amount, with its tails remaining vertical and pointing upwards. Otherwise there was no indication of any lipids leaving the flat interface.

Flatness was lost at 40 Å<sup>2</sup>/lipid (Figure 5.11). The lipid/water interface started out flat. As the simulation progressed the lipids on one side of the monolayer rose up. There was no corresponding rise on the other side of the monolayer (PBC). Some lipids on the other side did rise, but the two sides were independent of each other. This may have been the beginnings of the undulations seen in other monolayers. In Figure 5.11 the phosphorous atoms are used to illustrate headgroup position.

## 5.3 Water Layer Simulation

The approach used to simulate lipid monolayers was similar to the approach used by Kaznessis et. al. [16]. This approach resulted in two contributions to the surface tension. A surface tension is generated at the lipid/water interface, and at the water/vacuum interface (Figure 3.5). The surface tension calculated by GROMACS is the sum of the two surface tensions. To determine the lipid/water surface tension it

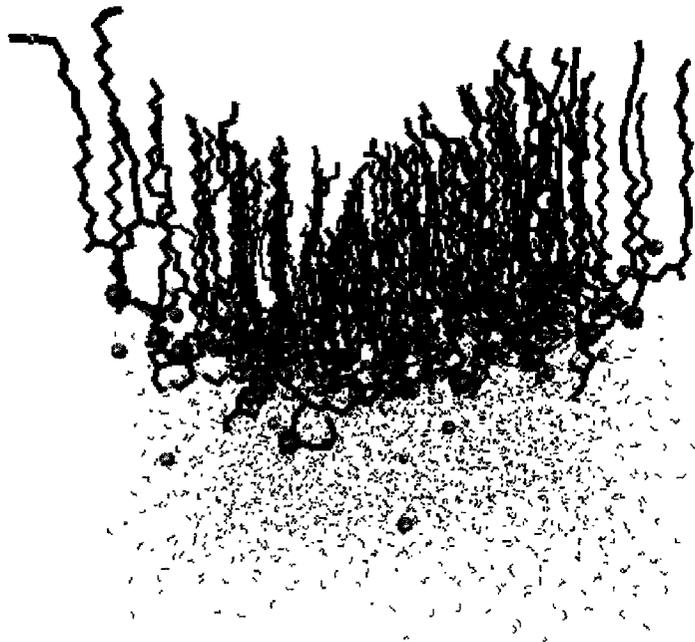


Figure 5.11: POPG Monolayer with Lipid Density  $40 \text{ \AA}^2/\text{lipid}$  after 10 ns Simulation was first necessary to calculate the water/vacuum surface tension.

MD simulations were performed on water layers without any lipids. The initial configuration files for DPPC monolayers ( $41.4$ ,  $45.9$ ,  $50.3$ ,  $59.2$  and  $77.0 \text{ \AA}^2/\text{lipid}$ ) were used. The DPPC lipids were deleted from the configuration files, leaving only the water layer with vacuum on both sides. Simulations were run for 10 ns, using the same parameters as for the lipid simulations. The results from the simulations are given in Table 5.1.

The surface tension quoted for a given original lipid density is the average of the surface tensions calculated during the simulation at each picosecond from 2 ns to 10 ns. The data from the first 2 ns was discarded from all surface tension calculations due to the changing total energy observed during this time (Figure 5.3). The simulated system has two water/vacuum interfaces, and the calculated surface tensions were for

Table 5.1: Surface Tensions on a Water Layer without Lipids

Original Lipid Density ( $\text{\AA}^2/\text{lipid}$ )	Surface Tension (mN/m)
41.4	$104.6 \pm 1.4$
45.9	$103.0 \pm 1.4$
50.3	$102.2 \pm 1.4$
59.2	$103.2 \pm 1.3$
77.0	$101.2 \pm 1.2$

both interfaces. To get the surface tension at one interface it was necessary to divide by two. The uncertainty was taken from the standard error in the mean, which is the method used by Allen and Tildesley [18, Page 194].

The surface tension values are very close to each other. The original lipid densities of the water layers with lipids did not seem to influence the calculation. This makes sense, since each simulated water layer has the same density. During the simulations the water layers remained intact with a constant volume. The layers did not appear to deform in the absence of lipids.

The average of the five values was taken to be the surface tension of a water/vacuum interface. This value was  $51.4 \pm 0.7$  mN/m.

The calculated value of the surface tension of water was different from the expected value of 72 mN/m. This was not unexpected. It is well known that the available force fields are not good at predicting the surface tension of an air/water interface [49]. Kaznessis in his simulations obtained a surface tension of  $56.6 \pm 2.3$  [16].

Different force fields model water in different ways. A test was made with three different water force fields to see what the difference in surface tension would be. The force fields were single point charge (SPC), extended single point charge (SPC/E) and TIP3P. The simulations were performed on the water layer that had come from the  $41.4 \text{\AA}^2/\text{lipid}$  DPPC monolayer. The results are given in Table 5.2.

Table 5.2: Surface Tensions on a Water Layer from Different Water Force Fields

Lipid Density ( $\text{\AA}^2/\text{lipid}$ )	Surface Tension (mN/m)
SPC	$52.4 \pm 0.7$
SPC/E	$58.3 \pm 0.8$
TIP3P	$48.2 \pm 0.7$

The three force fields gave different values, but none were close to the expected value of 72 mN/m. SPC/E was closer to the true value than SPC, but not by much. Tieleman and Berendsen [43] in their paper recommend SPC over SPC/E for studies involving lipids, due to SPC’s better chemical potential. Thus SPC was used.

## 5.4 Surface Tension

The surface tension for the various monolayers was calculated. GROMACS calculated the time averaged total surface tension of the system ( $\gamma_{sim}$ ) based on Equation 2.17,

$$\gamma_{sim} = \frac{1}{h_z} \left\langle P_{zz} - \frac{P_{xx} + P_{yy}}{2} \right\rangle \quad (5.1)$$

where  $\langle \rangle$  indicates averaging over time. The surface tension at the water/vacuum interface ( $\gamma_{wv}$ ) was calculated to be  $51.4 \pm 0.7$  mN/m. The monolayer surface tension was thus

$$\gamma_{mon} = \gamma_{sim} - \gamma_{wv}. \quad (5.2)$$

The surface tension results from the simulations with the DPPC monolayer are plotted in Figure 5.12. Simulations were performed at 300 K (squares) and 310 K (circles). The two simulations appeared to follow the same surface tension isotherm.

The calculated surface tensions were smaller than the surface tension of water, 72

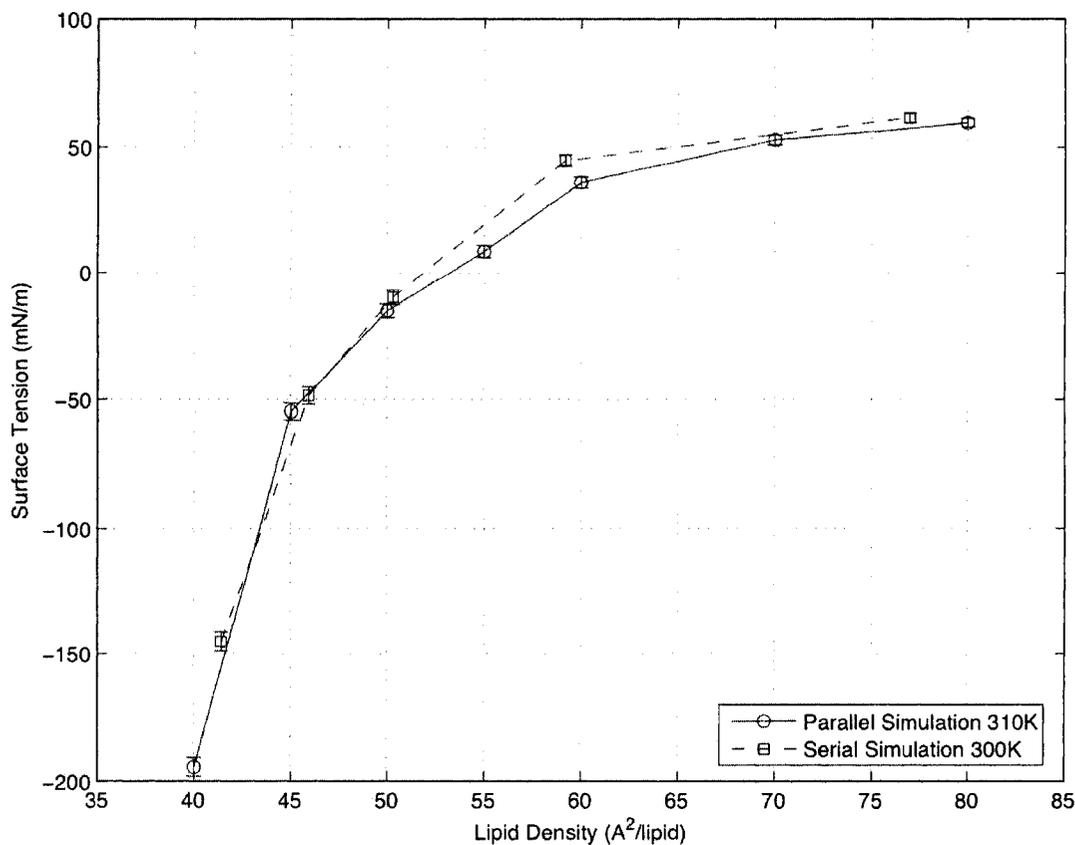


Figure 5.12: Surface Tension in a DPPC Monolayer

mN/m. At 80 Å<sup>2</sup>/lipid the surface tension was slightly lower than pure water. As the lipid packing increased the surface tension decreased slightly, up to 60 Å<sup>2</sup>/lipid. Past 60 Å<sup>2</sup>/lipid the surface tension started to fall more rapidly. The surface tension became negative at 52 Å<sup>2</sup>/lipid and plunged to almost -200 mN/m at 40 Å<sup>2</sup>/lipid.

The surface tension isotherm has the expected shape. The presence of lipids reduced the surface tension at the lipid/water interface. The surface tension fell as the lipids were pushed closer together. The actual surface tension values were not as expected. The values for loose packing (60 to 80 Å<sup>2</sup>/lipid) were alright. But for tight packing the calculated surface tensions took on negative values.

The negative surface tensions could be explained mathematically. In order to calculate a surface tension GROMACS used the diagonal components of the pressure tensor, from which it calculated the difference between the normal and lateral pressures acting on the system. The pressure components for the DPPC monolayer system are plotted in Figure 5.13. The z-component changed very little between different lipid densities. It was essentially constant and had a value very close to zero. The x and y-components changed a great deal with different lipid densities. For tightly packed monolayers these components became positive, which in turn made the surface tension negative.

Positive and negative pressures have a physical interpretation. A positive pressure indicates a tendency to expand, while a negative pressure indicates a tendency to contract. When the lipids become too tightly packed they try to push apart. This is seen as an expansive lateral pressure in the monolayer, and leads mathematically to a negative surface tension.

The surface tension isotherm calculated for the mixed DPPC/POPA monolayer is shown in Figure 5.14. The isotherm is similar to the isotherm from DPPC. Surface tension decreased as the lipids were more tightly packed. The decrease was gradual from 80 to 60  $\text{\AA}^2/\text{lipid}$ , and became steeper between 60 and 40  $\text{\AA}^2/\text{lipid}$ . The calculated surface tension became negative at around 47  $\text{\AA}^2/\text{lipid}$ . In this case the surface tension remained positive for more lipid densities than for DPPC.

Surface tension isotherms for a mixed DPPC/POPC monolayer are shown in Figure 5.15. As for the case with DPPC there were two isotherms, one at 300 K and one at 310 K. The two curves were very similar in shape. Surface tensions for 300 K seemed to be slightly higher than those at 310K, indicating that the temperature of the monolayer has an effect on the surface tension.

The isotherms followed the pattern seen in the previous monolayers. Surface

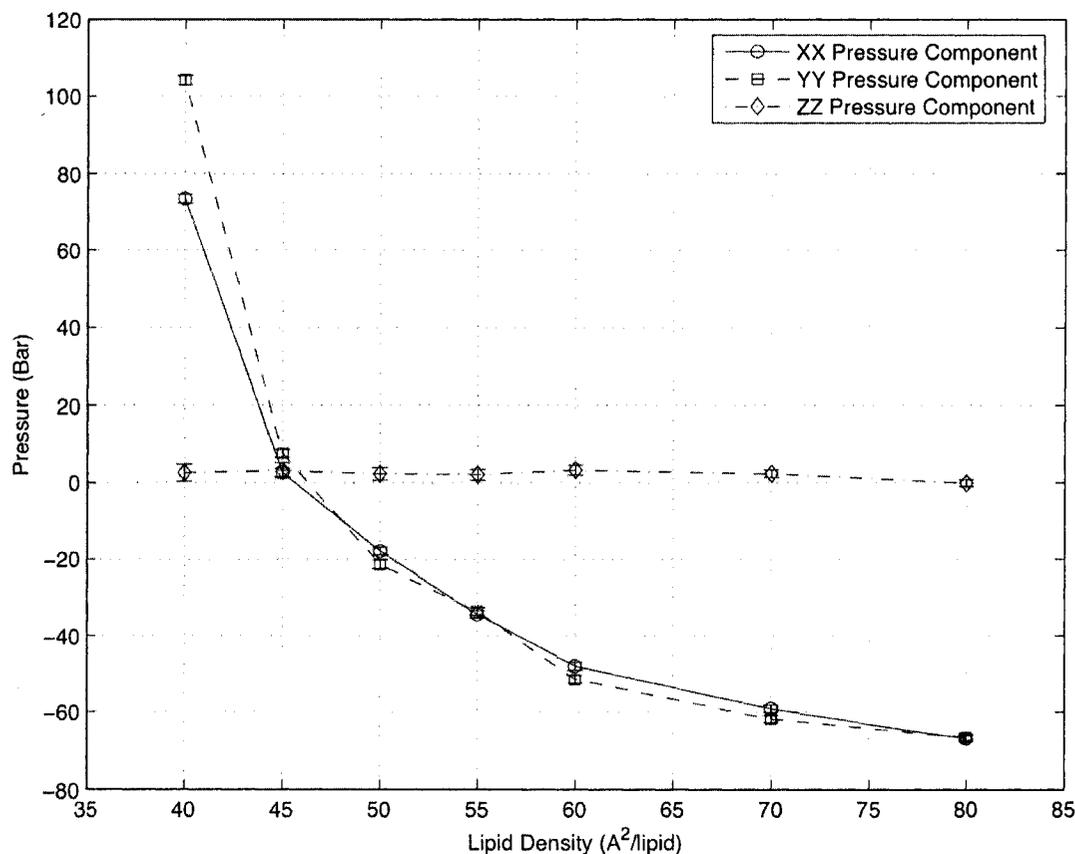


Figure 5.13: Time Averaged Pressure Components from a DPPC Monolayer as a Function of Lipid Density

tension decreased gradually from 80 to 60 Å<sup>2</sup>/lipid. The decrease then became more rapid, and the surface tension became negative between 50 and 55 Å<sup>2</sup>/lipid. At 40 Å<sup>2</sup>/lipid the surface tension plunged to -250 mN/m, which was lower than the other monolayers.

The mixed DPPC/POPG monolayer showed different behaviour from the previous monolayers (Figure 5.16). Surface tension decreased as packing increased, as was seen in the other monolayers. The range of gradual decrease was shorter, and the more rapid fall began at 70 Å<sup>2</sup>/lipid. The surface tension remained positive longer, and became negative at around 47 Å<sup>2</sup>/lipid. The surface tension at 40 Å<sup>2</sup>/lipid was -120

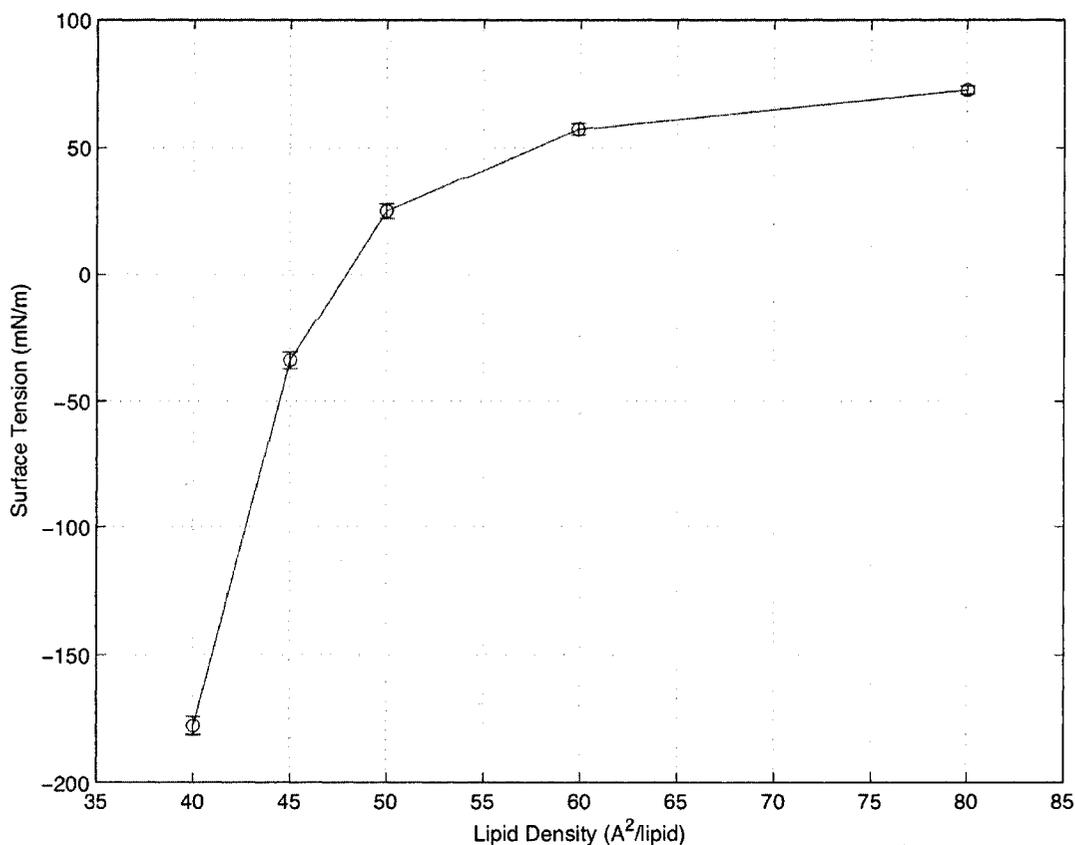


Figure 5.14: Surface Tension in a Mixed DPPC/POPA Monolayer

mN/m, which was much higher than the other monolayers.

The final monolayer was POPG. The surface tension isotherm for this monolayer, shown in Figure 5.17, was similar to the others. The surface tension decreased as the lipid packing increased. The decrease was gradual between 80 and 60  $\text{\AA}^2/\text{lipid}$ , and was rapid between 60 and 40  $\text{\AA}^2/\text{lipid}$ . This monolayer had the steepest fall of all the monolayers. The surface tension at 40  $\text{\AA}^2/\text{lipid}$  was almost -400 mN/m.

The isotherms for the five monolayers are plotted in Figure 5.18. The isotherms were made with a temperature coupling of 310 K, except for the DPPC/POPA isotherm which was made at a temperature of 300 K. For DPPC and DPPC/POPC

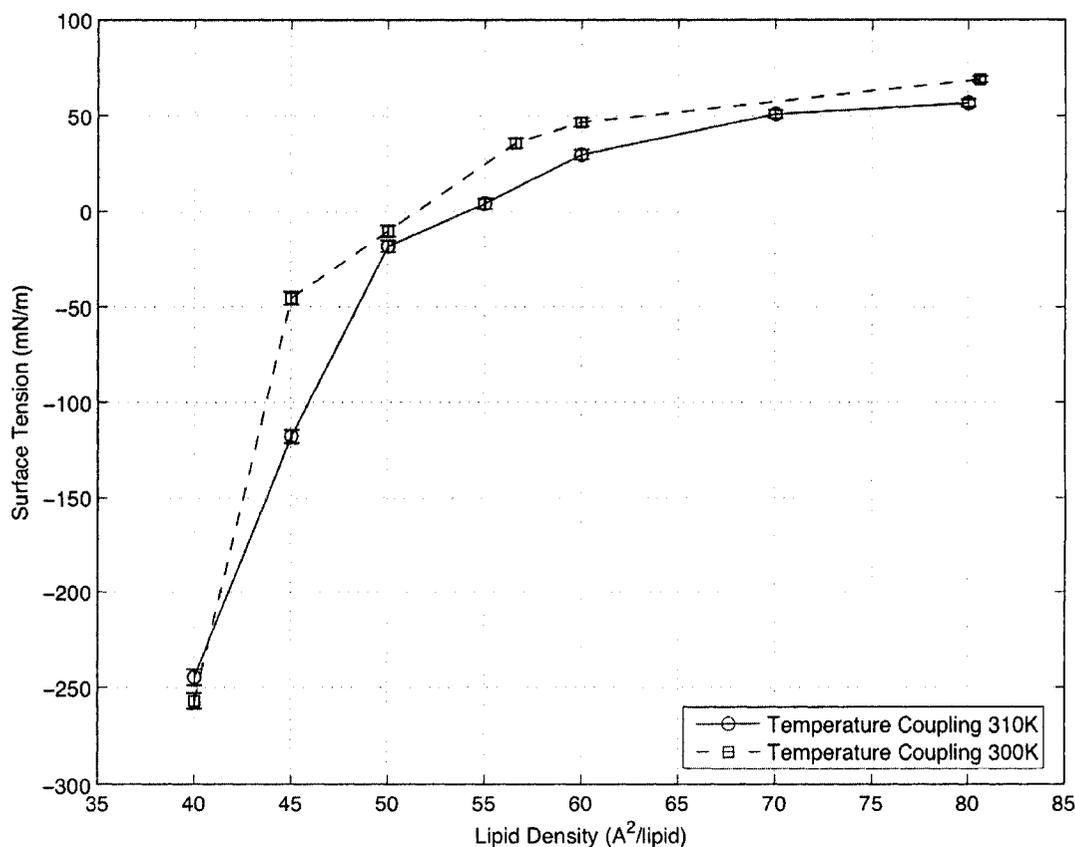


Figure 5.15: Surface Tension in a Mixed DPPC/POPC Monolayer

only the isotherm at 310 K was plotted.

The monolayer isotherms are similar in the loosely packed region of 60 to 80 Å<sup>2</sup>/lipid. Differences become more apparent between 40 and 60 Å<sup>2</sup>/lipid. All five monolayers follow the same general curve. The DPPC/POPG monolayer overall has surface tensions that are the least negative. POPG and DPPC/POPC have surface tensions that are the most negative.

Figure 5.19 shows the surface tension isotherms in the region between 50 and 80 Å<sup>2</sup>/lipid. The monolayers with only PC's (DPPC and DPPC/POPC) were very similar, both in isotherm shape and in surface tension values. This was expected,

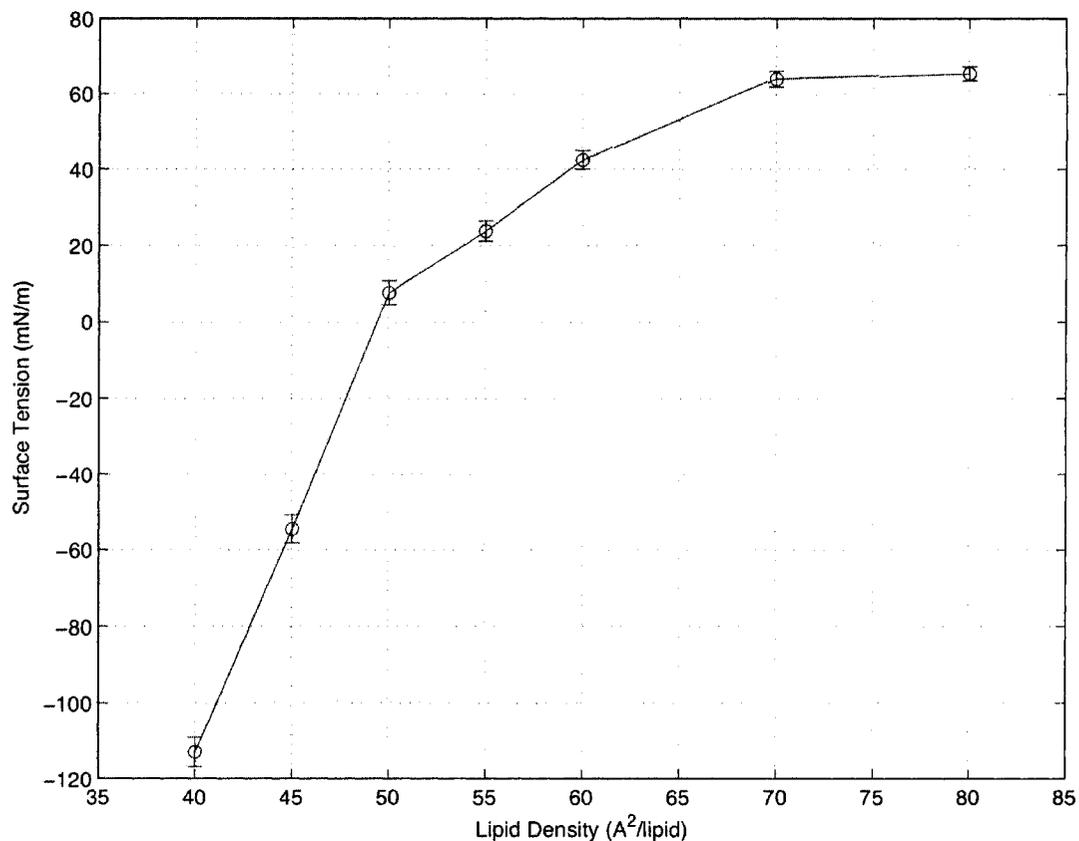


Figure 5.16: Surface Tension in a Mixed DPPC/POPG Monolayer

since the two PC lipids have the same headgroup and would interact with water in the same way. Differences in surface tension in the more tightly packed monolayers would be due to the double bond in the POPC tail, which would resist compression more than the saturated DPPC tails. The other mixed monolayers (DPPC/POPG and DPPC/POPA) had a similar isotherm shape, but their surface tensions were greater. The POPG monolayer saw surface tension increase and then fall at a very rapid rate.

The surface tension calculations provide some interesting results. The calculations showed that as the monolayers were compressed they developed lateral pressures

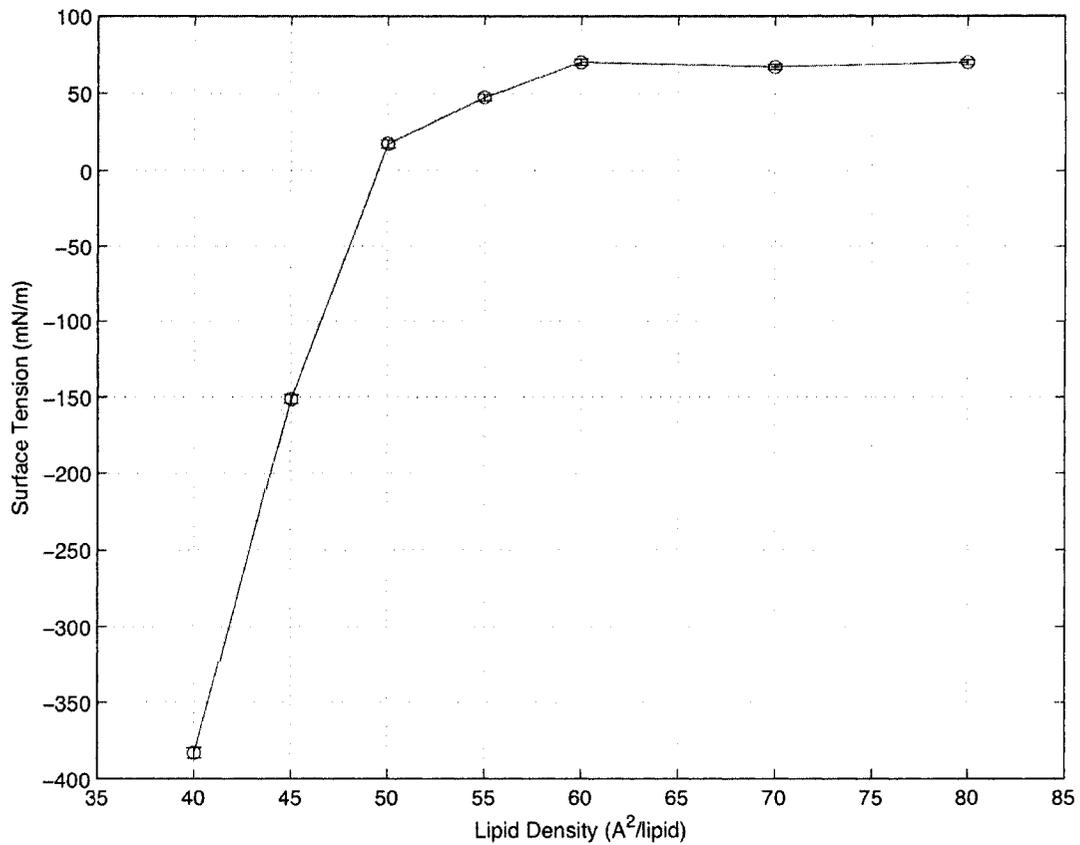


Figure 5.17: Surface Tension in a POPG Monolayer

which resisted the compression and tried to expand the monolayer. Different lipids generated different levels of pressure.

## 5.5 Comparison to Experiment

Measurements of the surface tension in a DPPC monolayer and in a mixed DPPC/POPG monolayer were performed by Mauricia Garcia in Kaushik Nag's lab. The experiments were done on a Langmuir balance [2]. These experiments measured a surface pressure,

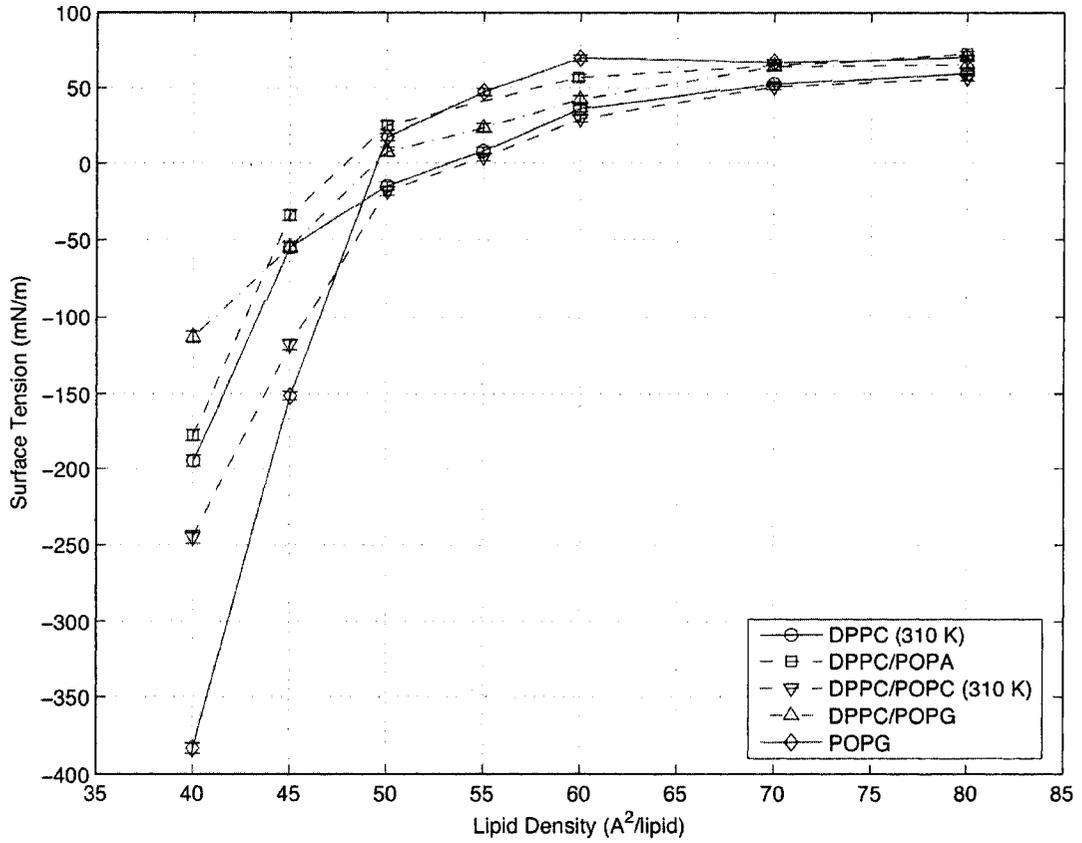


Figure 5.18: Surface Tension in All Monolayers

which was converted into a surface tension by

$$\gamma_{mon} = \Pi - \gamma_{water} \quad (5.3)$$

where  $\gamma_{mon}$  was the monolayer surface tension,  $\Pi$  was the monolayer surface pressure, and  $\gamma_{water}$  was the surface tension of pure water, 72 mN/m.

The computational results for the DPPC monolayer are plotted alongside the experimental results in Figure 5.20. Also included in the figure is data from the Handbook of Monolayers [61]. This data consisted of surface pressure measurements that were obtained from previous experiments. The surface pressures were converted

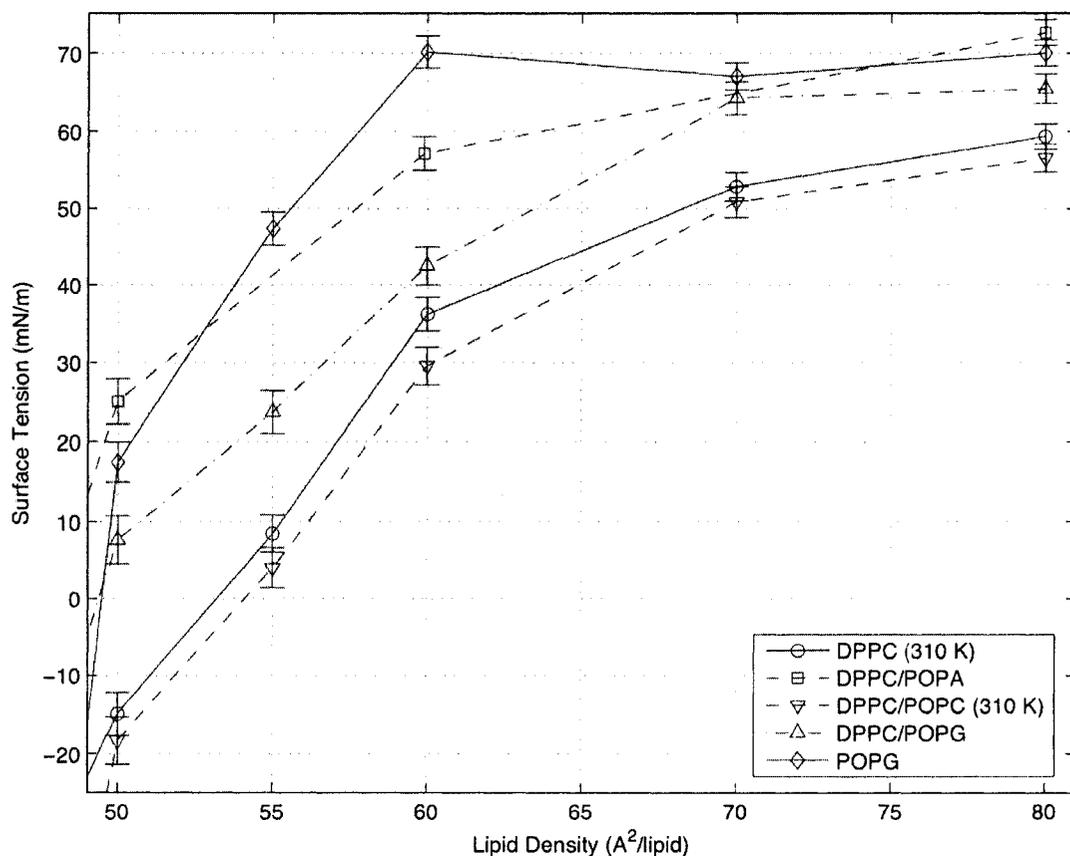


Figure 5.19: Surface Tension in All Monolayers between 50 and 80 Å<sup>2</sup>/lipid

to surface tensions.

There was very good agreement between experiment and computation at 80 Å<sup>2</sup>/lipid. As the lipid packing tightened the agreement between experiment and computation started to worsen. While both showed a gradual decrease in surface tension, the computed surface tension fell at a slightly faster rate than experiment.

The rate of decrease picked up at 60 Å<sup>2</sup>/lipid and continued to 50 Å<sup>2</sup>/lipid. Again, the decrease was faster in the computations than experiment. This widened the gap between computation and experiment even further. It was at this point that the experiment started to differ from the Handbook of Monolayers data. The data showed

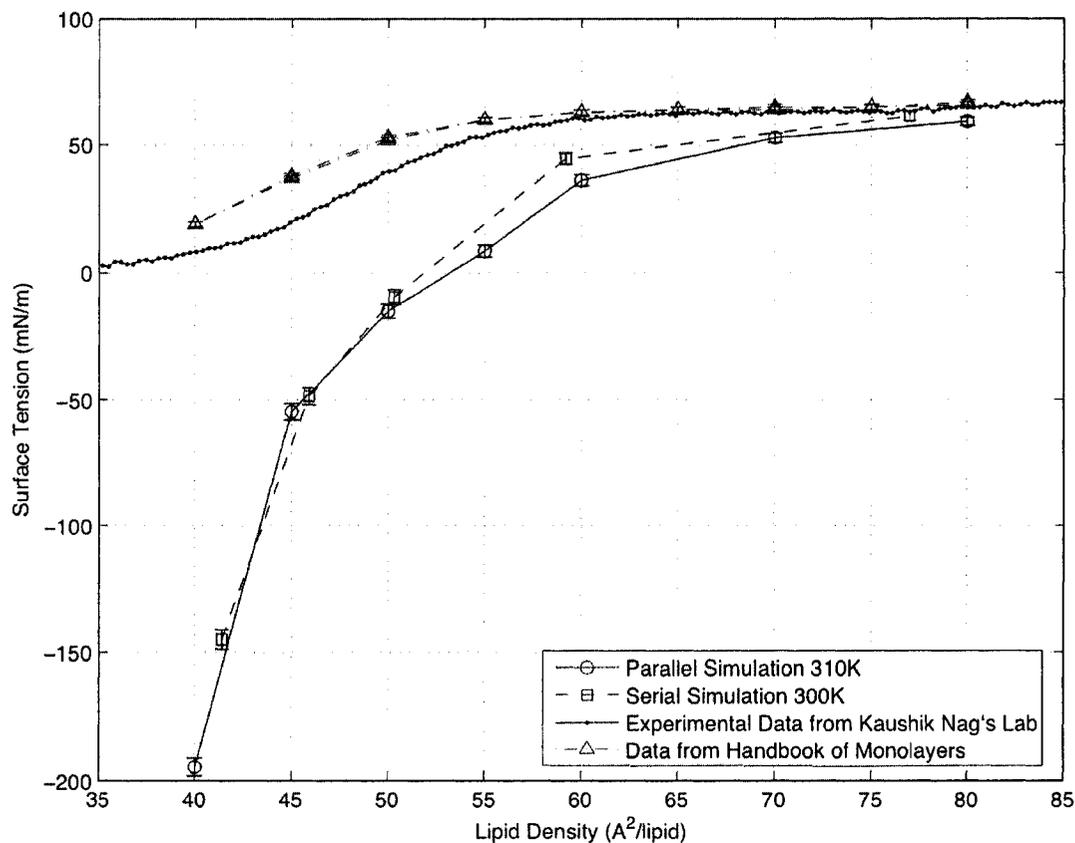


Figure 5.20: Computational and Experimental Measurements of a DPPC Monolayer in Water

a slower fall in surface tension than the experiment.

At  $45 \text{ \AA}^2/\text{lipid}$  the experimental surface tension started to level off. Surface tension fell at a slower rate, and eventually stopped falling at  $37 \text{ \AA}^2/\text{lipid}$ . The experimental surface tension at the densest packing was close to but still greater than zero. The surface tension data from the handbook did not level off. However, the data stopped at  $40 \text{ \AA}^2/\text{lipid}$ . Also, the data had fallen at a slower rate than the experiment, and was still greater than zero. The computed surface tension continued to fall at an ever increasing rate, and ended at  $40 \text{ \AA}^2/\text{lipid}$  with a surface tension close to  $-200 \text{ mN/m}$ .

From this we see that MD was good for predicting the surface tension for loosely

packed monolayers, but was not good for tightly packed monolayers. Previous MD work, which looked at the loosely packed region, also found good agreement between simulation and experiment [16] [68].

The ability of the MD simulations to predict experimentally measured surface tensions should not be given too much importance. Experiments measure macroscopic properties, such as the pull a monolayer exerts on a dipping plate [2]. MD measures the pressures acting on the system. The comparison assumes that the two quantities are equivalent, which may not be the case.

The deviation from experiment was greatest for tightly packed monolayers. For these monolayers there was a slight difference between the surface tension measured in the experiment and the surface tension data from previous experiments. This difference indicated that the behaviour of the tightly packed monolayers were more complex than for loosely packed monolayers. This greater complexity was observed in the model, in the form of vertical movement among the lipids.

Figure 5.21 shows the experimentally measured surface tension from the mixed DPPC/POPG monolayer alongside the computational results. For loosely packed monolayers (60 to 80  $\text{\AA}^2/\text{lipid}$ ) the experimental surface tension decreased more rapidly than the computations predicted. The gradual decline predicted by the computations was not seen, and the surface tension values were overestimated. The situation switched at 60  $\text{\AA}^2/\text{lipid}$ . Between 50 and 60  $\text{\AA}^2/\text{lipid}$  the calculated surface tension decreased more rapidly than experiment. This was in spite of the fact that the experimental surface tension also decreased at a faster rate. The experimental surface tension started to level off at 47  $\text{\AA}^2/\text{lipid}$ , while the computed surface tension continued to fall more rapidly and go negative.

In this case MD did not do as well in predicting the measured surface tensions. This was not a concern. The rising lateral pressures which gave rise to the negative

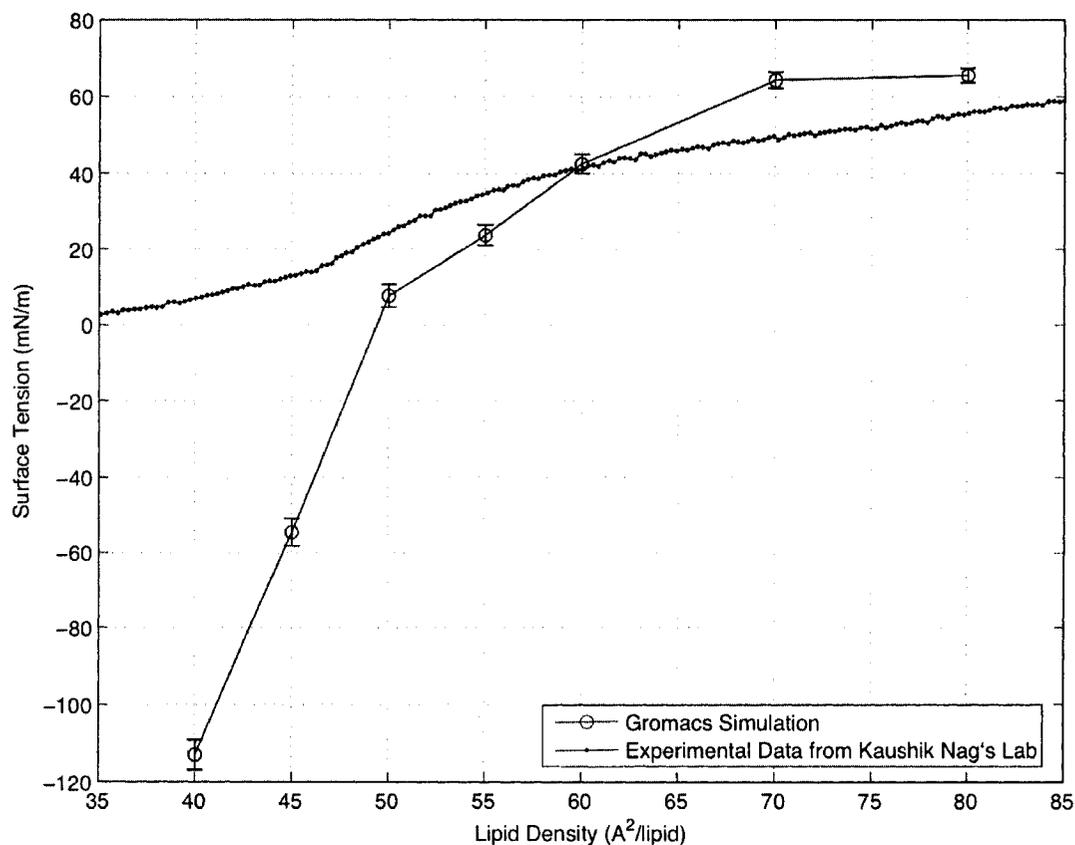


Figure 5.21: Computational and Experimental Measurements of a Mixed DPPC/POPG Monolayer in Water

surface tensions were more important than the fit to experiment.

## 5.6 Comparison to Previous Simulations

This research has studied the behaviour of phospholipid monolayers under different levels of compression. Similar research has been done by Kaznessis *et. al.* [16]. They studied monolayers of DPPC and DPPG under different levels of compression in the loosely packed region.

Kaznessis' group published their surface tension results in the form of surface

pressures. The conversion from surface tension was made using Equation 5.3. Their surface pressure isotherms showed that the surface pressure of DPPC ranged between 40 and 50 mN/m from 70 to 60 Å<sup>2</sup>/lipid, and then rose rapidly to 80 mN/m at 55 Å<sup>2</sup>/lipid. A mixed monolayer of DPPC/DPPG in a 7:3 ratio had a similar isotherm shape but lower surface pressures (higher surface tensions).

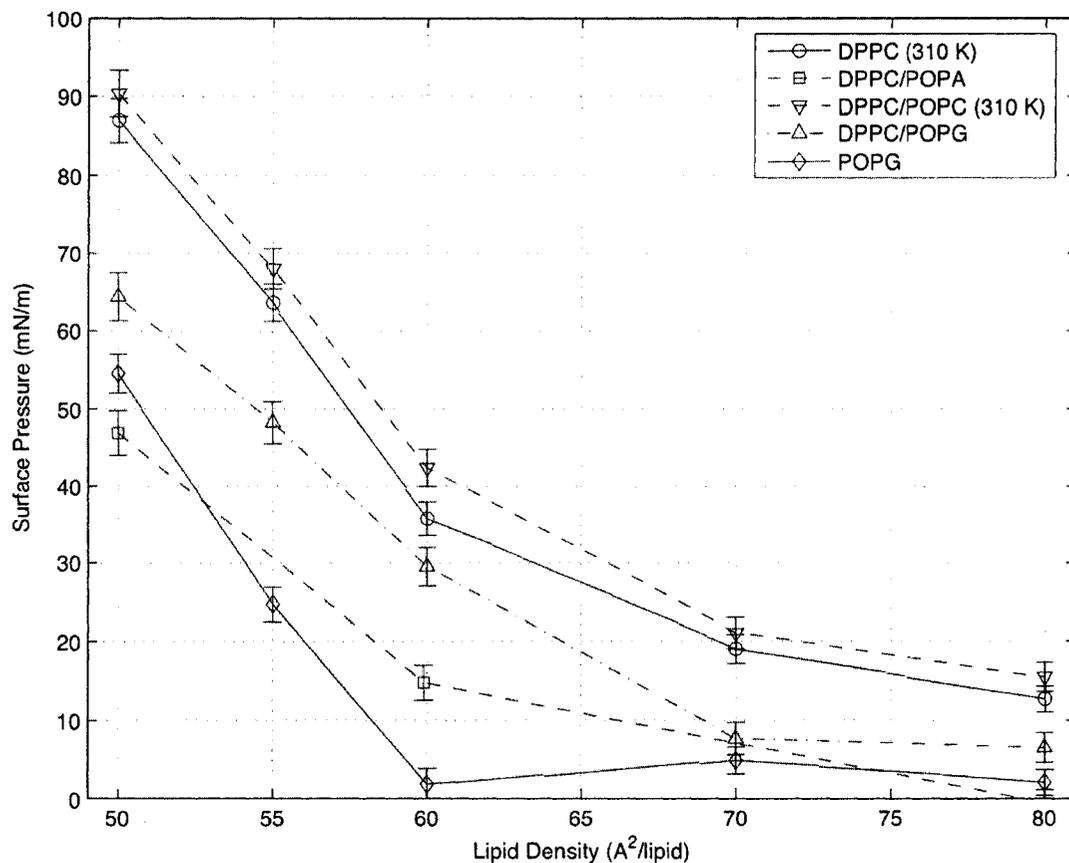


Figure 5.22: Surface Pressures

The surface tensions from Figure 5.19 have been converted to surface pressures in Figure 5.22. The results are similar to the Kaznessis results. Monolayers of DPPC (or DPPC/POPC) had higher surface pressures than the other monolayers. Surface pressure rose as lipid packing was increased. The rate of increase became greater as

the monolayers were more tightly packed.

The actual measured surface pressures were slightly lower than the Kaznessis results. This was not unexpected. The Kaznessis simulations were performed with a different force field, the CHARMM force field. Also, the Kaznessis simulations had 0.15 M NaCl added to the monolayers. The Kaznessis simulations showed that the addition of ions had an effect on the surface pressures [16].

It is worth noting that the Kaznessis group calculated a surface pressure of about 80 mN/m in the DPPC monolayer at 55 Å<sup>2</sup>/lipid. This implied a negative surface tension from Equation 5.3.

# Chapter 6

## Conclusion and Future Work

### 6.1 Conclusion

The simulations on lipid monolayers yielded some interesting results with regards to lipid behaviour. When the lipids were loosely packed the tails were spread out and tilted downwards. As the packing density increased the tails had less room to move and arranged themselves in a vertical manner. The lipid headgroups positioned themselves on top of the water layer, which remained flat.

When the lipids were tightly packed, as they would be during expiration, they exhibited behaviours that were not observed in the loosely packed monolayers. During the simulations some lipids were observed to start to rise up out of the lipid/water interface plane. At the tightest packing densities the interface exhibited undulations, which persisted for the duration of the simulation. In mixed monolayers there did not seem to be a preference to which type of lipid rose.

The rising lipids rose to different heights. Some lipids rose as high as the tips of the tails of the stationary lipids. No lipids were observed to actually leave the monolayer. It seemed that the tightly packed lipids were rearranging themselves in

three dimensions, since they did not have enough room in the original two dimensional plane.

The rising lipids observed in the simulations are consistent with theories such as the squeeze-out theory. It is possible that the rise out of the interface plane may be the first step in removing lipids from the monolayer. Proteins would then complete the process and fully remove these lipids. Also, since the proteins interact with anionic lipids [8], it is possible that they would selectively remove lipids such as POPG, leaving a DPPC enriched monolayer.

Another interpretation of the observed undulations was that the monolayers were about to collapse. Only DPPC can actually be compressed to  $40 \text{ \AA}^2/\text{lipid}$ . Such compression was not natural for the other monolayers. It would be of interest to run these simulations for a longer time to see if the undulations really are stable.

It is also important to note the presence of expansive lateral pressures in the tightly packed monolayers. It has been suggested that LS prevents alveolar by resisting the compressive forces of the exhaling lung [3]. The observed lateral pressures are consistent with this theory.

The main role of lung surfactant is to reduce surface tension. Thus the surface tension generated by the simulated monolayers was of interest. The calculated surface tension values differed from experimental measurements. Values were reasonably close for loosely packed monolayers but were different for tightly packed monolayers. Similarly, the calculated isotherm curves were close to the measured curves when the monolayers were loosely packed, but differed greatly when the monolayers were tightly packed.

There are several reasons why the simulated surface tensions would not agree with experiment. It is possible that the surface tension quantity being calculated in the simulations is not the same quantity that is being measured in experiments.

MD surface tensions are derived from the difference between calculated normal and lateral pressures. It is a localized calculation that applies to an extremely small section of monolayer. Experimental surface tensions are calculated from apparatus such as the Langmuir-Wilhelmy surface balance [5]. This apparatus measures the pull on a platinum strip suspended in LS on a layer of water [2]. This is actually a surface pressure, local to the area around the plate, which is later converted into a surface tension for the entire interface. The appropriateness of this interpretation of the relationship between surface pressure and surface tension has been questioned [3].

It is known that the existing force fields do not do a good job in modeling the surface properties of water [49]. Indeed, it was found that different force fields gave different surface tensions. Also, the cut-off methods that were used generated an error that had an effect on the surface tension calculations. There is also the question as to whether or not the force field being used is valid for tight packing densities. For example, the Berger force field parameters [28] were optimized for lipids at 50°C at a lipid density of about 60 Å<sup>2</sup>/lipid. Compressing lipids to 40 Å<sup>2</sup>/lipid may take the force field out of its applicable range.

Changing monolayer structure might also affect the surface tension calculation. It is assumed in surface tension calculations that the surface is flat. For tightly packed monolayers this did not hold true. Undulations resulted in highly curved water surfaces. Surface tension is created by the attraction of water molecules in the water layer on water molecules at the surface, and undulations would affect this attraction by having some surface water molecules further from the layer than others.

Related to this is the fact that experimental monolayers are unstable at tight packing densities. While such monolayers show near zero surface tension the measurements can be inconsistent, and it is not certain that it is a monolayer that is responsible for the low surface tension. Alternate theories have been proposed [3].

Finally, It has been suggested that at  $40 \text{ \AA}^2/\text{lipid}$  the monolayers may enter a solid phase, and that there may be more than one interface [3]. If this happens it would be necessary to rethink the notion of surface tension, assuming there is still a surface tension to speak of. Also, the parameters in the force fields, which were determined for lipids in the fluid phase, would no longer be valid.

The purpose of this MD work is to gain insight into the workings of LS on the molecular level, by modeling the atomic interactions. Comparison to experiment was proposed as a way to validate the model. Matching surface tension calculations to experimental measurements was not critical, and may not even be appropriate. What is important is the observation that expansive lateral pressures are generated, which lead to rising lipids and monolayer deformation. The model shows that theories on LS function likely have a basis in the molecular interactions.

## 6.2 Future Work

There are a number of avenues open for future work.

- Force fields need to be studied to determine the best force field or combination of force fields to use. The quality of the simulations depend on the force field. The Berger parameters have been proven to work well with lipids, but the current protein parameters are not good. A new force field, or the existing one with new protein parameters, is needed.

Related to this are the electrostatic interaction parameters used with PME. The cutoffs can be increased to give more accurate results.

- Proteins need to be added to the simulation. The lipids are not the only active component in lung surfactant. The role played by the proteins is important and

needs to be studied.

- Perform simulations with large monolayers (about 500 Lipids) to see what effect system size has.
- Increase simulation time and observe the behaviour of the lipids to see if they form stable structures.
- Implement parallelization. Parallel computing will allow for longer simulations and for simulations with larger systems.

# Bibliography

- [1] Sarker, Muzaddid. NMR Structural Studies of Lung Surfactant Protein B (SP-B) Peptides. M.Sc. Thesis. Memorial University of Newfoundland. St. John's. 2006.
- [2] Possmayer, F., Yu, S.-H., Weber, J.M. and Harding, P.G.R. Pulmonary Surfactant. *Can. J. Biochem. Cell Biol.* 62: 1121 - 1133, 1984.
- [3] Bangham, A.D. Lung Surfactant: How It Does and Does Not Work. *Lung.* 165: 17 - 25, 1987.
- [4] Pérez-Gil, J. Molecular Interactions in Pulmonary Surfactant Films. *Biology of the Neonate.* 81 (suppl 1): 6 - 15, 2002.
- [5] Veldhuizen, R., Nag, K., Orgeig, S. and Possmayer, F. The role of lipids in pulmonary surfactant. *Biochim. Biophys. Acta.* 1408: 90 - 108, 1998.
- [6] Kuchel, P.W. and Ralston, G.B. Coordinating Authors. *Schaum's Outline of Theory and Problems of Biochemistry.* McGraw-Hill Book Company. New York. 1998. Page 147.
- [7] Christie, W.W. *The Lipid Library.* 15 December 2006. 15 December 2006. <<http://www.lipidlibrary.co.uk/index.html>>

- [8] Pérez-Gil, J. and Keough, K.M.W. Interfacial properties of surfactant proteins. *Biochim. Biophys. Acta.* 1408: 203 - 217, 1998.
- [9] von Neergaard, K., Neue Auffassungen über einen Grundbegriff der Atemmechanik. Die Retraktionskraft der Lunge, abhängig von der Oberflächenspannung in den Alveolen. *Z. Gesamte Exp. Med.* 66: 373 - 394, 1929.
- [10] Prattle, R. Properties, functions and origin of the alveolar lining layer. *Nature.* 175: 1125 - 1126, 1955.
- [11] Clements, J. Surface tension of lung extracts. *Proc. Soc. Exp. Biol. Med.* 95: 170 - 172, 1957.
- [12] Banghan, A.D., Miller, N.G.A, Davies, R.J., Greenough, A. and Morley, C.J. Introductory Remarks About Artificial Lung Expanding Compounds (ALEC). *Colloids and Surfaces.* 10: 337 - 341, 1984.
- [13] Banghan, A. Breathing Made Easy. *New Scientist.* 7: 408 - 410, 1980.
- [14] Ercolessi, F. A molecular dynamics primer. June 1997. 6 December 2006. <<http://www.fisica.uniud.it/ercolessi/md/md/>>
- [15] van der Spoel, D., and Berendsen, H. J. C. Aspects of Computational Science. National Computing Facilities Foundation. The Netherlands. 1995. Pages 367 - 378.
- [16] Kaznessis, Y. N., Kim, S. and Larson, R. G. Simulations of Zwitterionic and Anionic Phospholipid Monolayers. *Biophysical Journal.* 82: 1731 - 1742, 2002.
- [17] van der Spoel, D., Lindahl, E., Hess, B., van Buuren, A. R., Apol, E., Meulenhoff, P. J., Tieleman D. P., Sijbers, A. L. T. M., Feenstra, K. A., van Drunen, R.,

and Berendsen, H. J. C. GROMACS User Manual version 3.3. [www.gromacs.org](http://www.gromacs.org). 2005.

- [18] Allen, M. P. and Tildesley, D.J. *Computer Simulation of Liquids*. Oxford University Press. Oxford. 1987. Pages 78-82.
- [19] Tieleman, D.P., MacCallum, J.L., Ash, W.L., Kandt, C., Xu, Z. and Monticelli, L. Membrane protein simulations with a united-atom lipid and all-atom protein model: lipid-protein interactions, side chain transfer free energies and model proteins. *Journal of Physics: Condensed Matter*. 18: S1221 - S1234. 2006.
- [20] Chandrasekhar, I., Kastenholz, M., Lins, R. D., Oostenbrink, C., Schuler, L. D., Tieleman, D. P. and van Gunsteren, W. F. A consistent potential energy parameter set for lipids: dipalmitoylphosphatidylcholine as a benchmark of the GROMOS96 45A3 force field. *Eur Biophys Journal*. 32: 67-77. 2003.
- [21] Schuler, L.D., Daura, X. and van Gunsteren, W.F. An Improved GROMOS96 Force Field for Aliphatic Hydrocarbons in the Condensed Phase. *Journal of Computational Chemistry*. 22: 1205 - 1218. 2001.
- [22] Weiner, S.J., Kollman, P.A., Case, D.A., Singh, U.C., Ghio, C., Alagona, G., Profeta, S. Jr, and Weiner, P. A New Force Field for Molecular Mechanical Simulation of Nucleic Acids and Proteins. *J. Am. Chem. Soc.* 106: 765-784, 1984.
- [23] Cornell, W.D., Cieplak, P., Bayly, C.I., Gould, I.R., Merz, K.M. Jr., Ferguson, D.M., Spellmeyer, D.C., Fox, T., Caldwell, J.W. and Kollman P.A. A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules. *J. Am. Chem. Soc.* 117: 5179-5197, 1995.

- [24] Brooks, B.R., Bruccoleri, R.E., Olafson, B.D., States D.J., Swaminathan, S., Karplus M. CHARMM: a program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.* 4: 187 - 217, 1983.
- [25] MacKerell, A.D. Jr., *et al.* All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins. *J. Phys. Chem. B* 102: 3586-3616, 1998
- [26] van Gunsteren, W. F. and Berendsen, H. J. C. Groningen Molecular Simulation (GROMOS) Library Manual. Biomos BV Nijenborgh 4, 9747 AG Groningen (University of Groningen), The Netherlands 1987.
- [27] Jorgensen, W. L., Tirado-Rives, J. The OPLS potential functions for proteins. Energy minimizations for crystals of cyclic peptides and crambin. *J. Am. Chem. Soc.* 110: 1657 - 1666, 1988.
- [28] Berger, O., Edholm, O. and Jähnig, F. Molecular Dynamics Simulations of a Fluid Bilayer of Dipalmitoylphosphatidylcholine at Full Hydration, Constant Pressure, and Constant Temperature. *Biophys. J.* 72: 2002 - 2013, 1997.
- [29] Bekker, H. Dijkstra. E.J., Renardus, M.K.R. and Berendsen, H.J.C. An efficient, box shape independent non-bonded force and virial algorithm for molecular dynamics. *Molecular Simulation.* 14: 137 - 151. 1995.
- [30] Bekker, H., Berendsen, H. J. C., Dijkstra, E. J., Achterop, S., van Drunen, R., van der Spoel, D., Sijbers, A., Keegstra, H., Reitsma, B., Renardus, M. K. R. GROMACS: A parallel computer for molecular dynamics simulations. In *Physics Computing 92* (Singapore, 1993). de Groot, R. A., Nadrchal, J., eds. World Scientific.

- [31] Berendsen, H. J. C., van der Spoel, D., van Drunen, R. GROMACS: A message-passing parallel molecular dynamics implementation. *Comp. Phys. Comm.* 91: 43 - 56, 1995.
- [32] Lindahl, E., Hess, B., van der Spoel, D. GROMACS 3.0: A package for molecular simulation and trajectory analysis. *J. Mol. Mod.* 7: 306 - 317, 2001.
- [33] van der Spoel, D., Lindahl, E., Hess, B., Groenhof, G., Mark, A. E., Berendsen, H. J. C. GROMACS: Fast, Flexible and Free. *J. Comp. Chem.* 26: 1701 - 1718, 2005.
- [34] Berendsen, H. J. C., Postma, J. P. M., DiNola, A., Haak, J. R. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.* 81: 3684 - 3690, 1984.
- [35] Hess, B., Bekker, H., Berendsen, H. J. C., Fraaije, J. G. E. M. LINCS: A Linear Constraint Solver for molecular simulations. *J. Comp. Chem.* 18: 1463-1472, 1997.
- [36] Miyamoto, S., Kollman, P. A. SETTLE: An Analytical Version of the SHAKE and RATTLE Algorithms for Rigid Water Models. *J. Comp. Chem.* 13: 952-962, 1992.
- [37] Ryckaert, J. P., Ciccotti, G. and Berendsen, H. J. C. Numerical integration of the cartesian equations of motion of a system with constraints; molecular dynamics of n-alkanes. *J. Comp. Phys.* 23: 327 - 341, 1977.
- [38] Ewald, P. P. Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Ann. Phys.* 64:253287, 1921.
- [39] Darden, T., York, D., Pedersen, L. Particle mesh Ewald: An N-log(N) method for Ewald sums in large systems. *J. Chem. Phys.* 98:1008910092, 1993.

- [40] Essman, U., Perela, L., Berkowitz, M. L., Darden, T., Lee, H., Pedersen, L. G. A smooth particle mesh Ewald method. *J. Chem. Phys.* 103: 8577 - 8592, 1995.
- [41] Bekker, H., Berendsen, H. J. C., Dijkstra, E. J., Achterop, S., van Drunen, R., van der Spoel, D., Sijbers, A., Keegstra, H., Reitsma, B., Renardus, M. K. R. GROMACS Method of Virial Calculation Using a Single Sum. In *Physics Computing 92* (Singapore, 1993). de Groot, R. A., Nadrchal, J., eds. World Scientific.
- [42] Chiu, S.-W., Clark, M., Balaji, V., Subramaniam, S., Scott, H.L. and Jakobsson, E. Incorporation of Surface Tension into Molecular Dynamics Simulation of an Interface: A Fluid Phase Lipid Bilayer Membrane. *Biophysical Journal.* 69: 1230 - 1245, 1995.
- [43] Tieleman, D.P. and Berendsen, H.J.C. Molecular dynamics simulations of a fully hydrated dipalmitoylphosphatidylcholine bilayer with different macroscopic boundary conditions and parameters. *J. Chem. Phys.* 105: 4871 - 4880. 1996.
- [44] Knecht, V., Müller, M., Bonn, M., Marrink, S. J. and Mark, A. E. Simulation Studies of Pore and Domain Formation in a Phospholipid Monolayer. *The Journal of Chemical Physics.* 122. 2005.
- [45] Lee, H., Kandasamy, S. K. and Larson, R. G. Molecular Dynamics Simulations of the Anchoring and Tilting of the Lung-Surfactant Peptide SP-B<sub>1-25</sub> in Palmitic Acid Monolayers. *Biophysical Journal.* 89: 3807 - 3821. 2005.
- [46] Kandasamy, S. K. and Larson, R. G. Molecular Dynamics Study of the Lung Surfactant Peptide SP-B<sub>1-25</sub> with DPPC Monolayers: Insights into Interactions and Peptide Position and Orientation. *Biophysical Journal.* 88: 1577 - 1592. 2005

- [47] Kaznessis, Y. N., Kim, S. and Larson, R. G. Specific Mode of Interaction Between Components of Model Pulmonary Surfactants Using Computer Simulations. *J. Mol. Biol.* 322: 569 - 582, 2002.
- [48] Freites, J.A., Choi, Y. and Tobias, D.J. Molecular Dynamics Simulations of a Pulmonary Surfactant Protein B Peptide in a Lipid Monolayer. *Biophysical Journal.* 84: 2169 - 2180, 2003.
- [49] Zakharov, V.V., Brodskaya, E.N. and Laaksonen, A. Surface properties of water clusters: a molecular dynamics study. *Molecular Physics.* 95:203-209, 1998.
- [50] Tieleman, P. Structures and Topologies. 15 August 2006. 17 December 2006. <[http://moose.bio.ucalgary.ca/index.php?page=Structures\\_and\\_Topologies](http://moose.bio.ucalgary.ca/index.php?page=Structures_and_Topologies)>
- [51] Zhao, W., Rog, T., Gurtovenko, A. A., Vattulainen, I., Karttunen, M. Atomic-scale structure and electrostatics of anionic POPG lipid bilayers with Na<sup>+</sup> counterions. Submitted.
- [52] Eisenberg, D. and McLachlan, A.D. Solvation energy in protein folding and binding. *Nature.* 319: 199-203, 1986
- [53] de Groot, Bert. lipid params for use with ffgmx. 14 Jan. 2003. 15 Dec. 2006. <[http://www.gromacs.org/contributed\\_by\\_users/task,doc\\_download/gid,40/](http://www.gromacs.org/contributed_by_users/task,doc_download/gid,40/)>
- [54] In-Chul Y., Berkowitz, M. L. Ewald summation for systems with slab geometry. *J. Chem. Phys.* 111: 3155 - 3162, 1999.
- [55] Hess, B., Feenstra, A. and Shrivastava, I. [gmx-users] rvdw parameters. GROMACS mailing list archive. 14 August 2002. 17 December 2006. <<http://www.gromacs.org/pipermail/gmx-users/2002-August/002320.html>>

- [56] Advanced Computational Research Laboratory. Advanced Computational Research Laboratory - Content. "chorus" (Sun V60 cluster). 16 June 2004. 27 Nov. 2006.  
<<http://acrl.cs.unb.ca/php/modules.php?name=Content&pa=showpage&pid=9>>
- [57] Advanced Computational Research Laboratory. Advanced Computational Research Laboratory - Content. Software. 7 July 2004. 27 November 2006.  
<<http://acrl.cs.unb.ca/php/modules.php?name=Content&pa=showpage&pid=23>>
- [58] van der Spoel, David and Yuhua Song. [gmx-users] mpich - mdrun problem and md using mpich. GROMACS mailing list archive. 10 August 2005. 27 November 2006. <<http://www.gromacs.org/pipermail/gmx-users/2005-August/016573.html>>
- [59] ACE-net support team. A Quick User's Guide to ACE-net. 5 July 2006. 27 November 2006. <[http://www.ace-net.ca/user\\_guide.pdf](http://www.ace-net.ca/user_guide.pdf)>
- [60] Humphrey, W., Dalke, A., Schulten, K. VMD - Visual Molecular Dynamics. J. Molec. Graphics. 14.1: 33-38, 1996.
- [61] Mingotaud, AF., Mingotaud, C. and Patterson, L.K. Handbook of Monolayers Volume 1. Academic Press Inc. London. 1993.
- [62] Lindahl, E. Message to GROMACS mailing list . [gmx-users] Re: DPPC lipid force field difference between upload one, benchmark one, and lipid.itp. 2003.
- [63] Lindahl, E. Readme file included with ffG43a2x software. Uploaded by Erik Lindahl. 2001.
- [64] Lindahl, E. Message to GROMACS mailing list . [gmx-users] aliphatic hydrocarbon parameters. 2002.

- [65] van der Spoel, David. [gmx-users] parallel problem. GROMACS mailing list archive. 3 Sept. 2003. 27 Nov. 2006. <<http://www.gromacs.org/pipermail/gmx-users/2003-September/006922.html>>.
- [66] Carsten Kutzner and Michael Haverty. [gmx-developers] GROMACS 3.3.1 parallel benchmarking. GROMACS mailing list archive. 15 August 2006. 27 November 2006. <<http://www.gromacs.org/pipermail/gmx-developers/2006-August/001759.html>>.
- [67] Dongarra, J.J., Duff, I.S., Sorensen, D.C. and van der Vorst, H.A. Numerical Linear Algebra for High-Performance Computers. Society for Industrial and Applied Mathematics. Philadelphia. 1998.
- [68] Kaznessis, Y. N. and Larson, R. G. Lung Surfactants: A Molecular Perspective from Computation. In Lung Surfactant Function and Disorder (Boca Raton, 2005). Nag, K., ed. Taylor & Francis.

# Appendix 1 - LINCS Algorithm

The LINCS algorithm demonstrates the optimized numerical algorithms that have gone into the development of GROMACS. Positions and forces in GROMACS are stored in vectors. Calculations are done on these vectors. The LINCS algorithm is based on matrices but avoids matrix-matrix multiplication and matrix inversion, two very time consuming operations. Instead of using these operations the LINCS algorithm has been designed to make use of much quicker matrix-vector operations.

For a system of  $N$  atoms, with positions given by a  $3N$  vector  $\mathbf{r}(t)$ , the equation of motion is

$$\frac{d^2\mathbf{r}}{dt^2} = \mathbf{M}^{-1}\mathbf{F} \quad (1)$$

where  $\mathbf{M}$  is a  $3N \times 3N$  matrix containing the masses of the atoms, and  $\mathbf{F}$  is the  $3N$  force vector. The system is constrained by  $K$  time-independent constraint equations

$$g_i(\mathbf{r}) = |\mathbf{r}_{i_1} - \mathbf{r}_{i_2}| - d_i = 0 \quad (2)$$

where  $d_i$  is the length of the bond between atoms  $i_1$  and  $i_2$ . Define the matrix  $\mathbf{B}$  by

$$B_{hi} = \frac{\partial g_h}{\partial r_i}. \quad (3)$$

$\mathbf{B}$  is a  $K \times 3N$  matrix that contains the directions of the constraints. It can be shown that, at time step  $n$ , for the first step of the LINCS algorithm

$$\begin{aligned}\mathbf{r}_{n+1}^{Step1} &= \mathbf{r}_{n+1}^{unc} - \mathbf{M}^{-1}\mathbf{B}_n^T (\mathbf{B}_n\mathbf{M}^{-1}\mathbf{B}_n^T)^{-1} (\mathbf{B}_n\mathbf{r}_{n+1}^{unc} - \mathbf{d}) \\ &= (\mathbf{I} - \mathbf{T}_n\mathbf{B}_n)\mathbf{r}_{n+1}^{unc} + \mathbf{T}_n\mathbf{d}\end{aligned}\quad (4)$$

where  $\mathbf{r}_{n+1}^{unc}$  is the vector of positions after the unconstrained update and we define the matrix  $\mathbf{T}_n$  as

$$\mathbf{T}_n = \mathbf{M}^{-1}\mathbf{B}_n^T (\mathbf{B}_n\mathbf{M}^{-1}\mathbf{B}_n^T)^{-1}.\quad (5)$$

For the second step let  $l_i$  be the bond length after step one ( $l_i = d_i \cos \theta$ , Figure 2.2) and define

$$p_i = \sqrt{2d_i^2 - l_i^2}.\quad (6)$$

Then

$$\mathbf{r}_{n+1}^{Step2} = (\mathbf{I} - \mathbf{T}_n\mathbf{B}_n)\mathbf{r}_{n+1}^{Step1} + \mathbf{T}_n\mathbf{p}.\quad (7)$$

The calculation of position starts with the vectors  $\mathbf{r}$  and  $\mathbf{p}$  and moves left. Each multiplication of a vector with a matrix results in a vector, and thus the calculation uses only matrix-vector multiplication. It remains to determine  $(\mathbf{B}_n\mathbf{M}^{-1}\mathbf{B}_n^T)^{-1}$ .  $\mathbf{B}_n\mathbf{M}^{-1}\mathbf{B}_n^T$  is a  $K \times K$  matrix with  $1/m_{i_1} + 1/m_{i_2}$  on the diagonal. The only non-zero elements off the diagonal occur when two bonds are connected. To calculate  $\mathbf{T}$  we introduce a  $K \times K$  matrix  $\mathbf{S}$ , which is the inverse square root of the diagonal of  $\mathbf{B}_n\mathbf{M}^{-1}\mathbf{B}_n^T$ ,

$$\mathbf{S} = \text{Diag} \left( \sqrt{\frac{1}{m_{1_1}} + \frac{1}{m_{1_2}}}, \dots, \sqrt{\frac{1}{m_{K_1}} + \frac{1}{m_{K_2}}} \right).\quad (8)$$

Then

$$\begin{aligned}
(\mathbf{B}_n \mathbf{M}^{-1} \mathbf{B}_n^T)^{-1} &= \mathbf{S} \mathbf{S}^{-1} (\mathbf{B}_n \mathbf{M}^{-1} \mathbf{B}_n^T)^{-1} \mathbf{S}^{-1} \mathbf{S} \\
&= \mathbf{S} (\mathbf{S}^{-1} \mathbf{B}_n \mathbf{M}^{-1} \mathbf{B}_n^T \mathbf{S})^{-1} \mathbf{S} \\
&= \mathbf{S} (\mathbf{I} - \mathbf{A}_n)^{-1} \mathbf{S}
\end{aligned} \tag{9}$$

The matrix  $\mathbf{A}_n$  is sparse and has zeros on the diagonal. In a molecule with only bond constraints, such as lipids or water, the connectivity is very low and as a result the eigenvalues of  $\mathbf{A}_n$  will all be less than one. The inverse can then be expanded as

$$(\mathbf{I} - \mathbf{A}_n)^{-1} = \mathbf{I} + \mathbf{A}_n + \mathbf{A}_n^2 + \mathbf{A}_n^3 + \dots \tag{10}$$

This expansion eliminates the need for matrix inversion. For most MD simulations a fourth order expansion will yield sufficient accuracy. This expansion also eliminates the need for matrix-matrix multiplication, since  $\mathbf{A}_n^{k+1} \mathbf{x} = \mathbf{A}_n^k (\mathbf{A}_n \mathbf{x})$  and  $\mathbf{A}_n \mathbf{x}$  is a vector.





