國立臺灣大學理學院物理學系

碩士論文

Department of Physics

College of Science

National Taiwan University

Master Thesis

最大似然法在時間序列資料分析的應用與全身麻醉藥（三氟氯溴乙烷）在細胞膜內的分子動力學模擬

Maximum Likelihood Method in Time Series Data Analysis and Molecular Dynamics Simulations of a General Anesthetic inside a Membrane

涂楷旻
Kai-Ming Tu

指導教授：龐寧寧 博士
Advisor: Ning-Ning Pang, Ph.D.

共同指導教授：梁國淦 博士
Coadvisor: Kuo-Kan Liang, Ph.D.

中華民國 97 年 7 月
July, 2008

Acknowledgments

I am very grateful to my supervisor, Prof. Pang for her guidance, to Dr. Liang for his kind assistance, to Dr. Chau for his patience and help, to Prof. Tzeng and Prof. Lu for their advice and to Prof. Matubayasi for his clear explanation of his method. Also, I would like to thank my family and my girlfriend for their unwavering support.

For the second part of this thesis, Dr. Chau wrote the program and did most of the calculations for 1 atm and 400 atm, and Prof. Matubayasi did all the analysis from energy distribution histograms to energy values. This work has been accepted for publication in *Chemical Physics Letters* on July 11th, 2008.

致謝

非常感謝我的指導老師，龐教授對我的諄諄教誨，梁老師給我的慷慨協助，周博士對我的耐心幫忙，兩位口試委員曾教授及陸教授提供的寶貴意見，以及松林教授對於他發展的方法的詳細解釋。我也要感謝我的家人以及女朋友對我不變的支持，你們是我前進的力量。

此篇論文第二部份裡，程式的撰寫以及 1 大氣壓、400 大氣壓大部分的計算是由周博士完成，而松林教授則負責由能量分佈得出自由能的分析。這部份的工作已在 2008 年 7 月 11 日 被 *Chemical Physics Letters* 接受.

Abstract

Two topics are covered by this thesis: the maximum likelihood method and molecular dynamics simulations of general anesthetics, halothane.

For the first topic, I introduce the maximum likelihood method and its application in analyzing time series data, and then test it with simulated data.

For the second topic, I introduce molecular dynamics (MD) simulations and the method of energy representation to general anesthetic research. Using the MD simulations and the method of energy representation, I have calculated the free-energy change of inserting a halothane molecule into different depths of a hydrated dimyristoylphosphatidylcholine (DMPC) bilayer at pressures of 1 atm, 200 atm and 400 atm. It is found that halothane preferentially resides in the region between the headgroup and the lipid tails, between 10 Å and 15 Å from the centre of the membrane. It is also found that pressure has no detectable effect on the free-energy change of inserting a halothane from bulk water to DMPC, and does not change the regional preference of halothane, either.

Keywords: maximum likelihood, data analysis, energy representation, molecular dynamics simulations, halothane, DMPC, free-energy change

摘要

此篇論文包含兩個部份：最大似然法以及全身麻醉藥（三氟氯溴乙烷）的分子動力學模擬。

第一部份，介紹最大似然法在分析時間序列資料上的應用，並且以電腦模擬的資料加以測試。

第二部份，介紹分子動力學模擬以及「交互作用能量泛函法」應用在全身麻醉藥的研究。利用分子動力學模擬以及交互作用能量泛函法，我計算了在不同壓力下將三氟氯溴乙烷（Halothane）溶入 dimyristoylphosphatidylcholine（DMPC）水合物的自由能改變，壓力分別為 1 大氣壓、200 大氣壓以及 400 大氣壓。結果顯示三氟氯溴乙烷偏好待在 DMPC 的前頭原子群與碳鏈尾端之間的區域，介於距細胞膜中央 10 埃至 15 埃的距離。另外也發現壓力對於三氟氯溴乙烷從純水溶入 DMPC 的自由能改變所造成的影響很小，且對於三氟氯溴乙烷的偏好區域沒有可觀的影響。

關鍵詞：最大似然法，資料分析，交互作用能量泛函法，分子動力學模擬，全身麻醉藥，三氟氯溴乙烷，自由能變化

Contents

Part I

Maximum Likelihood Method in Time Series Data Analysis

Chapter 1

Introduction

When seeking the mechanism behind Nature, we often have to analyze a series of observations ordered in time. This is called a time series.

The data in a time series have an internal structure: the observables could be random, or there may be correlation between the observables. The structure of the time series data could reveal to us the underlying mechanism driving the processes which, in turn, give rise to the observables. If we already have a number of hypotheses about the mechanism, how do we know which one is more probable and what are the values of the parameters related to this hypothesis?

In this work, we have to propose candidate mechanisms, and apply statistical methods to see which mechanism would be in the best position to give the observations. We use statistical methods to achieve this.

In a seminal paper on the application of statistics in the natural sciences, R.A. Fisher divided the problems of statistics into three types [1]:

(1) Problems of Specification. These arise in the choice of the mathematical form of the population (the term ‘population’ in this case refers to the distribution of the values of observables).

(2) Problems of Estimation. These involve the choice of methods of calculating from a sample statistical derivates, or as we shall call them statistics, which are designed to estimate the values of the parameters of the hypothetical population.

(3) Problems of Distribution. These include discussions of the distribution of statistics derived from samples, or in general any functions of quantities whose distribution is known.

The problem of generating candidate mechanisms or hypotheses is not done mathematically; it relies on humans to suggest them. The problem of proposing mathematical forms for candidate mechanisms belongs to the first problem. Once the mathematical form has been determined, the values of the parameters are evaluated, and this is the problem of the second type, that of estimation. In this work, I focus mainly on problems of this type, and the data I aim to analyze is a time series.

A time series data with an internal structure can usually be described by a probability distribution function. For example, take a time series data of an ion channel which has an open and a closed state. In this case, the data are the time durations of the open state, or open time durations. Suppose an ion channel is closed at the beginning of the time series, and remains closed for the next second, then opens for 2 s, closes for 1 s and then opens for 3 s, we obtain two opening time durations, one of 2 s and the other of 3 s (Fig. 1.1). In my research, the time series is much longer and consists of a large number of observations of open time durations. In the case of ion channels, it has been observed that the open time durations has a large variation. Its order ranges from several μs to several ms. However, for most ion channels, the probability of having a short open time duration is large, and it drops as the open time duration increases. It can be shown that the probability distribution of the open time durations follows an exponential decay.

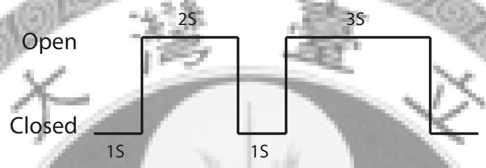


Figure 1.1: An example of time series data.

The analysis of ion channel data is, of course, not limited to quantifying the probability distribution of ion channel opening time duration. We could justifiably define the probability distribution of ion channel closed time durations. More sophisticated analysis would investigate the probability distribution of time duration of a sequential state of the channel being open and then closed. This is known as two-dimensional analysis, because if the data are plotted out, in addition to the time axis, there are two more axes, one for open time durations and the other for closed time durations. Naturally, this kind of analysis can be extended to more complicated sequential states, leading to higher-dimensional analyses.

Chapter 2

Methods

When we have a candidate mechanism to account for a time series, we propose a probability distribution for these data. We then need to know how well the probability distribution generated from this candidate mechanism agrees with the observed probability distribution. In this process, we need to determine the numerical values for the parameters of the probability distribution.

2.1 Maximum Likelihood

The probability distribution we wish to know can be written as a probability density function (pdf) p of the observables, $y_1, y_2, ..., y_n$, with some parameters, $\theta_1, \theta_2, ..., \theta_m$ which describe this probability density function:

$$p(y_1, y_2, ..., y_n | \theta_1, \theta_2, ..., \theta_m) \equiv p(\hat{y}|\hat{\theta}) \tag{2.1.1}$$

where $\hat{y}$ is an n-dimensional vector representing *one* datum and $\hat{\theta}$ an m-dimensional vector representing a set of parameters. The line between $\hat{y}$ and $\hat{\theta}$ means "$\hat{y}$ given $\hat{\theta}$", and denotes a conditional probability. In practice, however, we are given a number of data $\hat{y}$ and we wish to find what the best values for the parameters $\hat{\theta}$ are.

So R.A. Fisher introduced a quantity named "likelihood" [1]:

> The likelihood that any parameter (or set of parameters) should have any assigned value (or set of values) is proportional to the probability that if this were so, the totality of observations should be that observed.

Simply put, likelihood is a quantity to show how likely, in a relative sense, a set of parameters are responsible for a set of observed data. If the total number of data are k, and $\hat{y}^i$ denotes the ith datum, we can define likelihood as:

$$\begin{aligned} L(\hat{\theta}|\{\hat{y}^i\}) &\equiv p(\{\hat{y}^i\}|\hat{\theta}) \\ &= p(\hat{y}^1|\hat{\theta}) \times p(\hat{y}^2|\hat{\theta}) \times ... \times p(\hat{y}^k|\hat{\theta}) \\ &= \prod_{i=1}^{k} p(\hat{y}^i|\hat{\theta}) \end{aligned} \tag{2.1.2}$$

where $\{\hat{y}^i\}$ denotes a set of data, $i = 1, ..., k$.

Given a set of observed data $\{\hat{y}^i\}$, and a postulated pdf $p(\hat{y}|\hat{\theta})$, with undetermined parameters $\hat{\theta}$, the most probable values of those parameters will maximize the likelihood $L(\hat{\theta}|\{\hat{y}^i\})$. Hence we can find the most probable values of the parameters by maximizing its likelihood.

In practice, we often maximize the logarithm of the likelihood or log-likelihood, $l(\hat{\theta}|\{\hat{y}^i\})$, instead of the likelihood because of the less computing time cost in doing addition than doing multiplication. The log-likelihood is defined as:

$$l(\hat{\theta}|\{\hat{y}^i\}) \equiv \log L(\hat{\theta}|\{\hat{y}^i\}) = \sum_{i=1}^{k} \log p(\hat{y}^i|\hat{\theta}) \tag{2.1.3}$$

2.2 Maximization of Functions

In this work the downhill simplex method [2, 3] was used to maximize the log-likelihood function.

The problem of maximization is actually equivalent to the problem of minimization. By adding a negative sign, the maximum becomes minimum. To make the picture easier, minimization is discussed instead of maximization in this chapter.

If there are N undetermined parameters, $\theta_1 ... \theta_N$, we can say they form an N-dimensional parameter space. Each point in the parameter space represents a set of parameters and in the log-likelihood case, there exist a corresponding log-likelihood with respect to that point. What we wish to find is a point with the maximum value of log-likelihood, or equivalently, the minimum value of negative log-likelihood.

The concept of the downhill simplex method is simple. Like rolling a ball down from a hill, this method 'rolls' a simplex in an N-dimensional parameter space, so that in the end the simplex will stop at a minimum. A simplex in an N-dimensional space is a geometric object with $N + 1$ points, or vertices. For example, a simplex in two dimensions is a triangle and in three dimensions a tetrahedron. To find the minimum value of the negative log-likelihood, we first give an initial simplex, or initial guess, in the N-dimensional parameter space, then let the simplex roll. For each computational step, a simplex will try to take one of the following four actions in order to make the highest (worst) point lower (better): (a) reflection, (b) reflection and expansion, (c) contraction, (d) multiple contraction. See Fig. 2.1 for the four possible actions in a two-dimensional space, in which case the simplex is a triangle, and Fig. 2.2 for the algorithm flowchart.

The advantage of the downhill simplex method is that there is no need to evaluate the first derivatives of the function, only the values of the function itself is needed. Therefore this method can be used regardless whether the function is differentiable. In situations where we do not know the ‘landscape’ of the function, a method which does not require the function to be differentiable is more suitable and ‘safer’.

Like many other minimization methods, the downhill simplex method also suffers from a local minimum problem. The ‘rolling’ simplex can be easily ‘trapped’ by a local minimum (Fig 2.3). Unfortunately there is no simple solution to the local minimum problem. We can only use some techniques to enhance our confidence in the minimum we found. For example, we can choose several different initial guesses and perform the minimization several times to see if the results are consistent, or we can draw a profile of log-likelihood with respect to each parameter to see if the minimum we found, at least in this particular scale and resolution, is really the global minimum.

2.3 Log Likelihood Ratio Test

In most cases, we cannot suggest a specific mechanism to account for the observed data. Hence we do not have a complete pdf to describe the experimental data. All that we can do is to suggest that the hypothetical pdf has a certain mathematical form. But the pdf can be an addition of multiple components, say, an addition of three exponential terms. If we do not know how many components there are in the pdf, we do not know how many parameters we should use to evaluate the log-likelihood. Sometimes

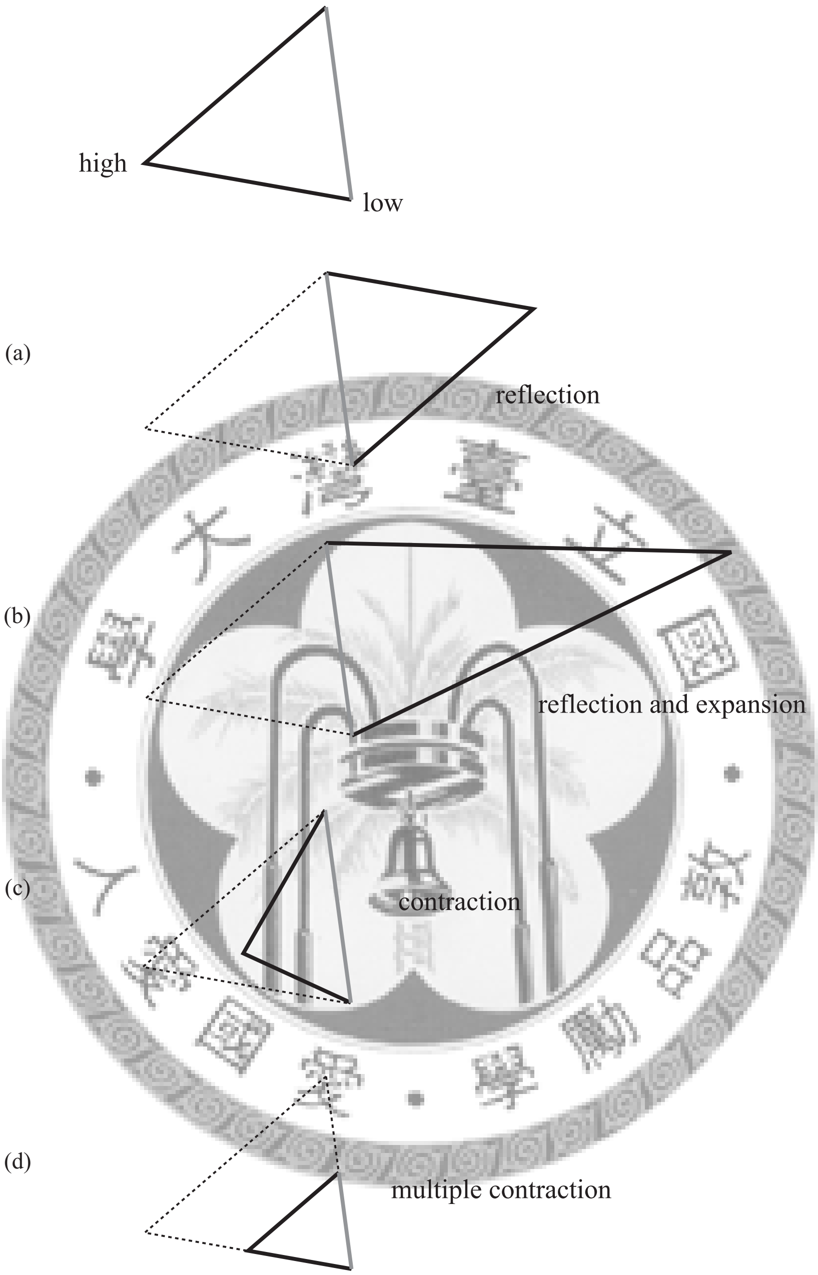


Figure 2.1: Four possible actions for a step in the downhill simplex method are shown. At the beginning of the step, the initial simplex, in this case a triangle, is shown on top. At the end of the step, the simplex can take any one of the four actions according to the log-likelihood change of the highest point, (a) a reflection away from the high point, (b) a reflection and expansion away from the high point, (c) an one-dimensional contraction from the high point, or (d) an overall contraction towards the low point.

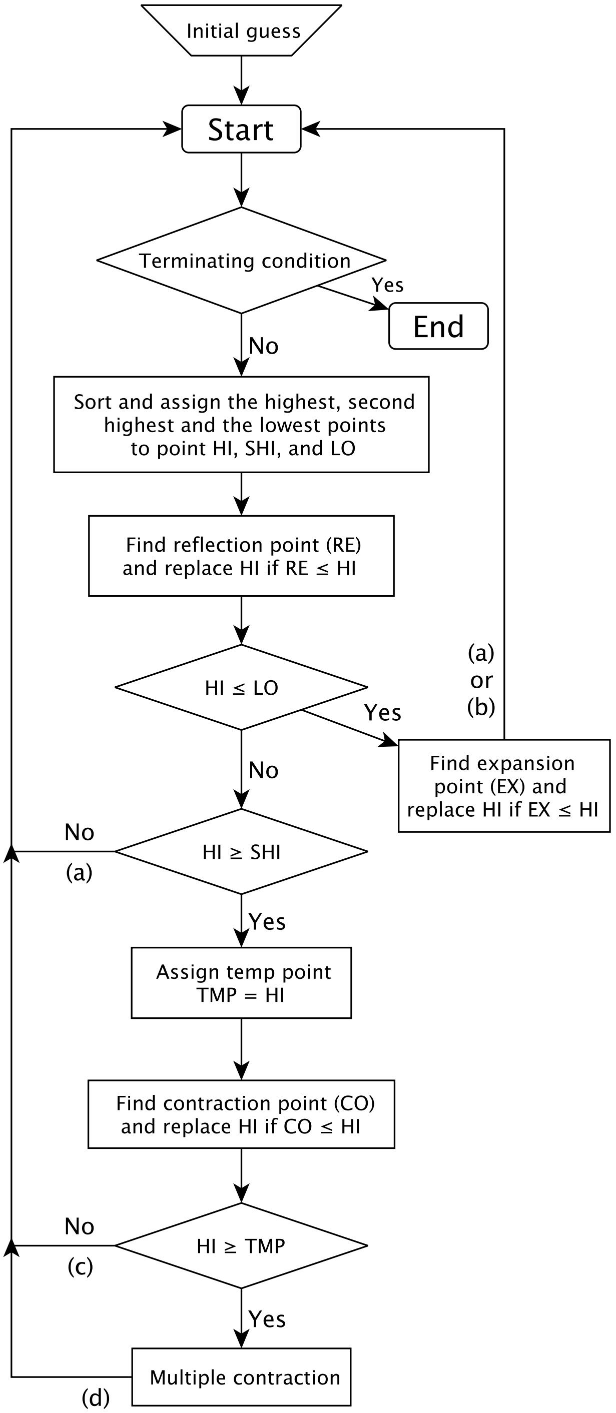

Initial guess
Start
Terminating condition
Yes
End
No
Sort and assign the highest, second highest and the lowest points to point HI, SHI, and LO
Find reflection point (RE) and replace HI if RE ≤ HI
(a) or (b)
HI ≤ LO
Yes
Find expansion point (EX) and replace HI if EX ≤ HI
No
No
(a)
HI ≥ SHI
Yes
Assign temp point TMP = HI
Find contraction point (CO) and replace HI if CO ≤ HI
No
(c)
HI ≥ TMP
Yes
Multiple contraction
(d)

Figure 2.2: The algorithm flowchart of the downhill simplex method. The terminating condition can be (1) whether the fractional change of the vector distance moved in this step is less than a pre-defined tolerance or (2) whether the fractional difference between the highest and the lowest log-likelihood is less than a pre-defined tolerance. The symbols (a) (b) (c) (d) denote the four actions.

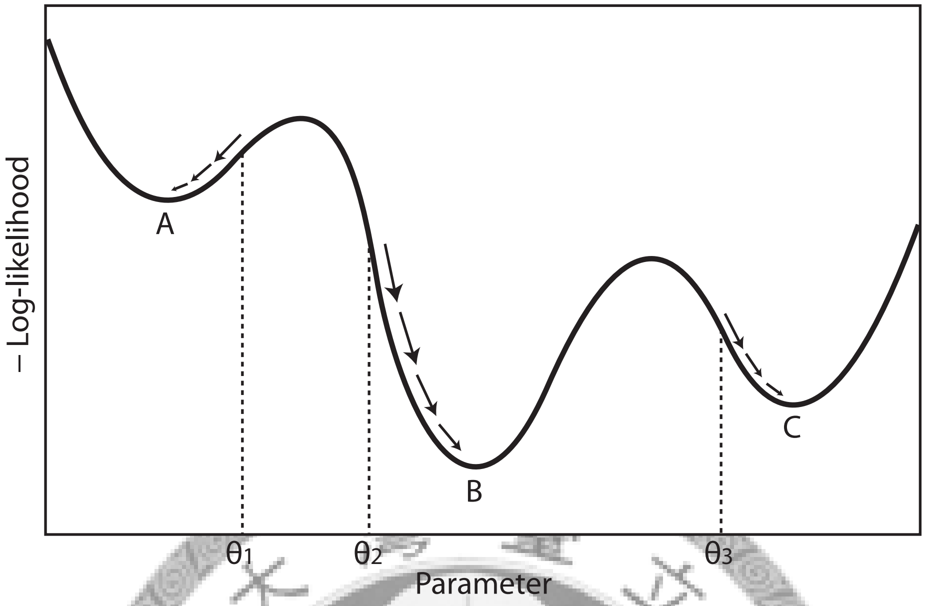


Figure 2.3: A schematic plot of the log-likelihood function with only one parameter. Point B is the global minimum while points A and C are local minima. Three initial guesses are shown. When the initial guesses are θ_1 and θ_3, the searching will be trapped at A and C respectively. Only when the searching is starting from θ_2, the global minimum B can be found.

we can determine the number of components by visually inspecting the diagram of the experimental data, but most of the time it is not that simple. What we then need is a statistical approach: the log likelihood ratio test.

The log likelihood ratio test is based on the following theorem. Let $l_1(\hat{\theta}_{n_1})$ and $l_2(\hat{\theta}_{n_2})$ be two log-likelihood functions with n_1 and n_2 parameters, respectively, and $n_2 > n_1$. Define a quantity R, called the *log likelihood ratio* (logarithm of the ratio of the likelihoods)

$$R \equiv l_2 - l_1 = \log L_2 - \log L_1 = \log\left(\frac{L_2}{L_1}\right) \ . \tag{2.3.1}$$

It can be shown [4] that:

If the most appropriate number of parameters is n_1 and the number of data

> is large, the quantity $2R$ will have a χ^2 distribution with $n_2 - n_1$ degrees of freedom.

The χ^2 distribution is explained in the next paragraph. Note that R is defined only when $L_2/L_1 > 1$, i.e. $R > 0$. This is very reasonable since the more parameters we use to fit the data, the higher the likelihood can be. From the definition, we can see that R is a quantity describing the increase of the 'goodness of fit' when the number of parameters is increased.

Before proceeding further, the χ^2 distribution should be defined. A χ^2 distribution is actually a special case of Γ distribution with parameters $\alpha = \nu/2$ and $\beta = 2$. The cumulative distribution function (cdf) of a Γ distribution is

$$F_\gamma(x; \alpha, \beta) = \frac{\gamma(\alpha, x/\beta)}{\Gamma(\alpha)} \equiv P(\alpha, x/\beta) \tag{2.3.2}$$

where $\Gamma(\alpha)$ is the gamma function, $x \geq 0$ and α, $\beta > 0$

$$\Gamma(\alpha) \equiv \int_0^\infty x^{\alpha-1} e^{-\alpha} dx \ , \tag{2.3.3}$$

$\gamma(\alpha, x)$ is the incomplete gamma function ,

$$\gamma(\alpha, x) \equiv \int_0^x x'^{\alpha-1} e^{-\alpha} dx' \tag{2.3.4}$$

and $P(\alpha, x)$ is the regularized incomplete gamma function

$$P(\alpha, x) \equiv \frac{\gamma(\alpha, x)}{\Gamma(\alpha)} \tag{2.3.5}$$

and ν, in the log likelihood case mentioned above, equals $n_2 - n_1$, which means 'the degrees of freedom'. Therefore, by inserting the parameters $\alpha = \nu/2$ and $\beta = 2$ into the cdf of a Γ distribution, Eq. (2.3.2), we have the cdf of a χ^2 distribution

$$F_{\chi^2}(\chi^2;\nu) = P\left(\frac{\nu}{2}, \frac{\chi^2}{2}\right) \tag{2.3.6}$$

By differentiating the cdf we obtain the pdf

$$p_{\chi^2}(\chi^2;\nu) = \frac{(1/2)^{k/2} x^{k/2-1} e^{-x/2}}{\Gamma(\nu/2)} \tag{2.3.7}$$

Fig. 2.4 shows the pdf and cdf of a χ^2 distribution.

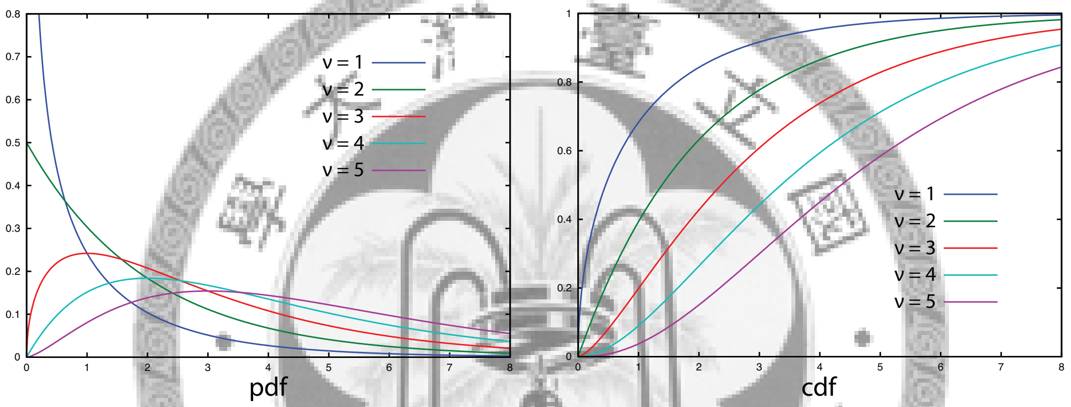


Figure 2.4: χ^2 distribution

With the value $2R$ and its χ^2 distribution, how do we use them to decide the most appropriate number of parameters ? First we need to ask the question: what does it mean by 'the most appropriate'? Since we know in general the fitting will be better and better if we use more and more parameters, it is natural to say that when the number of parameters reaches the most appropriate one, any further increase of the number of parameters will have a large probability to make only an insignificant increase in the likelihood and hence a small value of $2R$. An insignificant increase means there is a large probability that this amount of increase is only a result of chance. The value of that probability can be obtained from another function $Q(\alpha, x)$, also confusingly named the regularized incomplete gamma function, or the regularized complementary incomplete gamma function for discrimination. $Q(\alpha, x)$ is defined as:

$$Q(\alpha, x) \equiv \frac{\Gamma(\alpha, x)}{\Gamma(\alpha)} = 1 - P(\alpha, x) \tag{2.3.8}$$

where $\Gamma(\alpha, x)$ is the (complementary) incomplete gamma function

$$\Gamma(\alpha, x) \equiv \int_x^\infty x^{\alpha-1} e^{-\alpha} dx = \Gamma(\alpha) - \gamma(\alpha, x) \tag{2.3.9}$$

We can write $Q(\alpha, x)$ in the χ^2 case as

$$Q(\alpha, x) = Q\left(\frac{\nu}{2}, \frac{\chi^2}{2}\right) \equiv Q(\chi^2|\nu) \tag{2.3.10}$$

This so called χ^2 *probability function*, $Q(\chi^2|\nu)$, is the probability that an increase of the 'goodness of fit', $2R$, due to the increase of the number of parameters, ν, is only a result of chance. So when Q is small, it means that the increase of the number of parameters is necessary to give a better description of the data. Usually the value of Q is also called P value. See Fig. 2.5 for pictures of P and Q.

In practice, we often require the acceptance value of Q to be smaller than 0.05, 0.01 or even 0.001. For example, when we increase the number of parameters from four to five and obtain the log likelihood ratio $2R_1$, we have $Q(2R_1|1) = Q\left(\frac{1}{2}, R_1\right) \equiv Q_1$. We again increase the number of parameters from five to six and obtain Q_2. If we set our standard to be 0.05, and $Q_1 < 0.05$ while $Q_2 > 0.05$, this means that the most appropriate number of parameters is five. However, if Q_2 is still less than 0.05 when the number of parameters is six, then we shall need to increase the number of parameters further.

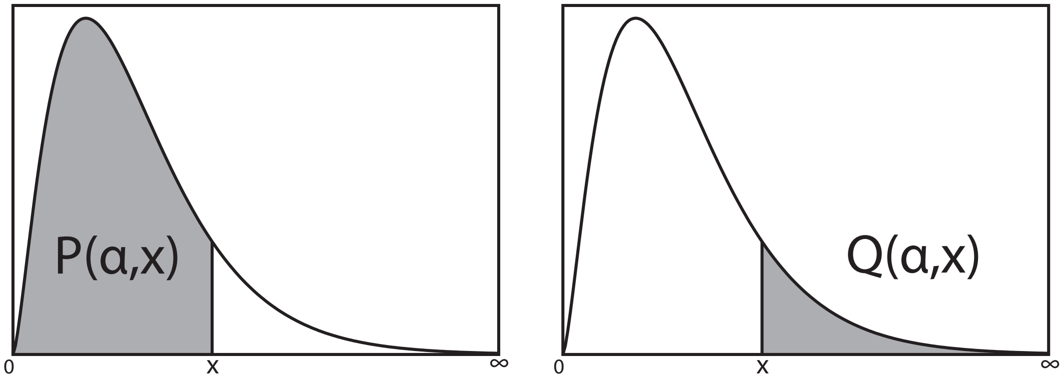


Figure 2.5: $P(\alpha, x)$ and $Q(\alpha, x)$ are actually integrations of the pdf of χ^2 distribution. The integration range of P is from 0 to x and that of Q is from x to ∞.

2.4 Generation of Non-Uniform Random Numbers

In order to test the maximum likelihood method with some simulated data, we must generate some random numbers with a certain pre-defined pdf, which is more complex than a simple uniform pdf.

Continuous uniform random (actually pseudo-random) numbers can be easily obtained by using the intrinsic random number generators of many programming languages. It is not that easy, however, to obtain non-uniform random numbers directly. We have to use some techniques to generate non-uniform random numbers out of uniform random numbers. There are two common methods for doing so, the inverse transform method [3] and the rejection method [3, 5].

2.4.1 Inverse Transform Method

The inverse transform method is based on the proposition that, given an invertible cumulative distribution function (cdf) $F(x)$ and a uniform random number U with range $(0,1)$, the random number $X \equiv F^{-1}(U)$ has a cdf $F(x)$. It can be proved as follows. Define the cdf of X as $F_x(x)$:

$$
\begin{aligned}
F_x(x) &\equiv \mathrm{Prob}(X < x) \\
&= \mathrm{Prob}(F^{-1}(U) < x) \\
&= \mathrm{Prob}(F(F^{-1}(U)) < F(x)) \\
&= \mathrm{Prob}(U < F(x)) \\
&\equiv F(x)
\end{aligned}
\tag{2.4.1}
$$

at the third equality, we have used a property of cdf that it is a monotonically increasing function.

With the knowledge above, we can generate a random number X with a cdf $F(x)$ by the following procedure:

Step 1: Find the inverse function $F^{-1}(u) = x$.

Step 2: Generate a uniform random number U.

Step 3: $X \equiv F^{-1}(U)$ is the random number we wish to find.

For example, to obtain a random number X with pdf

$$
f(x) = \frac{1}{\lambda} e^{-\frac{x}{\lambda}} \ , \tag{2.4.2}
$$

we first integrate the pdf to obtain its cdf

$$
F(x) = \int_0^x f(x')dx' = 1 - e^{-\frac{x}{\lambda}} \tag{2.4.3}
$$

and find the inverse function

$$
F^{-1}(u) = -\lambda \ln(1-u) \ , \tag{2.4.4}
$$

so we obtain

$$\begin{aligned} X &= -\lambda \ln(1-U) \\ &= -\lambda \ln(U) \ . \end{aligned} \tag{2.4.5}$$

Since U is a uniform random number with range $(0,1)$, $1-U$ is also uniform on $(0,1)$. We can therefore equivalently use $1-U$ to save computing time.

2.4.2 Rejection Method

This method is first proposed by John von Neumann in 1951 [5]. Suppose we already have a method to generate a random number with a pdf $g(x)$, and we wish to generate a random number X with a pdf $f(x)$. We can first generate a random number Y with a pdf $g(x)$ and accept this Y with a probability proportional to $f(Y)/g(Y)$. The accepted Y, defined to be X, will then have a pdf $f(x)$.

The protocol is as follows:

Step 0: Find a constant c (the lower the better) such that $\dfrac{f(x)}{g(x)} \leq c$ for all x.

Step 1: Generate Y with pdf $g(x)$.

Step 2: Generate a uniform random number U.

Step 3: If $U \leq \dfrac{f(Y)}{cg(Y)}$, $X = Y$, otherwise return to Step 1.

Note that c can never be less than 1. We can see this by integrating the inequality of Step 0, $f(x) \leq c\,g(x)$, over all values of x.

The validity of this protocol can be proved as follows. Define the pdf of X as $p_{_X}(x)$:

$$
\begin{aligned}
p_{X}(x)dx &= \text{Prob}\{X \to x + dx\} \\
&= \text{Prob}\{Y \to x + dx \mid \text{Acceptance}\} \\
&= \frac{\text{Prob}\{Y \to x + dx,\, \text{Acceptance}\}}{\text{Prob}\{\text{Acceptance}\}} \\
&= \frac{\text{Prob}\left\{Y \to x + dx,\, U \leq \dfrac{f(x)}{cg(x)}\right\}}{\text{Prob}\{\text{Acceptance}\}} \\
&= \frac{\text{Prob}\{Y \to x + dx\}\,\text{Prob}\left\{U \leq \dfrac{f(x)}{cg(x)}\right\}}{\text{Prob}\{\text{Acceptance}\}} \quad \text{by independence} \\
&= \frac{\{p_{Y}(x)dx\}\left\{\dfrac{f(x)}{cg(x)}\right\}}{1/c} \\
&= g(x)dx\frac{f(x)}{g(x)} \\
&= f(x)dx \qquad (2.4.6)
\end{aligned}
$$

where

$$
\text{Prob(Acceptance)} = \frac{\int f(x)dx}{\int cg(x)dx} = \frac{1}{c} \qquad (2.4.7)
$$

See Fig. 2.6 for some specific pictures about this method.

In order to generate the target random numbers quickly, the number of the rejection events should be as few as possible. Since the more the rejection events, the longer the time we need to collect a required number of random numbers. Therefore the efficiency of the rejection method is proportional to the probability of acceptance, $1/c$. Hence we should always try to make the constant c be as close to unity as possible.

Compared with the inverse transform method, the rejection method may be slower since it has to generate two random numbers (often several times) to get only one

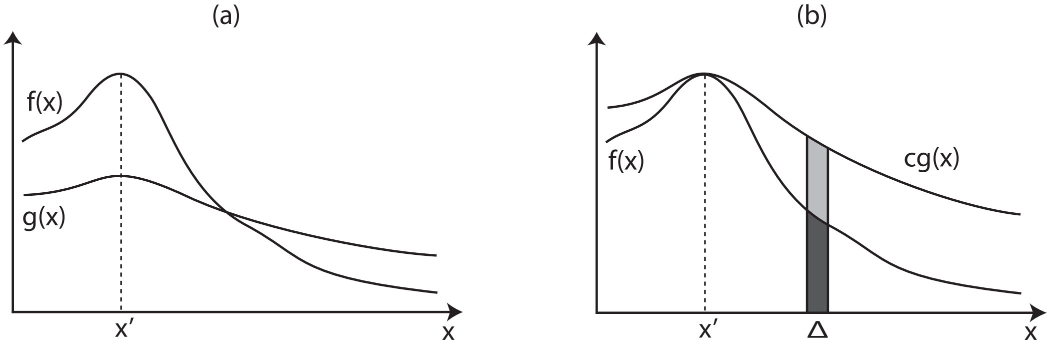


Figure 2.6: (a) $f(x)$ is the desired pdf, $g(x)$ the pdf we can already get, x' the point that makes $\frac{f(x)}{g(x)}$ maximum, i.e. $\frac{f(x)}{g(x)} \leq \frac{f(x')}{g(x')} \equiv c$. Note there is always at least one cross point between $f(x)$ and $g(x)$ because the areas of $f(x)$ and $g(x)$ are both normalized to unity. (b) The multiplier c makes the curve $cg(x)$ cover all the area under $f(x)$. The Prob$\left\{U \leq \frac{f(x)}{cg(x)}\right\}$ is equal to the ratio of the dark gray area $f(x)\Delta$, to the whole gray area $cg(x)\Delta$.

desired random number. But the rejection method is more versatile than the inversion method since the target pdf of the latter has to be an invertible function while that of the former does not.

Chapter 3

Simulation and Analysis

In this chapter, a demonstration is presented, in which the maximum likelihood method and the log likelihood ratio test were used to analyze the simulated time series data whose pdf was known.

3.1 Simulated Data

The simulated data used have a pdf with five exponential components. The pdf $p(t)$ is:

$$p(t) = \sum_{i=1}^{5} a_i p_i(t) = \sum_{i=1}^{5} a_i \left(\frac{1}{\tau_i} e^{-\frac{t}{\tau_i}} \right) \tag{3.1.1}$$

where a_i is the weight of ith component with a condition $\sum_{i=1}^{5} a_i = 1$, and τ_i is the ith characteristic time constant. The total number of independent parameters are therefore $5 \times 2 - 1 = 9$ with one dependent parameter $a_5 = 1 - \sum_{i=1}^{4} a_i$.

The procedures of the generation of the simulated data are as follows. First, one of those five exponential components is chosen with a probability proportional to its weight, a_i. The inverse transform method is then used to generate a random number obeying the exponential pdf just chosen, $\frac{1}{\tau_i}e^{-\frac{t}{\tau_i}}$. This whole process of choosing an exponential component and then generating a random number obeying that exponential pdf is repeated until the required number of random numbers has been generated. These procedures, of course, can be easily extended to generate random numbers whose pdf have arbitrary number of exponential components. In this work the uniform random numbers, which were needed by the inverse transform method, were generated by the intrinsic subroutines, RANDOM_SEED() and RANDOM_NUMBER(), of Intel® Fortran compiler.

The parameters of the simulated data used are listed in Table 3.1, and the histogram of the data is shown in Fig. 3.1a.

number of data points: 100000			
$a_1 =$	0.023	$\tau_1 =$	0.002
$a_2 =$	0.068	$\tau_2 =$	0.04
$a_3 =$	0.227	$\tau_3 =$	0.5
$a_4 =$	0.455	$\tau_4 =$	0.8
$a_5 =$	0.227	$\tau_5 =$	5

Table 3.1: The parameters of the simulated data are listed here. The values of the weights a_i's are actually assigned in a relative manner, which before normalization and rounding are 0.1, 0.3, 1, 2, 1, respectively.

3.2 Logarithmic Histograms

Histograms with linear axes such as Fig. 3.1a are actually not so informative. Another more illuminative way to plot the histograms with a logarithmic time axis was first

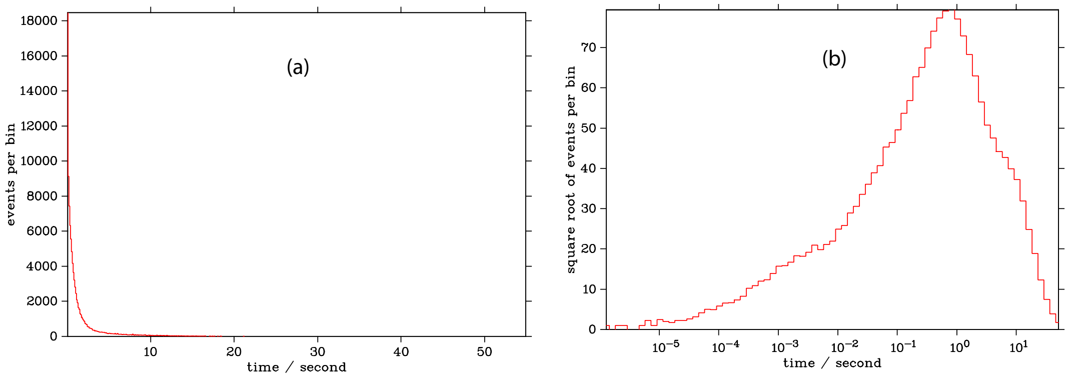


Figure 3.1: Two histograms of the same simulated data with different binning scales are shown. (a) Linear histogram. Events are binned with a constant bin width in linear scale. (b) Logarithmic histogram. Events are binned with a constant bin width in logarithmic scale and the histogram is plotted with a logarithmic x-axis and a square-root y-axis. The pdf and the corresponding parameters of the simulated data are shown in Eq. (3.1.1) and Table 3.1, respectively.

proposed by Blatz and Magleby [6] and developed further by Sigworth and Sine [7]. In this work Sigworth and Sine's method is used to plot the logarithmic histograms. With this method, histograms are plotted with bins which have a constant width in the logarithmic scale. Note that this is not the same as simply doing a logarithmic transformation to the time axis of a linear histogram, because the transformation is done *before* the events are binned.

Here is the explanation of this method. Considering a pdf with n components:

$$p(t) = \sum_{i=1}^{n} \frac{a_i}{\tau_i} e^{-\frac{t}{\tau_i}} \tag{3.2.1}$$

where the symbols are defined in the same way as Eq. (3.1.1). A logarithmic transformation to the time axis is:

$$x = \log_{10}(t) \tag{3.2.2}$$

Here base 10 was used for consistency. First we note that if $t < t'$, $\log_{10}(t) < \log_{10}(t')$.

Therefore the cdf becomes

$$\begin{aligned} F(t) &\equiv \text{Prob}\{t < t'\} \\ &= \text{Prob}\{\log_{10}(t) < \log_{10}(t')\} \\ &= \text{Prob}\{x < x')\} \\ &\equiv F_{\log}(x) \end{aligned} \tag{3.2.3}$$

where $F_{\log}(x)$ is the cdf with a logarithmic time axis. In other words, we found that the cdf's of t and x are the same. Finally, the pdf of x, $p_{\log}(x)$ can be obtained by differentiating the cdf $F_{\log}(x)$ with respect to x:

$$\begin{aligned} p_{\log}(x) &\equiv \frac{dF_{\log}(x)}{dx} \\ &= \frac{dF(t)}{d\log_{10}(t)} \\ &= \frac{dt}{d\log_{10}(t)}\frac{dF(t)}{dt} \\ &= \sum_{i=1}^{n} t\ln(10)\ p(t) \\ &= \sum_{i=1}^{n} \frac{a_i \ln(10)}{\tau_i}\ \exp\left(\ln t - \frac{t}{\tau_i}\right) \\ &= \sum_{i=1}^{n} \frac{a_i \ln 10}{\tau_i}\ \exp\left(\ln 10^x - \frac{10^x}{\tau_i}\right) \end{aligned} \tag{3.2.4}$$

The shape of $p_{\log}(x)$ when $n = 1$ is like a skewed-bell. If $n > 1$ the shape will be a superposition of each component's shape (Fig 3.1b). The advantage of this kind of histograms is that the peak of each component's shape indicates the corresponding characteristic time constant of that component. This can be seen by differentiating $p_{\log}(x)$ with respect to x and equating it to zero.

In addition to the transformation of the time axis, Sigworth and Sine also changed the ordinate of the histograms from the linear scale to the square-root scale. It is done by simply taking the square-root of the events number *after* they are binned. This change of the ordinate makes the standard deviation for each bin equal. [7]

3.3 Data Analysis

After generating 100000 simulated time series data, the test analysis was performed without using the pdf as part of the input. It was tested how the proposed method could recover the underlying patterns of the simulated time series data.

Initially I assessed the logarithmic histogram of the simulated data to give a rough solution (Solution 1). The other three solutions (Solutions 2, 3 and 4) were obtained by simply adding more parameters with some deviation from the existing parameters. Using these four rough solutions as the initial guesses of the maximum likelihood, four optimized solutions with respect to each number of components were obtained. The results are shown in Fig. 3.2 and Table 3.3. From Fig. 3.2 we see that the data were not well fitted with three components, but they seemed to be better fitted with four, five and six components. So which one was the most appropriate choice for describing the simulated data? With the help of the log likelihood ratio test, we can see from Table 3.2 that the most appropriate number of components should be five. When the number of components increased from five to six, the fit slightly improved. However, the probability of this improvement being due to chance became much larger. This means that the increase of the number of components from five to six was unnecessary. In contrast, the other two increases (from three to four, and then from four to five) were necessary because the probability of the improvement of the fit being due to chance was negligibly small. Hence this demonstration also showed the advantage of using the log likelihood ratio test: even if all the fitting histograms seem very good, the log likelihood ratio test can still give us quantitative indexes to help us make judgement.

It is found that the most appropriate number of components is five. The optimized values of the corresponding parameters are listed in Table 3.3, Solution 3. We can compare the original parameters with the parameters determined from the simulated data. In the following discussion, we define the parameter values we originally put into the simulation to be the 'original values', and we define the parameter values obtained from the simulated data using the maximum likelihood method the 'recovered values'. It can be seen that, for the first, second and fifth components, the difference between the original values and the recovered values are small (<5% of each other). However, for the third and fourth components, the difference between the original and recovered values can be larger than 20%. Note also that the characteristic time constants for the first, second and fifth components are very different from each other; they are orders of magnitude apart. The characteristic time constants for the third and fourth components are more or less the same. This makes it difficult to separate the third and fourth components. This is a general problem of data analysis, and is not an indication of any specific weakness of this method.

The other difficulty in using this method arises when the number of parameters is large, say 15 or 20. Under these conditions, the maximization of the log-likelihood will often become unstable. In other words, different initial guesses will easily give wildly different maximization points. However, the problem is not in the maximum likelihood method itself, it is in the maximization. The maximization can be trapped in a local maximum. Future work includes investigating other maximization methods to see if they will make the maximum likelihood method more stable. The Monte Carlo method, for example, allows the searching to jump out of the local maximum and hence can avoid the local maximum problem.

components	R	P value
$3 \to 4$	647.934	4.034×10^{-282}
$4 \to 5$	18.649	7.959×10^{-9}
$5 \to 6$	0.150	8.603×10^{-1}

Table 3.2: The results of the log likelihood ratio test. The first column is the change of the number of components, the second column the log likelihood ratio and the third column the value of the χ^2 probability function $Q(\chi^2 = 2R|\nu = 2)$. Note that the degrees of freedom ν are the number of the increased independent parameters, which are 2 for all three cases here.

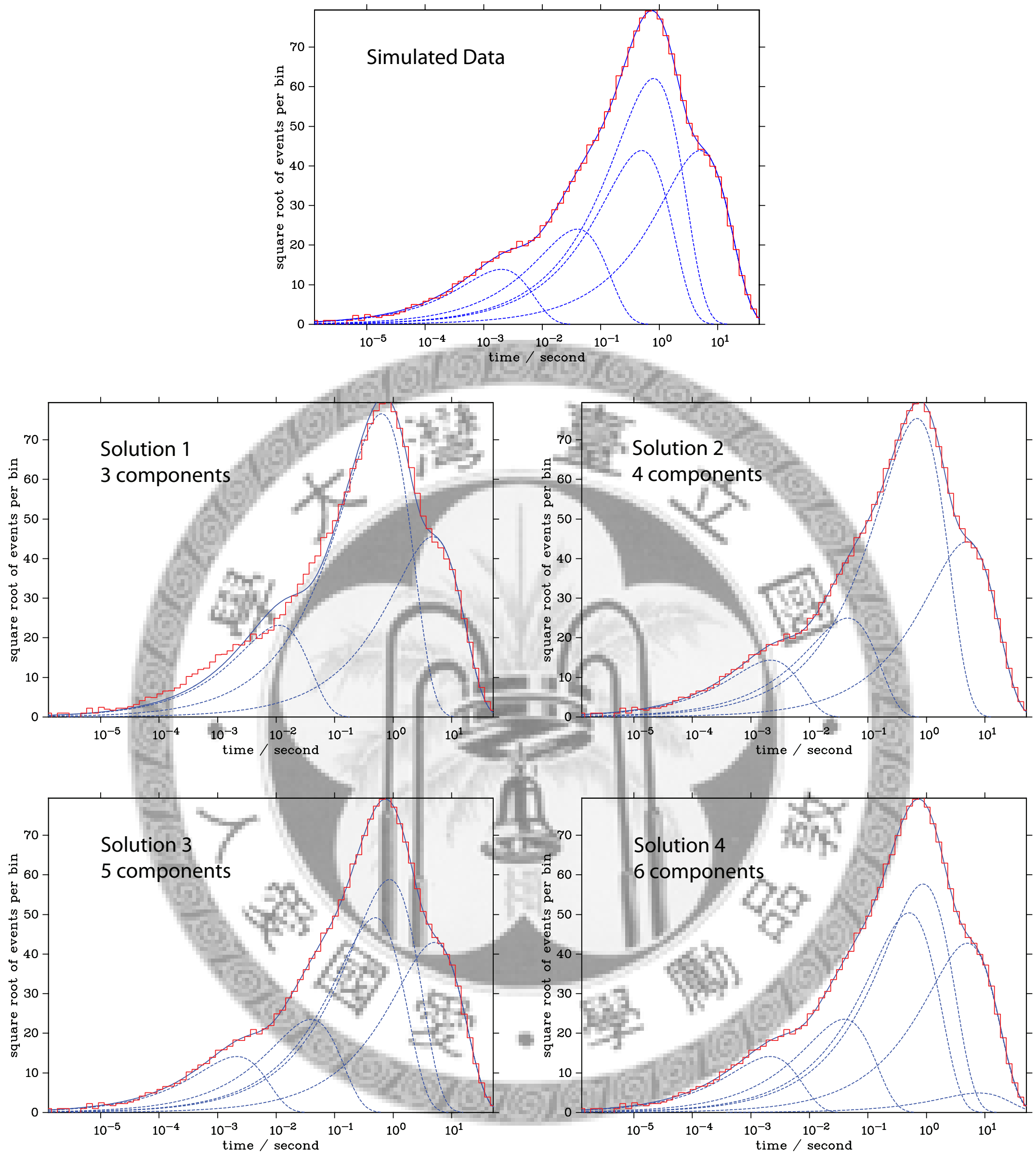


Figure 3.2: The logarithmic histogram of the simulated data is shown on top. The other four histograms are obtained by the maximum likelihood method with different solutions of number of components. In these five histograms, the red step-lines all denote the distribution of the events and the superimposed blue solid lines are the sum of the blue dashed lines. In the top histogram the blue dashed lines denotes the five pdf components of the simulated data, while in the other four histograms the blue dashed lines are the pdf components obtained by the maximum likelihood method.

Simulated data			
$a_1 =$	0.023	$\tau_1 =$	0.002
$a_2 =$	0.068	$\tau_2 =$	0.04
$a_3 =$	0.227	$\tau_3 =$	0.5
$a_4 =$	0.455	$\tau_4 =$	0.8
$a_5^* =$	0.227	$\tau_5 =$	5

Solution 1 : 3 components

initial guess				results			
$a_1 =$	0.167	$\tau_1 =$	0.01	$a_1 =$	0.063	$\tau_1 =$	0.011
$a_2 =$	0.5	$\tau_2 =$	1	$a_2 =$	0.692	$\tau_2 =$	0.633
$a_3^* =$	0.333	$\tau_3 =$	10	$a_3^* =$	0.245	$\tau_3 =$	4.7
Log-likelihood =				−118321.732			

Solution 2 : 4 components

initial guess				results			
$a_1 =$	0.133	$\tau_1 =$	0.01	$a_1 =$	0.024	$\tau_1 =$	0.002
$a_2 =$	0.2	$\tau_2 =$	0.1	$a_2 =$	0.074	$\tau_2 =$	0.045
$a_3 =$	0.4	$\tau_3 =$	1	$a_3 =$	0.671	$\tau_3 =$	0.689
$a_4^* =$	0.267	$\tau_4 =$	10	$a_4^* =$	0.231	$\tau_4 =$	4.87
Log-likelihood =				−117673.798			

Solution 3 : 5 components

initial guess				results			
$a_1 =$	0.118	$\tau_1 =$	0.01	$a_1 =$	0.024	$\tau_1 =$	0.002
$a_2 =$	0.176	$\tau_2 =$	0.1	$a_2 =$	0.065	$\tau_2 =$	0.039
$a_3 =$	0.118	$\tau_3 =$	0.2	$a_3 =$	0.285	$\tau_3 =$	0.495
$a_4 =$	0.353	$\tau_4 =$	1	$a_4 =$	0.408	$\tau_4 =$	0.867
$a_5^* =$	0.235	$\tau_5 =$	10	$a_5^* =$	0.218	$\tau_5 =$	5.033
Log-likelihood =				−117655.149			

Solution 4 : 6 components

initial guess				results			
$a_1 =$	0.105	$\tau_1 =$	0.01	$a_1 =$	0.024	$\tau_1 =$	0.002
$a_2 =$	0.158	$\tau_2 =$	0.1	$a_2 =$	0.065	$\tau_2 =$	0.039
$a_3 =$	0.105	$\tau_3 =$	0.2	$a_3 =$	0.300	$\tau_3 =$	0.504
$a_4 =$	0.105	$\tau_4 =$	1	$a_4 =$	0.392	$\tau_4 =$	0.872
$a_5 =$	0.316	$\tau_5 =$	5	$a_5 =$	0.216	$\tau_5 =$	4.982
$a_6^* =$	0.211	$\tau_6 =$	10	$a_6^* =$	0.003	$\tau_6 =$	7.833
Log-likelihood =				−117654.999			

Table 3.3: The parameters of the simulated data are shown in the top table for comparison. The other four tables contain the initial guesses and the corresponding maximum likelihood results. *These a_n's are not free parameters and are obtained by the equation: $a_n = 1 - \sum_{i=1}^{n-1} a_i$.

Chapter 4

Conclusion

In this work, the maximum likelihood method is described, which is a powerful tool for time series analysis. It can provide the user with numerical results to describe the observed events. However, experiments are still required to define the actual physical processes responsible for these events. Unfortunately, limitation of time does not allow me to apply this method on real experimental data. Future work would include using this method on real data to further study its performance.

Part II

Molecular Dynamics Simulations of a General Anesthetic inside a Membrane

Chapter 5

Introduction

General anesthesia (GA) is a state of complete unconsciousness caused by drugs. It was first publicly demonstrated by Morton in Massachusetts General Hospital in 1846. The first general anesthetics used were chloroform and ether. However, the former was quite toxic and the latter was highly inflammable. Later, safer alternatives such as isoflurane and enflurane were developed, and they remain some of the most commonly used drugs in clinical medicine. Unlike regional or local anesthesia, the exact mechanism of GA is still an unsolved problem.

At the turn of the 19th and 20th centuries, Meyer and Overton independently proposed a hypothesis which described the correlation between the efficacy of an anesthetic and its solubility in lipids. This hypothesis, which states that the logarithm of the efficacy of a general anesthetic was proportional to the logarithm of its lipophilicity, was subsequently called the Meyer-Overton rule. This rule suggests a lipid-rich region of the body was involved in GA.

There was little progress in elucidating the mechanism of action of general anesthetics until the middle of the 20th century. Johnson and Flagler [8] placed tadpoles in a container and administered ethanol to them. The animals became anesthetized and fell to the bottom of the container at atmospheric pressure. When the pressure was raised to over 100 atm, they observed that GA was reversed, and the tadpoles started to swim. This was repeated on a number of living organisms and on a large variety of general anesthetics [9, 10, 11]. The only report of pressure reversal in humans appeared in 1979 [12].

Pressure reversal is not a universal phenomenon, as species variation has been observed [13]. For example, Paton and his co-workers discovered that in the common frog, *Rana temporaria*, in the presence of general anesthetics, activity increased as pressure increased. This is what we would expect from pressure reversal. However, in the freshwater shrimp, *Gammarus pulex*, increased pressure reduced the swimming activity in the presence of general anesthetics.

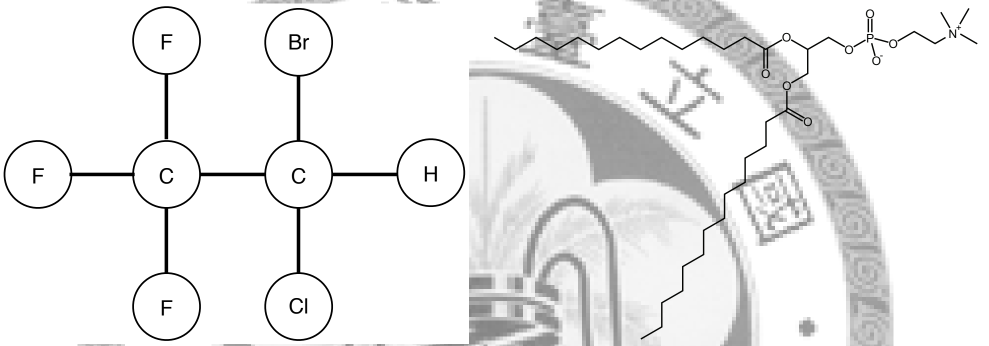
J.R. Trudell et al. spin-labelled phosphatidylcholine, mixed them with water and organic solvents, sonicated the mixture and produced a translucent vesicle suspension. Halothane was added to this suspension at concentrations of 49mmol, 147mmol or 490mmol per mole of lipid, whilst the concentration of methoxyfluorane was 58mmol, 174mmol or 580mmol per mole of lipid. Electron spin resonance (ESR) spectra were measured, and they showed that anisotropic motion of phosphatidylcholine within the phospholipid bilayer was increased. There was a concomitant decrease of the order parameter $S'{}_n$ as the concentration of anesthetic increased [14]. On application of pressure up to 274 atm by increasing helium, a non-anesthetic gas at these pressures, in the container, these changes were reversed: S'_n increased and the spectra shifted back [15]. The mechanism of this change, however, remained unclear, but the most likely cause seemed to be restriction of motion caused by phospholipid molecules coming closer together.

The results of these experiments lead to two conclusions: the cell membrane is probably involved in general anesthetic action, and ambient pressure affects general anesthetic action. To clarify the effect of pressure on the distribution of general anesthetics in the cell membrane, many scientists have performed experiments and simulations to determine the location of general anesthetics in the membrane.

K. Tu et al. [16] performed a 1.6-ns simulation of 64 DPPC molecules hydrated in 1792 water molecules, with 4 halothane molecules in this system. After equilibration, they found that halothane was preferentially located about 10 Å from the center of the membrane. The halothane molecules exhibited no orientational preference. Koubi et al. [17] performed a 2-ns simulation of 64 DPPC molecules hydrated in 1792 water molecules, but with 32 halothane molecules. After equilibration, they found that halothane was preferentially located about 10 Å from the center of the membrane. However, the concentration of halothane used in these studies were several times in excess of the concentration of halothane used in clinical work.

Subsequently, Pickholz et al. [18] developed a coarse-grained model, and applied it to simulate hydrated DOPC at different halothane concentrations. Their results also showed that halothane was always preferentially located at about 10 Å from the center of the membrane. They also evaluated the potential of mean force of extracting a halothane from its equilibrium position into the solution. They found that this was largest when the halothane was just below the headgroup of the phospholipid.

However, none of them have explored the effect of pressure. In this work, we performed a series of molecular dynamics simulations on a model membrane, DMPC, and a model general anesthetic, halothane. The chemical structure of these two molecules are shown in Fig. 5.1(a) and Fig. 5.1(b). We evaluated the free energy change of inserting halothane into a membrane of fully hydrated DMPC at different depths, and at different pressures. The following diagrams show the molecules used in our study:



(a) Halothane (b) Dimyristoylphosphatidylcholine

Figure 5.1: (a) Halothane, $CF_3CHClBr$, IUPAC name 1,1,1-trifluoro-2-chloro-2-bromo-ethane, is a nonflammable, halogenated, hydrocarbon anesthetic [19]. It is in liquid state and volatile at room temperature. (b) Dimyristoylphosphatidylcholine, a phospholipid which consists of polar head and two non-polar tails.

Chapter 6

Methods

Methods used in this work are explained in this chapter. We used the molecular dynamics (MD) simulation method to generate samples of the configurations of the solute (halothane) and solvent (hydrated DMPC bilayer) system. Then we apply the method of energy representation of solution theory (ERnST) to calculate the solvation free energy of halothane into the lipid bilayer or in bulk water. To carry out MD simulations, we used the program DL_POLY version 2.15, developed by the Daresbury Laboratory, UK [20]. In the following sections, the methods of molecular dynamics (MD) simulations employed in the DL_POLY program is firstly introduced. These include the Verlet integration scheme, the empirical force fields, the Nosé-Hoover thermostats and barostats, and the Ewald summation method [21]. Although the Monte Carlo method is not used in this work, it is still introduced together with the MD simulations for completeness. Then, the introduction to the ERnST follows.

6.1 Molecular Dynamics Simulations

What is a simulation? In a simulation, the researcher invokes a system, and allows components of the system to interact according to defined rules. These days, all the required calculations are usually performed by a computer. By making these rules similar to how a real system would behave, the researcher can collect useful data on these artificial systems, and use them to gain insight into the real system under study. The results of the simulation can be validated by evaluating macroscopic parameters from the simulation, and comparing their values from experiments. Commonly used parameters including free energy changes, molecular diffusion coefficients and re-orientational coefficients.

There are many atomistic simulations methods, but they come under two main categories: molecular dynamics and Monte Carlo. Both methods are the same in that a large number of molecular configurations are generated, but the procedures of moving from one configuration to another are different.

In molecular dynamics, the position and velocity of each molecule are noted, and the force on each evaluated. Each step consists of using this information to calculate the position, velocity and force on each atom a very short time interval away, typically of the order of femtoseconds. This is repeated many many times, and one builds up a time series of molecular configurations. The whole trajectory describes the evolution of the molecular system in time. Observing these configurations would be similar to watching a film depicting molecular motion. The advantage of molecular dynamics is that one observes how a system changes in time, and one can also obtain velocity and force data from the simulation. However, if the model used is such that the velocity and the force cannot be calculated, then molecular dynamics would fail. Monte Carlo becomes then the only method possible.

In Monte Carlo simulations, a random process determines how a configuration is changed to another. The advantage of Monte Carlo is that, under certain circumstances, it samples more configurations than molecular dynamics, so it is useful for exploring, *e.g.*, the possible conformers of a molecule. Neither velocity nor force calculations are required, so many more models can be applied to this method than molecular dynamics. One usually assumes that the system is ergodic. Hence, it does not matter what the starting position is, because one will always obtain the same result for the ensemble averages if a Monte Carlo simulation is performed for a sufficiently large number of steps.

Otherwise the main ingredients of the two methods are essentially the same. Both require a model describing how these atoms interact, sometimes called a potential or a force-field. Both require a method of advancing from one configuration to the other. And, last but not least, both require good analysis methods to analyze the collection of configurations produced. Since performing a simulation and obtaining molecular configurations have become much less difficult with the rapid increase of computer program packages, the challenge is then to perform intelligent analyses by optimally using the collection of configurations (or, in the case of molecular dynamics, the trajectory), and thus to obtain useful scientific understanding.

The general use of simulations should not be simply to reproduce experimental findings. After all, if an observable can be obtained by experiments, there is little point in obtaining that information using a more roundabout and potentially less reliable way. Simulations are best used to perform ‘impossible’ experiments or to gain insight into a system. For example, if one wants to study the effect of size on the hydration of non-polar solutes, one could, of course, place inert gases of different sizes in water, and study their hydration pattern. However, by replacing one molecule with another, one is not merely changing the size, but also altering the electronic properties. Simulations allow us to circumvent this problem by allowing us to invent spheres whose only difference is the size, with everything else kept identical [22]. Another example is to study the solvation properties of simple solutes with different electric charges. Simulations allow scientists to keep other properties constant, whilst the charge and size of the particles are considered as dynamic variables using an extended Lagrangian [23]. The selectivity of micropores for ions of different sizes has been used as models to study the selectivity of ion channels [24]. Thus simulations can be used to isolate factors contributing to an effect, and give researchers a better understanding.

Simulations can be used to investigate events and mechanisms inaccessible by experiments. However, in this sort of simulations, there are still certain quantities which are measurable by experiments. The simulation values of those quantities should be compared with the experimental values. If they agree, there is a higher probability that the simulation potentials are correct. The scientist can then use the system to generate data which can only be obtained by simulations.

6.1.1 Development of the Monte Carlo method

The first simulation method invented was the Monte Carlo method. It was developed by Stanislaw Ulam and Nicholas Metropolis [25]. Ulam was convalescing from an illness when he started to ponder about the probabilities of winning a game of solitaire. He realized that, instead of going through the combinatorics, he could just lay out the cards many many times and observe the outcome. This was also the era when the first computers were being used, so he discussed the idea with John von Neumann and started planning actual calculations.

This work was expanded a few years later to calculate the equation of state of N hard discs in two dimensions. Metropolis *et al.* [26] then applied their own method to a system of 224 idential discs in a square with periodic boundary conditions, and calculated the area occupied by the discs. Periodic boundary conditions [27] mean that the squares are laid out in a 'container', so that a particle leaving the 'container' on the left side will re-enter it on the right side. The effect is to create an infinite space using a finite number of particles. Note that in the evaluation of interaction between atoms, there is a maximum distance, r_{cut}. Beyond the distance r_{cut}, even though two atoms could interact, their interaction is considered zero, to simplify calculations.

The work of Metropolis *et al.* [26] was extended by Wood and Parker [28] a few years later, in a paper which studied the equation of state of Lennard-Jones molecules in three dimensions. These are spherical molecules which interacted with each other according to the relation:

$$U_{LJ}(r) = \epsilon \left[\left(\frac{r}{r_o} \right)^{-12} - 2 \left(\frac{r}{r_o} \right)^{-6} \right] \ , \tag{6.1.1}$$

where ϵ is a constant, r_o the radius of the Lennard-Jones potential well, and r the distance between the two atoms. They chose a set of Lennard-Jones parameters to model argon, and set the temperature at 55°C. Using 32 or 108 molecules in a cubic box with periodic boundary conditions, they performed the simulation 31 times, each lasting from 27000 to 261000 configurations, at different densities. The authors discovered that their simulation results agreed well with some experimental data but not with others. They also observed a phase transition, but not at the pressure that it would occur under experimental conditions. This work shows that even a simple simulation system is capable of reproducing features of an experimental system qualitatively, and sometimes even quantitatively.

Wood and Parker [28] also carefully considered the problem of using periodic boundary conditions. They chose a truncation scheme called the minimum-image distance method, namely any pairwise interactions among the fundamental set of N molecules are included, but for each pair, only the smallest distance between any images of the two molecules are taken. This method of minimum-image distance was subsequently adopted in almost all simulations using periodic boundary conditions.

Wood and Jacobson [29] subsequently repeated some of the Monte Carlo simulations with longer chain length, and compared their results with the first molecular dynamics simulation of the same system, the work of Alder and Wainwright [30]. They found good agreement between the two methods.

6.1.2 Development of Molecular Dynamics

The first detailed description of the molecular dynamics method was not published until two years later [31]. In this and a subsequent paper [32], Alder and Wainwright described the scheme used for molecular dynamics simulations, and evaluated the properties of a number of elastic spheres at different densities.

The work of Alder and Wainwright used hard spheres, which have not generally been used to simulate biological molecules. Atoms in biological systems are usually approximated by the Lennard-Jones potential, which was first applied by Rahman [33]:

$$V(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \,, \tag{6.1.2}$$

where ϵ is a constant, σ the optimum distance between two atoms of these types and r the distance between the two atoms. In this work, Rahman developed a time-step algorithm to solve the time-dependent differential equations.

In any molecular dynamics simulations, the position of each atom is pre-determined, the velocity either assigned from a Maxwell distribution or from previous trajectory, and the force on each atom can thus be evaluated. A time-step algorithm uses these data at time t to obtain the positions, velocities and forces of the atoms at $t + \delta t$. In Rahman's original algorithm [33], three equations were applied:

$$\begin{aligned}
\mathbf{r}_p(t+\delta t) &= \mathbf{r}(t-\delta t) + 2\delta t\mathbf{v}(t) && (6.1.3)\\
\mathbf{v}(t+\delta t) &= \mathbf{v}(t) + \frac{\delta t\left[\mathbf{a}(t+\delta t) + \mathbf{a}(t)\right]}{2} && (6.1.4)\\
\mathbf{r}(t+\delta t) &= \mathbf{r}(t) + \frac{\delta t\left[\mathbf{v}(t+\delta t) + \mathbf{v}(t)\right]}{2} \,, && (6.1.5)
\end{aligned}$$

where $\mathbf{r}(t)$, $\mathbf{v}(t)$, $\mathbf{a}(t)$ are the position, velocity and acceleration of a particle at time t,

$\mathbf{r}(t+\delta t)$, $\mathbf{v}(t+\delta t)$, $\mathbf{a}(t+\delta t)$ are the position, velocity and acceleration of a particle at time $t+\delta t$, and $\mathbf{r}_p(t+\delta t)$ the predicted position of the particle at time $t+\delta t$.

Eq. (6.1.3) provides a guess at the new position, and from this the acceleration $\mathbf{a}(t+\delta t)$ can be evaluated. The new velocity and new calculated position can be evaluated using Eq. (6.1.4) and Eq. (6.1.5). The acceleration is then re-calculated and Eq. (6.1.4) and Eq. (6.1.5) iterated to provide a more accurate result.

Rahman's algorithm is not the most efficient, because it requires a few passes through Eq. (6.1.4) and Eq. (6.1.5), and it also requires force calculations which are expensive in computer time. A few years later, more efficient time-step algorithms were developed by Verlet [34] and Gear [35].

Verlet's method is as follows. Let the position of a particle at time t in the simulation be $\mathbf{r}(t)$, and its acceleration be $\mathbf{a}(t)$. Then the position of this particle at a short time δt before and after t can be evaluated using Taylor's expansion:

$$\begin{aligned} \mathbf{r}(t+\delta t) &= \mathbf{r}(t) + \delta t \mathbf{v}(t) + \frac{\delta t^2 \mathbf{a}(t)}{2} + \ldots && (6.1.6) \\ \mathbf{r}(t-\delta t) &= \mathbf{r}(t) - \delta t \mathbf{v}(t) + \frac{\delta t^2 \mathbf{a}(t)}{2} - \ldots\ . && (6.1.7) \end{aligned}$$

Adding Eq. (6.1.6) and Eq. (6.1.7) and re-arranging, one obtains:

$$\mathbf{r}(t+\delta t) = 2\mathbf{r}(t) - \mathbf{r}(t-\delta t) + \delta t^2 \mathbf{a}(t)\ . \qquad (6.1.8)$$

Thus the estimate of position $\mathbf{r}(t+\delta t)$ is correct except for errors of order δt^4.

The Gear predictor-corrector method, on the other hand, uses a series of equations for predicted values (subscripted with $_p$):

$$\begin{aligned}
\mathbf{r}_p(t+\delta t) &= \mathbf{r}(t) + \delta t\mathbf{v}(\mathbf{t}) + \frac{\delta t^2\mathbf{a}(t)}{2} + \frac{\delta t^3\mathbf{b}(t)}{6} + \ldots && (6.1.9)\\
\mathbf{v}_p(t+\delta t) &= \mathbf{v}(t) + \delta t\mathbf{a}(\mathbf{t}) + \frac{\delta t^2\mathbf{b}(t)}{2} + \ldots && (6.1.10)\\
\mathbf{a}_p(t+\delta t) &= \mathbf{a}(t) + \delta t\mathbf{b}(\mathbf{t}) + \ldots && (6.1.11)\\
\mathbf{b}_p(t+\delta t) &= \mathbf{b}(t) + \ldots\ , && (6.1.12)
\end{aligned}$$

where $\mathbf{b}$ denotes the third time derivative of $\mathbf{r}$. This is the predictor step.

In order to obtain the correct values, corrections are required. For example, from the predicted positions $\mathbf{r}_p(t+\delta t)$, one can calculate the forces on the particle, and a corrected acceleration $\mathbf{a}_{cor}$ is obtained. This will, in general, be different from the predicted acceleration. The corrected position is:

$$\mathbf{r}_{cor} = \mathbf{r}_p(t+\delta t) + f_{cor}\left[\mathbf{a}_{cor}(t+\delta t) - \mathbf{a}_p(t+\delta t)\right]\ , \tag{6.1.13}$$

where f_{cor} is a correction factor, and its choice has been extensively discussed by Gear [35]. Using a similar method, the corrected velocities and accelerations at $t+\delta t$ can be evaluated. This is a second-order predictor-corrector, because the acceleration (second-order time derivative) is used to correct the position. Higher-order predictor-correctors can be constructed using the same principles. In practical use, often the correction protocol is executed more than once to obtain better corrected values.

The advantages of the Verlet method is that it requires less storage space in the computer, and it is executed in one step. The Gear predictor-corrector requires more storage, and at least two steps are executed to obtain the corrected values. Van Gunsteren and Berendsen [36] performed simulations on the bovine pancreatic trypsin inhibitor, with or without constrainted bond lengths, and with the Gear predictor-corrector or the Verlet algorithm. They found that for non-constrained dynamics, a high-order (at least fourth order) Gear predictor-corrector was more accurate than the Verlet algorithm when the time-step was smaller than about 1.5 fs, but at larger time-steps, the Verlet algorithm was more accurate. For constrained dynamics, the Verlet algorithm was more accurate when the time-step was longer than 1 fs. A detailed evaluation of the two methods can be found in Fincham and Heyes [37]. In this work, the Verlet method was used for all molecular dynamics simulations.

6.1.3 Simulation of Biological Molecules

By the 1960's, both Monte Carlo and molecular dynamics methods had been well developed. The simulation engine of going from one configuration to the other was efficient. Nevertheless, it was difficult to simulate biological molecules because they often consisted of long chains of atoms, capable of dihedral rotations. Potentials had to be developed to describe the properties of biological molecules before any simulation could be done.

6.1.3.1 Pure protein simulations

The incentive to develop a potential to describe the properties of biological molecules came from crystal structure refinement. In the 1960's, the structure of a protein was refined using a geometric method [38]. Levitt and Lifson [39] proposed an alternative refinement method using an energy minimization scheme. The energy of the protein was considered to be a sum of different contributions:

$$E = \sum_{\text{bonds}} \frac{k_b}{2}(b - b_o)^2 + \sum_{\text{angles}} \frac{k_\theta}{2}(\theta - \theta_o)^2 + \sum_{\text{dihedrals}} \frac{k_\phi}{2}\left[1 + \cos(n\phi - \delta)\right]$$

$$+ \sum_{\text{non-bonded}} \epsilon \left[\left(\frac{r^o}{r}\right)^{12} - 2\left(\frac{r^o}{r}\right)^{6}\right] + \sum_{\text{all coords}} \frac{w}{2}(x_i - x_o)^2 \; . \tag{6.1.14}$$

The first term describes the bond energy, where b_o is the 'ideal' bond length and b the actual bond length, and k_b an empirical constant. The second term is similar to the first term, except that bond angles deviations are being considered. The third is a dihedral angle term, where k_ϕ, n and δ are constants. The fourth term considers the interaction between atoms separated by at least three bonds, and takes on the usual form of the Lennard-Jones 12-6 potential (Cf. Eq. (6.1.1)), with r_o being a constant and r the distance between the non-bonded atoms. The last term is a constraining force, where x_o is the crystallographic coordinate of an atom, and x_i the coordinate after refinement.

In Levitt and Lifson's scheme, hydrogen atoms were combined with the heavy atoms (united atom potential). The energy was minimized by steepest descent. They applied this method to the heavy atom coordinates of lysozyme and myoglobin, and obtained refined coordinates which deviate, respectively, from the crystallographic coordinates by 0.22 Å and 0.086 Å r.m.s. The results for myoglobin were in good agreement with the uncertainty of the experimental data which was 0.1 Å r.m.s. Levitt and Lifson subsequently simplified the energy function in the same paper, and eliminated the w-term, and modified the Lennard-Jones potential:

$$E = \sum_{\text{bonds}} \frac{k_b}{2}(b-b_o)^2 + \sum_{\text{angles}} \frac{k_\theta}{2}(\theta-\theta_o)^2 + \sum_{\text{dihedrals}} \frac{k_\phi}{2}\left[1+\cos(n\phi-\delta)\right]$$

$$+ \sum_{\text{non-bonded}} \epsilon \left[\left(\frac{R^o}{R_{ij}}\right)^{12} - 2\lambda \left(\frac{R^o}{R_{ij}}\right)^{6} \right] , \qquad (6.1.15)$$

where λ was an interaction function ($0 < \lambda < 1$), R_o was a constant and R_{ij} was the distance between the non-bonded atoms. Using this modified energy scheme, they obtained refined myoglobin coordinates which deviate by 0.15 Å from the crystallographic coordinates.

This work showed that this energy function, LL69, obtained from educated guesses, was capable of reproducing essential features of a protein. The work was later extended by Levitt [40] with a more complicated bond angle term. Better refinement of the structures were obtained.

6.1.3.2 Simulation of membrane bilayers

The first simulation of membrane bilayers was performed by van der Ploeg and Berendsen [41]. In this system, they used a simple model for the phospholipid. Only three types of atoms were defined: head groups, methylene units and methyl end groups. Even with such a united-atom potential, the researchers were able to reproduce some essential features of the structure of phospholipids, for example, the bond order parameter of the alkyl CH_2 groups.

A few years later, Egberts and Berendsen [42] performed a simulation of another system consisting of 76 decanol molecules, 52 decanoate ions, 52 Na^+ ions and 526 water molecules. Unlike the previous work, the system was treated in full atomic detail in the head groups, and the whole system had total 3166 atoms. Due to the increase of the detail, this research gave a better match between the dynamical variables of simulations and that of experiments than previous simulations of membrane bilayers did.

Subsequently, more sophisticated models were developed for phospholipid molecules. In this work, we used the CHARMM potential for DMPC [43].

6.1.4 Thermodynamics Ensembles

For more sophisticated simulations, we might want to simulate a system at constant pressure and constant temperature, to mimic most experimental conditions. To achieve this, molecular dynamics methods allow the incorporation of thermostats and barostats into the propagation protocol. In this work, we used exclusively the Nosé-Hoover [44] thermostat and a modified form of the Nosé-Hoover barostat, mainly because they give a true NPT ensemble, rather than just keeping the temperature and pressure constant.

6.1.4.1 Thermostat

Normally, Newton's equations of motion can be written as:

$$\frac{d\mathbf{r}(t)}{dt} = \mathbf{v}(t) \tag{6.1.16}$$

$$\frac{d\mathbf{v}(t)}{dt} = \frac{\mathbf{F(t)}}{m} \ , \tag{6.1.17}$$

where $\mathbf{r}$(t), $\mathbf{v}$(t) and m are the position, velocity and mass of each atom in the system respectively.

In the Nosé-Hoover formulation, an additional term is added to the second equation:

$$\frac{d\mathbf{v}(t)}{dt} = \frac{\mathbf{F(t)}}{m} - \chi(t)\mathbf{v}(t) \ , \tag{6.1.18}$$

where the frictional constant $\chi(t)$ is defined by

$$\frac{d\chi(t)}{dt} = \frac{1}{\tau_T^2}\left(\frac{T}{T_{req}} - 1\right) \tag{6.1.19}$$

with τ_T a constant and T_{req} the requested temperature.

The integration of these equations requires several iterations.

6.1.4.2 Barostat

The barostat we used in our work is a modified form of the Nosé-Hoover barostat. The constant-temperature equations of motion become:

$$\begin{aligned}
\frac{d\mathbf{r}(t)}{dt} &= \mathbf{v}(t) + \eta(\mathbf{r}(t) - \mathbf{R}_0) \\
\frac{d\mathbf{v}(t)}{dt} &= \frac{\mathbf{F(t)}}{m} - [\chi(t) + \eta(t)]\mathbf{v}(t) \\
\frac{d\chi(t)}{dt} &= \frac{1}{\tau_T^2}\left(\frac{T}{T_{req}} - 1\right) \\
\frac{d\eta(t)}{dt} &= \frac{1}{Nk_B T_{req}\tau_P^2} V(t)(P - P_{req}) \\
\frac{dV(t)}{dt} &= [3\eta(t)]V(t) \ ,
\end{aligned} \tag{6.1.20}$$

where η is the barostat friction constant, $\mathbf{R}_0$ center of mass of the system, τ_P a time constant, P the instantaneous pressure, P_{req} the requested pressure and V the volume of the system. As the case of thermostat, the integration needs several iterations to achieve convergence.

6.1.5 Ewald Summation

The Coulomb potential of two charged atoms is a long-range potential proportional to the inverse of the distance between them. Since we use periodic boundary conditions in our simulations, the Coulomb potential could sum to infinity. To prevent this from happening, we use the Ewald summation method [45]. In this method, each charged atom is firstly neutralized by superposing a spherical gaussian cloud of opposite charge on its center which makes the combined-potential short range and summable. Secondly another set of gaussian charges are superposed again to nullify the effect of the first gaussian charges. The two sets of Gaussian charges cancel each other, so the total charges summed are not changed. However, the original infinite sum is changed to two finite sums, and this can avoid the problem of summing to infinity.

6.2 Energy Representation of Solution Theory

In the evaluation of free energy change of solvation, in principle only the initial and final states of a process are required for the calculation. However, the two commonly used methods, thermodynamic integration (TI) and free energy perturbation (FEP), both require a large number of intermediate states for evaluating free energy changes. The method of energy representation of solution theory (ERnST), which was developed by Matubayasi [46, 47, 48], does not require intermediate states, so is 40–200 times faster than either TI or FEP, and yet its results are within 5% those obtained from FEP.

6.2.1 Basic Derivation

In this method, we first construct an instantaneous distribution of the solvent molecules relative to a solute molecule $\hat{\rho}^f$ defined as:

$$\hat{\rho}^f(\mathbf{x}) = \sum_i \delta(\mathbf{x} - \mathbf{x}_i) \ , \tag{6.2.1}$$

where $\mathbf{x}_i$ is the full coordinates of ith solvent molecule which include positions, orientations and, if they exist, the intramolecular degrees of freedom, and δ the Dirac delta function. The summation is taken over all the solvent molecules. The superscript f emphasizes that the full coordinates are used.

When the solute-solvent interaction potential is u, the distribution function ρ^f can be generated from taking an ensemble average over the instantaneous distribution function:

$$\rho^f(\mathbf{x}; u) = \left\langle \hat{\rho}^f(\mathbf{x}) \right\rangle_u \ , \tag{6.2.2}$$

where the subscript u denotes that the ensemble average is taken in the solution with solute-solvent interaction potential u. This potential u can take a subscript λ so that when $\lambda = 1, u_\lambda$ is the normal potential, but when $\lambda = 0, u_\lambda$ denotes a zero potential. In FEP and TI, λ is varied gradually to take the system from the initial state to the final state. This is not performed in ERnST, but this variable is used in the derivation.

ERnST is introduced by changing the coordinates of the solvent molecules from full coordinates to solute-solvent interaction energies:

$$\hat{\rho}^e(\epsilon) = \int d\mathbf{x}\delta\left(v^f(\mathbf{x}) - \epsilon\right)\hat{\rho}^f(\mathbf{x}) = \sum_i \delta\left(v^f(\mathbf{x}_i) - \epsilon\right) \ , \tag{6.2.3}$$

where $v \equiv u_{(\lambda=1)}$

$$\rho^e(\epsilon; u) = \langle \hat{\rho}^e(\epsilon) \rangle_u = \int d\mathbf{x} \delta \left(v^f(\mathbf{x}) - \epsilon \right) \rho^f(\mathbf{x}; u) \ . \tag{6.2.4}$$

ERnST evaluates the free-energy change $\Delta\mu$ of a process, without having the need to invoke intermediate states. The basic principle of this method draws analogy from density functional theory, and starts from the Kirkwood charging formula [49, 50, 51], which is an integration over the coupling parameter λ:

$$\Delta\mu = \int_0^1 d\lambda \int d\mathbf{x} \frac{\partial u_\lambda(\mathbf{x})}{\partial \lambda} \rho^f(\mathbf{x}; u_\lambda) \ , \tag{6.2.5}$$

where λ changes from 0 to 1 in going from the initial to the final state. Integration by parts gives

$$\begin{aligned} \Delta\mu &= \int d\mathbf{x} v(\mathbf{x}) \rho^f(\mathbf{x}; v) - \int_0^1 d\lambda \int d\mathbf{x} u_\lambda(\mathbf{x}) \frac{\partial \rho^f(\mathbf{x}; u_\lambda)}{\partial \lambda} \\ &\equiv \int d\mathbf{x} v(\mathbf{x}) \rho^f(\mathbf{x}; v) - F^f \left[\rho^f(\mathbf{x}; u_\lambda) \right] \ , \end{aligned} \tag{6.2.6}$$

where F^f is a density-functional as defined above. By introducing the indirect part ω^f of the potential of mean force, and defining $\rho^f(\mathbf{x}; u_\lambda) \equiv \rho^f_\lambda(\mathbf{x})$:

$$\rho^f_\lambda(\mathbf{x}) = \rho^f_0(\mathbf{x}) \exp\left\{ -\beta \left[u(\mathbf{x}; \rho^f_\lambda) + \omega^f(\mathbf{x}; \rho^f_\lambda) \right] \right\} \ . \tag{6.2.7}$$

Eq. (6.2.6) and Eq. (6.2.7) lead to

$$\begin{aligned} F^f[\rho^f(\mathbf{x})] = k_B T \int d\mathbf{x} \Bigg[&\Big(\rho^f(\mathbf{x}) - \rho^f_0(\mathbf{x}) \Big) - \rho^f(\mathbf{x}) log \left(\frac{\rho^f(\mathbf{x})}{\rho^f_0(\mathbf{x})} \right) \\ &- \beta \Big(\rho^f(\mathbf{x}) - \rho_0(\mathbf{x}) \Big) \int_0^1 d\lambda \omega^f(\mathbf{x}; \rho^f_\lambda) \Bigg] \ . \end{aligned} \tag{6.2.8}$$

After applying ERnST we can show that the Kirkwood charging formula becomes

$$\Delta\mu = \int_0^1 d\lambda \int d\epsilon \frac{\partial u_\lambda(\epsilon)}{\partial \lambda} \rho^e(\epsilon; u_\lambda) \ . \tag{6.2.9}$$

It can be proved that

$$\Delta\mu = \int d\epsilon\, \epsilon\rho^e(\epsilon) - k_B T \int d\epsilon \Bigg[\big(\rho^e(\epsilon) - \rho_0^e(\epsilon)\big) - \rho^e(\epsilon) \log\left(\frac{\rho^e(\epsilon)}{\rho_0^e(\epsilon)}\right) \\ - \Big\{\alpha(\epsilon)F(\epsilon) + \big(1-\alpha(\epsilon)\big)F_0(\epsilon)\Big\}\big(\rho^e(\epsilon) - \rho_0^e(\epsilon)\big)\Bigg]\ , \tag{6.2.10}$$

where

$$F(\epsilon) = \begin{cases} \beta w^e(\epsilon) + 1 + \dfrac{\beta w^e(\epsilon)}{\exp\big(-\beta w^e(\epsilon)\big) - 1} & \text{when } w^e(\epsilon) \leq 0 \\ \frac{1}{2}\beta w^e(\epsilon) & \text{when } w_0^e(\epsilon) \geq 0 \end{cases} \tag{6.2.11}$$

$$F_0(\epsilon) = \begin{cases} -log\big(1 - \beta w_0^e(\epsilon)\big) + 1 + \dfrac{\log\big(1-\beta w_0^e(\epsilon)\big)}{\beta w_0^e(\epsilon)} & \text{when } w_0^e(\epsilon) \leq 0 \\ \frac{1}{2}\beta w^e(\epsilon) & \text{when } w_0^e(\epsilon) \geq 0 \end{cases} \tag{6.2.12}$$

$$\alpha(\epsilon) = \begin{cases} 1 & \text{when } \rho^e(\epsilon) \geq \rho_0^e(\epsilon) \\ 1 - \left(\dfrac{\rho^e(\epsilon) - \rho_0^e(\epsilon)}{\rho^e(\epsilon) + \rho_0^e(\epsilon)}\right)^2 & \text{when } \rho^e(\epsilon) \leq \rho_0^e(\epsilon)\ . \end{cases} \tag{6.2.13}$$

In addition,

$$w^e(\epsilon) = -k_B T \log\left(\frac{\rho^e(\epsilon)}{\rho_0^e(\epsilon)}\right) - \epsilon \tag{6.2.14}$$

and

$$w_0^e(\epsilon) = -k_B T \int d\eta \left(\frac{\delta(\epsilon-\eta)}{\rho_0^e(\epsilon)} - \left(\chi_0^e\right)^{-1}(\epsilon,\eta)\right)\big(\rho^e(\eta) - \rho_0^e(\eta)\big)\ , \tag{6.2.15}$$

where χ_0^e is the correlation matrix defined as

$$\chi_0^e(\epsilon,\eta) = \big\langle \hat{\rho}^e(\epsilon)\hat{\rho}^e(\eta)\big\rangle_0 - \big\langle \hat{\rho}^e(\epsilon)\big\rangle_0 \big\langle \hat{\rho}^e(\eta)\big\rangle_0\ . \tag{6.2.16}$$

The solvation free energy can be evaluated from Eq. (6.2.10) – (6.2.15), with the inputs (ρ^e, ρ_0^e, χ_0^e) given by Eq. (6.2.4) and Eq. (6.2.16).

Chapter 7

Simulation and Analysis

Previous work suggests the relation between general anesthetics and membranes, so the following simulations were performed to study this question.

7.1 Approach

We used hydrated DMPC as a model of the cell membrane, halothane as a model GA molecule and pure water as a model of the body fluid. The DMPC lipid bilayer, the halothane molecule, and the surrounding water were put in an orthorhombic box whose size is 54 Å(x direction) by 54 Å(y direction) by 63 Å(z direction). The lipid bilayer is positioned perpendicularly to the z direction and thus the DMPC molecules are arranged in the x-y direction. The origin of the z-axis is so chosen that the x-y plane bisects the box in the z direction. The bisecting plane of the lipid bilayer is then fixed to coincide with the x-y plane. Applying the orthorhombic periodic boundary condition [21], the system we actually simulated looks like a set of infinitely extending bilayers in the x-y direction, and such planes repeat themselves for infinite number of times in the z direction. The initial configuration of the hydrated DMPC bilayer we used was obtained from previous work done by Zubrzycki et al. [52].

To find out where the halothane molecule is most likely to be in the cell membrane, the membrane was divided along its z-axis into four regions. They are defined as follows:

1. region I

   $-5\,\text{Å} < z < 5\,\text{Å}$

2. region II

   $-10\,\text{Å} < z < -5\,\text{Å}$ and $5\,\text{Å} < z < 10\,\text{Å}$

3. region III

   $-15\,\text{Å} < z < -10\,\text{Å}$ and $10\,\text{Å} < z < 15\,\text{Å}$

4. region IV

   $-20\,\text{Å} < z < -15\,\text{Å}$ and $15\,\text{Å} < z < 20\,\text{Å}$

The hydrated DMPC membrane and the divisions are visualized in Fig. 7.1.

The free-energy change of a halothane molecule being moved from vacuum into any of these regions in the membrane was evaluated. Another set of calculations are also performed to evaluate the free-energy change of inserting a halothane from vacuum into water. The complete procedure was first performed at atmospheric pressure (10^5 Pa), and then repeated at 2×10^7 Pa (200 atm) and at 4×10^7 Pa (400 atm).

To evaluate the free energy change, ERnST was used. In this method, there were two kinds of simulations. One is to insert the solute from vacuum into an equilibrated

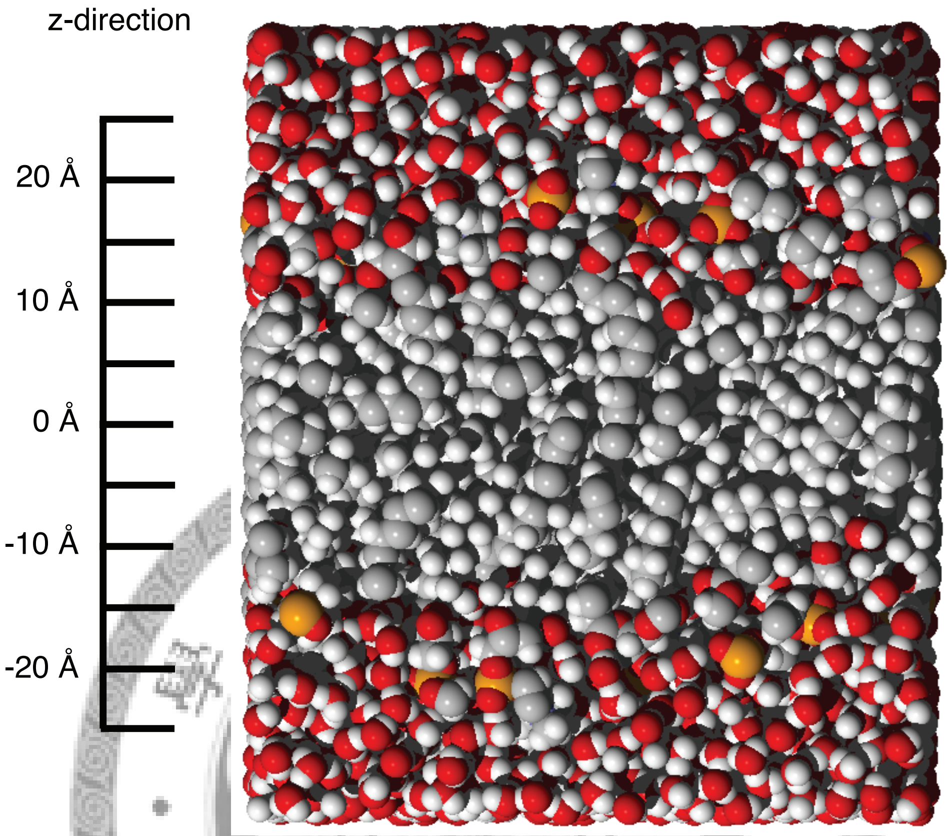


Figure 7.1: Diagram of a hydrated DMPC membrane. The oxygen atoms are shown in red, the hydrogen atoms in white, the phosphorus atoms in orange and the carbon atoms in gray. The scale on the left shows the divisions of the membrane into different regions.

environment (called the 'reference solvent' system) and calculate the interaction energy between the solute and molecules in the environment. The other is to equilibrate a system consisting of the solute dissolved in the environment, and evaluate the interaction energy between the solute and molecules in this equilibrated environment. This system is called the solution system.

The interaction energy distributions of both the reference solvent and the solution are needed by ERnST to evaluate the free-energy change of inserting a solute into the environment. The interaction energies were binned to obtain the interaction energy distributions. The binning was performed in a special way which had four segments:

1. From -1.3×10^{-19} J to -1.5×10^{-21} J with interval 2.5×10^{-22} J
2. From -1.5×10^{-21} J to $+1.5 \times 10^{-21}$ J with interval 1.0×10^{-23} J
3. From $+1.5 \times 10^{-21}$ J to $+7.0 \times 10^{-20}$ J with interval 2.5×10^{-22} J
4. From $+7.0 \times 10^{-20}$ J to $+7.0 \times 10^{-11}$ J with 200 logarithmic intervals.

   It means that the bin width is a constant under the logarithmic axis and the number of bins is 200.

The free-energy change of moving a halothane molecule from water into the membrane was divided into two parts: one is the free-energy change of moving a halothane from water into vacuum, and the other is the free-energy change of moving it from vacuum into the membrane. The addition of both parts is the free-energy change we wish to get. Finally, the layer with the most negative value of the free energy change would be the one the halothane most likely to be in.

7.2 Conditions of the Simulations

All the simulations were performed in the isotropic NPT ensemble [53], where the number of molecules, the pressure and the temperature were kept constant. The temperature used was 310 K, and the thermostat time constant was 1 ps. The pressure used was either 10^5 Pa, 2×10^7 Pa or 4×10^7 Pa. The barostat time constant was 5 ps.

The CHARMM potential [43], the TIP3P model [54, 55] and the Scharf-Laasonen model [56] were used for DMPC, water and halothane, respectively. Non-bonded forces were cut off at 13 Å with long-range correction. Particle-mesh Ewald summation was applied to evaluate electrostatics, with a real-space cut-off of 13 Å, the Ewald convergence parameter being 0.24374, and the k-vector (8, 8, 8). The details of simulations are shown in Table 7.1.

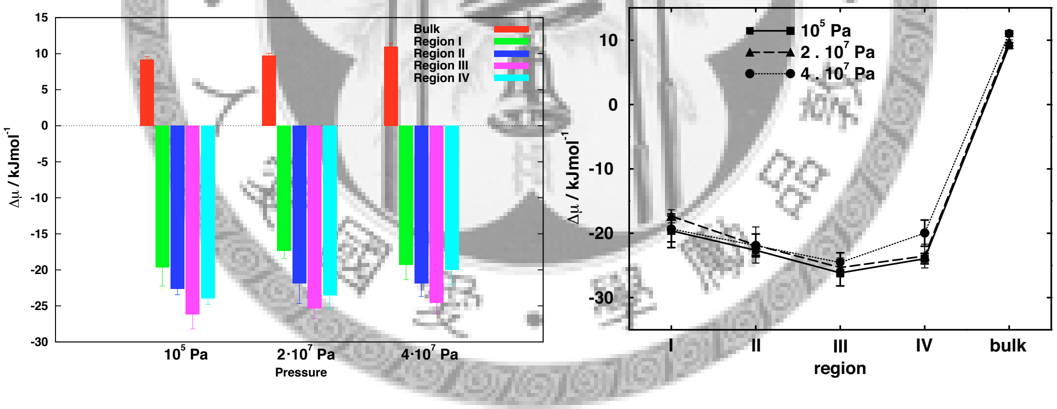
	pressure (atm)	timestep (fs)	starting configurations	number of dumped config.
p.w.	1	2 fs	2742 water	600, 1 config./100 fs
	200	2 fs	2742 water	600, 1 config./100 fs
	400	2 fs	2742 water	600, 1 config./100 fs
h.w.	1	2 fs	2742 water, 1 halothane	10000, 1 config./10 fs
	200	2 fs	2742 water, 1 halothane	10000, 1 config./10 fs
	400	2 fs	2742 water, 1 halothane	10000, 1 config./10 fs
p.D.	1	2 fs	2742 water, 100 DMPC	600, 1 config./1 ps
	200	1→2 fs[†]	2742 water, 100 DMPC	600, 1 config./1 ps
	400	2 fs	2742 water, 100 DMPC	600, 1 config./1 ps
h.D.	1	2 fs	2742 water, 100 DMPC, 1 halothane	5000 × 3, 1 config./10 fs
	200	1→2 fs[†]	2742 water, 100 DMPC, 1 halothane	5000 × 3, 1 config./10 fs
	400	2 fs	2742 water, 100 DMPC, 1 halothane	5000 × 3, 1 config./10 fs

Table 7.1: Here p.w. stands for "pure water", h.w. stands for "halothane in water", p.D. stands for "pure hydrated DMPC" and h.D. stands for "halothane in hydrated DMPC". [*]These simulations were done by Dr. Chau. [†]The timesteps were 1 fs before 900 ps when equilibrating, and changed to 2 fs after 900 ps. Each system was equilibrated for a total of 1 ns.

Chapter 8

Results and Discussion

Fig. 8.1 shows the free-energy change of inserting a halothane into different environments and Fig. 8.2 shows the structures of the membrane at different pressures.



(a) Histogram

(b) Line plot

Figure 8.1: Two diagrams showing the same results with different presentation methods. Bulk: free-energy change of moving a halothane from vacuum into bulk water. Region I–IV: free-energy change of moving a halothane from vacuum into the respective regions. The standard deviations are shown by the error bars.

The free-energy change of inserting a halothane from vacuum into bulk water is about $10\,\mathrm{kJmol^{-1}}$. This value increases from $9.1\,\mathrm{kJmol^{-1}}$ at $10^5\,\mathrm{Pa}$, to $11.0\,\mathrm{kJmol^{-1}}$ at

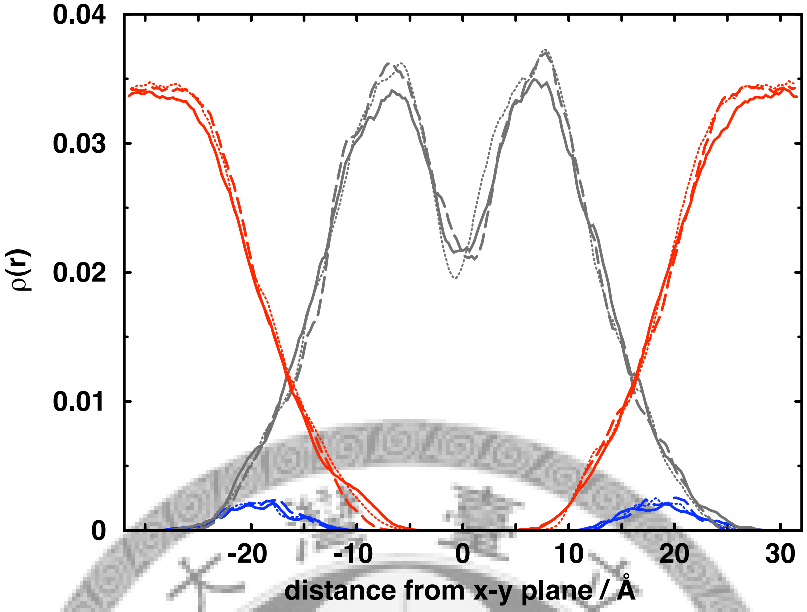


Figure 8.2: Diagram showing the probability of finding an atom in the z-direction of the hydrated membrane. The blue line shows the distribution of the choline nitrogen, the gray line displays the distribution of the methylene carbon of the alkyl tail, and the red line shows the distribution of water oxygen atoms. The solid line denotes data at 10^5 Pa, the dashed line denotes data at 2×10^7 Pa, and the dotted line denotes data at 4×10^7 Pa. We can see from this diagram that the structures of the membrane are basically the same at different pressures.

4×10^7 Pa. This change is statistically significant.

The free-energy change of inserting a halothane from vacuum into hydrated DMPC is dependent on the region. In region I ($-5\,\text{Å} < z < 5\,\text{Å}$), as the pressure increases from 10^5 Pa to 2×10^7 Pa, the free-energy change increases from $-19.6\,\text{kJmol}^{-1}$ to $-17.4\,\text{kJmol}^{-1}$. However, at 4×10^7 Pa, the free-energy change is -19.3 kJmol^{-1}. In region II ($-10\,\text{Å} < z < -5\,\text{Å}$ and $5\,\text{Å} < z < 10\,\text{Å}$), as the pressure increases from 10^5 Pa to 2×10^7 Pa, the free-energy change increases from $-22.7\,\text{kJmol}^{-1}$ to $-21.8\,\text{kJmol}^{-1}$. However, at 4×10^7 Pa, the free-energy change slightly reduces to $-21.9\,\text{kJmol}^{-1}$. In region III ($-15\,\text{Å} < z < -10\,\text{Å}$ and $10\,\text{Å} < z < 15\,\text{Å}$), as the pressure increases from

10^5 Pa to 2×10^7 Pa then to 4×10^7 Pa, the free-energy change increases monotonically from $-26.2\,\mathrm{kJmol^{-1}}$ to $-25.3\,\mathrm{kJmol^{-1}}$ then to $-24.6\,\mathrm{kJmol^{-1}}$. In region IV ($-20\,\text{Å} < z < -15\,\text{Å}$ and $15\,\text{Å} < z < 20\,\text{Å}$), the free-energy change also increases monotonically from $-24.0\,\mathrm{kJmol^{-1}}$ to $-23.6\,\mathrm{kJmol^{-1}}$ then to $-20.0\,\mathrm{kJmol^{-1}}$ when the pressure increases from 10^5 Pa to 2×10^7 Pa then to 4×10^7 Pa.

The free-energy change of inserting a halothane from bulk water into the most favorable region of hydrated DMPC is $-35.3 \pm 2.5\,\mathrm{kJmol^{-1}}$, $-35.0 \pm 1.8\,\mathrm{kJmol^{-1}}$ and $-35.6 \pm 1.9\,\mathrm{kJmol^{-1}}$ at 10^5 Pa, 2×10^7 Pa and 4×10^7 Pa, respectively. These values most reflect the movement of the general anesthetic from the bloodstream into the brain. With the result of Fig. 8.2, we thus conclude that we are not able to detect any statistically significant difference in the free-energy change of halothane going from bulk water to hydrated DMPC nor the structure of the hydrated membrane when the pressure changes.

Chapter 9

Conclusion

The site and mechanism of action of general anesthetics are still not fully understood. Research work dating back to the 19th century suggests that the phospholipid membrane is involved, but its role is unclear. Recent view has shifted towards a site of action in the membrane proteins [57]. In this work, we aim to quantify the interaction of general anesthetics with a model membrane, the hydrated DMPC bilayer.

Previous experimental studies of the location of halothane have used different amphipathic model molecules. Baber et al. applied nuclear magnetic resonance (NMR) methods to measure the effect of halothane and a number of halo-alkane general anesthetics on a palmitoyl-oleoyl-phosphatidylcholine (POPC) membrane, and they discovered that halothane preferentially stays in the region of the membrane-water interface, and in the region of the upper portion of the acyl chain [58]. Yoshino et al. used sodium dodecyl sulphate (SDS), and applied NMR to locate the anesthetic near the polar head of the SDS molecules. They concluded that halothane did not penetrate into the hydrophobic core [59].

Tu et al. performed molecular dynamics simulations on a halothane-DPPC system. They placed 4 halothane molecules into 64 phospholipids and 1792 water molecules [16], or 32 halothane molecules into the same hydrated DPPC system [17], and carried out simulations for 1.5 ns on both systems. Halothane did not appear to exhibit any preferred location from their first set of results, but they reported that halothane was located preferentially to the more peripheral part of the acyl chains. In clinical use, the concentration of halothane is about 0.1 that of the membrane phospholipids. Therefore, the concentration of halothane used in the later simulation was considerably higher than clinical dosage.

Koubi et al. performed molecular dynamics simulations of 3.7 ns on a system consisting of 64 molecules of 1-stearoyl-2-docosahexaenoyl-*sn*-glycero-3-phosphocholine (SDPC), 1760 water molecules and 16 molecules of halothane [60]. They discovered that halothane exhibited three peak distributions within the membrane, one at the centre of the membrane in its lipid core, and two at the head of the acyl chains of SDPC. Pickholz et al. developed a coarse-grained model of DMPC, and performed simulations on system consisting of 512 phospholipids, 4384 water molecules and halothane [18]. The number of halothane molecules used were, respectively, 0, 64, 128, 256, 384 and 512. The authors observed that halothane always partitioned to the upper part of the acyl chain. Subsequently, Vemparala et al. performed molecular dynamics on system consisting of 1,2-dioleoyl-*sn*-glycero-3-phosphocholine (DOPC) with embedded helical peptide bundles based on the transmembrane domains of the nicotinic acetylcholine receptor [61]. Using steered molecular dynamics, they also evaluated the free energy profile of halothane in a hydrated system of 72 DOPC molecules.

This work has applied ERnST to evaluate the free-energy change of inserting a halothane from bulk water to different regions of a hydrated DMPC bilayer, at pressures of 10^5 Pa, 2×10^7 Pa and 4×10^7 Pa. The simulations results indicate that the influence of pressure on the free-energy change is very small and so is that on the regional preference of halothane in membrane.

This work has only touched upon the effect of pressure on the free-energy change of insertion of halothane. Previous work [62] suggests that as the pressure increases, halothane tends to aggregate inside the cell membrane. However, those simulations were rather short (less than 1 ns) and the concentration of halothane used was well above clinical concentrations. It would be interesting to perform longer simulations of hydrated DMPC with a clinical concentration of halothane, to see if aggregation is indeed observed at higher pressures. If so, we could extend the application of ERnST to evaluate the free-energy change of aggregation. In addition, the phospholipid molecules are not limited to DMPC, other molecules such as DPPC, POPC, etc. should also be used in simulations for comparison.

Bibliography

[1] R. A. Fisher, “On the mathematical foundations of theoretical statistics,” *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 222, pp. 309–368, 1922.

[2] J. Nelder and R. Mead, “A simplex method for function minimisation,” *Computer Journal*, vol. 7, pp. 308–313, 1965.

[3] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran 77.* Cambridge, UK: Cambridge University Press, 2nd ed., 1992.

[4] C. R. Rao, *Linear Statistical Inference and Its Applications.* Wiley-Interscience, 1965.

[5] J. von Neumann, “Various techniques used in connection with random digits, monte carlo method,” *National Bureau Standandards*, vol. 12, pp. 36–38., 1951.

[6] A. L. Blatz and K. L. Magleby, “Quantitative description of three modes of activity of fast chloride channels from rat skeletal muscle.,” *J Physiol*, vol. 378, no. 1, pp. 141–174, 1986.

[7] F. J. Sigworth and S. M. Sine, "Data transformations for improved display and fitting of single-channel dwell time histograms.," *Biophys. J.*, vol. 52, no. 6, pp. 1047–1054, 1987.

[8] F. H. Johnson and E. A. Flagler, "Hydrostatic pressure reversal of narcosis in tadpoles," *Science*, vol. 112, pp. 91–92, 1950.

[9] M. J. Lever, K. W. Miller, W. D. M. Paton, and E. B. Smith, "Pressure reversal of anaesthesia," *Nature*, vol. 231, no. 5302, pp. 368–371, 1971.

[10] K. W. Miller, W. D. M. Paton, R. A. Smith, and E. B. Smith, "The pressure reversal of general anesthesia and the critical volume hypothesis," *Mol Pharmacol*, vol. 9, no. 2, pp. 131–143, 1973.

[11] M. J. Halsey and B. Wardley-Smith, "Pressure reversal of narcosis produced by anaesthetics, narcotics and tranquillisers," *Nature*, vol. 257, no. 5529, pp. 811–813, 1975.

[12] C. R. Dundas, "Alphaxalone/alphadolone in diving-chamber anesthesia," *Lancet*, vol. Pt 1, p. 378, 1979.

[13] E. B. Smith, F. Bowser-Riley, S. Daniels, I. T. Dunbar, C. B. Harrison, and W. D. M. Paton, "Species variation and the mechanism of pressure-anaesthetic interactions," *Nature*, vol. 311, no. 5981, pp. 56–57, 1984.

[14] J. Trudell, W. Hubbell, and E. Cohen, "The effect of two inhalation anesthetics on the order of spin-labeled phospholipid vesicles," *Biochimica et Biophysica Acta*, vol. 291, pp. 321–327, 1973.

[15] J. Trudell, W. Hubbell, and E. Cohen, “Pressure reversal of inhalation anesthetic-induced disorder in spin-labeled phospholipid vesicles,” *Biochimica et Biophysica Acta*, vol. 291, pp. 328–334, 1973.

[16] K. Tu, M. Tarek, M. L. Klein, and D. Scharf, “Effects of anesthetics on the structure of a phospholipid bilayer: Molecular dynamics investigation of halothane in the hydrated liquid crystal phase of dipalmitoylphosphatidylcholine,” *Biophys. J.*, vol. 75, no. 5, pp. 2123–2134, 1998.

[17] L. Koubi, M. Tarek, M. L. Klein, and D. Scharf, “Distribution of Halothane in a Dipalmitoylphosphatidylcholine Bilayer from Molecular Dynamics Calculations,” *Biophys. J.*, vol. 78, no. 2, pp. 800–811, 2000.

[18] M. Pickholz, L. Saiz, and M. L. Klein, “Concentration effects of volatile anesthetics on the properties of model membranes: A coarse-grain approach,” *Biophys. J.*, vol. 88, no. 3, pp. 1524–1534, 2005.

[19] American Medical Association Division of Drugs and Toxicology, *Drug Evaluations Annual*, American Medical Association Press, 1994.

[20] W. Smith and T. R. Forester, “DL_POLY_2.0: A general-purpose parallel molecular dynamics simulation package,” *Journal of Molecular Graphics*, vol. 14, no. 3, pp. 136–141, 1996.

[21] W. Smith, T. Forester, and I. Todorov, *The DL_POLY_2.0 User Manual.* STFC Daresbury Laboratory, Daresbury, Warrington WA4 4AD, Cheshire, UK, 2.18 ed., August 2007.

[22] P.-L. Chau, "Curvature effects on hydrophobic solvation," *Molecular Physics*, vol. 89, no. 4, pp. 1033–1055, 1996.

[23] R. M. Lynden-Bell and J. C. Rasaiah, "From hydrophobic to hydrophilic behaviour: A simulation study of solvation entropy and free energy of simple solutes," *The Journal of Chemical Physics*, vol. 107, no. 6, pp. 1981–1991, 1997.

[24] D. Goulding, J.-P. Hansen, and S. Melchionna, "Size selectivity of narrow pores," *Phys. Rev. Lett.*, vol. 85, pp. 1132–1135, Jul 2000.

[25] N. Metropolis and S. Ulam, "The monte carlo method," *Journal of the American Statistical Association*, vol. 44, no. 247, pp. 335–341, 1949.

[26] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[27] M. Born and T. von Karman, "Über schwingungen in raumgittern," *Physikalische Zeitschrift*, vol. 13, pp. 297–309, 1912.

[28] W. W. Wood and F. R. Parker, "Monte carlo equation of state of molecules interacting with the lennard-jones potential. i. a supercritical isotherm at about twice the critical temperature," *The Journal of Chemical Physics*, vol. 27, no. 3, pp. 720–733, 1957.

[29] W. W. Wood and J. D. Jacobson, "Preliminary results from a recalculation of the monte carlo equation of state of hard spheres," *The Journal of Chemical Physics*, vol. 27, no. 5, pp. 1207–1208, 1957.

[30] B. J. Alder and T. E. Wainwright, “Phase transition for a hard sphere system,” *The Journal of Chemical Physics*, vol. 27, pp. 1208–1209, 1957.

[31] B. J. Alder and T. E. Wainwright, “Studies in molecular dynamics. I. general method,” *The Journal of Chemical Physics*, vol. 31, pp. 459–466, 1959.

[32] B. J. Alder and T. E. Wainwright, “Studies in molecular dynamics. II. behavior of a small number of elastic spheres,” *The Journal of Chemical Physics*, vol. 33, no. 5, pp. 1439–1451, 1960.

[33] A. Rahman, “Correlations in the motion of atoms in liquid argon,” *Physical Review*, vol. 136, pp. A405–A411, Oct 1964.

[34] L. Verlet, “Computer ”experiments” on classical fluids. i. thermodynamical properties of lennard-jones molecules,” *Phys. Rev.*, vol. 159, p. 98, Jul 1967.

[35] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations.* Upper Saddle River, NJ, USA: Prentice Hall PTR, 1971.

[36] W. F. van Gunsteren and H. J. C. Berendsen, “Algorithms for macromolecular dynamics and constraint dynamics,” *Molecular Physics*, vol. 34, no. 5, pp. 1311–1327, 1977.

[37] D. Fincham and D. Heyes, “Integration algorithms in molecular dynamics,” *CCP5 Quarterly*, vol. 6, pp. 4–10, 1982.

[38] R. Diamond, “A mathematical model-building procedure for proteins,” *Acta Crystallographica*, vol. 21, no. 2, pp. 253–266, 1966.

[39] M. Levitt and S. Lifson, “Refinement of protein conformations using a macromolecular energy minimization procedure,” *Journal of Molecular Biology*, vol. 46, no. 2, pp. 269–279, 1969.

[40] M. Levitt, “Energy refinement of hen egg-white lysozyme,” *Journal of Molecular Biology*, vol. 82, no. 3, pp. 393–420, 1974.

[41] P. van der Ploeg and H. J. C. Berendsen, “Molecular dynamics simulation of a bilayer membrane,” *The Journal of Chemical Physics*, vol. 76, no. 6, pp. 3271–3276, 1982.

[42] E. Egberts and H. J. C. Berendsen, “Molecular dynamics simulation of a smectic liquid crystal with atomic detail,” *The Journal of Chemical Physics*, vol. 89, no. 6, pp. 3718–3732, 1988.

[43] A. MacKerell, D. Bashford, M. Bellott, R. Dunbrack, J. Evanseck, M. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. Lau, C. Mattos, S. Michnick, T. Ngo, D. Nguyen, B. Prodhom, W. Reiher, B. Roux, M. Schlenkrich, J. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus, “All-atom empirical potential for molecular modeling and dynamics studies of proteins,” *Journal of Physical Chemistry B*, vol. 102, no. 18, pp. 3586–3616, 1998.

[44] W. G. Hoover, “Canonical dynamics: Equilibrium phase-space distributions,” *Phys. Rev. A*, vol. 31, pp. 1695–1697, Mar 1985.

[45] P. P. Ewald, "Die Berechnung optischer und elektrostatischer Gitterpotentiale," *Annalen der Physik*, vol. 369, pp. 253–287, 1921.

[46] N. Matubayasi and M. Nakahara, "Theory of solutions in the energetic representation. I. Formulation," *The Journal of Chemical Physics*, vol. 113, no. 15, pp. 6070–6081, 2000.

[47] N. Matubayasi and M. Nakahara, "Theory of solutions in the energy representation. II. Functional for the chemical potential," *The Journal of Chemical Physics*, vol. 117, pp. 3605–3616, Aug 2002.

[48] N. Matubayasi and M. Nakahara, "Theory of solutions in the energy representation. III. Treatment of the molecular flexibility," *The Journal of Chemical Physics*, vol. 119, pp. 9686–9702, Nov 2003.

[49] J. P. Hansen and I. R. McDonald, *Theory of Simple Liquids*. Academic Press, 2006.

[50] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*. Oxford University Press, 1989.

[51] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, 2002.

[52] I. Z. Zubrzycki, Y. Xu, M. Madrid, and P. Tang, "Molecular dynamics simulations of a fully hydrated dimyristoylphosphatidylcholine membrane in liquid-crystalline phase," *The Journal of Chemical Physics*, vol. 112, no. 7, pp. 3437–3441, 2000.

[53] S. Melchionna, G. Ciccotto, and B. L. Holian, "Hoover npt dynamics for systems varying in shape and size," *Molecular Physics*, vol. 78, pp. 533–544, 1993.

[54] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, "Comparison of simple potential functions for simulating liquid water," *The Journal of Chemical Physics*, vol. 79, no. 2, pp. 926–935, 1983.

[55] A. R. Leach, *Molecular Modelling: Principles and Applications.* Prentice Hall, 2nd ed., 2001.

[56] D. Scharf and K. Laasonen, "Structure, effective pair potential and properties of halothane," *Chemical Physics Letters*, vol. 258, no. 1-2, pp. 276–282, 1996.

[57] L. Nelson, T. Guo, J. Lu, C. Saper, N. Franks, and M. Maze, "The sedative component of anesthesia is mediated by $\mathrm{GABA_A}$ receptors in an endogenous sleep pathway," *Nature Neuroscience*, vol. 5, pp. 979–984, August 2002.

[58] J. Baber, J. F. Ellena, and D. S. Cafiso, "Distribution of general anesthetics in phospholipid bilayers determined using 2h nmr and 1h-1h noe spectroscopy," *Biochemistry*, vol. 34, no. 19, pp. 6533–6539, 1995.

[59] A. Yoshino, T. Yoshida, H. Okabayashi, H. Kamaya, and I. Ueda, "19f and1h nmr and noe study on halothane-micelle interaction: Residence location of anesthetic molecules," *Journal of Colloid and Interface Science*, vol. 198, no. 2, pp. 319–322, 1998.

[60] L. Koubi, L. Saiz, M. Tarek, D. Scharf, and M. Klein, "Influence of anesthetic and nonimmobilizer molecules on the physical properties of a polyunsaturated

lipid bilayer," *Journal of Physical Chemistry B*, vol. 107, no. 51, pp. 14500–14508, 2003.

[61] S. Vemparala, L. Saiz, R. G. Eckenhoff, and M. L. Klein, "Partitioning of Anesthetics into a Lipid Bilayer and their Interaction with Membrane-Bound Peptide Bundles," *Biophys. J.*, vol. 91, no. 8, pp. 2815–2825, 2006.

[62] P.-L. Chau, P. N. Hoang, S. Picaud, and P. Jedlovszky, "A possible mechanism for pressure reversal of general anaesthetics from molecular simulations," *Chemical Physics Letters*, vol. 438, pp. 294–297, 2007.