Molecular Dynamics Simulation of Electroporation

A thesis submitted in partial fulfillment of the requirement
for the degree of Bachelor of Science with Honors in
Physics from the College of William and Mary in Virginia,

by

Joshua J Waterfall

Accepted for _____
(Honors, High Honors or Highest Honors)

_____
Advisor: Dr. Manos

_____
Dr. Saha

_____
Dr. Griffioen

Williamsburg, Virginia
May 2001

Abstract

When an external electric field is pulsed at a cell membrane the permeability of the membrane drastically increases due to a phenomenon known as electroporation. This is a technique often used in the laboratory to introduce foreign molecules to a cell. The exact mechanism which causes the increase in permeability is not well understood. This computational study is presented as a simulation of the behavior of biological membrane molecules in the presence of a pulsed external electric field. My research has shown that the electric field varies substantially over the cell membrane and that when a single molecule is studied through the interaction of its dipole moment with the applied field, pulse duration has minimal effect and that dipole - external field interactions alone are too slow to account for pore formation in the time scales observed experimentally.

Acknowledgements

I would like to thank Dr. Manos for his guidance in this project and in general over the last three years. I would also like to thank Dr. Morgan for introducing me to and helping me with Gaussian and to Dr. Poutsma for helping me both with Gaussian and its graphical interface, GaussView. Dr. Zhang was an immense help with the molecular dynamics simulations. Dr. Bradley and Dr. Saha explained the interest in electroporation from a biologist's point of view. I would like to thank Dr. Griffioen for helping me with all the administrative factors of this research.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Overview of Cells 2

1.3 Overview of Cell Membranes 3

1.4 Electrophysiology and Electroporation 5

2 MAGIC **8**

2.1 Introduction 8

2.2 Initialization 9

2.3 MAGIC Results 11

3 Gaussian **20**

3.1 Introduction 20

3.2 The DOPC Molecule 21

3.3 Calculations with Gaussian 22

3.4 Gaussian Results 24

4 Molecular Dynamics Simulation **24**

4.1 Introduction 24

4.2 My Molecular Dynamics Simulation 26

4.3 MD Results 28

5 Discussion and Future Work **31**

5.1 MAGIC 31

5.2 Gaussian 32

5.3 MD Simulations 33

6 Appendix A **36**

6.1 MAGIC Input File 36

7 Appendix B **46**

7.1 Gaussian Input 46

7.2 Gaussian Output 52

8 Appendix C **88**

8.1 MD Simulation Code 88

1 Introduction

1.1 Motivation

Although the computer, nearly since its invention, has been used to study physics, using it to study biology has been a relatively recent and explosive event. The computer does not take the place of laboratory experiment but it can be used to both motivate further experiments and to interpret results. According to a recent article in the New York Times, "All Science is Computer Science," computer simulations have been put to use in biology only within the last decade. One reason for this is that the types of problems of interest to biologists have only become affordable on supercomputers within the last ten years.[2]

The amount of data which streams in from the Human Genome Project and from protein crystallography studies demand that the programs used to interpret the data be as efficient as possible. In dynamic simulations, the amount of data present and the number of calculations required limit the time scales to being on the order of nanoseconds. This is at the bottom end of the time scale for biological processes of interest.

The present study involves dynamics which occur over time periods ranging over twelve orders of magnitude: from picosecond time scales to several seconds. At the quickest end of the spectrum is the molecular vibrations of individual phospholipid molecules in the cell membrane. From there, the pulsed electric field transients last on the order of picoseconds. The pulses themselves last anywhere from nanoseconds to milliseconds while phospholipid realignment on the nanosecond time scale leads eventually to pore formation. Once pores exist, ions and small molecules diffuse into and out of the cell up until the pores close and membrane permeability returns to preschock levels, a process which can take up to several seconds to occur.

There are no computer resources available today that can track all the parameters

of interest in a full electroporation simulation over the entire time scale. It is possible, however, to make certain simplifying assumptions that make the problem tractable. The first thing to do is to model the membrane as a lumped circuit in which the important electronic features are included but the geometry of the problem can not respond to the field. Studying the effects of a pulsed electric field on a single molecule representative of those present in a biological membrane offers valuable information as well. In order to do this, the researcher must have as much information about the molecule in its ground state, before it is pulsed with the external field, as possible. This research program has taken just such a modular approach to electroporation, studying the dynamics of the system on different time scales separately and combining the information from each study to get a better picture of the process as a whole. This project has been the first step which researchers at William and Mary intend to make into the investigation of the interactions between electromagnetic fields and biological materials. The purpose of this thesis has been to get preliminary results and set up the facilities for full molecular dynamics simulations of the interaction between electromagnetic fields with biological materials.

1.2 Overview of Cells

The structure and organization of the eukaryotic cell as a whole and its subcellular organelles is a pattern which nature uses repeatedly. All eukaryotic cells are surrounded by a lipid membrane which separates the internal cytosol, nucleus and other organelles from the extracellular space. All eukaryotic cells contain a nucleus, inside of which is stored the organism's DNA. While cells can differ substantially in their exact components, the presence of mitochondria, the endoplasmic reticulum(ER), the Golgi complex, lysosomes, peroxisomes and cytoskeletal proteins is common. The component of the eukaryotic cell of most concern to this project is the phospholipid bilayer, the major constituent of the outer membrane, nuclear membrane and the

mitochondrial membrane.

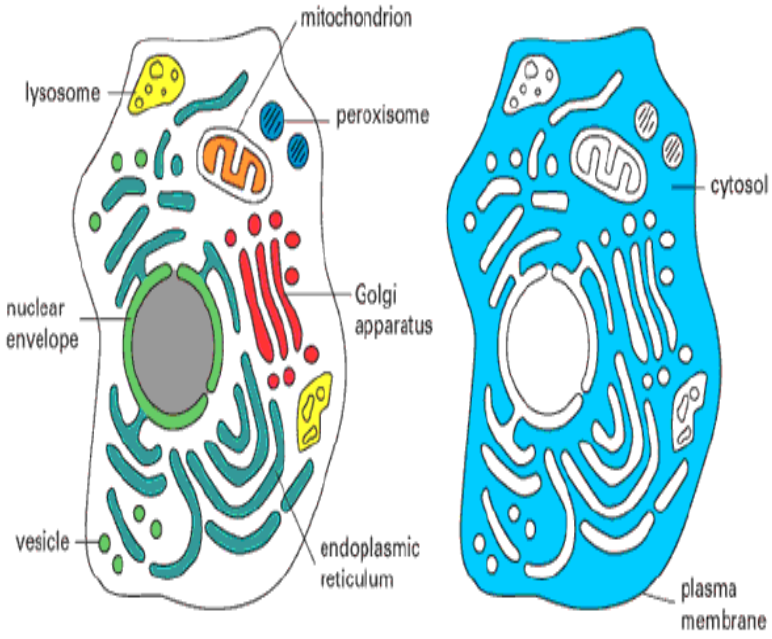

Figure 1: Typical Eukaryotic Cell

1.3 Overview of Cell Membranes

Cells are very efficient at controlling precisely what gets into and out of them. One of the fundamental defenses is the phospholipid bilayer membrane which surrounds all cells. A phospholipid is a molecule with a charged or polar head group and a long hydrocarbon chain tail (Fig. 2). The polar or charged nature of the head groups causes both the interior (cytosolic) and extracellular faces of the membrane to be hydrophilic while the non-polar hydrocarbon chains within the membrane, between the faces, are highly hydrophobic (Fig. 3).

Due to the hydrophobic/hydrophilic forces in a phospholipid bilayer only gases, such as CO_2 and O_2, and small uncharged polar molecules, such as ethanol and urea, can passively diffuse across the membrane. The passage of anything else across the membrane is highly regulated by transport proteins. Routine transport methods can

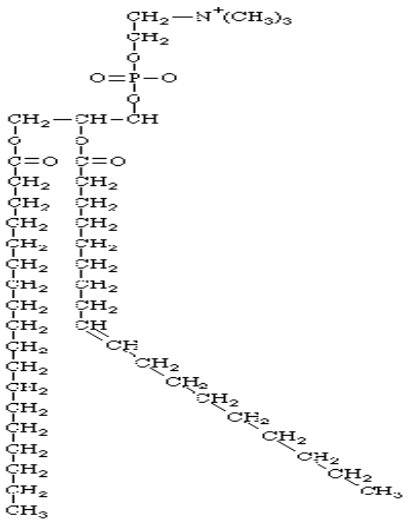

Figure 2: Typical Phospholipid Molecule

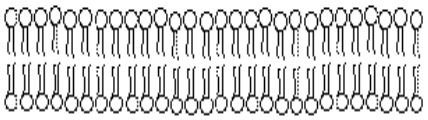

Figure 3: Phospholipid Bilayer

be classified into three major categories: pumps, channels and transporters. By regulating the passage of molecules and ions into and out of the cell, the membrane maintains both an electric potential gradient and an chemical concentration gradient between the inside and the outside of the cell. Protein pumps use energy, usually from ATP hydrolysis, to drive ions or small molecules against their electro-chemical gradient. Protein channels undergo conformational changes upon ligand binding to allow water or specific ions to travel down their electro-chemical gradient. Since this is a energetically favorable process, channels do not require energy input the way pumps do. A transporter binds to only a single molecule or a small number of molecules at a time and can be further categorized into three groups. Uniporters are proteins which change shape so as to move the molecules down their electro-chemical gradi-

ent. This is an energetically favorable reaction which requires no ATP hydrolysis or other energy input. Symporters and antiporters couple the energetically favorable movement of one molecule down its electrochemical gradient to the energetically unfavorable movement of another molecule up its electrochemical gradient as an energy source. The difference between symporters and antiporters is that symporters move both molecules in the same direction (into or out of the cytosol) while the molecules transported by antiporters are going in different directions.

Although this highly regulated nature of transmembrane transport is essential to life, it can pose a substantial problem in the laboratory. Experimental programs in clinical drug delivery, antibody production and DNA plasmid introduction for genetic modification all rely on introducing foreign molecules into cells. Such molecules most likely will not passively diffuse across the membrane. Unless they are structurally equivalent to a molecule the cell typically deals with, these molecules will not be admitted by the transmembrane proteins. The laboratory researcher has a great need of effective methods by which the molecule of choice can be placed into the cell without causing irreparable damage.

1.4 Electrophysiology and Electroporation

External electric fields are known to have certain effects on biological tissues and other biological materials. The field of electrophysiology is concerned with the how electric and magnetic fields affect, and are affected by, biological substances. One example is the injury potential: the phenomenon by which an electric potential causes cells to regress towards the stem cell state. The naturally occurring electric potential resulting when tissues are damaged is thought to initiate the process by which salamanders regenerate limbs. The application of external potentials has led to a method by which bone healing can be accelerated[8]. Currently, possible effects from the use of cell phones is under study as well.

The process of electroporation is another example of electric fields causing changes in biological tissues. In electroporation, a short pulse of a very strong electric field is applied. If the applied field is too strong the cell membrane will rupture and the cell will die. If the field is too weak nothing significant will be induced. For applied potentials on the order of 1 kV and pulses on the time scale of 10^{-6} to 10^{-3} seconds, the cell will experience "reversible electrical breakdown" (REB) where nearly all ions and molecules are allowed to travel between the extracellular space and the cytosol, and vice versa, for a period of time on the order of several seconds before the permeability returns to pre-shock levels.

This process occurs at time scales and spatial dimensions outside the reach of direct observation. Due to this lack of experimental observation of the mechanism for electroporation, several theories have been put forward to explain it.

The most widely accepted explanation for electroporation is that thermal fluctuations in the lipid bilayer cause "holes" between adjacent hydrophilic head groups to increase randomly and transiently. In the presence of an electric field, these "holes" can be turned into hydrophilic pores consisting strictly of lipid molecules. The potential across the membrane causes the lipid molecules to rearrange such that the head groups form the lining of a pore through the membrane. This type of pore is thought to be stable for a time period on the order of seconds before thermal vibrations force the lipid molecules back to the original configuration. Details, however, are lacking.

The theoretical work concerned with this subject consists primarily of solving partial differential equations for pore density as a function of pore radius[8,12] or modeling the cell as an equivalent electrical circuit and computing electric fields and currents[17]. Much of the focus of that work is to derive equations from experimental data instead of *ab initio*. The primary equations which dominate the first type of study are Maxwell's Equations governing electric and magnetic fields in matter taken in combination with driven-diffusion in the Smoluchowski equation.

$$\nabla \cdot \mathbf{D} = \rho_f \tag{1}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{2}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{3}$$

$$\nabla \times \mathbf{H} = \mathbf{J}_f + \frac{\partial \mathbf{D}}{\partial t} \tag{4}$$

$$\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P} \tag{5}$$

$$\mathbf{H} = \frac{1}{\mu_0}\mathbf{B} - \mathbf{M} \tag{6}$$

$$n_t + D\partial_r \left(-\frac{\phi_r}{kT} n - n_r \right) = S(r) \tag{7}$$

In Maxwell's Equations $\mathbf{D}$ is the electric displacement, ρ_f is the free charge, $\mathbf{E}$ is the total electric field, $\mathbf{B}$ is the total magnetic field, $\mathbf{J}_f$ is the free current, $\mathbf{P}$ is the polarization of the material, $\mathbf{M}$ is the magnetization of the material, ϵ_0 is permittivity of free space, and μ_0 is the permeability of free space. In the Smoluchowski equation D is the diffusion constant of molecules across the membrane in the presence of pores (not to be confused with the electric displacement of Maxwell's Equations), $\phi(r)$ is the pore energy, k is the Boltzmann constant, T is the temperature in Kelvin and $S(r)$ is a source term controlling the opening and closing of pores. Derivatives with respect to time are denoted by the subscript t and both ∂_r and the subscript r represent differentiation with respect to pore radius. The solution to the Smoluchowski equation is in quantities not directly measurable in the lab and is in terms of several constants which also can not be measured directly and are only known from theory to order of magnitude precision.

The second type of theoretical exploration of electroporation is done by devising an driven RC electronic circuit which mimics the applicable properties of the cell. Once a circuit has been devised, Kirchhoff's rules and Ohm's Law yield differential equations for the current in the circuit which corresponds to the ion flux across the

membrane. Kirchhoff's rules and Ohm's Law are as follows:

Rule 1 *When any closed circuit loop is traversed, the algebraic sum of the changes in potential must equal zero.*

Rule 2 *At any junction point in a circuit where the current can divide, the sum of the currents into the junction must equal the sum of the currents out of the junction.*

$$I = \frac{\mathcal{E}}{R} e^{\frac{-t}{RC}} \tag{8}$$

where I is the current, $\mathcal{E}$ is the potential, R is the resistance and C is the capacitance of the circuit.

2 MAGIC

2.1 Introduction

Software tools such as SPICE make it possible for modern electrical engineers to model the cell as a circuit with a very large number of components. It turns out that many of the most important features of a cell in an electric field can be modeled with just a few electronic components such as resistors and capacitors and thus an equivalent circuit can be constructed and studied with software such as SPICE. The first part of this thesis consists of studying electroporation with MAGIC[2], an electromagnetic particle-in-cell(PIC) software package. MAGIC is a finite-difference, time-domain code for simulating processes involving discrete, mobile charges and electromagnetic fields, it is not an circuit analyzer. The user inputs the initial state of the system and the code evolves the process through time by solving Maxwell's equations, the Lorentz force equation for charged particle trajectories and the continuity equation for current and charge densities.

$$\mathbf{F} = Q[\mathbf{E} + (\mathbf{v} \times \mathbf{B})] \tag{9}$$

$$\nabla \cdot \mathbf{J} = -\frac{\partial \rho}{\partial t} \tag{10}$$

The Lorentz force equation gives the force, $\mathbf{F}$, on a charge, Q in electric field $\mathbf{E}$ and magnetic field $\mathbf{B}$ travelling with velocity $\mathbf{v}$. The continuity equation relates the current flowing through an area, $\mathbf{J}$ to the time variation of the enclosed charge, ρ. As compared to the equivalent-circuit approach, MAGIC allows the geometry of the system to affect the electromagnetic fields. This is very important for electroporation research as the distribution of the electric field will have significant effects on pore formation and ion diffusion. PIC also has the advantage over lumped element circuit analysis that the fast transients of the applied field are not lost. Since pulse duration is a controllable parameter in electroporation experiments, calculations of how the fast transients of the field evolve in time is very important.

2.2 Initialization

The initial geometry set up for the MAGIC simulations was that of two parallel plates between which a voltage was applied and two concentric spheres to represent the outer membrane and the cytosol (Fig. 4). The radius of the outer sphere was defined to be 15 μm, the bilayer was defined to be 1.5 μm thick and the plates were made as opposing faces of a cube of length 120 μm. The value for the radius of the cell is fairly reasonable for a typical eukaryotic cell but the membrane thickness had to be increased for computational reasons. The thickness of a typical phospholipid bilayer is just a few nanometers but to create a grid that fine over the entire sphere was prohibitively time expensive for the software. The dielectric and capacitance properties for each area were defined to have physiologically relevant values. The relative permittivity ($\frac{\epsilon}{\epsilon_0}$) of the cytosol and extracellular space was 80 (modeling these areas as basically salty water) and that of the outer membrane was 4.958. This number is the predefined value for Teflon in MAGIC, a reasonable material to model the membrane as. The dielectric permeability of the cytosol and extracellular

space were defined as 0.3 F/m and the membrane was defined as 3×10^{-7} F/m. The values for the membrane were intentionally diminished to compensate for the increased thickness.

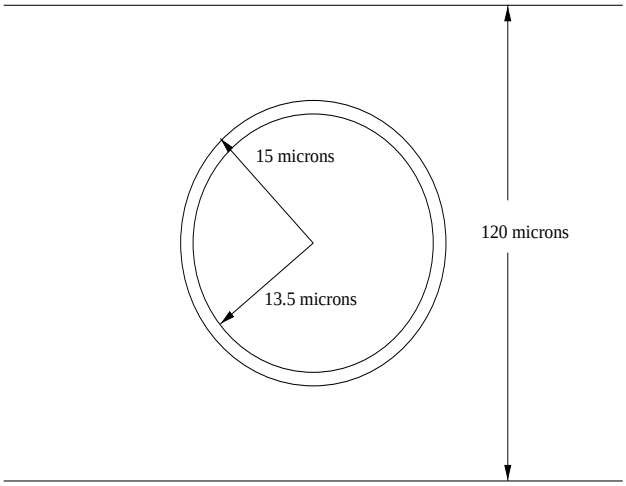

Figure 4: MAGIC Geometry: Not to Scale

Even with the thick membrane a rather complicated griding system had to be used to get as close to exact solutions as possible for the membrane while not wasting time with the same degree of accuracy for the largely invariant extracellular and cytosolic spaces. The membrane was marked to a 0.375 μm grid while the rest of the system was on roughly a 20 μm grid. The simulations were carried out until a roughly steady state was achieved, which proved to occur around 45 ps. The time integration was carried out in roughly 0.6 fs time steps. If too large a time step is used the results can not be relied upon; therefore 0.6 fs is used as the default calculated by MAGIC to maintain the Courant criterion given by $\chi < 1$:

$$\chi^2 = c^2 \delta t^2 \sum_{i=1}^{N} \frac{1}{(\delta x_i)^2} \tag{11}$$

Where c is the speed of light, δt is the time step and δx is the most restrictive spatial grid size in the simulation. MAGIC sets $\chi = 0.8$ to be safe and solves the Courant criterion as soon as all of the geometry is specified. This ensures that the calculations

are stable over successive iterations [2]. The potential applied between the plates was 150 kV, keeping very well within typical electroporator parameters. These simulations were run on a Hewlett-Packard 1 GHz Athlon processor desktop computer.

To construct this model, some details of the cell and cell membrane had to be lost. The coalescing of the individual phospholipid molecules into a sheath of membrane is the most obvious. This prevents the formation of pores in response to the field, but MAGIC does not allow the geometry to react to the fields even if discrete molecules are included. This model does not include any internal structure for the cell. As will be noted later, this is one aspect which can be improved upon in future research. The distribution of the field around the nucleus is of particular interest. This should not alter the results for this model; however, it would only be additional, yet useful, information. The input file for MAGIC is Appendix A. Typical computation time was sixteen hours for a simulatoin of a 45 ps time interval.

2.3 MAGIC Results

The simulations in MAGIC showed that the electric field varied widely around the surface of the cell, with the highest values being at the poles, defined by the intersections of the cell surface with the perpendicular line connecting the centers of the two plates. This means that pore density should not be expected to be uniform around the surface of the cell. It is also a result which would never be achieved through the study of artificial planar bilayers either experimentally or theoretically. The steady state transmembrane voltage for spherical cells in an external field was derived analytically in the 1950s by H.P. Schwan[12] to be

$$\Delta\Phi = \frac{3}{2} ERcos\phi \tag{12}$$

where $\delta\Phi$ is the voltage across the membran, E is the applied field, R is the cell radius and *phi* is the polar angle from the center of the cell with respect to the applied field. This result is for cells with nonconducting membranes. In the 1990s Kotnik et al.

extended this result to conducting membranes and derived[12]

$$\Delta\Phi = \frac{3}{2}\frac{\sigma_e[3dR^2\sigma_i + (3d^2R - d^3)(\sigma_m - \sigma_i)]}{R^3(\sigma_m + 2\sigma_e)(\sigma_m + \frac{1}{2}\sigma_i) - (R-d)^3(\sigma_e - \sigma_m)(\sigma_i - \sigma_m)} ERcos\phi \qquad (13)$$

Where σ_i, σ_m, and σ_e are the conductivities of the cytoplasm, membrane and extracellular space, respectively, and d is the thickness of the membrane.

In the MAGIC simulation the z component of the electric field at the two poles leveled out at approximately 17.5×10^4 V/m (Fig. 8) while at the equator the maximum z component of the field was 35×10^3 V/m (Fig. 9, 10). This is in contrast to the predictions made by Kotnik's equation where the potential across the membrane goes as the cosine of the polar angle and would therefore rigorously be zero at the equator.

If the medium between the two plates was uniform, there would only be a z component to the electric field. The presence of the cell causes refraction of the field and hence the x and y components are no longer zero. The MAGIC results show that at the equator, the x and y components are of order 10^{-10} V/m (Fig. 11, 12, 13, 14) which is small compared to the z component of the field at the equator. At the poles, however, both the x and y components have magnitude 15×10^3 V/m (Fig. 15, 16), only one order of magnitude smaller than the z component of the field at the poles. At the "south pole" both x and y components are negative while at the "north pole" both the x and y components are positive. This lack of preference between the x and y coordinates is reassuring that the symmetry of the system is being preserved in the long numerical sequence of calculations. There are two possible explanations for the change of sign. The first possibility is that it is related to the curvature of the membrane at the poles - where the curvature is positive the x and y components are negative while where the curvature is negative the x and y components are positive. The other possibility is that these are residues of the model initial conditions. The way MAGIC is constructed, the potential difference between the plates is ported into the system from one of the "invisible" faces between the conductor plates. In this

simulation the potential enters at y=0 and travels in the positive y direction at a velocity equal to the speed of light in the medium. All transient effects of this propagation are removed from the fields long before 45 picoseconds have passed but it is still possible that the direction of the x and y components of the field at the poles are related to this initial asymmetry. The best way to check this would be to port the potential from the opposite face, propagating in the negative y direction.

If the electric field is not constant over the face of the cell then pore density should not be constant either. The type of field distribution observed in MAGIC should lead to substantially higher pore density at the polar regions than the pore density at the equatorial regions. This result has been experimentally verified by several recent fluorescence imaging experiments[4]. Another interesting feature of the spatial distribution of the electric field is that just outside the polar areas of the membrane the field dips before it spikes throughout the width of the membrane. Many different griding systems were used and various applied voltages to see if the dip was a numerical artifact of the PIC method. The dip was not only consistently present, but the ratios of the dip depth and width to the spike height and width were approximately constant. This provides an interesting point to pursue in future studies, both experimentally and theoretically.

The steady state of the simulation was achieved in less than 45 ps. This is an overestimate however because MAGIC applies the voltage as a traveling wave which enters through one of the side faces of the cube. This leads to reflection and refraction of the field as it encounters the cell which would not occur in a laboratory electroporator. It is in the direction that the field travels that 45 ps was necessary for equilibrium to be achieved while the other two directions achieved a constant state within just a few ps. Even 45 ps is very much shorter than the shortest pulse times used in electroporation. Although the field does have a ramping-up time, the associated transients are removed by the system too quickly for them to affect the

geometry. The complicated transient properties of the applied field can be ignored when examining pore formation.

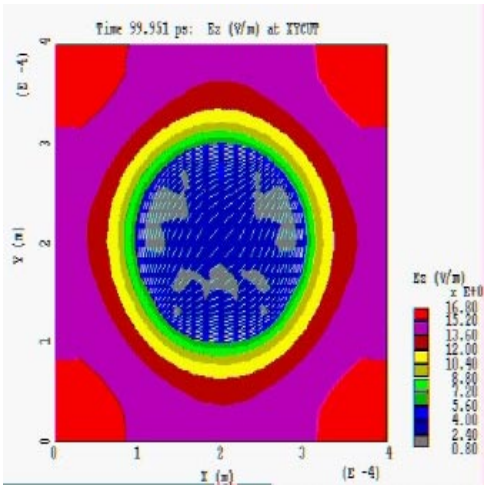

Figure 5: Z component of electric field in XY plane through center of cell

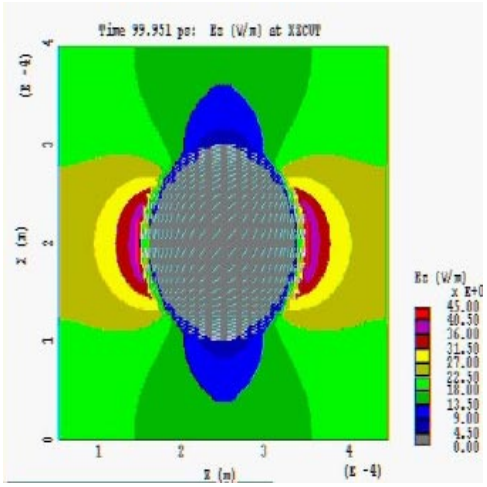

Figure 6: Z component of electric field in XZ plane through center of cell

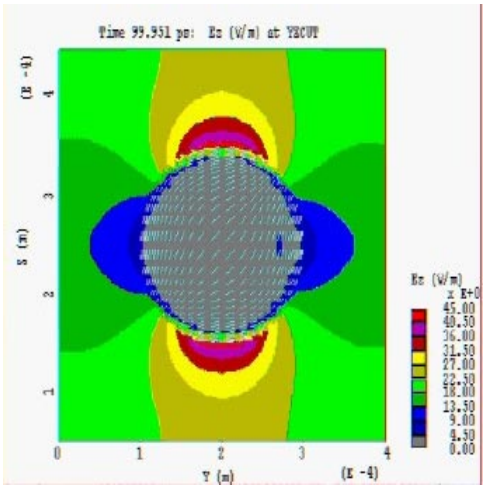

Figure 7: Z component of electric field in YZ plane through center of cell

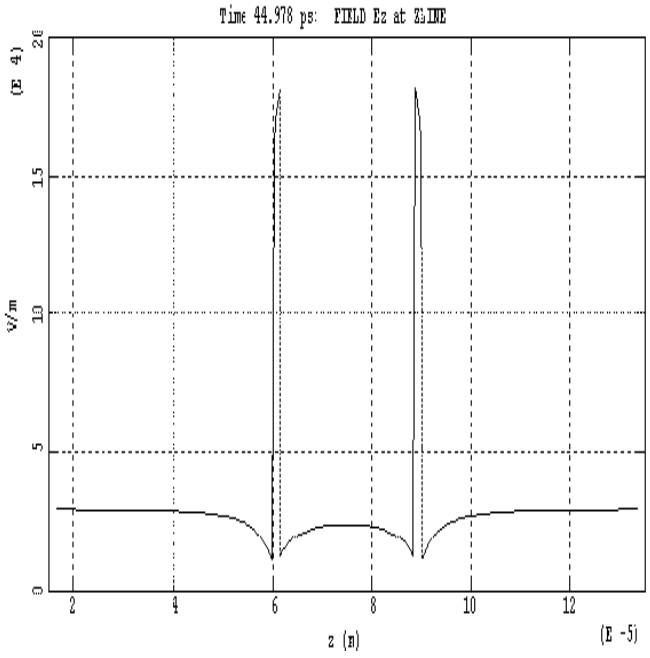

Figure 8: Z component of electric field along Z axes through center of cell

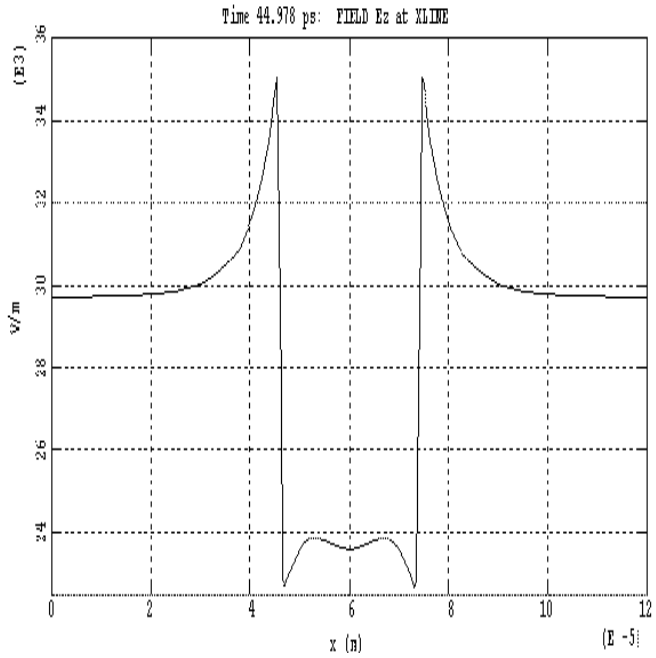

Figure 9: Z component of electric field along X axes through center of cell

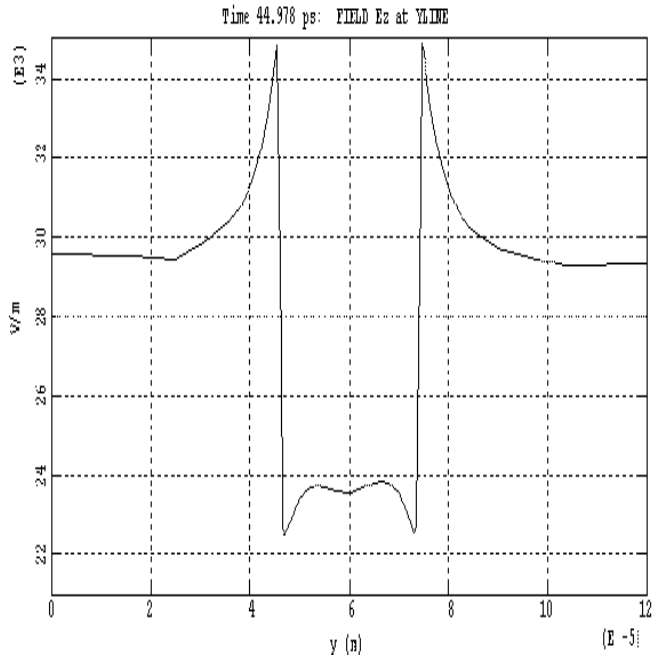

Figure 10: Z component of electric field along Y axes through center of cell

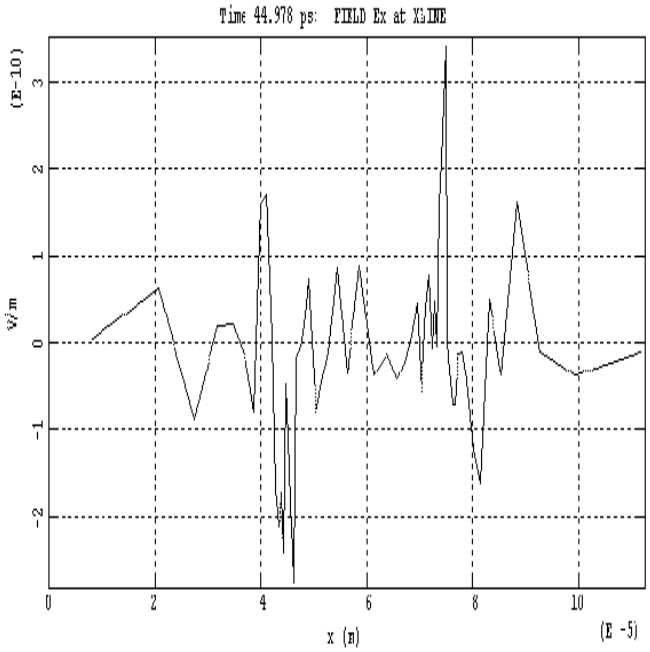

Figure 11: X component of electric field along X axes through center of cell

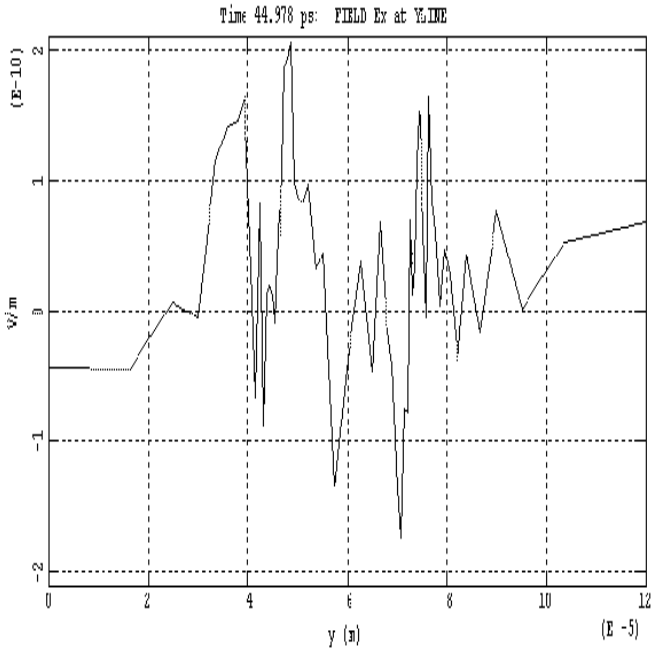

Figure 12: X component of electric field along Y axes through center of cell

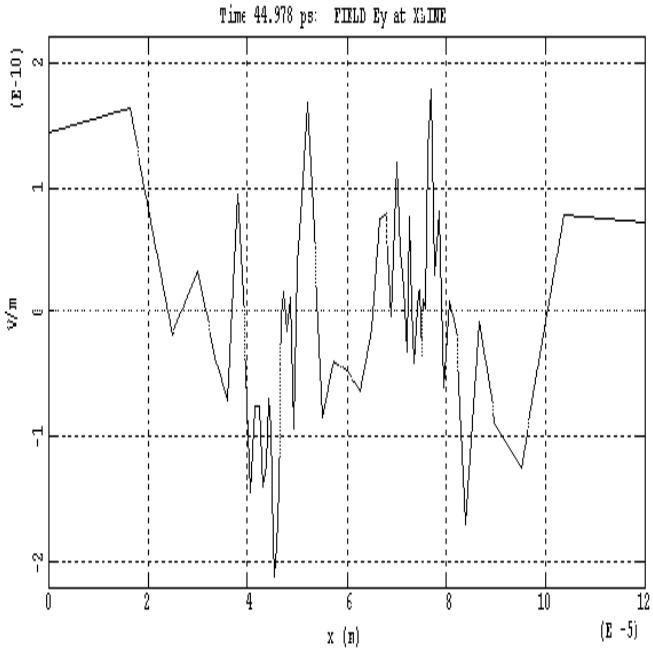

Figure 13: Y component of electric field along X axes through center of cell

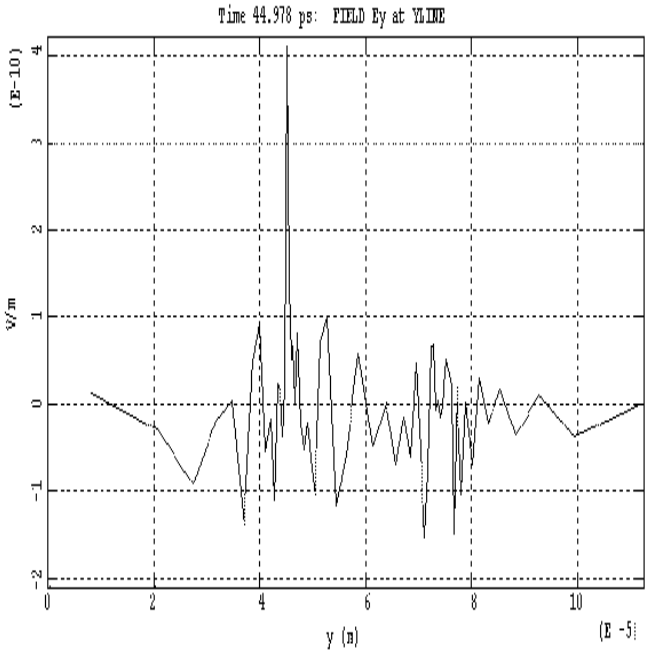

Figure 14: Y component of electric field along Y axes through center of cell

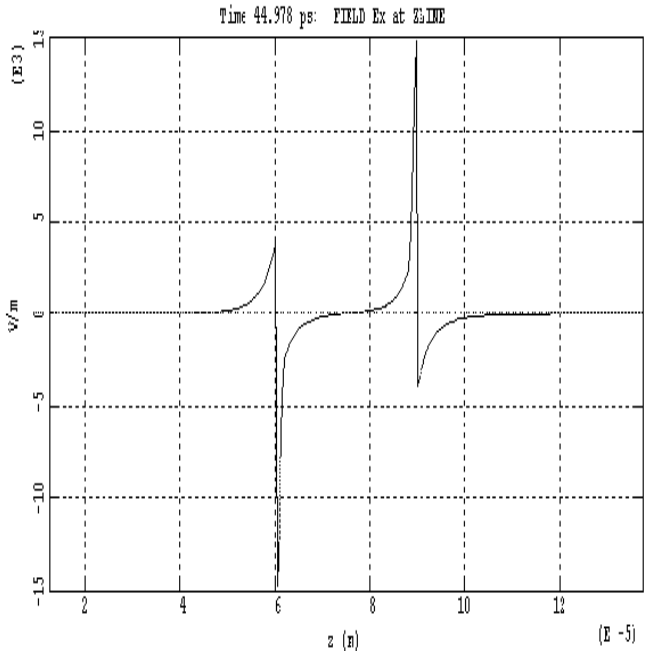

Figure 15: X component of electric field along Z axes through center of cell

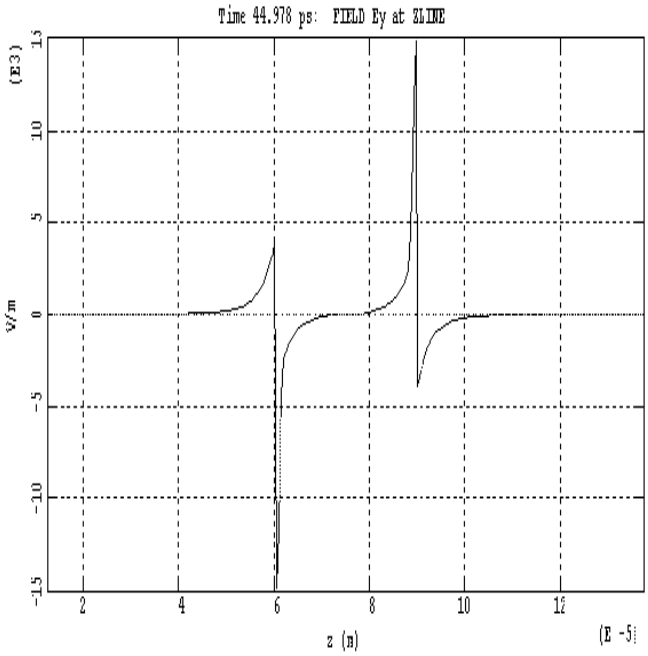

Figure 16: Y component of electric field along Z axes through center of cell

3 Gaussian

3.1 Introduction

There is one obvious problem with using the equivalent electrical circuit approach to electroporation - it does not allow the system's structure to change in response to the field. Since the results from MAGIC demonstrated that the transient effects of the applied field are present for only a few tens of picoseconds, the fields can be taken to be stationary for single pulse electroporation. The electric field spatial distribution from MAGIC can be used as input to a second module of simulation to determine the effects on the cell membrane. Before attempting to compute how an entire phospholipid bilayer would respond to an electric field, it is useful to consider the effects on selected regions of it. Motion of the layer will result from Coulomb interactions of the field so knowledge about the charge distribution (or dipole moments) and the force constants between atoms and molecules is fundamental to the response of the system as a whole.

The next step in this thesis was to turn to computational quantum chemistry to gather as much information as possible about a single phospholipid molecule taken as a representative of active regions of the membrane. Theoretically, getting this information requires solving the Schrödinger equation for the whole lipid molecule.

$$\imath\hbar\frac{\partial\psi}{\partial t} = -\frac{\hbar^2}{2m}\nabla^2\psi + V(\mathbf{r},t)\psi \tag{14}$$

Where V($\mathbf{r}$, t) is the potential energy function of the entire system. To obtain the best approximation to the solution for the Schrödinger equation we used the software package Gaussian[4]. Gaussian is a very comprehensive program which can study large molecules to various degrees of accuracy and with many different theoretical schemes.

3.2 The DOPC Molecule

Any cellular membrane consists of a wide variety of phospholipids, therefore, a specific membrane component is needed which is both common in the cell membrane and for which initial atomic positions are available. This thesis focuses on the dioleoyl phosphatidylcholine (DOPC) molecule (Fig. 17). DOPC is an unsaturated member of the phosphatidylcholine family which is a common constituent of biological membranes. DOPC is well-studied, and the equilibrium coordinates for each of the 138 different atoms are available [6] . These coordinates for the atoms in a DOPC molecule are the end product of a 1500 ps simulation of a purely DOPC lipid bilayer in physiological conditions[5]. This time scale is long enough to assure relatively stable positions corresponding to free molecules in equilibrium. The next step is to input these coordinates into Gaussian to calculate the features of interest to us, namely the dipole moment, the charge distribution and the force parameters. The input and output files for Gaussian are contained in Appendix B.

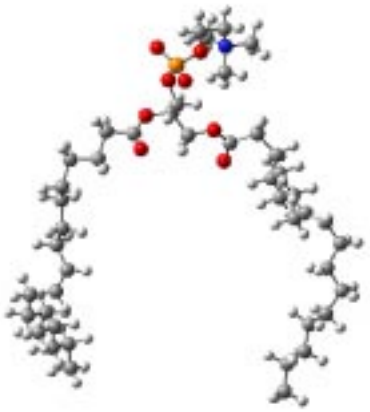

Figure 17: DOPC molecule from Gaussian

3.3 Calculations with Gaussian

Gaussian requires that the user input the method and basis set to be employed. The method corresponds to whatever level of theory is chosen to solve the Schrödinger equation. Gaussian has many such options. For this thesis Restricted Hartree-Fock theory was applied. Hartree-Fock Theory makes the assumption that the total wave function can be decomposed into the product of molecular orbitals for each electron.

$$\psi(\mathbf{r}) = \phi_1(\mathbf{r_1})\phi_2(\mathbf{r_2})\cdots\phi_n(\mathbf{r_n}) \tag{15}$$

Where the individual wave functions are normalized and mutually orthogonal,

$$< \phi_i \mid \phi_i >= \delta_{ij} \tag{16}$$

This formulation only considers the spatial wave functions of the electrons. The spin component must also be included. The "Restricted" in Restricted Hartree-Fock means that electrons are grouped into pairs and a single wave function, consisting of a spatial component and a spin component, is defined for each pair. Individual electrons can have either $spin = +\frac{1}{2}(\uparrow)$ or $spin = -\frac{1}{2}(\downarrow)$ Hartree-Fock theory defines two spin functions for the electron, α and β such that

$$\begin{aligned} \alpha(\uparrow) &= 1 \qquad \alpha(\downarrow) = 0 \\ \beta(\uparrow) &= 0 \qquad \beta(\downarrow) = 1 \end{aligned} \tag{17}$$

Since the total wave function for the molecule must be antisymmetric under interchange of electrons, the Restricted Hartree-Fock wave function for a molecule with n electrons can be written as the Slater determinant of the $n \times n$ matrix composed of every combination of spatial and spin wave functions (with a normalization factor):

$$\psi(\mathbf{r}) = \frac{\det}{\sqrt{n!}} \begin{bmatrix} \phi_1(\mathbf{r_1})\alpha(1) & \phi_1(\mathbf{r_1})\beta(1) & \phi_2(\mathbf{r_1})\alpha(1) & \phi_2(\mathbf{r_1})\beta(1) & \cdots & \phi_{\frac{n}{2}}(\mathbf{r_1})\beta(1) \\ \phi_1(\mathbf{r_2})\alpha(2) & \phi_1(\mathbf{r_2})\beta(2) & \phi_2(\mathbf{r_2})\alpha(2) & \phi_2(\mathbf{r_2})\beta(2) & \cdots & \phi_{\frac{n}{2}}(\mathbf{r_2})\beta(2) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{r_n})\alpha(n) & \phi_1(\mathbf{r_n})\beta(n) & \phi_2(\mathbf{r_n})\alpha(n) & \phi_2(\mathbf{r_n})\beta(n) & \cdots & \phi_{\frac{n}{2}}(\mathbf{r_n})\beta(n) \end{bmatrix}$$

For this thesis the 6-31G(d) basis set was used, which includes 982 basis functions and 1872 primitive gaussians. One basis function is assigned per atom as its molecular orbital. Each basis function is made up of several primitive gaussians. This basis set is widely used in computational chemistry for up to medium atomic number atoms because it includes functions to account for d type orbitals. Since phosphorous has the largest atomic number of all the elements in a DOPC molecule, relativistic effects do not really need to be accounted for as would be necessary with larger nuclei.

Gaussian has to calculate the coefficients for each of its basis functions to get the proper linear combination for the ground state of the molecule. According to the variational principle, the ground state of any antisymmetric set of orthogonal wave functions will give an energy expectation value greater than the energy of the true ground state. This means that as the combination of basis functions is refined by iteration and lower energies are calculated, the true ground state energy of the system is approached as a lower bound. Gaussian employs the *Self-Consistent Field*(SCF) method to converge to this ground state value. In SCF, the current values for the wave functions are used to calculate the field potential they generate. The effect of this field on the same wave functions is then used to refine the wave functions. This process is repeated until the fields generated do not alter the wave functions and the wave functions do not alter the fields to within a certain convergence limit.

Once the Restricted Hartree-Fock method and 6-31G(d) basis set were chosen, the position and type of each atom in the DOPC molecule were specified. For this simulation we specified that the net charge on the molecule was 0 and that the multiplicity was 1. Real membranes may be charged and this could be an area of future investigation. The multiplicity of a molecule is related to the number of unpaired electrons, with 1 meaning there are no lone electrons. These Gaussian calculations were also run on a Hewlett-Packard 1GHz Athlon processor desktop machine and typical clock time requirements were on the order of a few hours.

3.4 Gaussian Results

From the Gaussian simulation we extract the charge distribution and the dipole moment of the DOPC molecule. As one might expect, the dipole term is dominated largely by the single bonded oxygen in the phosphate which tends to gather electrons and the nitrogen in the choline which tends to lose electrons. Combining this with the arrangement of these atoms in the molecule it is apparent that the net dipole moment is roughly perpendicular to what would be the radial vector of a cell, i.e. it is tangential to the membrane. The net dipole moment is of magnitude 21.195 Debye. Gaussian outputs the fractional charge on each atom and also sums the charges of the hydrogens into the heavier atoms, very useful information for future molecular dynamics simulations. The fractional charge data shows that the hydrogens tend to lose electrons, the phosphorus atom has the largest fractional positive charge of all atoms (roughly 1.5 Coulombs) and that the carbons and oxygens tend to become negative. The SCF energy calculated by Gaussian is -2709 Hartrees. It took 8 cycles to converge to this value within one part in one hundred thousand.

4 Molecular Dynamics Simulation

4.1 Introduction

The closest one has come to directly observing electroporation has been using fluorescence techniques. These experiments visualize molecules that have travelled through the membrane but the pores themselves are not visible.[3] This lack of sufficient microscopy techniques to directly image in real time the process of electroporation makes computer simulations of the phenomenon necessary. Proper simulations allow the researcher to mimic laboratory situations and see how individual molecules in the membrane react to an applied field. While a full scale molecular dynamics simulation would provide valuable information about electroporation it is not currently feasi-

ble on even the fastest modern systems. The present upper limit on biomembrane molecular dynamics simulations is around one nanosecond. Not only is the return to preshock permeability beyond this limit by many orders of magnitude, even pore formation is outside this range. For these reasons the simulations carried out in this thesis reduce the massively complex process of electroporation to the interactions of the dipole moment of a single DOPC phospholipid with an external field.

When a dipole is placed in an electric field it feels a torque and a force,

$$\tau = \mathbf{p} \times \mathbf{E} \tag{18}$$

And a force,

$$\mathbf{F} = (\mathbf{p} \cdot \nabla)\mathbf{E} \tag{19}$$

Where $\mathbf{p}$ is the dipole moment. This torque tends to align $\mathbf{p}$ parallel to $\mathbf{E}$. In electroporation, the dominant term is the torque although the electric field spatial distribution from MAGIC suggests that the force on the dipole should also be investigated. In electroporation, since the dipole moment of the phospholipid is roughly perpendicular to the radial vector of the cell, the phospholipids at $\theta = \frac{\pi}{2}$ relative to the applied field should not rotate while the molecules at $\theta = 0, \pi$ relative to the external field will be twisted to align with the external field. The capacitive nature of the membrane will cause a build up of oppositely signed charges on either side of the membrane which will induce local alignments of the dipole moments. These two factors will cause the phospholipids to rotate into the interior of the membrane, thus forming a pore.

This type of motion is very complicated. It is the motion of an asymmetric top with a constantly varying torque and orientation. The precise torque on the phospholipid at any one moment in time is calculable with out a computer but tracking the rotation through time necessitates a computational approach.

4.2 My Molecular Dynamics Simulation

The program which was developed for this thesis performs a simple Euler-Cromer method time-domain integration to study the rotation of a DOPC molecule in a uniform external field. The first step is to initialize the program by inputting the relevant information from Gaussian and MAGIC: the rotational constants, the dipole moment and the values for the external electric field. The moments of inertia are calculated from the rotational constants by the equation

$$I = \frac{\hbar}{4\pi B} \tag{20}$$

B is a spectroscopically determined rotational energy coefficient, which can be calculated in Gaussian. These rotational constants define the coordinate system in which the moment of inertia matrix for the molecule is strictly diagonal. As the torque on the molecule is computed at each time step the molecule is rotated about each of the three axes defined by the rotational constants. This new alignment of the dipole moment with the external field leads to a new torque at the next time step. The following are the kinematic equations of rotation:

$$\begin{aligned} \omega &= \omega_0 + \alpha t \\ \theta &= \theta_0 + \omega_0 t + \frac{1}{2}\alpha t^2 \end{aligned} \tag{21}$$

The Euler-Cromer method takes these differential equations for the angle and angular velocity of the particle of interest and solves it according to the following procedure:

$$\begin{aligned} \omega_{i+1} &= \omega_i + \frac{\tau_i}{I}\delta t - \frac{6\pi\eta r^3 \omega_i}{I}\delta t \\ \theta_{i+1} &= \theta_i + \omega_{i+1}\delta t \end{aligned} \tag{22}$$

Where η is the absolute viscosity of the medium the phospholipid rotates through. The last term in the equation for ω is a drag term to account for pushing water out of the way and any frictional forces slowing down the molecule from how it would spin

in a vacuum. It is derived from Stokes equation for three dimensional fluid flow.

$$\rho\frac{\partial \mathbf{u}}{\partial t} + \rho\mathbf{u}\cdot\nabla\mathbf{u} = \rho\mathbf{F} - \nabla p + \frac{1}{3}\eta\nabla\nabla\cdot\mathbf{u} + \eta\nabla\cdot\nabla\mathbf{u} \tag{23}$$

In the Stokes equation, ρ is the density of the liquid, $\mathbf{u}$ is the velocity of the liquid, $\mathbf{F}$ is the net external force per unit mass acting on the flow, p is the hydrostatic pressure of the liquid and η is the absolute viscosity. The equation can be much simplified for the type of dynamics being invesitgated with this simulation. First of all the medium in which the phospholipid rotates is stationary so the first term of the Stokes equation drops out. The velocity of the phospholipid is small enough that there is no energy loss to wave formation so $\mathbf{u}\cdot\nabla\mathbf{u}$ goes to zero because it depends on the square of the magnitude of the velocity. The applied external field acts dominantly on the phospholipid, not on the surrounding medium so the net external force, $\mathbf{F}$ is zero. Since the surrounding fluid is taken to be an incompressible liquid, rho is uniform everywhere and $\nabla\cdot\mathbf{u}$ is zero so the third term on the right hand side of the equation disappears. The two terms left describe the forces acting on the rotating phospholipid due to the changing hydrostatic pressure over space and due to friction. Integrating over the surface of a sphere representing an idealized phospholipid, the total drag force comes to $-6\pi\eta r\mathbf{u}$. The correction to this equation for a prolate ellipsoid is a factor of

$$\frac{(1-b^2/a^2)^{1/2}}{(b/a)^{2/3} ln\frac{1+(1-b^2/a^2)^{1/2}}{(b/a)}} \tag{24}$$

for a prolate ellipsoid with semi-axes a, b, b[9]. The viscosity of the medium is the only parameter which is variable in this simulation. Testing the relationship between viscosity and rotation rate is a significant application of this simulation code. Viscosity values for applicable fluids are easily available for testing. Water has a typical viscosity on the order of $10^{-3}Ns/m^2$ and glycerine and castor oil, as models for the lipid bilayer, have viscosities on the order of $0.1Ns/m^2$[10]. This damping term does account for all the major sources of energy loss in the dynamics being invesitgated and should offer a good check on the effects of viscosity on lipid rotation.

The Euler-Cromer algorithm was chosen over the simpler Euler method because of energy conservation concerns; in simulations of oscillatory motion the Euler method does not conserve energy no matter what time step is used, but the Euler-Cromer method does[7]. The difference between the two methods being the use of ω_{i+1} to calculate θ_{i+1}. At each time step the new angle, angular velocity and torque are calculated. Primary variables of interest are the time step and the drag coefficient since everything else is output of calculation either by MAGIC or by Gaussian. Since these simulations can run for longer than the nanosecond, or even millisecond, time scale, realistically long electroporation pulses of the electric field can be included in this module of the investigation. The results from MAGIC show that the transient effects of the field do not have important implications for the conformational state of the phospholipids so the MD simulation need not include the transients. The output parameter of interest is the time progression of the rotation of the dipole moment. Since this program models the phospholipid as a rigid body, the rotation of the entire molecule can be inferred from the dipole. These simulations were carried out on the Camelot Cluster of Pentium-II machines at the William and Mary Physics department. The C++ code is available in Appendix C. Depending on the magnitude of the viscosity in the simulation, runs took from less than a minute to on the order of an hour.

4.3 MD Results

Based on the MAGIC and Gaussian results, the most dynamic area of the cell in an electroporation event would be the polar regions near the plates with the applied voltage. The electric field is oriented nearly perpendicular to the dipole moment of the individual phospholipid molecules and is parallel to their long axes, roughly normal to the membrane. The rotation of the phospholipid is critically damped even for viscosity values well below that of water. For a viscosity of $10^{-5} Ns/m^2$ a DOPC

molecule is found to rotate by $\frac{\pi}{2}$ in roughly 100 nanoseconds (Fig. 18). The time for rotation increases dramatically with viscosity with a viscosity of $10^{-4}Ns/m^2$ requiring nearly one microsecond (Fig. 19). These viscosity values are orders of magnitude less than that of the phospholipid membrane or even water. When the viscosity of water is used in the simulation, rotation by $\frac{\pi}{2}$ takes just less than 10 microseconds (Fig. 20). Viscosities larger than that of water were not investigated due to computer clock time constraints. The water simulation took several hours and increasing the viscosity by two orders of magnitude to get values representative of a phospholipid bilayer[10, 11] would have increased the computational time to on the order of a day or more.

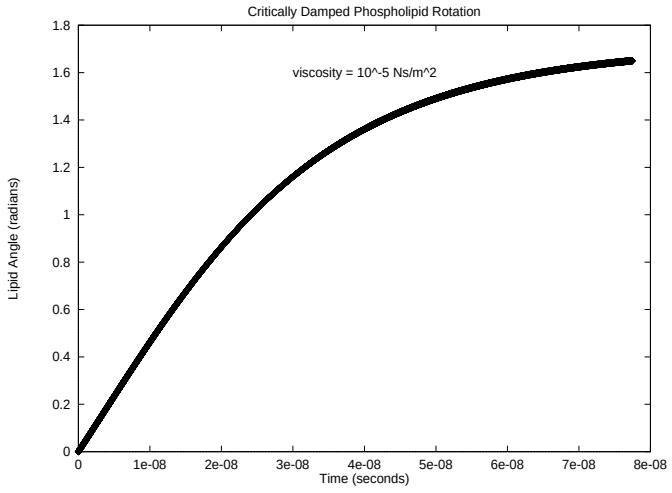

Figure 18: Lipid Rotation from MD Simulation for a viscosity of $10^{-5}Ns/m^2$

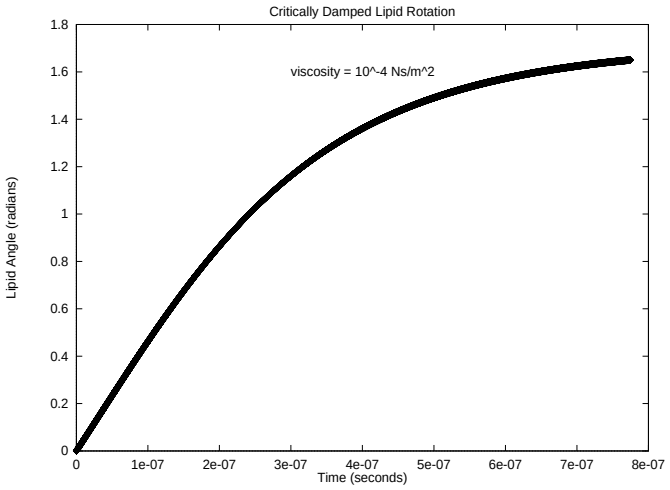

Figure 19: Lipid Rotation from MD Simulation for a viscosity of $10^{-4}Ns/m^2$

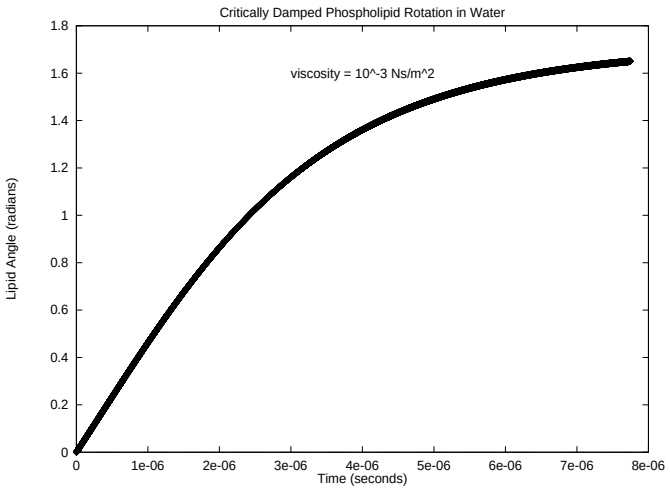

Figure 20: Lipid Rotation from MD Simulation for a viscosity of water, $10^{-3}Ns/m^2$

The code developed for this simulation is easily scalable. The input of data from Gaussian and MAGIC is very flexible and many different variables of the simulation can be outputted for further study. This code estimates electroporative phenomena in a highly simplified model to yield easily intelligible results. The closest type of experimental verification of these results would have to be by flow birefringence where macromolecules are oriented by the shearing force of a moving liquid[9]. The space between two concentric cylinders is filled with solution of interest. One of the cylinders is then spun while the other remains stationary. Since the molecules next to the

spinning cylinder will move with the same velocity as the cylinder while the molecules next to the stationary cylinder will remain at rest a velocity gradient is established in the solution. Macromolecules in the solution will feel a torque because, due to their extended nature, they experience a different flow velocity at different ends of the molecule. This torque will reach steady state with rotational diffusion, a sort of Brownian motion for rotation, and the molecules will all be aligned. This alignment can then be probed by shining polarized light on the apparatus and observing the polarization of the transmitted light. This type of experiment has been done for decades and is the best resource for determining the rate of rotational diffusion of macromolecules. Flow birefringence does not offer a specific check on the results of this simulation, however, because it does not involve any external field. Another problem is that the rates derived from flow birefringence are for solutions relatively dilute to the environment in which pore formation takes place. The fact that the equivalent experiment can not be conducted in the lab for verification of results is the strongest motivating factor to perform such a calculation.

5 Discussion and Future Work

5.1 MAGIC

While the results from MAGIC are very interesting there is, of course, much more that can be investigated. The model used in this thesis reduced the highly structured eukaryotic cell to a water filled sphere. If the program were run on a massively parallel system the problem of griding the thin membrane would be greatly reduced. Different geometries for the cell can be investigated in the future. There is evidence to show that electroporated cells are stretched along the direction on the field into a more elliptical structure which could be further investigated by MAGIC. Intracellular organelles could be included in the geometry. Of most interest would be to include a

nucleus inside the cell. Since one of the major uses for electroporation is to introduce DNA plasmids for genetic research, the presence of pores in the nuclear membrane would be an area of important research. The presence of other organelles such as the ER and Golgi complex could have a significant impact on the spatial distribution of the electric field. Small pores could even be introduced into the membrane to study how the system progresses assuming these pores are present.

5.2 Gaussian

The degree of accuracy and the amount of information that can be calculated by Gaussian is limited by the computing power of the machine and the amount of time one is willing to devote to a calculation. Future work with Gaussian could be to use higher order theories for higher accuracy than Restricted Hartree-Fock. Allowing the program to solve for each electron individually instead of as pairs could improve the results. Using a theory which allows for electron-electron interactions would also be an improvement. Methods such as Møller-Plesset perturbation theory where the Hamiltonian for the system is divided into an exactly-solvable part and a small perturbation, or configuration interaction methods which evaluate several orbitals of the Hartree-Fock type but with various orbitals replaced by empty orbitals to get a better picture of the possible positions of the electrons would be an improvement. Full scale density functional theory is an even higher order theory which separates the energy of the system into kinetic energy terms, terms for the nucleus-electron energy, electron-electron repulsion terms, and terms for the asymmetry of the wave function and any correlations in the motions of electrons. The other way to improve the Gaussian calculations is to use a better basis set. New basis sets are regularly being published, along with improvements on current ones. The 6-31G(d) is rather good for the types of atoms in a DOPC molecule and the only real improvement would be to use one such as 6-31+G(d) which includes diffuse wave functions, allowing orbitals

to cover a greater area of space. Since it is very unlikely that electrons in any of the atoms which compose a DOPC molecule are in anything higher than a d-orbital, highly polarized orbitals need not be included. Since the masses of the atoms are all relatively small, the high angular momentum orbitals are unnecessary.

5.3 MD Simulations

This thesis represents a preliminary step into the the study of the interactions between electromagnetic fields and biological materials at the College of William and Mary. It has been conducted in such a manner as to provide tools and results which can be employed and furthered in future research. Along with the improvements already outlined which could be made to the work done with MAGIC and Gaussian, the MD simulation is scalable for different types of updates.

The MAGIC results show that the components of the electric field parallel to the cell surface at the poles is an order of magnitude less than the perpendicular portion and is antialigned at the opposite poles. These components were not included in the simulation, however, because further investigation, as outlined above, is needed to verify the values for these induced components of the electric field. If they are of significant magnitude then they can be easily incorporated into the code as it stands already.

Improving the drag term in the simulation is an important refinement needed for the code. The drag term is the rate defining step as far as runtime is concerned. To study phospholipid rotation in a medium of viscosity similar to that of the lipid bilayer would require over a day on the machine the code is currently run on. Moving this simulation to a faster system or improving the code so that larger time steps can be used would drastically reduce this restraint. Calculating the drag coefficient from an *ab initio* standpoint would be very difficult since the nature of the medium through which the phospholipid molecule is moving is highly uncertain. If the lipid molecules

which eventually form the pore are uniformly packed then rotation would require the head groups to push water out of the way while the tails pushed through the fatty interior of the membrane. If the molecules which eventually form the pore are already slightly angled to cover water molecules which have leaked into the membrane prior to electroporation, the drag term would be quite different. In this case the number of water molecules already trapped in the membrane would have a large impact on the drag term as well. If there are only a few molecules then they can behave quite unlike bulk water; the hydrogen bonds between them could cause the formation of an ice-like substance. In this case, breaking the surface tension of the trapped material would be the rate limiting step in the rotation. Once the surface tension was broken it would be relatively easy to move through. This type of drag force would not be linear in v, as that of this simulation is. If, on the other hand, the hidden pore is on the scale of tens of Angstroms, then the water inside would have properties much more similar to those of bulk water and the conventional value for the viscosity of water could be used. Both of these scenarios provide for more rapid movement than when the initial state is the classic membrane. In this final case the polar head groups of the pore-forming lipids would be in much closer contact with the fatty interior of the membrane. The torque from the electric field required to overcome these repulsions would be much greater than that required to push aside already present water molecules.

The force of an electric field on a dipole, as opposed to the torque, is not included in this simulation. The MAGIC results could be used to construct a spatial griding of the electric field over the simulation area. This force could lead to shape deformation of the membrane which in turn would affect pore formation and ionic diffusion. On the single molecule level the force should be immaterial in relation to the torque, but in a multi-molecule simulation it would need to be included. The classical view of electroporation, where hydrophilic pores are formed strictly by cytosolic facing phospholipids and extracellular-space facing phospholipids rotating into the interior

of the membrane, in fact, requires a combination of translation and rotation. Rotation alone could never form a pore because it would only decrease the distance between phospholipids, not increase it.

Other possible improvements would be to move from Euler-Cromer to a more accurate and efficient time-integration method such as Runge-Kutta. This method offers a dramatic increase in the accuracy of the solution at minimal extra computational cost. The Runge-Kutta algorithm uses the fourth order Taylor approximation of a function and is integrated in time by

$$\frac{dy}{dt} = f(y,t) \tag{25}$$

$$\begin{aligned} y_{i+1} &= y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 &= \delta t f(t_i, y_i) \\ k_2 &= \delta t f(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}) \\ k_3 &= \delta t f(t_i + \frac{h}{2}, y_i + \frac{k_2}{2}) \\ k_4 &= \delta t f(t_i + h, y_i + k_3) \end{aligned} \tag{26}$$

This version of the MD simulation also does not include important factors such as the explicit presence of water molecules, elastic strain in the membrane, and ion flow. The clock time for the simulations could also be decreased dramatically by being ported to a massively parallel system such as the Beowulf cluster being designed at the College. This simulation could also be addended with the presence of proteins or nanofabricated artificial pores.

6 Appendix A

6.1 MAGIC Input File

```
! ============================================================
GRAPHICS PAUSE ;
! ============================================================
HEADER AUTHOR       "Josh Waterfall" ; ! ... put your name here
HEADER ORGANIZATION "College of William and Mary" ;
HEADER DEVICE       "Electroporation of Single Cell" ;
HEADER REMARKS      "Cold Test" ;
! ============================================================
! ... cell and plate parameters



CELL_RADIUS = 15._Micron ;
BILAYER     = 0.1*CELL_RADIUS ;

PLATE_XWIDTH    = 8*CELL_RADIUS;
PLATE_YWIDTH    = 8*CELL_RADIUS;
PLATE_SPACING   = 8*CELL_RADIUS ;
PLATE_THICKNESS = PLATE_YWIDTH/8;

phv = sqrt(1 / 80) ;



! ============================================================
! ... computed coordinates
```

```
XBGN_PLATE = 0.0_MM ; ! arbitrary
YBGN_PLATE = 0.0_MM ; ! arbitrary

XEND_PLATE = XBGN_PLATE + PLATE_XWIDTH ;
YEND_PLATE = YBGN_PLATE + PLATE_YWIDTH ;

ZBGN_PLATELO = 0.0_MM ; ! arbitrary
ZEND_PLATELO = ZBGN_PLATELO + PLATE_THICKNESS ;

ZBGN_PLATEUP = ZEND_PLATELO + PLATE_SPACING ;
ZEND_PLATEUP = ZBGN_PLATEUP + PLATE_THICKNESS ;

XCENTER = XBGN_PLATE + (PLATE_XWIDTH / 2.) ;
YCENTER = YBGN_PLATE + (PLATE_YWIDTH / 2.) ;
ZCENTER = ZEND_PLATELO + (PLATE_SPACING / 2.) ;

CYT_RADIUS = CELL_RADIUS - BILAYER ;

! ============================================================
! ... grid sizes

DX   = 0.2*PLATE_XWIDTH ;
DY   = 0.2*PLATE_YWIDTH ;
DZ   = 0.2*PLATE_THICKNESS ;
DCEN = 0.25*BILAYER ;
DBIG = 8*DCEN ;
```

```
! ============================================================
! ... geometry objects

SYSTEM CARTESIAN ;

VOLUME LO_PLATE CONFORMAL XBGN_PLATE,YBGN_PLATE,ZBGN_PLATELO
                          XEND_PLATE,YEND_PLATE,ZEND_PLATELO ;

VOLUME UP_PLATE CONFORMAL XBGN_PLATE,YBGN_PLATE,ZBGN_PLATEUP
                          XEND_PLATE,YEND_PLATE,ZEND_PLATEUP ;

POINT CELLCENTER XCENTER YCENTER ZCENTER ;

VOLUME CYTOSOL SPHERICAL XCENTER, YCENTER, ZCENTER, CYT_RADIUS ;

VOLUME OUTER_MEM SPHERICAL XCENTER, YCENTER, ZCENTER, CELL_RADIUS ;

VOLUME CHAMBER CONFORMAL XBGN_PLATE, YBGN_PLATE, ZEND_PLATELO
                         XEND_PLATE, YEND_PLATE, ZBGN_PLATEUP;

AREA FRONTWALL CONFORMAL XBGN_PLATE, YBGN_PLATE, ZEND_PLATELO
                         XEND_PLATE, YBGN_PLATE, ZBGN_PLATEUP;

AREA RIGHTWALL CONFORMAL XEND_PLATE, YBGN_PLATE, ZEND_PLATELO
                         XEND_PLATE, YEND_PLATE, ZBGN_PLATEUP;
```

```
AREA BACKWALL CONFORMAL XEND_PLATE, YEND_PLATE, ZEND_PLATELO
                        XBGN_PLATE, YEND_PLATE, ZBGN_PLATEUP;

AREA LEFTWALL CONFORMAL XBGN_PLATE, YEND_PLATE, ZEND_PLATELO
                        XBGN_PLATE, YBGN_PLATE, ZBGN_PLATEUP;

! ============================================================
! ... cross section area to see efield

XZXBGN = XBGN_PLATE;
XZXEND = XEND_PLATE;
XZYBGN = YBGN_PLATE + 0.5*PLATE_YWIDTH;
XZYEND = YBGN_PLATE + 0.5*PLATE_YWIDTH;
XZZBGN = ZEND_PLATELO;
XZZEND = ZBGN_PLATEUP;

AREA XZCUT CONFORMAL XZXBGN, XZYBGN, XZZBGN
                     XZXEND, XZYEND, XZZEND;

YZXBGN = XBGN_PLATE + 0.5*PLATE_XWIDTH;
YZXEND = XBGN_PLATE + 0.5*PLATE_XWIDTH;
YZYBGN = YBGN_PLATE;
YZYEND = YEND_PLATE;
YZZBGN = ZEND_PLATELO;
YZZEND = ZBGN_PLATEUP;

AREA YZCUT CONFORMAL YZXBGN, YZYBGN, YZZBGN
```

```
                    YZXEND, YZYEND, YZZEND;

XYXBGN = XBGN_PLATE;
XYYBGN = YBGN_PLATE;
XYZBGN = ZEND_PLATELO + 0.5*PLATE_SPACING;
XYXEND = XEND_PLATE;
XYYEND = YEND_PLATE;
XYZEND = ZEND_PLATELO + 0.5*PLATE_SPACING;

AREA XYCUT CONFORMAL XYXBGN, XYYBGN, XYZBGN
                    XYXEND, XYYEND, XYZEND;

LINE XLINE CONFORMAL XZXBGN, XZYBGN, XYZBGN
                    XZXEND, XZYEND, XYZEND;

LINE YLINE CONFORMAL YZXBGN, YZYBGN, XYZBGN
                    YZXEND, YZYEND, XYZEND;

LINE ZLINE CONFORMAL YZXBGN, XZYBGN, XZZBGN
                    YZXEND, XZYEND, XZZEND;

! ============================================================
! ... gridding

MARK LO_PLATE X1 SIZE DX ;
MARK LO_PLATE X2 SIZE DY ;
MARK LO_PLATE X3 SIZE DZ ;
```

```
MARK UP_PLATE X1 SIZE DX ;
MARK UP_PLATE X2 SIZE DY ;
MARK UP_PLATE X3 SIZE DZ ;

MARK CELLCENTER SIZE DBIG ;
MARK CYTOSOL X1 SIZE DCEN ;
MARK CYTOSOL X2 SIZE DCEN ;
MARK CYTOSOL X3 SIZE DCEN ;

MARK OUTER_MEM X1 SIZE DCEN ;
MARK OUTER_MEM X2 SIZE DCEN ;
MARK OUTER_MEM X3 SIZE DCEN ;

AUTOGRID ;

! ============================================================
! ... construction

CONDUCTOR LO_PLATE ;
CONDUCTOR UP_PLATE ;

DIELECTRIC CHAMBER 80. ;
DIELECTRIC OUTER_MEM 4.958 ;
DIELECTRIC CYTOSOL 80. ;

CONDUCTANCE CHAMBER 0.30 ;
```

```
CONDUCTANCE OUTER_MEM 3.0E-7 ;
CONDUCTANCE CYTOSOL 0.30 ;

! ============================================================
! ... diagnostic to see cross section of cell

XBGNCUBE = XBGN_PLATE;
YBGNCUBE = YBGN_PLATE;
ZBGNCUBE = ZEND_PLATELO;
XENDCUBE = XEND_PLATE * 0.5;
YENDCUBE = YEND_PLATE;
ZENDCUBE = ZBGN_PLATEUP;

VOLUME CHECKER CONFORMAL XBGNCUBE, YBGNCUBE, ZBGNCUBE
                         XENDCUBE, YENDCUBE, ZENDCUBE ;

!VOID CHECKER ;

! ============================================================
! ... ports

! positive voltage on front wall
Vmax = +150kilovolts;
Trise = 100. femtosecond;
RUNTIME = 45._PICOSECOND ;

xfull = xend_plate - xbgn_plate;
```

```
xhalf = xbgn_plate + 0.5*xfull;

function dc(t) = Vmax * smooth_Ramp(t/Trise);
function gx(x,y,z) = 0;
function gz(x,y,z) = (1-((x-xhalf)/xhalf)**2)**6;
Z PORT FRONTWALL POSITIVE PHASE_VELOCITY phv
    incoming dc(t) function e3 gz e1 gx;
PORT FRONTWALL POSITIVE
    incoming dc(t) function e3 gz e1 gx;
PORT RIGHTWALL NEGATIVE; ! PHASE_VELOCITY phv;
PORT BACKWALL NEGATIVE; ! PHASE_VELOCITY phv;
PORT LEFTWALL POSITIVE; ! PHASE_VELOCITY phv;


! ============================================================
! ... timing

DURATION RUNTIME ;
time_interval = 50;

TIMER SNAPSHOT PERIODIC integer 150,78000,150 INTEGRATE time_interval ;

GRAPHICS PAUSEOFF TSYS$FIRST ;
GRAPHICS PAUSEON  TSYS$LAST ;


!VECTOR FIELD e2,e3 CHKAREA SNAPSHOT DENSITY 20 30 nodump;
```

```
!vector field e1, e2 xysctn snapshot density 20 30 nodump;

!CONTOUR FIELD e3 xyCUT SNAPSHOT  shade movie display conductor nodisplay dielectr
!CONTOUR FIELD e3 yzCUT SNAPSHOT  shade movie display conductor nodisplay dielectr
!CONTOUR FIELD e3 xzCUT SNAPSHOT  shade movie display conductor nodisplay dielectr

!CONTOUR FIELD e2 xyCUT SNAPSHOT  shade movie display conductor nodisplay dielectr
!CONTOUR FIELD e2 yzCUT SNAPSHOT  shade movie display conductor nodisplay dielectr
!CONTOUR FIELD e2 xzCUT SNAPSHOT  shade movie display conductor nodisplay dielectr

!CONTOUR FIELD e1 xyCUT SNAPSHOT  shade movie display conductor nodisplay dielectr
!CONTOUR FIELD e1 yzCUT SNAPSHOT  shade movie display conductor nodisplay dielectr
!CONTOUR FIELD e1 xzCUT SNAPSHOT  shade movie display conductor nodisplay dielectr

RANGE FIELD e3 xLINE SNAPSHOT movie;
RANGE FIELD e3 yLINE SNAPSHOT movie;
RANGE FIELD e3 zLINE SNAPSHOT movie;

RANGE FIELD e2 xLINE SNAPSHOT movie;
RANGE FIELD e2 yLINE SNAPSHOT movie;
RANGE FIELD e2 zLINE SNAPSHOT movie;

RANGE FIELD e1 xLINE SNAPSHOT movie;
RANGE FIELD e1 yLINE SNAPSHOT movie;
RANGE FIELD e1 zLINE SNAPSHOT movie;

!OBSERVE FIELD_INTEGRAL E.DL xLINE SUFFIX XPOTENTIAL;
```

```
!OBSERVE FIELD_INTEGRAL E.DL yLINE SUFFIX YPOTENTIAL;
!OBSERVE FIELD_INTEGRAL E.DL zLINE SUFFIX ZPOTENTIAL;

!CONTOUR FIELD e2 CHKAREA SNAPSHOT AXIS Z -50E+3 0 5 shade nodump;
!CONTOUR FIELD e1 CHKAREA SNAPSHOT AXIS Z -50E+3 0 5 shade nodump;


! ============================================================
! ... look at geometry


!DISPLAY_3D OSYS$MIDPLANE1 OBJECTS ;
!DISPLAY_3D OSYS$MIDPLANE1 OBJECTS ;
!DISPLAY_3D OSYS$MIDPLANE2 OBJECTS ;
!DISPLAY_3D OSYS$MIDPLANE3 OBJECTS ;
!DISPLAY_3D OSYS$MIDPLANE3 OBJECTS SPATIAL_GRID ;
!VIEW_3D ;

! ============================================================
START ;
STOP ;
! ============================================================
```

7 Appendix B

7.1 Gaussian Input

```
%chk=dopcespot
#P RHF/6-31G(d) Test

coordinates for DOPC molecule from Feller for input to Gaussian

0 1

C 25.238 -18.36 17.677
O 24.822 -17.975 16.59
C 26.74 -18.416 18.137
H 26.959 -19.23 18.862
H 26.983 -17.433 18.594
C 27.714 -18.558 16.924
H 28.671 -18.788 17.441
H 27.868 -17.514 16.578
C 27.441 -19.714 15.932
H 27.886 -19.527 14.932
H 26.361 -19.868 15.722
C 27.953 -21.085 16.512
H 27.341 -21.421 17.377
H 29.005 -20.927 16.83
C 27.961 -22.204 15.44
H 28.391 -21.823 14.489
H 26.948 -22.606 15.225
```

C 28.768 -23.469 15.87

H 28.269 -23.889 16.769

H 29.788 -23.081 16.074

C 28.823 -24.604 14.83

H 29.259 -24.29 13.857

H 27.778 -24.912 14.613

C 29.486 -25.874 15.435

H 28.879 -26.26 16.268

C 30.566 -26.522 15.0

H 30.742 -27.509 15.453

C 31.303 -26.299 13.745

H 32.256 -26.859 13.854

H 31.599 -25.236 13.618

C 30.517 -26.776 12.513

H 29.586 -26.175 12.434

H 30.148 -27.818 12.625

C 31.231 -26.581 11.167

H 31.89 -27.463 11.022

H 31.759 -25.605 11.103

C 30.29 -26.638 9.934

H 31.003 -26.524 9.09

H 29.662 -25.725 10.016

C 29.428 -27.918 9.724

H 28.583 -27.952 10.444

H 30.03 -28.846 9.818

C 29.072 -28.003 8.247

H 30.018 -28.0 7.664

H 28.481 -27.104 7.969

C 28.182 -29.236 7.901

H 27.188 -29.16 8.392

H 28.533 -30.212 8.299

C 27.938 -29.399 6.406

H 28.885 -29.636 5.876

H 27.39 -28.573 5.903

H 27.369 -30.349 6.322

N 25.292 -17.626 22.886

C 25.325 -18.521 24.083

C 24.234 -16.64 23.178

C 25.186 -18.332 21.604

C 26.657 -16.932 22.799

H 25.417 -17.874 24.944

H 26.152 -19.195 24.247

H 24.189 -15.812 22.485

H 24.538 -16.142 24.087

H 23.268 -17.12 23.227

H 24.46 -19.131 21.566

H 26.141 -18.829 21.519

H 25.084 -17.611 20.806

H 27.372 -17.548 22.274

H 26.99 -16.562 23.758

H 26.607 -16.022 22.219

C 24.041 -19.441 24.222

H 24.088 -19.854 25.253

H 23.098 -18.859 24.139

P 22.784 -21.242 22.848

O 21.799 -21.266 23.899

O 23.252 -22.495 22.264

O 22.239 -20.244 21.802

O 24.085 -20.459 23.348

C 22.592 -20.254 20.418

H 22.113 -21.072 19.837

H 23.7 -20.328 20.407

C 22.224 -18.936 19.664

H 22.646 -18.112 20.278

O 20.812 -18.719 19.51

C 23.024 -18.904 18.361

H 22.691 -18.019 17.778

H 22.717 -19.754 17.715

O 24.45 -18.949 18.579

C 20.132 -19.511 18.765

O 20.523 -20.466 18.101

C 18.643 -19.206 18.887

H 18.145 -19.864 19.63

H 18.549 -18.143 19.197

C 18.121 -19.501 17.454

H 18.403 -20.508 17.079

H 18.358 -18.707 16.714

C 16.611 -19.456 17.477

H 16.337 -18.521 18.012

H 16.263 -20.16 18.263

C 16.106 -19.898 16.106

H 15.099 -20.366 16.112

H 16.786 -20.685 15.716

C 15.963 -18.753 15.087

H 16.713 -17.947 15.231

H 14.947 -18.324 15.222

C 15.947 -19.355 13.653

H 15.278 -20.242 13.656

H 16.993 -19.664 13.441

C 15.527 -18.366 12.512

H 16.095 -17.435 12.722

H 14.458 -18.063 12.52

C 16.026 -18.792 11.177

H 17.068 -19.144 11.146

C 15.343 -18.841 10.018

H 15.803 -19.348 9.157

C 13.906 -18.335 9.79

H 13.516 -17.663 10.583

H 13.116 -19.109 9.897

C 13.692 -17.61 8.463

H 12.655 -17.237 8.605

H 13.614 -18.229 7.544

C 14.57 -16.319 8.26

H 15.653 -16.567 8.267

H 14.29 -15.697 9.137

C 14.084 -15.574 7.007

H 12.988 -15.391 7.027

H 14.407 -16.295 6.226

C 14.886 -14.281 6.943

H 15.974 -14.462 7.072

H 14.6 -13.57 7.747

C 14.67 -13.552 5.617

H 13.593 -13.299 5.506

H 15.008 -14.327 4.896

C 15.461 -12.3 5.46

H 16.522 -12.464 5.744

H 15.057 -11.531 6.153

C 15.478 -11.854 4.005

H 14.421 -11.594 3.784

H 15.771 -12.718 3.372

H 16.164 -11.001 3.816

7.2 Gaussian Output

```
Entering Link 1 = C:\G98W\l1.exe PID=       732.

Copyright (c) 1988,1990,1992,1993,1995,1998 Gaussian, Inc.
                    All Rights Reserved.

This is part of the Gaussian(R) 98 program.  It is based on
the Gaussian 94(TM) system (copyright 1995 Gaussian, Inc.),
the Gaussian 92(TM) system (copyright 1992 Gaussian, Inc.),
the Gaussian 90(TM) system (copyright 1990 Gaussian, Inc.),
the Gaussian 88(TM) system (copyright 1988 Gaussian, Inc.),
the Gaussian 86(TM) system (copyright 1986 Carnegie Mellon
University), and the Gaussian 82(TM) system (copyright 1983
Carnegie Mellon University). Gaussian is a federally registered
trademark of Gaussian, Inc.

This software contains proprietary and confidential information,
including trade secrets, belonging to Gaussian, Inc.

This software is provided under written license and may be
used, copied, transmitted, or stored only in accord with that
written license.

The following legend is applicable only to US Government
contracts under DFARS:

                    RESTRICTED RIGHTS LEGEND
```

Use, duplication or disclosure by the US Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Gaussian, Inc.
Carnegie Office Park, Building 6, Pittsburgh, PA 15106 USA

The following legend is applicable only to US Government contracts under FAR:

RESTRICTED RIGHTS LEGEND

Use, reproduction and disclosure by the US Government is subject to restrictions as set forth in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR 52.227-19.

Gaussian, Inc.
Carnegie Office Park, Building 6, Pittsburgh, PA 15106 USA

Warning -- This program may not be used in any manner that competes with the business of Gaussian, Inc. or will provide assistance to any competitor of Gaussian, Inc. The licensee

of this program is prohibited from giving any competitor of
Gaussian, Inc. access to this program. By using this program,
the user acknowledges that Gaussian, Inc. is engaged in the
business of creating and licensing software in the field of
computational chemistry and represents and warrants to the
licensee that it is not a competitor of Gaussian, Inc. and that
it will not use this program in any manner prohibited above.

Cite this work as:
Gaussian 98, Revision A.9,
M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria,
M. A. Robb, J. R. Cheeseman, V. G. Zakrzewski, J. A. Montgomery, Jr.,
R. E. Stratmann, J. C. Burant, S. Dapprich, J. M. Millam,
A. D. Daniels, K. N. Kudin, M. C. Strain, O. Farkas, J. Tomasi,
V. Barone, M. Cossi, R. Cammi, B. Mennucci, C. Pomelli, C. Adamo,
S. Clifford, J. Ochterski, G. A. Petersson, P. Y. Ayala, Q. Cui,
K. Morokuma, D. K. Malick, A. D. Rabuck, K. Raghavachari,
J. B. Foresman, J. Cioslowski, J. V. Ortiz, A. G. Baboul,
B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi,
R. Gomperts, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham,
C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill,
B. Johnson, W. Chen, M. W. Wong, J. L. Andres, C. Gonzalez,
M. Head-Gordon, E. S. Replogle, and J. A. Pople,
Gaussian, Inc., Pittsburgh PA, 1998.

```
******************************************
Gaussian 98:  x86-Win32-G98RevA.9 19-Apr-2000
                 26-Feb-2001
******************************************
Default route:  MaxDisk=2000MB
--------------------
#T RHF/6-31G(d) Test
--------------------
---------------------------------------
first attempt at DOPC lipid from Feller
---------------------------------------
Symbolic Z-matrix:
Charge =  0 Multiplicity = 1
C                    25.238   -18.36    17.677
O                    24.822   -17.975   16.59
C                    26.74    -18.416   18.137
H                    26.959   -19.23    18.862
H                    26.983   -17.433   18.594
C                    27.714   -18.558   16.924
H                    28.671   -18.788   17.441
H                    27.868   -17.514   16.578
C                    27.441   -19.714   15.932
H                    27.886   -19.527   14.932
H                    26.361   -19.868   15.722
C                    27.953   -21.085   16.512
H                    27.341   -21.421   17.377
H                    29.005   -20.927   16.83
```

C	27.961	-22.204	15.44
H	28.391	-21.823	14.489
H	26.948	-22.606	15.225
C	28.768	-23.469	15.87
H	28.269	-23.889	16.769
H	29.788	-23.081	16.074
C	28.823	-24.604	14.83
H	29.259	-24.29	13.857
H	27.778	-24.912	14.613
C	29.486	-25.874	15.435
H	28.879	-26.26	16.268
C	30.566	-26.522	15.
H	30.742	-27.509	15.453
C	31.303	-26.299	13.745
H	32.256	-26.859	13.854
H	31.599	-25.236	13.618
C	30.517	-26.776	12.513
H	29.586	-26.175	12.434
H	30.148	-27.818	12.625
C	31.231	-26.581	11.167
H	31.89	-27.463	11.022
H	31.759	-25.605	11.103
C	30.29	-26.638	9.934
H	31.003	-26.524	9.09
H	29.662	-25.725	10.016
C	29.428	-27.918	9.724
H	28.583	-27.952	10.444

H	30.03	-28.846	9.818
C	29.072	-28.003	8.247
H	30.018	-28.	7.664
H	28.481	-27.104	7.969
C	28.182	-29.236	7.901
H	27.188	-29.16	8.392
H	28.533	-30.212	8.299
C	27.938	-29.399	6.406
H	28.885	-29.636	5.876
H	27.39	-28.573	5.903
H	27.369	-30.349	6.322
N	25.292	-17.626	22.886
C	25.325	-18.521	24.083
C	24.234	-16.64	23.178
C	25.186	-18.332	21.604
C	26.657	-16.932	22.799
H	25.417	-17.874	24.944
H	26.152	-19.195	24.247
H	24.189	-15.812	22.485
H	24.538	-16.142	24.087
H	23.268	-17.12	23.227
H	24.46	-19.131	21.566
H	26.141	-18.829	21.519
H	25.084	-17.611	20.806
H	27.372	-17.548	22.274
H	26.99	-16.562	23.758
H	26.607	-16.022	22.219

C	24.041	-19.441	24.222
H	24.088	-19.854	25.253
H	23.098	-18.859	24.139
P	22.784	-21.242	22.848
O	21.799	-21.266	23.899
O	23.252	-22.495	22.264
O	22.239	-20.244	21.802
O	24.085	-20.459	23.348
C	22.592	-20.254	20.418
H	22.113	-21.072	19.837
H	23.7	-20.328	20.407
C	22.224	-18.936	19.664
H	22.646	-18.112	20.278
O	20.812	-18.719	19.51
C	23.024	-18.904	18.361
H	22.691	-18.019	17.778
H	22.717	-19.754	17.715
O	24.45	-18.949	18.579
C	20.132	-19.511	18.765
O	20.523	-20.466	18.101
C	18.643	-19.206	18.887
H	18.145	-19.864	19.63
H	18.549	-18.143	19.197
C	18.121	-19.501	17.454
H	18.403	-20.508	17.079
H	18.358	-18.707	16.714
C	16.611	-19.456	17.477

H	16.337	-18.521	18.012
H	16.263	-20.16	18.263
C	16.106	-19.898	16.106
H	15.099	-20.366	16.112
H	16.786	-20.685	15.716
C	15.963	-18.753	15.087
H	16.713	-17.947	15.231
H	14.947	-18.324	15.222
C	15.947	-19.355	13.653
H	15.278	-20.242	13.656
H	16.993	-19.664	13.441
C	15.527	-18.366	12.512
H	16.095	-17.435	12.722
H	14.458	-18.063	12.52
C	16.026	-18.792	11.177
H	17.068	-19.144	11.146
C	15.343	-18.841	10.018
H	15.803	-19.348	9.157
C	13.906	-18.335	9.79
H	13.516	-17.663	10.583
H	13.116	-19.109	9.897
C	13.692	-17.61	8.463
H	12.655	-17.237	8.605
H	13.614	-18.229	7.544
C	14.57	-16.319	8.26
H	15.653	-16.567	8.267
H	14.29	-15.697	9.137

C	14.084	-15.574	7.007
H	12.988	-15.391	7.027
H	14.407	-16.295	6.226
C	14.886	-14.281	6.943
H	15.974	-14.462	7.072
H	14.6	-13.57	7.747
C	14.67	-13.552	5.617
H	13.593	-13.299	5.506
H	15.008	-14.327	4.896
C	15.461	-12.3	5.46
H	16.522	-12.464	5.744
H	15.057	-11.531	6.153
C	15.478	-11.854	4.005
H	14.421	-11.594	3.784
H	15.771	-12.718	3.372
H	16.164	-11.001	3.816

Framework group C1[X(C44H84NO8P)]

Deg. of freedom 408

Standard orientation:

Center Number	Atomic Number	Atomic Type	Coordinates (Angstroms) X	Y	Z
1	6	0	2.136879	-1.907413	2.588809
2	8	0	1.121673	-1.280120	2.869365
3	6	0	3.529243	-1.804551	3.310945
4	1	0	4.397558	-2.002758	2.645419

5	1	0	3.523680	-2.536908	4.146295
6	6	0	3.731401	-0.410806	3.986828
7	1	0	4.811166	-0.452029	4.248452
8	1	0	3.236309	-0.515988	4.975356
9	6	0	3.523433	0.839975	3.099571
10	1	0	3.289387	1.747081	3.695704
11	1	0	2.663983	0.746719	2.401815
12	6	0	4.778509	1.094595	2.184073
13	1	0	4.882028	0.313597	1.399859
14	1	0	5.668254	1.085795	2.848203
15	6	0	4.729045	2.485148	1.501927
16	1	0	4.425725	3.262997	2.235010
17	1	0	4.022937	2.523177	0.645210
18	6	0	6.089561	2.913758	0.868157
19	1	0	6.331273	2.175155	0.074657
20	1	0	6.801485	2.893236	1.719804
21	6	0	6.098375	4.298694	0.193825
22	1	0	5.817098	5.125746	0.881059
23	1	0	5.327203	4.288203	-0.605651
24	6	0	7.437858	4.536047	-0.559796
25	1	0	7.550538	3.805146	-1.374916
26	6	0	8.318323	5.519548	-0.378035
27	1	0	9.092947	5.612640	-1.153693
28	6	0	8.165247	6.718717	0.462491
29	1	0	9.180280	7.163504	0.537073
30	1	0	7.878086	6.464003	1.504783
31	6	0	7.161786	7.722305	-0.128293

32	1	0	6.160936	7.241399	-0.163387
33	1	0	7.377174	7.957958	-1.192504
34	6	0	6.958182	9.003455	0.694346
35	1	0	7.760608	9.709649	0.393294
36	1	0	6.915249	8.804985	1.787151
37	6	0	5.658786	9.779765	0.350935
38	1	0	5.736285	10.669489	1.011304
39	1	0	4.835741	9.123421	0.706585
40	6	0	5.441228	10.236412	-1.122050
41	1	0	5.168392	9.376406	-1.769763
42	1	0	6.337663	10.741779	-1.538502
43	6	0	4.478026	11.414340	-1.107382
44	1	0	4.908325	12.202955	-0.453360
45	1	0	3.511460	11.080879	-0.672274
46	6	0	4.152226	11.957481	-2.532499
47	1	0	3.606652	11.198103	-3.132987
48	1	0	5.028101	12.147164	-3.189038
49	6	0	3.298271	13.219029	-2.512846
50	1	0	3.856966	14.065701	-2.060204
51	1	0	2.284283	13.121069	-2.068075
52	1	0	3.226042	13.519097	-3.579645
53	7	0	4.436844	-6.622691	2.199375
54	6	0	5.430831	-7.244654	1.271972
55	6	0	3.359632	-7.622668	2.327769
56	6	0	4.022484	-5.266351	1.822754
57	6	0	5.138091	-6.419601	3.548240
58	1	0	5.651843	-8.223045	1.674836

59	1	0	6.425420	-6.837082	1.173078
60	1	0	2.633452	-7.403255	3.097431
61	1	0	3.828488	-8.511808	2.723125
62	1	0	2.852455	-7.760499	1.384514
63	1	0	3.789739	-5.118941	0.778234
64	1	0	4.916706	-4.683355	1.986284
65	1	0	3.246432	-4.929896	2.494758
66	1	0	5.683166	-5.487443	3.564996
67	1	0	5.707784	-7.287320	3.848211
68	1	0	4.429170	-6.332680	4.358718
69	6	0	4.912380	-7.369846	-0.221317
70	1	0	5.631299	-8.052894	-0.723653
71	1	0	3.908470	-7.842976	-0.277652
72	15	0	4.040740	-5.826355	-2.110604
73	8	0	3.820513	-7.015101	-2.894049
74	8	0	4.638329	-4.648712	-2.732000
75	8	0	2.688463	-5.536429	-1.421811
76	8	0	4.945233	-6.179204	-0.840533
77	6	0	2.278383	-4.248062	-0.961257
78	1	0	1.973605	-3.554838	-1.775306
79	1	0	3.146530	-3.861056	-0.386968
80	6	0	1.069361	-4.287455	0.027570
81	1	0	1.344549	-5.029036	0.807560
82	8	0	-0.171739	-4.690640	-0.573812
83	6	0	1.021273	-2.946169	0.760696
84	1	0	0.106314	-2.934383	1.390548
85	1	0	0.828718	-2.130408	0.031641

86	8	0	2.231847	-2.666494	1.495079
87	6	0	-0.721201	-3.932213	-1.449922
88	8	0	-0.347626	-2.838280	-1.861692
89	6	0	-1.921914	-4.631192	-2.078210
90	1	0	-1.655740	-5.135328	-3.031109
91	1	0	-2.289996	-5.378342	-1.342556
92	6	0	-2.900796	-3.452561	-2.334366
93	1	0	-2.445543	-2.615533	-2.905616
94	1	0	-3.421410	-3.090671	-1.422114
95	6	0	-4.056353	-3.962874	-3.163174
96	1	0	-4.396157	-4.902299	-2.675794
97	1	0	-3.635490	-4.436954	-4.075664
98	6	0	-4.929566	-2.768617	-3.539045
99	1	0	-5.494195	-2.887916	-4.487766
100	1	0	-4.272195	-1.888718	-3.704908
101	6	0	-6.025098	-2.438955	-2.509012
102	1	0	-5.724785	-2.671771	-1.465709
103	1	0	-6.912272	-3.054311	-2.771275
104	6	0	-6.489995	-0.970092	-2.721912
105	1	0	-6.621972	-0.802000	-3.812172
106	1	0	-5.668742	-0.331409	-2.331824
107	6	0	-7.790302	-0.556704	-1.950734
108	1	0	-7.648649	-0.955822	-0.924029
109	1	0	-8.727123	-1.026978	-2.319302
110	6	0	-7.889304	0.913706	-1.748761
111	1	0	-6.963630	1.416024	-1.430273
112	6	0	-8.959457	1.704399	-1.953087

113	1	0	-8.820054	2.795264	-1.929623
114	6	0	-10.378277	1.232504	-2.323564
115	1	0	-10.567785	0.150569	-2.162207
116	1	0	-10.598725	1.226248	-3.412595
117	6	0	-11.502088	1.980948	-1.609959
118	1	0	-12.378091	1.377061	-1.930272
119	1	0	-11.752906	3.000923	-1.971274
120	6	0	-11.480730	1.874881	-0.039269
121	1	0	-10.549126	2.313270	0.378309
122	1	0	-11.524041	0.777447	0.128590
123	6	0	-12.783266	2.467062	0.521043
124	1	0	-13.683947	2.029527	0.038920
125	1	0	-12.618050	3.538863	0.280054
126	6	0	-12.751967	2.218511	2.023172
127	1	0	-11.784719	2.526315	2.473544
128	1	0	-12.873483	1.143035	2.272810
129	6	0	-13.878496	2.963770	2.738639
130	1	0	-14.858951	2.613324	2.348569
131	1	0	-13.649446	4.012474	2.451468
132	6	0	-13.884091	2.808207	4.219721
133	1	0	-12.868105	2.968841	4.638340
134	1	0	-14.174587	1.765209	4.469907
135	6	0	-14.777449	3.856740	4.866808
136	1	0	-15.799918	3.602841	4.514978
137	1	0	-14.501565	4.854836	4.465902
138	1	0	-14.709915	3.866263	5.975534

```
Rotational constants (GHZ):      0.0184286      0.0131872      0.0086962
Isotopes: C-12,O-16,C-12,H-1,H-1,C-12,H-1,H-1,C-12,H-1,H-1,C-12,H-1,H-1,C-12,H-1
,H-1,C-12,H-1,H-1,C-12,H-1,H-1,C-12,H-1,C-12,H-1,C-12,H-1,H-1,C-12,H-1,H-1,C-12,
H-1,H-1,C-12,H-1,H-1,C-12,H-1,H-1,C-12,H-1,H-1,C-12,H-1,H-1,C-12,H-1,H-1,H-1,N-1
4,C-12,C-12,C-12,C-12,H-1,H-1,H-1,H-1,H-1,H-1,H-1,H-1,H-1,H-1,H-1,C-12,H-1,H-1,P
-31,O-16,O-16,O-16,O-16,C-12,H-1,H-1,C-12,H-1,O-16,C-12,H-1,H-1,O-16,C-12,O-16,C
-12,H-1,H-1,C-12,H-1,H-1,C-12,H-1,H-1,C-12,H-1,H-1,C-12,H-1,H-1,C-12,H-1,H-1,C-1
2,H-1,H-1,C-12,H-1,C-12,H-1,C-12,H-1,H-1,C-12,H-1,H-1,C-12,H-1,H-1,C-12,H-1,H-1,
C-12,H-1,H-1,C-12,H-1,H-1,C-12,H-1,H-1,C-12,H-1,H-1,H-1
  982 basis functions     1872 primitive gaussians
  217 alpha electrons      217 beta electrons
      nuclear repulsion energy      6946.4399702385 Hartrees.
Projected CNDO Guess.
Warning!  Cutoffs for single-point calculations used.
SCF Done:  E(RHF) =  -2709.00186368     A.U. after    8 cycles
            Convg  =    0.2798D-05             -V/T =  2.0019
            S**2   =   0.0000

**********************************************************************

           Population analysis using the SCF density.

**********************************************************************

Alpha  occ. eigenvalues --  -79.97134 -20.64119 -20.63708 -20.57202 -20.54672
Alpha  occ. eigenvalues --  -20.54353 -20.53669 -20.39914 -20.39390 -15.77559
Alpha  occ. eigenvalues --  -11.41148 -11.38230 -11.36232 -11.35018 -11.33622
```

```
Alpha  occ. eigenvalues --  -11.32818 -11.32393 -11.31306 -11.30919 -11.28492
Alpha  occ. eigenvalues --  -11.28040 -11.25357 -11.23817 -11.23787 -11.23489
Alpha  occ. eigenvalues --  -11.22764 -11.22500 -11.22350 -11.22329 -11.22327
Alpha  occ. eigenvalues --  -11.22191 -11.22168 -11.22087 -11.22030 -11.21974
Alpha  occ. eigenvalues --  -11.21943 -11.21938 -11.21900 -11.21868 -11.21868
Alpha  occ. eigenvalues --  -11.21807 -11.21758 -11.21733 -11.21731 -11.21682
Alpha  occ. eigenvalues --  -11.21601 -11.21522 -11.21460 -11.21413 -11.21368
Alpha  occ. eigenvalues --  -11.21304 -11.21291 -11.21103 -11.20994  -7.50061
Alpha  occ. eigenvalues --   -5.39936  -5.39864  -5.39764  -1.49479  -1.48295
Alpha  occ. eigenvalues --   -1.42120  -1.40778  -1.36975  -1.35845  -1.35415
Alpha  occ. eigenvalues --   -1.22220  -1.16959  -1.10440  -1.09754  -1.09686
Alpha  occ. eigenvalues --   -1.09462  -1.09310  -1.08689  -1.07869  -1.07659
Alpha  occ. eigenvalues --   -1.07353  -1.06660  -1.06152  -1.05759  -1.03948
Alpha  occ. eigenvalues --   -1.03611  -1.01476  -1.00996  -0.98863  -0.98033
Alpha  occ. eigenvalues --   -0.96410  -0.95612  -0.94842  -0.94179  -0.92279
Alpha  occ. eigenvalues --   -0.91785  -0.89213  -0.88421  -0.87453  -0.86464
Alpha  occ. eigenvalues --   -0.85335  -0.83912  -0.82572  -0.81231  -0.80280
Alpha  occ. eigenvalues --   -0.79291  -0.78998  -0.78632  -0.78184  -0.78066
Alpha  occ. eigenvalues --   -0.77660  -0.77556  -0.77494  -0.76949  -0.76252
Alpha  occ. eigenvalues --   -0.75727  -0.75619  -0.74638  -0.72245  -0.71276
Alpha  occ. eigenvalues --   -0.71062  -0.70914  -0.70115  -0.69191  -0.68557
Alpha  occ. eigenvalues --   -0.67840  -0.67644  -0.66732  -0.66258  -0.66022
Alpha  occ. eigenvalues --   -0.65924  -0.65353  -0.64758  -0.64628  -0.64275
Alpha  occ. eigenvalues --   -0.64178  -0.63795  -0.62959  -0.62706  -0.62531
Alpha  occ. eigenvalues --   -0.62078  -0.61652  -0.61627  -0.61338  -0.61246
Alpha  occ. eigenvalues --   -0.60567  -0.59849  -0.59550  -0.59436  -0.59204
Alpha  occ. eigenvalues --   -0.58742  -0.58599  -0.58381  -0.57924  -0.57571
```

```
Alpha  occ. eigenvalues --   -0.57273  -0.56977  -0.56754  -0.56272  -0.56086
Alpha  occ. eigenvalues --   -0.55897  -0.55509  -0.55192  -0.54583  -0.54203
Alpha  occ. eigenvalues --   -0.53808  -0.53577  -0.53351  -0.53278  -0.52936
Alpha  occ. eigenvalues --   -0.52464  -0.52138  -0.52019  -0.51623  -0.51145
Alpha  occ. eigenvalues --   -0.50812  -0.50778  -0.50353  -0.49571  -0.49427
Alpha  occ. eigenvalues --   -0.49403  -0.49222  -0.48883  -0.48827  -0.48710
Alpha  occ. eigenvalues --   -0.48485  -0.47780  -0.47642  -0.47499  -0.47357
Alpha  occ. eigenvalues --   -0.47230  -0.47171  -0.47120  -0.46782  -0.46522
Alpha  occ. eigenvalues --   -0.46306  -0.46192  -0.45781  -0.45509  -0.45466
Alpha  occ. eigenvalues --   -0.45435  -0.45288  -0.45149  -0.44491  -0.44409
Alpha  occ. eigenvalues --   -0.44322  -0.44230  -0.43905  -0.43735  -0.43200
Alpha  occ. eigenvalues --   -0.43024  -0.42707  -0.42218  -0.41928  -0.41734
Alpha  occ. eigenvalues --   -0.40968  -0.39193  -0.37024  -0.34606  -0.33373
Alpha  occ. eigenvalues --   -0.32644  -0.32575
Alpha virt. eigenvalues --    0.12390   0.15424   0.17955   0.18247   0.18336
Alpha virt. eigenvalues --    0.18367   0.18575   0.18950   0.19599   0.20414
Alpha virt. eigenvalues --    0.20808   0.22260   0.22337   0.22619   0.22673
Alpha virt. eigenvalues --    0.22878   0.22950   0.23545   0.23564   0.23835
Alpha virt. eigenvalues --    0.24200   0.24262   0.24667   0.24752   0.24806
Alpha virt. eigenvalues --    0.25131   0.25567   0.25736   0.25814   0.25880
Alpha virt. eigenvalues --    0.26107   0.26289   0.26360   0.26729   0.26962
Alpha virt. eigenvalues --    0.27250   0.27554   0.27844   0.27925   0.28235
Alpha virt. eigenvalues --    0.28440   0.28786   0.29173   0.29199   0.29263
Alpha virt. eigenvalues --    0.29484   0.29579   0.29855   0.30000   0.30463
Alpha virt. eigenvalues --    0.30671   0.30972   0.31166   0.31563   0.31660
Alpha virt. eigenvalues --    0.31792   0.32112   0.32271   0.32423   0.32539
Alpha virt. eigenvalues --    0.32744   0.32923   0.33081   0.33225   0.33381
```

```
Alpha virt. eigenvalues --    0.33611   0.33647   0.33889   0.33972   0.34296
Alpha virt. eigenvalues --    0.34367   0.34405   0.34476   0.34625   0.34679
Alpha virt. eigenvalues --    0.35051   0.35254   0.35367   0.35454   0.35468
Alpha virt. eigenvalues --    0.35512   0.35924   0.36007   0.36141   0.36280
Alpha virt. eigenvalues --    0.36403   0.36701   0.37361   0.37402   0.37656
Alpha virt. eigenvalues --    0.37800   0.37961   0.38197   0.38475   0.38802
Alpha virt. eigenvalues --    0.39217   0.39575   0.39704   0.39949   0.40135
Alpha virt. eigenvalues --    0.40273   0.40504   0.40673   0.40788   0.40999
Alpha virt. eigenvalues --    0.41199   0.41858   0.41968   0.42457   0.42613
Alpha virt. eigenvalues --    0.43097   0.43525   0.43725   0.43796   0.44199
Alpha virt. eigenvalues --    0.44510   0.44626   0.45056   0.45501   0.46175
Alpha virt. eigenvalues --    0.46639   0.46959   0.47398   0.47448   0.48175
Alpha virt. eigenvalues --    0.48368   0.48924   0.49567   0.50034   0.50173
Alpha virt. eigenvalues --    0.50365   0.51662   0.52011   0.53532   0.54747
Alpha virt. eigenvalues --    0.55808   0.56302   0.57060   0.60068   0.60823
Alpha virt. eigenvalues --    0.60948   0.61710   0.61969   0.63405   0.63829
Alpha virt. eigenvalues --    0.65295   0.66477   0.67461   0.67678   0.68896
Alpha virt. eigenvalues --    0.69515   0.70251   0.70286   0.70665   0.70854
Alpha virt. eigenvalues --    0.71197   0.71274   0.71598   0.71921   0.71992
Alpha virt. eigenvalues --    0.72598   0.73144   0.73765   0.74000   0.74375
Alpha virt. eigenvalues --    0.74575   0.74697   0.74963   0.75579   0.75665
Alpha virt. eigenvalues --    0.76421   0.76525   0.76845   0.76931   0.77120
Alpha virt. eigenvalues --    0.77484   0.77793   0.78343   0.78606   0.78788
Alpha virt. eigenvalues --    0.78989   0.79416   0.79882   0.80416   0.80784
Alpha virt. eigenvalues --    0.80893   0.81593   0.82051   0.82661   0.82822
Alpha virt. eigenvalues --    0.83400   0.83744   0.83785   0.83874   0.84075
Alpha virt. eigenvalues --    0.84193   0.84264   0.84936   0.85071   0.85652
```

```
Alpha virt. eigenvalues --    0.85750   0.86374   0.86724   0.87454   0.87685
Alpha virt. eigenvalues --    0.87985   0.88397   0.88732   0.89122   0.89652
Alpha virt. eigenvalues --    0.90150   0.90292   0.90837   0.91249   0.91851
Alpha virt. eigenvalues --    0.92345   0.92831   0.93176   0.93501   0.93611
Alpha virt. eigenvalues --    0.94074   0.94193   0.94938   0.95352   0.95466
Alpha virt. eigenvalues --    0.95625   0.96003   0.96472   0.96803   0.97141
Alpha virt. eigenvalues --    0.97681   0.98044   0.98136   0.98509   0.98920
Alpha virt. eigenvalues --    0.99228   0.99495   0.99994   1.00312   1.00696
Alpha virt. eigenvalues --    1.00981   1.01289   1.01481   1.01843   1.02442
Alpha virt. eigenvalues --    1.02511   1.02927   1.03137   1.03567   1.03623
Alpha virt. eigenvalues --    1.04069   1.04590   1.05211   1.05737   1.05902
Alpha virt. eigenvalues --    1.06357   1.06688   1.06778   1.07013   1.07052
Alpha virt. eigenvalues --    1.07346   1.07511   1.07827   1.08476   1.08824
Alpha virt. eigenvalues --    1.09012   1.09575   1.09845   1.10023   1.10177
Alpha virt. eigenvalues --    1.10289   1.10381   1.10594   1.11061   1.11138
Alpha virt. eigenvalues --    1.11327   1.11368   1.11702   1.12020   1.12199
Alpha virt. eigenvalues --    1.12260   1.12349   1.12698   1.13097   1.13284
Alpha virt. eigenvalues --    1.13467   1.13612   1.13741   1.14111   1.14176
Alpha virt. eigenvalues --    1.14416   1.14521   1.14707   1.14742   1.14818
Alpha virt. eigenvalues --    1.15193   1.15369   1.15418   1.15777   1.15839
Alpha virt. eigenvalues --    1.16015   1.16078   1.16200   1.16399   1.16691
Alpha virt. eigenvalues --    1.16865   1.17072   1.17304   1.17425   1.17575
Alpha virt. eigenvalues --    1.17633   1.17785   1.17889   1.18039   1.18261
Alpha virt. eigenvalues --    1.18534   1.18627   1.18751   1.18901   1.18959
Alpha virt. eigenvalues --    1.19062   1.19094   1.19326   1.19561   1.19642
Alpha virt. eigenvalues --    1.19783   1.20054   1.20193   1.20499   1.20614
Alpha virt. eigenvalues --    1.20850   1.21022   1.21298   1.21327   1.21487
```

```
 Alpha virt. eigenvalues --    1.21625   1.21901   1.21949   1.21973   1.22169
 Alpha virt. eigenvalues --    1.22323   1.22762   1.23071   1.23208   1.23731
 Alpha virt. eigenvalues --    1.23838   1.23889   1.24480   1.24666   1.24898
 Alpha virt. eigenvalues --    1.25152   1.25485   1.25654   1.25847   1.26064
 Alpha virt. eigenvalues --    1.26345   1.26539   1.27068   1.27237   1.27682
 Alpha virt. eigenvalues --    1.28419   1.28743   1.28966   1.28981   1.30341
 Alpha virt. eigenvalues --    1.30791   1.31547   1.31755   1.32004   1.32718
 Alpha virt. eigenvalues --    1.34136   1.35088   1.36204   1.37699   1.38440
 Alpha virt. eigenvalues --    1.40226   1.40485   1.41423   1.41723   1.41973
 Alpha virt. eigenvalues --    1.42506   1.43757   1.44183   1.45580   1.46192
 Alpha virt. eigenvalues --    1.46345   1.46990   1.47556   1.48049   1.48733
 Alpha virt. eigenvalues --    1.48829   1.49268   1.49510   1.50627   1.50681
 Alpha virt. eigenvalues --    1.51677   1.52715   1.54143   1.54488   1.55409
 Alpha virt. eigenvalues --    1.55669   1.56273   1.57198   1.57921   1.58293
 Alpha virt. eigenvalues --    1.59533   1.59983   1.60812   1.62120   1.62599
 Alpha virt. eigenvalues --    1.63009   1.63523   1.63779   1.64484   1.65541
 Alpha virt. eigenvalues --    1.65564   1.65909   1.67730   1.68066   1.68271
 Alpha virt. eigenvalues --    1.68857   1.69161   1.69415   1.69629   1.70268
 Alpha virt. eigenvalues --    1.70942   1.71295   1.72549   1.73396   1.73802
 Alpha virt. eigenvalues --    1.74147   1.75268   1.75363   1.76127   1.76611
 Alpha virt. eigenvalues --    1.77300   1.78107   1.79099   1.79731   1.79846
 Alpha virt. eigenvalues --    1.80304   1.81004   1.81535   1.81915   1.83007
 Alpha virt. eigenvalues --    1.83338   1.83442   1.84008   1.84638   1.85401
 Alpha virt. eigenvalues --    1.86140   1.86681   1.87937   1.88497   1.88724
 Alpha virt. eigenvalues --    1.89419   1.89660   1.91414   1.92290   1.93047
 Alpha virt. eigenvalues --    1.93738   1.93854   1.94564   1.94865   1.95287
 Alpha virt. eigenvalues --    1.95956   1.96696   1.97447   1.97779   1.97876
```

```
Alpha virt. eigenvalues --    1.98188   1.98513   1.99266   2.00224   2.00642
Alpha virt. eigenvalues --    2.02312   2.02949   2.03601   2.04218   2.04693
Alpha virt. eigenvalues --    2.06045   2.06610   2.06656   2.07476   2.08221
Alpha virt. eigenvalues --    2.08722   2.09082   2.09548   2.09894   2.10659
Alpha virt. eigenvalues --    2.10696   2.10854   2.11351   2.12080   2.12092
Alpha virt. eigenvalues --    2.12739   2.12996   2.13777   2.14141   2.14501
Alpha virt. eigenvalues --    2.14887   2.15460   2.16180   2.16662   2.17084
Alpha virt. eigenvalues --    2.17185   2.17468   2.18832   2.19078   2.19519
Alpha virt. eigenvalues --    2.19677   2.20417   2.20984   2.21089   2.21335
Alpha virt. eigenvalues --    2.21767   2.22288   2.22436   2.23002   2.23348
Alpha virt. eigenvalues --    2.23408   2.23820   2.24204   2.24357   2.24802
Alpha virt. eigenvalues --    2.25225   2.25417   2.26055   2.26123   2.26676
Alpha virt. eigenvalues --    2.26928   2.27299   2.27930   2.28145   2.28194
Alpha virt. eigenvalues --    2.28371   2.28577   2.28770   2.29399   2.29438
Alpha virt. eigenvalues --    2.29747   2.30001   2.30184   2.30821   2.31034
Alpha virt. eigenvalues --    2.31094   2.31528   2.32007   2.32051   2.32298
Alpha virt. eigenvalues --    2.32570   2.32983   2.33031   2.33522   2.33803
Alpha virt. eigenvalues --    2.34242   2.34867   2.35046   2.35386   2.36024
Alpha virt. eigenvalues --    2.36353   2.36695   2.36806   2.36890   2.38186
Alpha virt. eigenvalues --    2.38408   2.39396   2.39647   2.39940   2.40277
Alpha virt. eigenvalues --    2.40761   2.40782   2.41341   2.41874   2.42211
Alpha virt. eigenvalues --    2.42749   2.43306   2.43982   2.44413   2.44742
Alpha virt. eigenvalues --    2.45540   2.46876   2.47349   2.48111   2.48537
Alpha virt. eigenvalues --    2.49189   2.49483   2.50180   2.51548   2.51755
Alpha virt. eigenvalues --    2.52543   2.52711   2.53055   2.54540   2.54732
Alpha virt. eigenvalues --    2.55258   2.55389   2.55904   2.56324   2.56522
Alpha virt. eigenvalues --    2.56957   2.57417   2.58180   2.58503   2.58899
```

```
Alpha virt. eigenvalues --    2.59088   2.59317   2.60989   2.61211   2.62013
Alpha virt. eigenvalues --    2.62606   2.63128   2.63388   2.64047   2.64379
Alpha virt. eigenvalues --    2.64884   2.65173   2.65598   2.66831   2.67226
Alpha virt. eigenvalues --    2.68461   2.69428   2.70137   2.70410   2.70760
Alpha virt. eigenvalues --    2.70817   2.71857   2.71955   2.72376   2.73019
Alpha virt. eigenvalues --    2.74219   2.74383   2.75678   2.75932   2.76207
Alpha virt. eigenvalues --    2.76475   2.76765   2.77006   2.78043   2.78538
Alpha virt. eigenvalues --    2.78855   2.79080   2.79691   2.80155   2.80233
Alpha virt. eigenvalues --    2.82251   2.83315   2.83852   2.83940   2.84236
Alpha virt. eigenvalues --    2.85555   2.86551   2.86892   2.87735   2.88283
Alpha virt. eigenvalues --    2.90690   2.90903   2.91129   2.91741   2.92066
Alpha virt. eigenvalues --    2.92904   2.93648   2.93799   2.94188   2.94426
Alpha virt. eigenvalues --    2.95384   2.95813   2.96773   2.97334   2.97547
Alpha virt. eigenvalues --    2.99049   2.99716   3.00406   3.00905   3.01237
Alpha virt. eigenvalues --    3.03160   3.03649   3.05177   3.05549   3.05976
Alpha virt. eigenvalues --    3.06330   3.07405   3.08697   3.08990   3.11146
Alpha virt. eigenvalues --    3.11699   3.12825   3.13092   3.15094   3.15916
Alpha virt. eigenvalues --    3.16582   3.19188   3.19863   3.19998   3.21650
Alpha virt. eigenvalues --    3.23623   3.23701   3.25578   3.25801   3.26090
Alpha virt. eigenvalues --    3.29660   3.30456   3.35602   3.37961   3.38535
Alpha virt. eigenvalues --    3.40363   3.41152   3.41505   3.46029   3.51412
Alpha virt. eigenvalues --    3.57516   3.96342   4.30509   4.35536   4.39735
Alpha virt. eigenvalues --    4.46249   4.47512   4.51597   4.52310   4.52597
Alpha virt. eigenvalues --    4.53656   4.53798   4.54963   4.57327   4.57630
Alpha virt. eigenvalues --    4.58564   4.59582   4.61436   4.61828   4.62290
Alpha virt. eigenvalues --    4.64526   4.65648   4.66554   4.68301   4.69315
Alpha virt. eigenvalues --    4.69861   4.71496   4.72112   4.74633   4.75888
```

```
Alpha virt. eigenvalues --    4.76302   4.76675   4.79706   4.80361   4.80913
Alpha virt. eigenvalues --    4.83057   4.83279   4.84720   4.85467   4.85660
Alpha virt. eigenvalues --    4.86706   4.88648   4.88917   4.90796   4.91558
Alpha virt. eigenvalues --    4.92262   4.94612   4.96365   4.97579   4.99292
Alpha virt. eigenvalues --    5.00027   5.01711   5.02368   5.03856   5.06762
          Condensed to atoms (all electrons):
Total atomic charges:
              1
 1  C    0.795606
 2  O   -0.575591
 3  C   -0.423503
 4  H    0.204097
 5  H    0.173702
 6  C   -0.306196
 7  H    0.156382
 8  H    0.181536
 9  C   -0.320282
10  H    0.173390
11  H    0.190434
12  C   -0.311937
13  H    0.164303
14  H    0.147315
15  C   -0.305749
16  H    0.149108
17  H    0.156283
18  C   -0.295442
19  H    0.157641
```

20	H	0.152617
21	C	-0.347401
22	H	0.165755
23	H	0.162770
24	C	-0.156492
25	H	0.177292
26	C	-0.177689
27	H	0.168304
28	C	-0.337282
29	H	0.158888
30	H	0.162696
31	C	-0.303343
32	H	0.151050
33	H	0.164798
34	C	-0.297023
35	H	0.145954
36	H	0.153646
37	C	-0.310150
38	H	0.152145
39	H	0.148512
40	C	-0.305834
41	H	0.167518
42	H	0.149039
43	C	-0.296087
44	H	0.156048
45	H	0.149771
46	C	-0.305138

```
47  H    0.152243
48  H    0.151862
49  C   -0.472962
50  H    0.151460
51  H    0.160831
52  H    0.151865
53  N   -0.550015
54  C   -0.227890
55  C   -0.349211
56  C   -0.343894
57  C   -0.349186
58  H    0.205592
59  H    0.255486
60  H    0.225753
61  H    0.215807
62  H    0.272472
63  H    0.311016
64  H    0.213365
65  H    0.209791
66  H    0.243108
67  H    0.233572
68  H    0.223514
69  C    0.010142
70  H    0.193792
71  H    0.180248
72  P    1.454594
73  O   -0.773047
```

74	O	-0.775696
75	O	-0.713312
76	O	-0.693999
77	C	-0.013747
78	H	0.245279
79	H	0.163936
80	C	0.130709
81	H	0.159173
82	O	-0.602178
83	C	-0.070913
84	H	0.208703
85	H	0.246538
86	O	-0.636538
87	C	0.819842
88	O	-0.610872
89	C	-0.425123
90	H	0.225800
91	H	0.189666
92	C	-0.322478
93	H	0.214903
94	H	0.155267
95	C	-0.321903
96	H	0.156955
97	H	0.151281
98	C	-0.315604
99	H	0.161980
100	H	0.165729

101	C	-0.302484
102	H	0.154059
103	H	0.150755
104	C	-0.308890
105	H	0.163261
106	H	0.155904
107	C	-0.346976
108	H	0.157566
109	H	0.174326
110	C	-0.157822
111	H	0.174229
112	C	-0.180575
113	H	0.174479
114	C	-0.342743
115	H	0.162243
116	H	0.164472
117	C	-0.316013
118	H	0.142403
119	H	0.153801
120	C	-0.303290
121	H	0.169632
122	H	0.145458
123	C	-0.303386
124	H	0.165980
125	H	0.153585
126	C	-0.301860
127	H	0.153761

```
128  H    0.145563
129  C   -0.288691
130  H    0.147996
131  H    0.145300
132  C   -0.297954
133  H    0.144026
134  H    0.150271
135  C   -0.466178
136  H    0.148483
137  H    0.155062
138  H    0.159078
 Sum of Mulliken charges=   0.00000
 Atomic charges with hydrogens summed into heavy atoms:
              1
  1  C    0.795606
  2  O   -0.575591
  3  C   -0.045704
  4  H    0.000000
  5  H    0.000000
  6  C    0.031723
  7  H    0.000000
  8  H    0.000000
  9  C    0.043542
 10  H    0.000000
 11  H    0.000000
 12  C   -0.000319
 13  H    0.000000
```

14	H	0.000000
15	C	-0.000359
16	H	0.000000
17	H	0.000000
18	C	0.014816
19	H	0.000000
20	H	0.000000
21	C	-0.018876
22	H	0.000000
23	H	0.000000
24	C	0.020801
25	H	0.000000
26	C	-0.009385
27	H	0.000000
28	C	-0.015698
29	H	0.000000
30	H	0.000000
31	C	0.012505
32	H	0.000000
33	H	0.000000
34	C	0.002577
35	H	0.000000
36	H	0.000000
37	C	-0.009493
38	H	0.000000
39	H	0.000000
40	C	0.010723

41	H	0.000000
42	H	0.000000
43	C	0.009732
44	H	0.000000
45	H	0.000000
46	C	-0.001032
47	H	0.000000
48	H	0.000000
49	C	-0.008806
50	H	0.000000
51	H	0.000000
52	H	0.000000
53	N	-0.550015
54	C	0.233188
55	C	0.364821
56	C	0.390279
57	C	0.351008
58	H	0.000000
59	H	0.000000
60	H	0.000000
61	H	0.000000
62	H	0.000000
63	H	0.000000
64	H	0.000000
65	H	0.000000
66	H	0.000000
67	H	0.000000

68	H	0.000000
69	C	0.384182
70	H	0.000000
71	H	0.000000
72	P	1.454594
73	O	-0.773047
74	O	-0.775696
75	O	-0.713312
76	O	-0.693999
77	C	0.395468
78	H	0.000000
79	H	0.000000
80	C	0.289882
81	H	0.000000
82	O	-0.602178
83	C	0.384328
84	H	0.000000
85	H	0.000000
86	O	-0.636538
87	C	0.819842
88	O	-0.610872
89	C	-0.009657
90	H	0.000000
91	H	0.000000
92	C	0.047692
93	H	0.000000
94	H	0.000000

95	C	-0.013667
96	H	0.000000
97	H	0.000000
98	C	0.012105
99	H	0.000000
100	H	0.000000
101	C	0.002331
102	H	0.000000
103	H	0.000000
104	C	0.010275
105	H	0.000000
106	H	0.000000
107	C	-0.015084
108	H	0.000000
109	H	0.000000
110	C	0.016407
111	H	0.000000
112	C	-0.006096
113	H	0.000000
114	C	-0.016028
115	H	0.000000
116	H	0.000000
117	C	-0.019810
118	H	0.000000
119	H	0.000000
120	C	0.011800
121	H	0.000000

```
122  H    0.000000
123  C    0.016179
124  H    0.000000
125  H    0.000000
126  C   -0.002535
127  H    0.000000
128  H    0.000000
129  C    0.004604
130  H    0.000000
131  H    0.000000
132  C   -0.003656
133  H    0.000000
134  H    0.000000
135  C   -0.003554
136  H    0.000000
137  H    0.000000
138  H    0.000000
 Sum of Mulliken charges=   0.00000
 Electronic spatial extent (au):  <R**2>=125889.5060
 Charge=     0.0000 electrons
 Dipole moment (Debye):
    X=     1.5820    Y=    -5.3556    Z=    20.4461  Tot=    21.1950


 Test job not archived.
 1|1|UNPC-UNK|SP|RHF|6-31G(d)|C44H84N1O8P1|PCUSER|26-Feb-2001|0||#T RHF
 /6-31G(D) TEST||first attempt at DOPC lipid from Feller||0,1|C,0,25.23
 8,-18.36,17.677|O,0,24.822,-17.975,16.59|C,0,26.74,-18.416,18.137|H,0,
```

26.959,-19.23,18.862|H,0,26.983,-17.433,18.594|C,0,27.714,-18.558,16.9
24|H,0,28.671,-18.788,17.441|H,0,27.868,-17.514,16.578|C,0,27.441,-19.
714,15.932|H,0,27.886,-19.527,14.932|H,0,26.361,-19.868,15.722|C,0,27.
953,-21.085,16.512|H,0,27.341,-21.421,17.377|H,0,29.005,-20.927,16.83|
C,0,27.961,-22.204,15.44|H,0,28.391,-21.823,14.489|H,0,26.948,-22.606,
15.225|C,0,28.768,-23.469,15.87|H,0,28.269,-23.889,16.769|H,0,29.788,-
23.081,16.074|C,0,28.823,-24.604,14.83|H,0,29.259,-24.29,13.857|H,0,27
.778,-24.912,14.613|C,0,29.486,-25.874,15.435|H,0,28.879,-26.26,16.268
|C,0,30.566,-26.522,15.|H,0,30.742,-27.509,15.453|C,0,31.303,-26.299,1
3.745|H,0,32.256,-26.859,13.854|H,0,31.599,-25.236,13.618|C,0,30.517,-
26.776,12.513|H,0,29.586,-26.175,12.434|H,0,30.148,-27.818,12.625|C,0,
31.231,-26.581,11.167|H,0,31.89,-27.463,11.022|H,0,31.759,-25.605,11.1
03|C,0,30.29,-26.638,9.934|H,0,31.003,-26.524,9.09|H,0,29.662,-25.725,
10.016|C,0,29.428,-27.918,9.724|H,0,28.583,-27.952,10.444|H,0,30.03,-2
8.846,9.818|C,0,29.072,-28.003,8.247|H,0,30.018,-28.,7.664|H,0,28.481,
-27.104,7.969|C,0,28.182,-29.236,7.901|H,0,27.188,-29.16,8.392|H,0,28.
533,-30.212,8.299|C,0,27.938,-29.399,6.406|H,0,28.885,-29.636,5.876|H,
0,27.39,-28.573,5.903|H,0,27.369,-30.349,6.322|N,0,25.292,-17.626,22.8
86|C,0,25.325,-18.521,24.083|C,0,24.234,-16.64,23.178|C,0,25.186,-18.3
32,21.604|C,0,26.657,-16.932,22.799|H,0,25.417,-17.874,24.944|H,0,26.1
52,-19.195,24.247|H,0,24.189,-15.812,22.485|H,0,24.538,-16.142,24.087|
H,0,23.268,-17.12,23.227|H,0,24.46,-19.131,21.566|H,0,26.141,-18.829,2
1.519|H,0,25.084,-17.611,20.806|H,0,27.372,-17.548,22.274|H,0,26.99,-1
6.562,23.758|H,0,26.607,-16.022,22.219|C,0,24.041,-19.441,24.222|H,0,2
4.088,-19.854,25.253|H,0,23.098,-18.859,24.139|P,0,22.784,-21.242,22.8
48|O,0,21.799,-21.266,23.899|O,0,23.252,-22.495,22.264|O,0,22.239,-20.
244,21.802|O,0,24.085,-20.459,23.348|C,0,22.592,-20.254,20.418|H,0,22.

113,-21.072,19.837|H,0,23.7,-20.328,20.407|C,0,22.224,-18.936,19.664|H
,0,22.646,-18.112,20.278|O,0,20.812,-18.719,19.51|C,0,23.024,-18.904,1
8.361|H,0,22.691,-18.019,17.778|H,0,22.717,-19.754,17.715|O,0,24.45,-1
8.949,18.579|C,0,20.132,-19.511,18.765|O,0,20.523,-20.466,18.101|C,0,1
8.643,-19.206,18.887|H,0,18.145,-19.864,19.63|H,0,18.549,-18.143,19.19
7|C,0,18.121,-19.501,17.454|H,0,18.403,-20.508,17.079|H,0,18.358,-18.7
07,16.714|C,0,16.611,-19.456,17.477|H,0,16.337,-18.521,18.012|H,0,16.2
63,-20.16,18.263|C,0,16.106,-19.898,16.106|H,0,15.099,-20.366,16.112|H
,0,16.786,-20.685,15.716|C,0,15.963,-18.753,15.087|H,0,16.713,-17.947,
15.231|H,0,14.947,-18.324,15.222|C,0,15.947,-19.355,13.653|H,0,15.278,
-20.242,13.656|H,0,16.993,-19.664,13.441|C,0,15.527,-18.366,12.512|H,0
,16.095,-17.435,12.722|H,0,14.458,-18.063,12.52|C,0,16.026,-18.792,11.
177|H,0,17.068,-19.144,11.146|C,0,15.343,-18.841,10.018|H,0,15.803,-19
.348,9.157|C,0,13.906,-18.335,9.79|H,0,13.516,-17.663,10.583|H,0,13.11
6,-19.109,9.897|C,0,13.692,-17.61,8.463|H,0,12.655,-17.237,8.605|H,0,1
3.614,-18.229,7.544|C,0,14.57,-16.319,8.26|H,0,15.653,-16.567,8.267|H,
0,14.29,-15.697,9.137|C,0,14.084,-15.574,7.007|H,0,12.988,-15.391,7.02
7|H,0,14.407,-16.295,6.226|C,0,14.886,-14.281,6.943|H,0,15.974,-14.462
,7.072|H,0,14.6,-13.57,7.747|C,0,14.67,-13.552,5.617|H,0,13.593,-13.29
9,5.506|H,0,15.008,-14.327,4.896|C,0,15.461,-12.3,5.46|H,0,16.522,-12.
464,5.744|H,0,15.057,-11.531,6.153|C,0,15.478,-11.854,4.005|H,0,14.421
,-11.594,3.784|H,0,15.771,-12.718,3.372|H,0,16.164,-11.001,3.816||Vers
ion=x86-Win32-G98RevA.9|HF=-2709.0018637|RMSD=2.798e-006|Dipole=4.3981
621,7.0638938,0.5397764|PG=C01 [X(C44H84N1O8P1)]||@

LIFE IS SO UNCERTAIN - EAT DESSERT FIRST.

```
Job cpu time:  0 days  3 hours 34 minutes 38.0 seconds.
File lengths (MBytes):  RWF=  432 Int=    0 D2E=    0 Chk=   39 Scr=    1
Normal termination of Gaussian 98.
```

8 Appendix C

8.1 MD Simulation Code

```
#include <iostream.h>
#include <math.h>
#include <fstream.h>
#include <assert.h>

void InitEM(double long[], double long[], double long[], double long[], ifstream&)
void TimeAdvance(double long[], double long[], double long[], double long[], ofstr
void CrossProd(double long[], double long[], double long[]);
void Rotate(double long[], double long[], double long);

main()
{
  double long p[3], E[3], MofI[3], radius[3]; //dipole moment, electric field,
                                    //moment of inertia and cell radius vector

  ifstream EMinput; //original dipole, moments of inertia and electric field
                    // from Gaussian and MAGIC
  ofstream ThetavTime;  //output file for theta vs time

  InitEM(radius, p, E, MofI, EMinput);  //initialize dipole and E field values

  cout << "initial radius x " << radius[0] << endl;
  cout << "initial radius y " << radius[1] << endl;
  cout << "initial radius z " << radius[2] << endl;
```

```
  cout << "Ex " << E[0] << endl;
  cout << "Ey " << E[1] << endl;
  cout << "Ez " << E[2] << endl;

  cout << "MofI x " << MofI[0] << endl;
  cout << "MofI y " << MofI[1] << endl;
  cout << "MofI z " << MofI[2] << endl;

  TimeAdvance(radius, MofI, p, E, ThetavTime); //do time integration

  cout << "final radius x " << radius[0] << endl;
  cout << "final radius y " << radius[1] << endl;
  cout << "final radius z " << radius[2] << endl;

}

void InitEM(double long radius[], double long p[], double long E[], double long Mo
{
  int i;
  double long align[3]; //angles to align radius and dipole vectors so we model th
  double long Chain1[3], Chain2[3]; //coordinates of last Carbons on fatty chains
  double long BtoI;  //conversion factor for B to I
  double long DebyetoCm; //conversion factor for Debye to Coulombmeter
  BtoI = (6.626076*pow(10,-34))/(8*3.14159*3.14159);
  //BtoI = h/(8*pi^2) with B in Hz and I in kg m^2
  DebyetoCm = 0.3335641*pow(10,-29);
```

```
EMinput.open("eminit.dat");
EMinput >> p[0] >> p[1] >> p[2];
EMinput >> E[0] >> E[1] >> E[2];
EMinput >> MofI[0] >> MofI[1] >> MofI[2];
EMinput >> Chain1[0] >> Chain1[1] >> Chain1[2];
EMinput >> Chain2[0] >> Chain2[1] >> Chain2[2];
EMinput.close();
for (i=0; i<3; i++) {
  MofI[i] = MofI[i]*pow(10,9); //because Gaussian outputs B in GHz
  MofI[i] = BtoI/MofI[i];
  p[i] = p[i]*DebyetoCm;

  cout << "MofI " << MofI[i] << endl;
  cout << "p    " << p[i] << endl;

  radius[i] = (Chain1[i] + Chain2[i])/2;
} //end for i
align[2] = -atan(radius[1]/radius[0]); //angle to rotate about z axes
align[1] = 0; //angle to rotate about y axes
align[0] = 0;  //angle to rotate about x axes
Rotate(radius, align, 1.0);
Rotate(p, align, 1.0);
align[2] = 0; //angle to rotate about z axes
align[1] = -atan(radius[0]/radius[2]); //angle to rotate about y axes
align[0] = 0;  //angle to rotate about x axes
Rotate(radius, align, 1.0);
Rotate(p, align, 1.0);
```

```
}

void TimeAdvance(double long radius[], double long MofI[], double long p[], double
{
  double long eta, deltat, lipidangle;
  double long torq[3], omega[3], theta[3], Fdrag[3];
  int i, timestep;
  double long DebyetoCm;
  deltat = 1*pow(10,-14);     //time step
  eta = 1*pow(10,-3);     //drag force coefficient

  ThetavTime.open("theta.out");

  CrossProd(p, E, torq);  //get torque

  for(i=0; i<3; i++) {
    omega[i]=0;
  } //initially at rest

  theta[0] = atan(p[2]/p[1]);
  theta[1] = atan(p[0]/p[2]);
  theta[2] = atan(p[1]/p[0]);

  cout << "theta x " << theta[0] << endl;
  cout << "theta y " << theta[1] << endl;
  cout << "theta z " << theta[2] << endl;
```

```
timestep=0;

lipidangle = atan(sqrt(radius[0]*radius[0] + radius[1]*radius[1])/radius[2]);
if (lipidangle < 0) {
  lipidangle = 2*3.141592654 + lipidangle;
} //renormalize from atan range of -pi/2 .. pi/2 to 0 .. pi

ThetavTime << timestep*deltat << " " << fabs(lipidangle) << endl;

timestep++;
CrossProd(p, E, torq);

theta[0] = theta[0] + omega[0]*deltat;
theta[1] = theta[1] + omega[1]*deltat;
theta[2] = theta[2] + omega[2]*deltat;
omega[0] = omega[0] + torq[0]*deltat/MofI[0];
omega[1] = omega[1] + torq[1]*deltat/MofI[1];
omega[2] = omega[2] + torq[2]*deltat/MofI[2];

Fdrag[0] = 6*3.141592654*eta*(10*pow(10,-10))*(10*pow(10,-10)*omega[0]);
Fdrag[1] = 6*3.141592654*eta*(10*pow(10,-10))*(10*pow(10,-10)*omega[1]);
Fdrag[2] = 6*3.141592654*eta*(10*pow(10,-10))*(10*pow(10,-10)*omega[2]);
//Stokes drag force of 6*pi*eta*R*v where eta is the absolute viscosity

theta[0] = theta[0] + omega[0]*deltat;
theta[1] = theta[1] + omega[1]*deltat;
theta[2] = theta[2] + omega[2]*deltat;
```

```
omega[0] = omega[0] - Fdrag[0]*(10*pow(10,-10))*deltat/MofI[0];
omega[1] = omega[1] - Fdrag[1]*(10*pow(10,-10))*deltat/MofI[0];
omega[2] = omega[2] - Fdrag[2]*(10*pow(10,-10))*deltat/MofI[0];

Rotate(p, omega, deltat); //rotate dipole vector by omega*deltat
Rotate(radius, omega, deltat); //rotate cell radius vector by omega*deltat

lipidangle = atan(sqrt(radius[0]*radius[0] + radius[1]*radius[1])/radius[2]);
if (lipidangle < 0) {
  lipidangle = 3.141592654 + lipidangle;
} //renormalize from atan range of -pi/2 .. pi/2 to 0 .. pi

while (lipidangle <= 1.65) {
//for(timestep=1; timestep<=pow(10,5); timestep++) {
  timestep++;
  if (timestep%10000000 == 0) cout << "timestep = " << timestep << endl;

  Fdrag[0] = 6*3.141592654*eta*(10*pow(10,-10))*(10*pow(10,-10)*omega[0]);
  Fdrag[1] = 6*3.141592654*eta*(10*pow(10,-10))*(10*pow(10,-10)*omega[1]);
  Fdrag[2] = 6*3.141592654*eta*(10*pow(10,-10))*(10*pow(10,-10)*omega[2]);
  //Stokes drag force of 6*pi*eta*R*v where eta is the absolute viscosity

  theta[0] = theta[0] + omega[0]*deltat;
  theta[1] = theta[1] + omega[1]*deltat;
  theta[2] = theta[2] + omega[2]*deltat;
  omega[0] = omega[0] + torq[0]*deltat/MofI[0] - Fdrag[0]*(10*pow(10,-10))*delta
  omega[1] = omega[1] + torq[1]*deltat/MofI[1] - Fdrag[1]*(10*pow(10,-10))*delta
```

```
    omega[2] = omega[2] + torq[2]*deltat/MofI[2] - Fdrag[2]*(10*pow(10,-10))*delta

    Rotate(p, omega, deltat); //rotate dipole vector by omega*deltat
    Rotate(radius, omega, deltat); //rotate cell radius vector by omega*deltat
    CrossProd(p, E, torq);

    lipidangle = atan(sqrt(radius[0]*radius[0] + radius[1]*radius[1])/radius[2]);
    if (lipidangle < 0) {
      lipidangle = 3.141592654 + lipidangle;
    } //renormalize from atan range of -pi/2 .. pi/2 to 0 .. pi

    if (timestep%100000 == 0) { //don't print every datapoint to save file size
      ThetavTime << timestep*deltat << " " << fabs(lipidangle) << endl;
    } //end if time step

    if (timestep*deltat > 10*pow(10,-3)) {  //pulse the electric field for 10ms
      E[0] = 0;
      E[1] = 0;
      E[2] = 0;
    }//end if pulse field
  } //end while lipidangle or for timestep
  ThetavTime.close();
  cout << "final lipid angle: " << lipidangle << endl;

} //end TimeAdvance

void CrossProd(double long first[], double long second[], double long third[]) {
```

```
  //third = first cross second all in cartesian coordinates
  third[0] = first[1]*second[2] - first[2]*second[1];
  third[1] = -first[0]*second[2] + first[2]*second[0];
  third[2] = first[0]*second[1] - first[1]*second[0];
}

void Rotate(double long vector[], double long omega[], double long deltat) {
  int i;
  double A[3][3]; //product of three rotation matrices
  double theta[3];
  double xtmp, ytmp, ztmp;

  for (i=0; i<3; i++) {
    theta[i] = omega[i] * deltat;
  }

  A[0][0] = cos(theta[1])*cos(theta[2]);
  A[0][1] = -cos(theta[1])*sin(theta[2]);
  A[0][2] = sin(theta[1]);
  A[1][0] = sin(theta[0])*sin(theta[1])*cos(theta[2]) + cos(theta[0])*sin(theta[2]
  A[1][1] = -sin(theta[0])*sin(theta[1])*sin(theta[2]) + cos(theta[0])*cos(theta[2
  A[1][2] = -sin(theta[0])*cos(theta[1]);
  A[2][0] = -cos(theta[0])*sin(theta[1])*cos(theta[2]) + sin(theta[0])*sin(theta[2
  A[2][1] = cos(theta[0])*sin(theta[1])*sin(theta[2]) + sin(theta[0])*cos(theta[2]
  A[2][2] = cos(theta[0])*cos(theta[1]);

  xtmp = A[0][0]*vector[0] + A[0][1]*vector[1] + A[0][2]*vector[2];
```

```
  ytmp = A[1][0]*vector[0] + A[1][1]*vector[1] + A[1][2]*vector[2];
  ztmp = A[2][0]*vector[0] + A[2][1]*vector[1] + A[2][2]*vector[2];

  vector[0] = xtmp;
  vector[1] = ytmp;
  vector[2] = ztmp;
} //end rotate
```

References

[1] G. Johnson *All Science is Computer Science.* New York Times [New York] 2001 Mar 25,

[2] B. Goplen, L. Ludeking, D. Smithe, and G. Warren, "User-Configurable MAGIC code for Electromagnetic PIC Calculations," Computer Physics Communications **87**, 54 (1995).

[3] J.R. Kinosita, M. Hibino, H. Itoh, M. Shigemori, H. Hirano, Y. Kirino, T. Hayakawa, "Events of membrane electroporation visualization on time scale from microsecond to second", in C. Chang, M. Chassy, J. Saunders, A. Sowers (Eds.), *Guide to electroporation and electrofusion* Academic Press, San Diego, CA (1992).

[4] *Gaussian 98* (Revision A.9), M.J. Frisch, G.W. Trucks, H.B. Schlegel, G.E. Scuseria, M.A. Robb, J.R. Cheeseman, V.G. Zakrzewski, J.A. Montgomery, R.E. Stratmann, J.C. Burant, S. Dapprich, J.M. Millam, A.D. Daniels, K.N. Kudin, M.C. Strain, O. Farkas, J. Tomasi, V. Barone, M. Cossi, R. Cammi, B. Mennucci, C. Pomelli, C. Adamo, S. Clifford, J. Ochterski, G.A. Petersson, P.Y. Ayala, Q. Cui, K. Morokuma, D.K. Malick, A.D. Rabuck, K. Raghavachari, J. B. Foresman, J. Cioslowski, J.V. Ortiz, B.B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. Gomperts, R.L. Martin, D.J. Fox, T. Keith, M.A. Al-Laham, C.Y. Peng, A. Nanayakkara, C. Gonzalez, M. Challacombe, P.M.W. Gill, B.G. Johnson, W. Chen, M.W. Wong, J.L. Andres, M. Head-Gordon, E.S. Replogle and J.A. Pople, Gaussian, Inc., Pittsburgh PA, 1998

[5] S.E. Feller, R.W. Pastor, D. Yin, A.D. MacKerell "Molecular Dynamics Simulation of Unsaturated Lipid Bilayer at Low Hydration: Parameterization and Comparison with Diffraction Studies," Biophys. J. **73**, 2269 (1997).

[6] S.E. Feller (2001, February) *DOPC 1500ps coordinate set.* Retrieved February, 2001 from http://persweb.wabash.edu/facstaff/fellers/coordinates

[7] N.J. Giordano, *Computational Physics.* Prentice Hall, Upper Saddle River, NJ (1997).

[8] R.O. Becker and G. Selden, it The body electric: electromagnetism and the foundation of life. Morrow, New York, NY (1985).

[9] C. Tanford, *Physical Chemistry of Macromolecules.* John Wiley and Sons, Inc. New York, NY (1961).

[10] W. Bober and R.A. Kenyon, *Fluid Mechanics.* John Wiley and Sons, Inc. New York, NY (1980).

[11] M. Ciofalo, M.W. Collins and T.R. Hennessy, *Nanoscale Fluid Dynamics in Physiological Processes: A Review Study.* WIT Press, Boston, MA (1999).

[12] T. Kotnik and D. Miklavčič "Analytical Description of Transmembrane Voltage Induced by Electric Field on Spheriodal Cells," Biophys. J. **79**, 670 (2000).

[13] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications.* Academic Press, San Diego, CA (1996).

[14] K.M. Merz and B. Roux, *Biological Membranes: A Molecular Perspective from Computation and Experiment.* Birkhäuser, Boston, MA (1996).

[15] A.K. Rappé and C.J. Casewit *Molecular Mechanics across Chemistry.* University Science Books, Sausalito, CA (1997).

[16] A.R. Leach *Molecular Modeling: Principles and Applications.* Longman, Harlow, Essex, England (1996).

[17] D.W. Scholfield, J.M. Gahl, and N. Shimomura "Effective Electric Field for an Arbitrary Electromagnetic Pulse," IEEE Trans. Plasma Science **27**, 628 (1999).

[18] T. Kotnik, F. Bobanović and D. Miklavčič "Sensitivity of transmembrane voltage induced by applied electric fields–a theoretical analysis," Bioelectrochem. Bioenerg. **43**, 285 (1997).

[19] K.A. DeBruin and W. Krassowska "Modeling Electroporation in a Single Cell. I. Effects of Field Strength and Rest Potential," Biophys. J. **77**, 1213 (1999).

[20] K.A. DeBruin and W. Krassowska "Modeling Electroporation in a Single Cell. II. Effects of Ionic Concentrations," Biophys. J. **77**, 1225 (1999).

[21] T. Kotnik, D. Miklavčič and T. Slivnik "Time course of transmembrane voltage induced by time-varying electric fields–a method for theoretical analysis and its application," Bioelectrochem. Bioenerg. **45**, 3 (1998).

[22] J.C. Neu and W. Krassowska "Asymptotic model of electroporation," Phys. Rev. E **59**, 3471 (1999).

[23] A. Barnett and J.C. Weaver "Electroporation: a unified, quantitative theory of reversible electrical breakdown and mechanical rupture in artificial planar bilayer membranes," Bioelectrochem. Bioenerg. **25**, 163 (1991).

[24] H. Lodish *et al. Molecular Cell Biology.* W.H. Freeman and Company, New York, NY (2000).

[25] B. Alberts *et al. Essential Cell Biology: An Introduction to the Molecular Biology of the Cell.* Garland Publishing, Inc, New York, NY (1998).

[26] K.H. Schoenbach, R.J. Barker and S. Liu "Nonthermal medical/biological treatments using electromagnetic fields and ionized gases,"

[27] K.H. Schoenbach, F.E. Peterkin, R.W Alden, III and S.J. Beebe "The Effects of Pulsed Electric Fields on Biological Cells: Experiments and Applications," IEEE Trans. Plasma Science **25**, 284 (1997).

[28] K.S. Cole "Electric Impedance of Marine Egg Membranes," Trans. Faraday Soc. **33**, 966 (1937).