BIOMEMBRANE MODELING: MOLECULAR DYNAMICS
SIMULATION OF PHOSPHOLIPID MONOLAYERS

by

Terence Randall Thompson

Submitted in Partial Fulfillment
of the
Requirements for the Degree

DOCTOR OF PHILOSOPHY

Supervised by

Dr. David A. Goldstein

Department of Radiation Biology and Biophysics
School of Medicine and Dentistry
University of Rochester
Rochester, New York

1979

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

BIOMEMBRANE MODELING: MOLECULAR DYNAMICS
SIMULATION OF PHOSPHOLIPID MONOLAYERS

by

Terence Randall Thompson

Submitted in Partial Fulfillment
of the
Requirements for the Degree

DOCTOR OF PHILOSOPHY

Supervised by

Dr. David A. Goldstein

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Department of Radiation Biology and Biophysics
School of Medicine and Dentistry
University of Rochester
Rochester, New York

MASTER

1979

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

fly

To Trink

The Heart is the capital of the Mind,
The Mind is a single State.
The Heart and the Mind together make
A single continent.

Emily Dickinson

Vitae

The author was born in 1946 and attended public schools in Warren, Ohio. He received the B.A. degree in philosophy from the University of Michigan (Ann Arbor) in 1968. After military service from 1969 to 1972, he completed requirements for the B.S. degree in physics at the University of California (Santa Barbara) in 1973. He then entered the University of Rochester as a graduate student in the Department of Radiation Biology and Biophysics. His research was supervised by Dr. David A. Goldstein.

Acknowledgments

Dr. David Goldstein has provided the ideal environment for a graduate student: great intellectual freedom and generous moral support.

The members of my thesis committee, Drs. Adil Shamoo, Robert McCrory, and Thomas Gunter, readily have taken time from their own research to serve as advisors and critics.

Marilyn Campbell and Drs. Irving Spar, Helen Eberle, and James Coleman of the RBB Education Office have given much aid in all matters administrative.

The RBB staff in general, and the computer-center and word-processing personnel in particular, have made the department a pleasant and efficient place in which to work.

Financial support has been provided by the RBB department, the National Institutes of Health (grants GM07088 and GM07230), and the Department of Energy.

Abstract

As a first step toward a computer model of a biomembrane-like bilayer, a dynamic, deterministic model of a phospholipid monolayer has been constructed. The model moves phospholipid-like centers of force according to an integrated law of motion in finite difference form. Forces on each phospholipid analogue are derived from the gradient of the local potential, itself the sum of Coulombic and short-range terms. The Coulombic term is approximated by use of a finite-difference form of Poisson's equation, while the short-range term results from finite-radius, pairwise summation of a Lennard-Jones potential. Boundary potentials are treated in such a way that the model is effectively infinite in extent in the plane of the monolayer. The two-dimensional virial theorem is used to find the surface pressure of the monolayer as a function of molecular area. Pressure-versus-area curves for simulated monolayers are compared to those of real monolayers. Dependence of the simulator's behavior on Lennard-Jones parameters and the specific geometry of the molecular analogue is discussed. Implications for the physical theory of phospholipid monolayers and bilayers are developed.

Table of Contents

	<u>Page</u>
Dedication	i
Vitae	ii
Acknowledgements	iii
Abstract	iv
Table of Contents	v
List of Figures	viii
List of Tables	x
I. BACKGROUND	
A. <u>Phospholipids and Biomembrane Structure</u>	
Models of Biomembrane Structure	2
The Nature of Phospholipid Molecules	6
B. <u>Interactions of Phospholipid Molecules</u>	
Electrostatic Interactions	11
Electrodynamic Interactions	14
Overlap Interactions	20
C. <u>Physical Chemistry of Monolayers</u>	
Structure of the Air-Film-Water System	21
Measurement of Monolayer Properties	28
D. <u>Simulation</u>	
Simulation in General	30
Elements of Molecular-dynamics Simulation	32
Deterministic versus Statistical Simulation	35

Table of Contents (continued)

	<u>Page</u>
II. CONSTRUCTION OF THE MONOLAYER SIMULATOR	
A. <u>The Particle-dynamics Program</u>	
General Plan of the Algorithm	37
The Equations of Motion	40
B. <u>Calculation of Forces</u>	
Construction of the Coulombic Potential Surface	45
Construction of the Non-Coulombic Potential Surface	67
Calculation of Potential Gradients	69
Application of Forces to the Molecular Analogue	70
C. <u>Calculation of Temperature, Surface Pressure, and Molecular Area</u>	
Temperature	71
Surface Pressure and Molecular Area	74
Generation of Pressure-Area Curves	79
III. COMPARISON OF EXPERIMENTAL AND SIMULATION RESULTS	
A. <u>Experimental Pressure-area Curves</u>	83
B. <u>Simulated Pressure-area Curves</u>	86
C. <u>Comparison of Pressure-area Curves</u>	95

Table of Contents (continued)

	<u>Page</u>
IV. CONCLUSIONS	
A. <u>Adequacy of the Model</u>	99
B. <u>Directions for Further Study</u>	101
APPENDIX A: Monolayer-simulation Program	
Main Program (MD1.LU)	104
Subroutines	
INIT	117
INIT32	119
INITSP	120
RANLOC	121
QDIST1	123
LUDEC3	126
FORCE1	130
FORCE2	131
LJFOR2	132
PRESS1	134
PHOTO	135
APPENDIX B: Program to Solve Matrix Equation (LUDEC3)	138
APPENDIX C: Program to Compute Potential Due to an	
Infinite Array of Discrete Charges (PWSUM3)	149
APPENDIX D: Pressure-Density Isotherms for Imperfect	
Gas of Stockmayer Particles	152
BIBLIOGRAPHY	159

List of Figures

<u>Figure</u>	<u>Title</u>	<u>Page</u>
I.1	Schematic diagram of dipalmitoyl phosphatidylcholine	8
I.2	Pressure-area isotherm for myristic acid on 0.01 N HCl at 290 ⁰ K	25
II.1	General flowchart for the particle-dynamics program	39
II.2	Schematic diagram of molecular analogues lying in the x-y plane	46
II.3	Schematic diagram of computational lattice, central array, and image arrays	49
II.4	Diagram of one x-y plane in the computational lattice	56
II.5	Schematic diagram of the z-dependence of the potential due to an infinite sheet of equal numbers of discrete positive and negative charges lying in the x-y plane	59

List of Figures (continued)

II.6	Comparison of actual and approximate potentials in the plane of the charges	66
II.7	Transformation of simulation results from pressure-versus-temperature form to pressure-versus-area form	81
III.1	Experimental pressure-area isotherms	85
III.2	Simulation results: surface pressure as a function of temperature at fixed molecular area	92
III.3	Simulation results: surface pressure as a function of molecular area at fixed temperature	94
III.4	Juxtaposition of experimental and simulation isotherms	97
D.1	Pressure-density isotherms for imperfect gas of Stockmayer particles	156

List of Tables.

<u>Table</u>	<u>Title</u>	<u>Page</u>
II.1	Results of direct-summation calculation of the potential due to an infinite array of 8A x 8A cells each containing one positive and one negative charge	63

SECTION I

BACKGROUND

I. A. Phospholipids and Biomembrane Structure

-Models of Biomembrane Structure

Since the middle of the nineteenth century, when Nageli first suggested that cells may be encapsulated in a membrane impermeable to some substances, much has been learned of the structure of biological membranes.

Plasmalemmal and intracellular membranes from various organs and organisms generally have the following common characteristics:

- (i) they are constructed in large part from lipids and proteins,
- (ii) their thickness is on the order of 50 angstroms,
- (iii) they are more permeable to lipid-soluble compounds than to lipid-insoluble compounds (disregarding metabolically-energized transport).

In addition, electron-microscopic and X-ray diffraction studies of various membranes, including sarcoplasmic reticulum, erythrocytes, myelin, and retinal rods (Lee, 1975; Jain, 1972), have led to a structural model which is very widely accepted among membranologists. In this model, the membrane is visualized as a two-dimensional matrix of lipid molecules in which proteins are embedded.

The matrix is bi-lamellar in form, with the polar ends of the lipids facing outward and the non-polar ends inward. Whether the proteins lie on the surface of the matrix or are deeply embedded in the matrix is not entirely clear, though the latter arrangement may sometimes be favored thermodynamically (Singer and Nicolson, 1972).

Furthermore, it seems likely that some proteins extend from one side of the membrane to the other, particularly in view of the trans-membrane transport of ions known to be mediated by such metabolically-powered, proteinaceous pumps as the $(Ca^{++} + Mg^{++})$ -dependent adenosinetriphosphatase of sarcoplasmic reticulum (Shamoo and MacLennan, 1974). Both proteins and lipids have some mobility in this structure (Cherry, 1976).

There are two energetic factors which favor the stability of this membrane structure: hydrophobic and hydrophilic contributions to the free energy of the system. More free energy is required to insert a non-polar moiety into a polar environment than into a non-polar environment (hydrophobicity) and, similarly, more free energy is required to insert a polar moiety into a non-polar environment than into a polar environment (hydrophilicity). If one were to move a lipid molecule out of the bi-lamellar membrane so that the hydrocarbon tails were partially exposed to the surrounding aqueous

solution, the hydrophobic increase in free energy would be approximately 1.2 kcal/mole for every pair of CH_2 groups exposed. Since transfer of zwitterionic glycine from water to acetone, a non-polar solvent, increases the free energy by 6.0 kcal/mole (Singer, 1971), the hydrophilic increase in free energy which would result if one moved the polar head of a phospholipid into the non-polar interior of the membrane is estimated to be several kilocalories per mole. $N_A kT$ is about 0.6 kcal at 300° K , so these stabilizing factors are five to ten times larger than thermal energies.

There are several ways to form molecular systems in the laboratory which have structures somewhat similar to the lipid matrix of cellular membranes. One may form multiple bilayers by dispersing phospholipids in water (Tanford, 1973), single bilayers by spreading a phospholipid-decane mixture over a small aperture in a submerged plastic cup and allowing it to thin until it appears black (Jain, 1972), or single monolayers by spreading phospholipids on the surface of a body of water (Rothfield and Fried, 1975). Investigation of these artificial systems provides various types of information on physical processes which may be important in natural membranes. In particular, the monolayer technique allows

one to measure several properties, such as the surface pressure of the film as it is expanded or contracted mechanically, and to derive some knowledge of molecular orientation (Adamson, 1976, p. 101), film-substrate interactions, and intra-film interactions (Gershfeld, 1974).

- The Nature of Phospholipid Molecules

From a biophysicist's point of view, the most important property of phospholipid molecules is that they are amphipathic. In the literal sense, this means that they have "two-sided feelings" or are affected in two contrary ways. In the physical sense, this means that they have both polar and non-polar regions, the first hydrophilic and the second hydrophobic. Figure I.1 is a schematic diagram of a common phospholipid, dipalmitoyl phosphatidylcholine. The polar head group has a dipole moment of approximately 19 debyes in vacuo, the hydrocarbon chains which form the non-polar tail contain 16 carbon atoms and no carbon-carbon double bonds, and the entire molecule has a weight of 733 daltons. The length of the fully-extended molecule is about 34 angstroms, the length of the fully-extended tails is about 22 angstroms, and the minimal cross-section is about 5 angstroms by 9 angstroms.

Two facets of the conformation of phosphatidylcholine and similar molecules are particularly relevant to membrane studies: the orientation of the head-group dipole and the extension of the tails. Both have received great attention in the literature: the first because the

Figure I.1

Schematic diagram of dipalmitoyl phosphatidylcholine

electrostatic interaction of the molecules in the lipid matrix will be much different if their dipoles lie in the plane of the matrix rather than perpendicular to it, and the second because the surface area effectively occupied by each molecule will vary according to the degree to which the tails are fully extended and co-linear. The question of dipole orientation remains controversial, but quite understandably so, since the head group, being exposed to the aqueous environment, is certain to be affected very significantly by the temperature, ionic strength, and pH of the solution, and by the state of the lipid matrix as a whole. In the matter of tail conformation in bilayers there is more agreement: the tails are fully extended below the calorimetrically-observed transition temperature (Hitchcock, et al., 1974), but show a greater tendency to bend, especially at locations near the terminal methyl group, above this temperature (Lee, 1975). The transition temperature is lowered if the tails contain carbon-carbon double bonds; it is raised if the tails contain more CH_2 groups (Melchior and Steim, 1976). In monolayers, the orientation of the lipids depends upon the surface pressure: the tails are perpendicular to the film at high pressures and lie in the plane of the film at low pressures (Gaines, 1966, p. 169). Furthermore, the

transition temperature in monolayers increases as the surface pressure increases, but, except at very low pressures, the tails are thought to be fully extended below that temperature and bent above it (Cadenhead, 1971).

I. B. Interactions of Phospholipid Molecules

- Electrostatic Interactions

Since phospholipid heads are polar, there is a charge-charge interaction between them which may be expressed in terms of the Coulombic potential:

$$\Phi_{ij}^c = \frac{Q_i Q_j}{4\pi\epsilon r_{ij}},$$

where ϵ is the permittivity and

$Q_i \equiv$ magnitude of the i^{th} charge

$r_{ij} \equiv$ distance between the i^{th} and j^{th} charges .

Several common phospholipids, including phosphatidyl ethanolamine and phosphatidyl choline, have one positive and one negative charge on each head group. Summing these interactions in pairs, we obtain the total Coulombic potential for two such molecules:

$$\Phi_{\text{total}}^c = \frac{e^2}{4\pi\epsilon} \left(\frac{1}{r_{13}} - \frac{1}{r_{14}} - \frac{1}{r_{23}} + \frac{1}{r_{24}} \right),$$

where charges 1 and 3 have values of $+e$ and charges 2 and 4 have values of $-e$.

We may use the last equation to compute a few representative, two-molecule interaction energies.

Suppose the arrangement were

Then, in a vacuum, with $a=4A$, we have

$b(A)$	$\phi_{total}^C (J)$
4	0.337×10^{-18}
8	0.607×10^{-19}
12	0.197×10^{-19}
16	0.858×10^{-20}

If one of the molecules were flipped end-for-end, so that charges of unlike sign lay closer than charges of like sign, ϕ_{total}^C would be of the same magnitude, but opposite in sign. This interaction energy is clearly dependent upon the orientation of the dipolar head

groups. For separations greater than several dipole lengths, it may be convenient to use a dipole-dipole interaction energy of the form

$$\Phi^{DD} = - \frac{\mu_a \mu_b}{4\pi\epsilon r_{ab}^3} (2 \cos \theta_a \cos \theta_b - \sin \theta_a \sin \theta_b \cos (\phi_b - \phi_a)),$$

where

$\mu_a, \mu_b \equiv$ dipole moments of molecules a and b

$r_{ab} \equiv$ distance between dipole centers,

and $\theta_a, \theta_b, \phi_a$, and ϕ_b describe the orientation of molecules a and b (Hirschfelder, Curtiss, and Bird, p. 849).

- Electrodynamic Interactions

Let us focus our attention upon the saturated hydrocarbon chains of a phospholipid molecule. The electronegativities of carbon and hydrogen are 2.5 and 2.1 on the Pauling scale and we may take the regions away from the carbonyl group to be non-polar. These regions will experience an effective binding force due to the interaction of induced dipoles on neighboring chains. This force is known as a dispersion force and it arises in the following manner:

- (i) a fluctuation in the electron cloud of a given molecule creates a short-lived dipole moment;
- (ii) this dipole moment polarizes a region of a neighboring molecule, creating a dipole moment opposite in orientation to the first;
- (iii) these induced dipoles attract one another.

The situation is somewhat more complicated than this since the fluctuations at a given point depend upon spontaneous fluctuations occurring at all other points in the system.

It is thus the correlation between fluctuations that determines the magnitude of the dispersion force.

The salient features of the interaction energy due to dispersion are expressed by

$$W_{\text{dis}} \propto \frac{\alpha_i \alpha_j}{r_{ij}^6}$$

where

$\alpha_i \equiv$ polarizability of the i^{th} molecule

$r_{ij} \equiv$ distance between the i^{th} and j^{th} molecules.

The polarizability tells one how much induced dipole moment appears when an external field is applied. If the polarizabilities are small, then the dispersion interaction must also be small, since it depends upon field-induced distortions of electron clouds.

There are two basic methods of calculating the magnitude of the dispersive interaction. The first of these is derived from the equations for the dispersive interaction between two point particles in a vacuum. In order to obtain W_{dis} for extended bodies, one assumes that the total interaction is the sum of interactions

between pairs of subunits making up the system (Hamaker, 1937; London, 1930a). It is known that this method is not rigorous.

The other method for calculating the dispersive interaction is based upon work of Lifshitz and co-workers (Lifshitz, 1956; Dzyaloshinskii, et al., 1961) in which the interacting bodies are treated as continuous media. The interaction is considered to take place through a fluctuating electromagnetic field present in the interior of material bodies and extending beyond their boundaries. This set of local fields is then treated as an electromagnetic field covering the entire system (Brenner and McQuarrie, 1973). This theory is thought to be rigorous, but its extension to non-planar geometries is difficult.

In order to obtain some estimate of the size of dispersion energies in bilayer systems, we revert to an application of the London-Hamaker summation technique (Salem, 1960, 1962). Consider two parallel, saturated hydrocarbon chains where

$L \equiv$ length of chain

$\lambda \equiv$ length of subunit in chain

$N \equiv$ number of subunits in chain

$D \equiv$ distance between chains ($D > \lambda$).

By assuming that individual subunits have an interaction energy of the form

$$w_i = \frac{A}{d^6},$$

where

$d \equiv$ distance between subunits

$A \equiv$ Hamaker "constant",

the interactions between all subunits are summed to give

$$w_{\text{dis}} = \frac{AL}{4\lambda^2 D^5} \left[3 \tan^{-1} \left(\frac{L}{D} \right) + \frac{L/D}{1 + L^2/D^2} \right]$$

This becomes, for $D \ll L$,

$$w_{\text{dis}} = \frac{3\pi AL}{8\lambda^2 D^5} = \frac{3\pi AN}{8\lambda D^5}$$

The subunit to be used in the calculation is

with $\lambda = 1.26 \text{ \AA}$. The interaction between two subunits, w_{bond} , is obtained by computing the following bond-bond interactions,

CC ---- CC (1 interaction)
CH ---- CH (4 interactions)
CC ---- CH (4 interactions)

according to

$$w_{\text{bond}} = - \frac{e^2}{d^6} \bar{\alpha} \bar{\alpha}' \left[\frac{\bar{\alpha}}{\langle (\vec{\Sigma} r_i)^2 \rangle} + \frac{\bar{\alpha}'}{\langle (\vec{\Sigma} r_{i'})^2 \rangle} \right]^{-1}$$

where the primes differentiate one bond from another,

$\langle (\vec{\Sigma} r_i)^2 \rangle$ is the expectation value of the operator

$(\vec{\Sigma} r_i)^2$ in the ground state, and r_i and $r_{i'}$ refer

to electronic coordinates of the bonds. This expression

is an approximation of London's formula for the

dipole-dipole contribution to the dispersion energy of two molecules (London, 1930b).

Using experimental values for the mean bond polarizabilities and calculating $\langle (\vec{\Sigma} r_i)^2 \rangle$ for the C-H and C-C bonds with local bond wavefunctions, Salem arrived at

$$w_i = - \frac{1.34 \times 10^3 \frac{\text{kcal}}{\text{mole}}}{d^6} \quad (d \text{ in angstroms}).$$

Hence, for two chains

$$w_{\text{dis}} = \frac{3\pi AN}{8\lambda D^5} = \frac{-N (1.24 \times 10^3 \frac{\text{kcal}}{\text{mole}})}{D^5} \quad (D \text{ in angstroms, } D \ll L).$$

N is the number of subunits, but N is equal to the number of carbon atoms in the chain since each carbon atom belongs to two subunits. Then, if we use values of 4 angstroms for D and 18 for N , we have, for two molecules,

$$W_{\text{dis}} = \frac{-18 (1.24 \times 10^3 \frac{\text{kcal}}{\text{mole}}) (4.19 \frac{\text{J}}{\text{kcal}})}{N_A (4)^5}$$

$$= 0.13 \times 10^{-21} \text{ J} ,$$

where N_A is Avogadro's number.

The attractive term of the Lennard-Jones potential will be used to simulate the dispersive interaction of phospholipids. The maximum depth of the potential well, β , should have a magnitude not greatly different from the value of W_{dis} just calculated.

It should be mentioned that interactions between permanent charges and induced dipoles also contribute to the electrodynamic interaction energy. For molecules of net charge C , polarizability α , and dipole moment μ , the leading term for this interaction is $C^2\alpha/r^4$ and the second term is $\mu^2\alpha/r^6$. However, for neutral molecules, the r^{-4} term disappears and the r^{-6} term may be accounted for in the attractive portion of the Lennard-Jones potential (Hirschfelder, et al., 1954, p. 987).

- Overlap Interactions

A third category of interaction is caused by overlap of the electron clouds of neighboring molecules. In quantum-mechanical terms, electrons are forced, by the Pauli exclusion principle, to occupy orbitals of higher energy as the molecules approach; this leads to a repulsive force between them.

The details of this interaction, also known as the valence force, are quite complicated (Hirschfelder, Curtiss, and Bird, pp. 26, 917). For my simulation purposes, it is sufficient that the repulsive force increase rapidly as the intermolecular distance decreases beyond a certain value; the r_{ij}^{-12} term of the Lennard-Jones potential will serve this function.

I. C. Physical Chemistry of Lipid Monolayers at the Air-Water Interface

-Structure of the Air-Film-Water System

The study of the surface phenomena of liquids has engaged many scientific luminaries since the seventeenth century, among them Young, Laplace, Gauss, Poisson, Kelvin, Maxwell, Gibbs, and Rayleigh (Rayleigh, 1910). In this century, the work of Langmuir, Harkins, Adam, Dervichian, Joly, Rideal, and Gorter has formed the basis of current understanding of films spread at the air-water interface (Gaines, 1966).

Since water consists of polar molecules and air of nonpolar molecules, it is to be expected that amphipathic substances might form stable layers at the air-water interface. A molecule with its hydrophilic part in water and its hydrophobic part in air is in a state of lower energy than would be the case if it had both parts in either water or air. As a result, molecules at the interface are resistant to both solvation and evaporation. In fact, Benjamin Franklin noted that a teaspoonful of oil would exert a remarkable calming effect as it spread to cover a half-acre of water surface

(Franklin, 1773). No more data are needed to show that such oil films are extremely thin: the volume of oil, about 5 cubic centimeters, divided by the area covered, about 2000 square meters, gives a thickness of 25 angstroms.

Modern film-balance experiments measure the horizontal pressure required to constrain the film to a certain area. Such experiments have demonstrated that many amphipathic substances, fatty acids and alcohols in particular, form stable films one molecule thick. Furthermore, since fatty molecules of various chain lengths have the same limiting area per molecule, the molecules are oriented with long axis approximately perpendicular to the interface, at least at high surface pressures (Langmuir, 1917). Work by Adam and others has shown that spontaneous spreading of an unconstrained film results from an attraction between film and substrate molecules greater than the mutual attraction of film molecules. In addition, the film molecules lie parallel to the interface at very low surface pressures (Adam, 1968, pp. 57, 212).

Between the high-pressure, perpendicular and low-pressure, parallel configurations lies an intermediate condition in which the film molecules show resistance to

compression greater than in the parallel configuration but less than in the perpendicular. This leads to pressure-area isotherms such as the one in Figure I.2 for myristic acid. There are five distinct regions:

- I - region of high-pressure, perpendicular configuration often labelled "condensed";
- II - transition region;
- III - intermediate region often labelled "expanded";
- IV - transition region; and,
- V - region of low-pressure, parallel configuration often labelled "gaseous".

The "condensed" phase of the monolayer is one of very tight molecular packing. The isotherm is approximately linear, very steep, and often exhibits a nearly vertical segment as the area per molecule approaches the area of a cross-section normal to the molecule's long axis. Hydrocarbon chain-length and temperature have only slight effect on the isotherms of condensed films of fatty alcohols and unionized fatty acids of fourteen to twenty carbons. However, branched or unsaturated chains will usually prohibit formation of condensed films (Gaines, 1966, p. 233).

The "expanded" phase of the monolayer and the transition to it from the "condensed phase" are poorly

Figure I.2

Pressure-area isotherm for myristic acid
on 0.01N HCl at 290°K
(after Adam and Jessop, 1926)

understood. Experiments with unsaturated fatty acids (Adam and Jessop, 1926; Nutting and Harkins, 1939) show that the expanded limb of the isotherm is raised very slightly by increased temperature or decreased chain length. The transition region, however, is much more affected: the transition to the "condensed" phase occurs at significantly higher surface pressures. In qualitative terms, this behaviour may be explained by taking the "expanded" phase to consist of more-or-less vertically oriented molecules whose hydrocarbon chains are moderately free to bend and twist about carbon-carbon bonds so that the area per molecule is about twice the cross-sectional area. At higher temperatures, these motions being more pronounced, additional pressure must be applied to condense the film. Shorter chains have less attraction for one another and may flop about more violently within a given volume, so again more pressure must be applied. It should be noted that the expanded region of myristic acid isotherms can be fitted with

$$\pi - \pi_0 = \frac{kT}{A - A_0} \quad ,$$

where π_0 is a negative constant and A_0 depends linearly on temperature (Langmuir, 1933). Adequate

theoretical justification of this equation has not yet been developed.

The "gaseous" phase of the monolayer occurs when the molecules are so far apart that they exert virtually no force on one another. The molecular area in this region is extremely large, perhaps several thousand square angstroms, and some systems are known to follow the two-dimensional-gas equation (Adam and Jessop, 1926):

$$\pi = \frac{kT}{A}$$

The transition from "expanded" to "gaseous" phases is marked by a constant surface pressure of approximately 0.1 dyne/cm as the film molecules, having been compressed to a certain density, adhere more closely due to their mutual attraction.

-Measurement of Monolayer Properties

Monolayers are most often constructed in the laboratory from a solution of the film-forming substance in a highly volatile, water-insoluble solvent, such as n-hexane. In most situations, essentially all of the solvent will evaporate within a few minutes, so monolayer properties will be unaltered by its use (Gershfeld, 1974). The film is formed by depositing a small amount of this solution on the surface of highly purified water; precautions are taken to minimize contamination of the surface by dissolved and airborne impurities. The temperature of the system is also controlled.

If we know the molecular weight of the film-forming substance, and if we carefully measure both the area of the constrained film and the amount of substance added, then we may easily calculate the film's surface area per molecule. It can be verified experimentally, in most cases, that the constrained film covers the surface uniformly and that the amount of film-forming substance dissolved in the substrate is minute (Gaines, 1966).

There are two principal methods for measurement of the film's surface pressure: the Wilhelmy and Langmuir techniques. In the former, a thin plate of known size and

density is suspended vertically in the liquid so that it pierces the surface. The force on the suspending cord depends on the size of the submerged portion of the plate and on the surface tension of the liquid. Spreading a monolayer on the surface will change the surface tension and, if the suspending force is constant, the surface-tension change can be computed from the measured change in the plate's vertical position. Alternatively, the surface-tension change may be computed from the measured change in suspending force when the vertical position is held constant. The surface pressure is the difference between the surface tensions of the pure and film-covered liquids.

The Langmuir technique (Langmuir, 1917) measures the force on a floating partition which separates film-covered surface from clean surface. The float is connected by a rigid vertical member to a torsion wire; after calibration, the horizontal force which brings about a given angular displacement from the vertical is known. Division of this force by the length of the float gives the surface pressure of the film.

I. D. Simulation

- Simulation in General

Simulation is the mimicry of one system by another. Of course, what shall be taken as mimicry depends upon the nature of these systems and the goals of the investigator. One may want to mimic one, several, or many characteristics of the original system. One may require that the simulator's internal processes be based on the same principles as the internal processes of the simulated system. Simulation of physical systems most often follows this approach. Alternatively, one may simply require that both systems produce similar output from similar input, regardless of internal processes.

Simulation of physical systems can be useful in two ways:

- (i) it can test the adequacy of explanations;
- (ii) it can provide information which is difficult to obtain from physical experiments.

Our explanations can be tested because simulator construction often demands that the principles which we believe to be operative in the simulated system be made explicit for implementation in the simulator. If the

simulator built on the basis of these principles fails to mimic the original system correctly, we may suspect a flaw in the explanation. Simulators which accurately mimic a physical system can often be used to study aspects of the system which are not easily observable, such as the dynamics of certain subunits, behavior at extremes of temperature or pressure, or behavior over very short or very long periods.

Evaluation of a simulator requires that the veracity of its mimicry be judged. What shall be taken as veracity depends, like the description of mimicry, upon the nature of the systems and the goals of the investigator. In general, simulation of physical systems, especially molecular systems, requires a high degree of correlation between simulator output and laboratory data on the behavior of the real system; we cannot ignore any areas in which they diverge significantly. Interpretation of simulator behavior outside the realm of known correlation must be done prudently.

- Elements of Molecular-dynamics Simulation

Molecular dynamics is a deterministic simulation method in which objects follow Newton's equation of motion while interacting via a force law. The objects that we speak of here are analogues of physical objects; they may correspond to particles or to assemblies of particles. In either case, movement is determined by a finite-difference equation of motion:

$$q_{n+1} = q_n + \dot{q}_n \Delta t + \ddot{q}_n \frac{(\Delta t)^2}{2},$$

where

$q_n \equiv$ generalized coordinate of a
certain object at time t_n

$\Delta t \equiv$ size of the time interval

$t_n \equiv t_0 + n\Delta t$,

and $\dot{q}_n$ and $\ddot{q}_n$ are the first and second time derivatives of the coordinate at time t_n . To obtain this equation, we integrate

$$\frac{d^2 q}{dt^2} = \ddot{q}_n$$

twice with respect to time, requiring that $\dot{q}$ and q have values of $\dot{q}_n$ and q_n at time t_n . Hence, given q , $\dot{q}$, and $\ddot{q}$ at time t_n , we are able to find q at time t_{n+1} .

The values of $\dot{q}_n$ and $\ddot{q}_n$ can be obtained in several ways. For first-order accuracy, we may use

$$\dot{q}_n = \frac{1}{\Delta t} (q_n - q_{n-1})$$

$$\ddot{q}_n = \frac{F_n}{m}$$

Then the equation for q_{n+1} becomes

$$q_{n+1} = 2q_n - q_{n-1} + \frac{F_n}{m} \frac{(\Delta t)^2}{2}$$

where

$F_n \equiv$ force along generalized-coordinate
axis, on a certain object at time t_n
 $m \equiv$ mass of the object

For higher-order accuracy, we may use more complicated formulae for $\dot{q}_n$ and $\ddot{q}_n$ (Schofield, 1973):

$$\dot{q}_n = \dot{q}_{n-1} + \frac{\Delta t}{6m} (2 F_n + 5 F_{n-1} - F_{n-2})$$

$$\ddot{q}_n = \frac{1}{3m} (4 F_n - F_{n-1})$$

In this case the equation for q_{n+1} becomes

$$q_{n+1} = q_n + \dot{q}_n \Delta t + \frac{\Delta t^2}{6m} (4 F_n - F_{n-1})$$

The force law governing the interaction of the analogues in a molecular-dynamics simulation is chosen according to the nature of the real objects they represent. If the objects are point charges, then the force between two of them has the familiar Coulombic form:

$$\vec{F}_{ij}^c = \frac{Q_i Q_j}{4\pi\epsilon r_{ij}^2} \cdot \frac{\vec{r}_{ij}}{r_{ij}}$$

where ϵ is the permittivity and

$Q_i \equiv$ magnitude of the i^{th} point charge

$\vec{r}_{ij} \equiv$ vector of magnitude r_{ij} from the i^{th} to the j^{th} charge .

If the objects are uncharged molecules, then the force law must contain components reflecting both the attraction due to electrodynamic influences and the repulsion due to electron-cloud overlap. This force may be expressed as the gradient of one of several potentials: the Lennard-Jones, the Kihara, or the Buckingham potential, for example (Hirschfelder, Curtiss, and Bird, 1964, pp. 31-35). In the case of the Lennard-Jones 6-12 potential,

$$\vec{F}_{ij}^{LJ} = - \vec{\nabla} \phi_{ij}^{LJ}$$

$$\phi_{ij}^{LJ} = 4\beta \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right]$$

where

$\beta \equiv$ maximum depth of the potential well

$\sigma \equiv$ value of r_{ij} at which the potential equals zero

- Deterministic versus Statistical Simulation

In deterministic, molecular-dynamics simulation, thermodynamic properties are derived from molecular properties by averaging them over all particles and over time. That is, the thermodynamic property, $\bar{P}(k_0, M, N)$ is given by

$$\bar{P}(k_0, M, N) = \frac{1}{M+1} \sum_{k=k_0}^{k=k_0+M} \left(\frac{1}{N} \sum_{i=1}^{i=N} p_i^k \right),$$

where

$p_i^k \equiv$ the microscopic property of particle i at time $k\Delta t$,

$N \equiv$ the number of particles in the system,

$M+1 \equiv$ the number of time-steps over which averaging is done.

All particles in the simulation contribute equally to the determination of P .

In contrast to deterministic methods, non-deterministic simulation obtains thermodynamic properties by averaging molecular properties over statistico-mechanical ensembles. If we use canonical ensembles, the thermodynamic property is given by (Metropolis, et al., 1953)

$$\bar{P} = \frac{\int P \exp(-E/kT) dV^{DN}}{\int \exp(-E/kT) dV^{DN}}$$

where

$E \equiv$ potential energy of a system in the ensemble

$dV^{DN} \equiv$ element of phase space for D space dimensions and N particles per system.

In such simulation, this integration is not carried out over all possible states, but over only a sample of them. This makes the computation less time-consuming, but the choice of sample must be performed very carefully in order to adhere to the principles of statistical mechanics.

SECTION II

CONSTRUCTION OF THE MONOLAYER SIMULATOR

II. A. The Particle-dynamics Program

- General Plan of the Algorithm

There are six basic steps in the repetitive algorithm which moves the molecular analogues through space. We begin the simulation with a preliminary step which is not repeated:

- (0) set initial positions and velocities of analogues.

We then enter the cyclic portion of the simulation:

- (1) place charges at interstices of computational lattice,
- (2) approximate the solution to Poisson's equation for the Coulombic potential at the interstices,
- (3) calculate the Lennard-Jones potentials in the region of each center of force,
- (4) calculate the force components acting upon each analogue,
- (5) calculate new positions and velocities after a small time increment,
- (6) calculate the temperature and surface pressure of the system.

Steps (1) through (6) are then repeated a thousand times or more.

Figure II.1

General flowchart for the particle dynamics program

- The Equations of Motion

As was mentioned in the introduction, the equations of motion consist of formulae which allow one to calculate new velocities and positions on the basis of old velocities, old positions and forces at the old positions. The equations may be written in various forms, all of which are finite-difference expansions of the function $f(t + \Delta t)$ in a Taylor series about $f(t)$. The complete expansion has an infinite number of terms:

$$f(t + \Delta t) = f(t) + \Delta t \dot{f}(t) + \frac{\Delta t^2}{2!} \ddot{f}(t) + \frac{\Delta t^3}{3!} \dddot{f}(t) + \dots$$

Using x_n for $x(n\Delta t)$, v_n for the first time-derivative of x , evaluated at $n\Delta t$, and a_n for the second time-derivative, we may write

$$v_{n+1} = v_n + \Delta t a_n$$

$$x_{n+1} = x_n + \Delta t v_n + \frac{\Delta t^2}{2} a_n$$

This set of equations makes no use of derivatives of the acceleration and so includes only terms out to Δt in v_{n+1} and Δt^2 in x_{n+1} .

More accurate equations may be derived which incorporate derivatives of the acceleration by use of finite-difference approximations. If we include terms out to Δt^2 in both v_{n+1} and x_{n+1} , we obtain

$$\begin{aligned}
v_{n+1} &= v_n + \Delta t a_n + \frac{\Delta t^2}{2} \left(\frac{\Delta a}{\Delta t} \right)_n \\
&= v_n + \Delta t a_n + \frac{\Delta t^2}{2} \left(\frac{a_{n+1} - a_{n-1}}{2\Delta t} \right) \\
&= v_n + \frac{\Delta t}{4} (a_{n+1} + 4a_n - a_{n-1})
\end{aligned}$$

and

$$x_{n+1} = x_n + \Delta t v_n + \frac{\Delta t^2}{2} a_n.$$

Furthermore, if we include terms out to Δt^3 in both v_{n+1} and x_{n+1} , we obtain

$$\begin{aligned}
v_{n+1} &= v_n + \Delta t a_n + \frac{\Delta t^2}{2} \left(\frac{\Delta a}{\Delta t} \right)_n + \frac{\Delta t^3}{6} \left[\frac{\Delta}{\Delta t} \left(\frac{\Delta a}{\Delta t} \right)_n \right]_n \\
&= v_n + \frac{\Delta t}{4} (a_{n+1} + 4a_n - a_{n-1}) \\
&\quad + \frac{\Delta t^3}{6} \left[\frac{1}{2\Delta t} \left(\frac{a_{n+1} - a_n}{\Delta t} - \frac{a_n - a_{n-1}}{\Delta t} \right) \right] \\
&= v_n + \frac{\Delta t}{6} (2a_{n+1} + 5a_n - a_{n-1})
\end{aligned}$$

and

$$\begin{aligned}
x_{n+1} &= x_n + \Delta t v_n + \frac{\Delta t^2}{2} a_n + \frac{\Delta t^3}{6} \left(\frac{\Delta a}{\Delta t} \right)_n \\
&= x_n + \Delta t v_n + \frac{\Delta t^2}{2} a_n + \frac{\Delta t^3}{6} \left(\frac{a_n - a_{n-1}}{\Delta t} \right) \\
&= x_n + \Delta t v_n + \frac{\Delta t^2}{6} (4a_n - a_{n-1}).
\end{aligned}$$

Note that a_{n+1} does not appear in the expression for x_{n+1} since forces are determined by position in the simulation.

The accuracy of these equations of motion is clearly dependent upon the size of the time step, Δt . Since the motion of a particle is determined by the force it experiences at a given position, and since this force is calculated only at a limited number of positions, it is crucial that the force not change drastically from one computed position to the next. Otherwise, unacceptably large errors in the computed trajectory might result.

We can estimate the appropriate size of Δt if we assume that the force should vary by no more than 10% between subsequent positions and that the force has the form

$$F(r) = \frac{c}{r^n} .$$

Then the change in F , ΔF , is given by

$$\Delta F = \Delta r \frac{\delta F}{\delta r} ,$$

or

$$0.1 F(r) = \Delta r \frac{\delta F}{\delta r} .$$

Thus,

$$\begin{aligned} \Delta r &= 0.1 \frac{F(r)}{\delta F / \delta r} \\ &= 0.1 \left(- \frac{r}{n} \right) . \end{aligned}$$

Forces which vary as r^{-n} will change most rapidly for small values of r . If we take the distance of closest approach to be one angstrom, then the largest allowed change in r , using the 10% limit on the change in F , is

$$|\Delta r|_{\max} = \frac{1}{n} \times 10^{-11} \text{ m} .$$

Then the maximum size of the time step will be

$$\Delta t_{\max} = \frac{|\Delta r|_{\max}}{\dot{r}_{\text{avg}}} .$$

The average velocity of a particle of mass M in an assembly at temperature T is

$$\dot{r}_{\text{avg}} = \sqrt{kT/M} ,$$

so that $\dot{r}_{\text{avg}}$ is approximately 50 m/sec if M is 800 daltons and T is 300° K. We then find that

$$\Delta t_{\max} \doteq \frac{2}{n} \times 10^{-13} \text{ seconds} .$$

Time steps of 10^{-16} to 10^{-13} seconds are used in most molecular-dynamics calculations.

When one simulates an isolated system of particles which interact via conservative forces, the most important test of the equations of motion is the degree to which the total energy of the system is conserved. In the case of one rotational and two translational degrees of freedom, the total energy of N particles of mass M and moment of inertia I is

$$E_{\text{tot}} = \sum_{i=1}^N \left[\frac{M}{2} \dot{x}_i^2 + \frac{M}{2} \dot{y}_i^2 + \frac{I}{2} \dot{\theta}_i^2 + P_i(x_i, y_i, \theta_i) \right] .$$

$P_i(x_i, y_i, \theta_i)$ is the total potential energy of the i th particle; it is related to the Coulombic and Lennard-Jones potentials by

$$P_i(x_i, y_i, \theta_i) = \sum_j Q_{ij} \phi_c(x_{ij}, y_{ij}) + \sum_k \phi_{LJ}(x_{ik}, y_{ik}) ,$$

where Q_{ij} is the magnitude of the j^{th} charge on the i^{th} particle, the summation on j is over all such charges, and the summation on k is over all Lennard-Jones force centers on the i^{th} particle. The position of the n^{th} force center on the m^{th} particle is (x_{mn}, y_{mn}) .

P_i 's dependence upon the positions of all particles in the system has not been expressed explicitly here.

The degree to which E_{tot} remains constant during a simulation is a complex function of the force laws, equations of motion, particle density, and time-step size chosen for the simulation algorithm. Since increases in the constancy of E_{tot} are attained, in general, by the use of higher-order equations of motion and smaller time steps, thus requiring more computer time to simulate a given system, exact constancy is not a practicable goal. It is practical, however, to construct the algorithm so that E_{tot} will show fluctuations of only a few percent about some average value as the system evolves in time.

II. B. Calculation of Forces

- Construction of the Coulombic Potential Surface

In its simplest form, the problem to be solved is this: find, as a function of position, the potential due to an infinite plane of discrete charges in free space.

We have several additional constraints:

- (i) the molecular analogues consist of coupled pairs of point charges; coupled charges ("brother charges") do not interact;
- (ii) the infinitude of the plane of charges is generated by replicating a small central square in two dimensions. The net charge in the central square and in each of its images is zero;
- (iii) the charges do not move out of the plane.

This situation is pictured in Figure II.2.

In order to compute a suitable solution for this problem, we must solve Poisson's equation:

$$\nabla^2 \phi(x,y,z) = - \frac{1}{4\pi\epsilon} \rho(x,y,z),$$

Figure II.2

Schematic diagram of molecular analogues
lying in the x-y plane.

where ϕ is the potential, ρ the charge density, and ϵ the permittivity. In order to compute a solution quickly enough to make dynamic simulation feasible, we shall solve the equation at only a finite number of points. These points will be the interstices of a three-dimensional lattice enclosing the central square, as shown in Figure II.3. α, β , and γ will be the numbers of points along the x, y , and z axes; there will therefore be $(\alpha\beta\gamma)$ points in the lattice.

To further enhance the speed of computation, we shall require that all charge be distributed upon the interstices of the lattice. It is thus important that $(\alpha\beta\gamma)$ be large enough to prevent significant distortion of the problem as it is converted from continuous to discrete form. Each charge on the molecular analogues will be apportioned to the four nearest lattice points by two-dimensional linear weighting.

In discrete form, Poisson's equation may be written for each lattice point as

$$\frac{1}{h_x^2}(\phi_W - 2\phi_P + \phi_E) + \frac{1}{h_y^2}(\phi_N - 2\phi_P + \phi_S) + \frac{1}{h_z^2}(\phi_U - 2\phi_P + \phi_D) = - \frac{1}{4\pi\epsilon} Q_P,$$

where

$Q_P \equiv$ charge density at lattice-point P

$\phi_P \equiv$ potential at lattice-point P

$\phi_W \equiv$ potential at the lattice-point one point
away from P in the negative x direction

Figure II.3

Schematic diagram of computational lattice,
central array, and image arrays

$\phi_E \equiv$ potential at the lattice-point one point
away from P in the positive x direction

$h_x \equiv$ distance between lattice points in the x
direction.

$\phi_S, \phi_N, h_y, \phi_D, \phi_U,$ and h_z are defined similarly
for the y and z directions. This equation is derived by
calculating gradients on the basis of functional values at
a point and at its six nearest neighbors. Higher-order
equations may be derived by including more distant points
in the calculation.

If we write the discrete equation for each lattice
point, having labeled them serially from 1 through
($\alpha\beta\gamma$), we obtain a system of ($\alpha\beta\gamma$) equations in
($\alpha\beta\gamma$) unknowns:

$$\frac{1}{h_x^2}(\tilde{\phi}_{1-\beta\gamma} - 2\phi_1 + \phi_{1+\beta\gamma}) + \frac{1}{h_y^2}(\phi_{1+\gamma} - 2\phi_1 + \tilde{\phi}_{1-\gamma}) + \frac{1}{h_z^2}(\phi_{1+1} - 2\phi_1 + \tilde{\phi}_{1-1}) = \frac{-Q_1}{4\pi\epsilon}$$

$$\frac{1}{h_x^2}(\tilde{\phi}_{2-\beta\gamma} - 2\phi_2 + \phi_{2+\beta\gamma}) + \frac{1}{h_y^2}(\phi_{2+\gamma} - 2\phi_2 + \tilde{\phi}_{2-\gamma}) + \frac{1}{h_z^2}(\phi_{2+1} - 2\phi_2 + \phi_{2-1}) = \frac{-Q_2}{4\pi\epsilon}$$

⋮

$$\begin{aligned} \frac{1}{h_x^2}(\phi_{\alpha\beta\gamma-\beta\gamma} - 2\phi_{\alpha\beta\gamma} + \tilde{\phi}_{\alpha\beta\gamma+\beta\gamma}) + \frac{1}{h_y^2}(\tilde{\phi}_{\alpha\beta\gamma+\gamma} - 2\phi_{\alpha\beta\gamma} + \phi_{\alpha\beta\gamma-\gamma}) \\ + \frac{1}{h_z^2}(\tilde{\phi}_{\alpha\beta\gamma+1} - 2\phi_{\alpha\beta\gamma} + \phi_{\alpha\beta\gamma-1}) = \frac{-Q_{\alpha\beta\gamma}}{4\pi\epsilon}, \end{aligned}$$

where a tilde marks those potentials lying outside the lattice. Boundary conditions must be specified which determine values for these external potentials, either "explicitly" or "implicitly". If the conditions are "explicit",

numerical values are provided and terms like $\frac{1}{h_x^2} \tilde{\phi}_{1-\beta\gamma}$

may be transferred to the right side of the first equation. If the conditions are "implicit", the external potential is given in terms of one of the $(\alpha\beta\gamma)$ internal potentials. In both cases we may write the system of equations as a single matrix equation:

$$\hat{\hat{A}}\hat{\phi} = \hat{Q}$$

where

$$\hat{\phi} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{\alpha\beta\gamma} \end{bmatrix}$$

$$\hat{Q} = \frac{-1}{4\pi\epsilon} \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_{\alpha\beta\gamma} \end{bmatrix}$$

and $\hat{\hat{A}}$ is an $(\alpha\beta\gamma \times \alpha\beta\gamma)$ matrix whose elements are derived from the left sides of the equations.

Before discussing the boundary conditions used in the simulator, let us see how the matrix equation is solved

for the potentials. We first decompose $\hat{A}$ into two matrices,

$$\hat{A} = \hat{L}\hat{U},$$

where all of $\hat{L}$'s elements are zero above the main diagonal and all of $\hat{U}$'s elements are zero below the main diagonal. In addition, $\hat{L}$'s main-diagonal elements are all equal to one. We now have:

$$\hat{L}(\hat{U}\phi) = \hat{Q},$$

or, writing $\hat{c}$ for $\hat{U}\phi$,

$$\hat{L}\hat{c} = \hat{Q}.$$

This equation corresponds to the system of equations

$$\begin{aligned} l_{11}c_1 &= Q_1 \\ l_{21}c_1 + l_{22}c_2 &= Q_2 \\ l_{31}c_1 + l_{32}c_2 + l_{33}c_3 &= Q_3 \\ &\vdots \\ l_{N1}c_1 + l_{N2}c_2 + l_{N3}c_3 + \dots + l_{NN}c_N &= Q_N, \end{aligned}$$

where

$$N \equiv \alpha\beta\gamma.$$

Since we know l_{11} and Q_1 , we can solve the first equation for c_1 . We can then solve the second equation for c_2 , then the third for c_3 , and so on until we know all the elements of $\hat{c}$. This process is called forward substitution. We then solve

$$\hat{U}\hat{\phi} = \hat{c}$$

for $\hat{\phi}$. The system of equations has similar structure:

$$\begin{aligned} u_{11}\phi_1 + u_{12}\phi_2 + u_{13}\phi_3 + \dots + u_{1N}\phi_N &= c_1 \\ &\vdots \\ u_{N-2,N-2}\phi_{N-2} + u_{N-2,N-1}\phi_{N-1} + u_{N-2,N}\phi_N &= c_{N-2} \\ u_{N-1,N-1}\phi_{N-1} + u_{N-1,N}\phi_N &= c_{N-1} \\ u_{NN}\phi_N &= c_N \end{aligned}$$

This time we solve the last equation for ϕ_N , then the next-to last for ϕ_{N-1} , and so on until we know all the elements of $\hat{\phi}$. This process is called backward substitution. Now that we have $\hat{\phi}$, we have reached our goal: we know the potential at all interstices of the computational lattice.

Treatment of the x and y boundaries in the monolayer simulator makes the system periodic in the x and y directions. This is accomplished by implicit boundary conditions which make potentials at points a distance h_x or h_y outside the central array equal to the potential at the nearest point of the array. The same is true for points in planes parallel to the central array. That is,

$$\tilde{\phi}_{p-\beta\gamma} = \phi_p \quad (\text{western boundary})$$

$$\tilde{\phi}_{p+\beta\gamma} = \phi_p \quad (\text{eastern boundary})$$

$$\tilde{\phi}_{p+\gamma} = \phi_p \quad (\text{northern boundary})$$

$$\tilde{\phi}_{p-\gamma} = \phi_p \quad (\text{southern boundary})$$

Figure II.4 depicts this situation for one x-y plane of the three-dimensional lattice.

The z boundary of the simulator is treated in different fashion since all charge lies in the monolayer plane; periodic boundary conditions would simulate an infinity of parallel monolayers, not just one. We therefore attempt to specify an implicit boundary condition which is consistent with the way in which the actual potential falls off with increasing distance from the monolayer plane.

Without performing any calculations, we may say the following things about the potential due to an infinite sheet of equal numbers of discrete positive and negative charges lying in the x-y plane:

- (i) the potential will be zero at an infinite distance from the plane;
- (ii) the gradient of the potential will also be zero at infinite distance;
- (iii) if $\phi(x,y,z)$ is plotted versus z, the profile will begin at $z=0$ with some positive or negative value, dependent upon x and y, and approach zero smoothly as z increases.

Figure II.4

Diagram of one x-y plane in the computational lattice

- - internal point
- - external point

Figure II.5 shows $\phi(x,y,z)$ as a function of z .

Further analysis results in the following points:

- (iv) If we compare the slopes of the $\phi(x,y,z)$ curves at a fixed value of z , we can expect great variation for small values of z and much less variation for large values.
- (v) If the $\phi(x,y,z)$ curves go to zero at approximately the same value of z , then, considering two curves which approach the z axis from the same side, the curve which lies further from the z axis will have a slope of greater absolute value.

Using (v) we may write

$$\frac{\delta\phi}{\delta z} \propto -\phi(x,y,z) = -k(x,y,z) \phi(x,y,z) .$$

Point (iv) suggests that $k(x,y,z)$ may be approximated by a constant for fixed z , if the value of Z , Z_c , is relatively large.

$$\text{Then} \quad \left. \frac{\delta\phi}{\delta z} \right|_{z=Z_c} \doteq -k_c \phi(x,y,Z_c) .$$

There is a way to construct an implicit boundary condition on z which gives normal derivatives of this form at the x - y planes bounding the computational lattice. We make potentials at external points a distance h_z away from the boundary planes equal to a fraction of the potential at the nearest point on the boundary plane.

Figure II.5

Schematic diagram of the z -dependence of the potential due to an infinite sheet of equal numbers of discrete positive and negative charges lying in the x - y plane

Potential at (x_0, y_0, z)

That is,

$$\tilde{\phi}_{p+1} = w\phi_p \quad (\text{upper boundary})$$

$$\tilde{\phi}_{p-1} = w\phi_p \quad (\text{lower boundary}),$$

where

$$0 \leq w \leq 1.$$

Then the slope at point p in the boundary plane will be

$$\left. \frac{\Delta\phi}{\Delta z} \right|_p = \frac{\tilde{\phi}_{p+1} - \phi_p}{h_z}.$$

Thus

$$\begin{aligned} \left. \frac{\Delta\phi}{\Delta z} \right|_p &= \frac{w\phi_p - \phi_p}{h_z} \\ &= -\left(\frac{1-w}{h_z}\right)\phi_p \\ &= -k_c \phi_p, \end{aligned}$$

where

$$k_c = \frac{1-w}{h_z}.$$

To test the validity of the assumption that

$$\frac{\partial \phi}{\partial z} \Big|_{z=z_c} = -k_c \phi(x, y, z_c)$$

for sufficiently large z_c , we consider the potential due to an infinite array of square cells which have an edge length of 8 angstroms and contain one positive and one negative charge separated by 5.66A. Table II.1 shows the results of direct-summation calculations of the potential at various points (see PWSUM program in Appendix C). The last column on the right shows a number which is directly proportional to k_c , since

$$-k_c \phi_z = \frac{\partial \phi}{\partial z} \Big|_{z=z} = \frac{-\Delta(z_1, z_2)}{h_z}$$

and

$$k_c = \frac{1}{h_z} \frac{\Delta(z_1, z_2)}{\phi_{z_1}}$$

When we are only 4A away from the plane of charges we see that the Δ/ϕ ratio varies from 0.3015 to 0.3266. When we move out to 8A and use an h_z of 2A rather than 1A, as is more reasonable for the simulation, the Δ/ϕ ratio varies from 0.53406 to 0.54291. It appears that k_c is approximately constant.

Table II.1

Results of direct-summation calculation of the
potential due to an infinite array of $8A \times 8A$ cells
each containing one positive and one negative charge

x	y	z_1	z_2	$\varphi(x, y, z_1)$	$\varphi(x, y, z_2)$	$\frac{\Delta(\varphi_{z_1} - \varphi_{z_2})}{\varphi_{z_1}}$
(Å)	(Å)	(Å)	(Å)	(volts)	(volts)	
8	7	4	5	4.0221	2.7354	0.3199
7				3.8874	2.6340	0.3224
6				3.4689	2.3360	0.3266
4				1.9385	1.3170	0.3206
2				0.4339	0.3031	0.3015
8	7	8	10	0.84698	0.38717	0.54288
7				0.81369	0.37193	0.54291
6				0.71870	0.32851	0.54291
4				0.40763	0.18676	0.54184
2				0.09660	0.04501	0.53406

In order to see how much error is introduced in the calculation of the potential by the assumption

$$\frac{\partial \phi}{\partial z} \bigg|_{z=z_c} \doteq -k_c \phi(x, y, z_c) \propto -\frac{1-w}{h_z} \phi(x, y, z_c)$$

we must compare the free-space potential obtained by direct summation with the potentials calculated using this assumption. In Figure II.6, we plot such potentials for a case which should show the approximate potentials in the worst possible light, that is, along a line in the x-y plane passing within one angstrom of one of the positive charges. Even though the assumption is quite inaccurate for such small values of z_c , approximate potentials C and E lie within ten percent of the free-space potential (A) and their gradients agree rather well with those of A for x less than $5A$. When the potential is calculated at points very close to the charge (within $1.732A$ for x between 5 and $9A$), and especially on the boundary of the lattice, agreement is poorer. These conditions do not occur in the dynamic simulation. Furthermore, when z_c is small, a decrease in w improves agreement, while, when z_c is large, an increase in w improves agreement. This is to be expected, since the slope of $\phi(x, y, z)$, as a function of z , decreases in absolute value as z is increased.

Figure II.6

Comparison of actual and approximate
potentials in the plane of the charges

- A - actual free-space potential (from PWSUM3)
 B - LWDEC3 result with $z_c = 2 \text{ \AA}$, $w = 0.95$
 C - $= 5 \text{ \AA}$ $= 0.95$
 D - $= 2 \text{ \AA}$ $= 0.90$
 E - $= 5 \text{ \AA}$ $= 0.90$

- Construction of the Non-Coulombic Potential Surface

In contrast to the particle-field interaction used to construct the Coulombic potential surface, we shall use direct summation of particle-particle interactions to construct the non-Coulombic surface. This is made feasible by the rapid decrease in strength of these interactions as the inter-particle distance increases. For example, using the Lennard-Jones 6-12 potential,

$$\phi_{ij}^{LJ} = 4\beta \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right],$$

We may tabulate the decrease as follows:

σ/r_{ij}	ϕ_{ij}^{LJ}
2.0	16028 β
1.5	473.4 β
1.0	0
$2^{-1/6}$	-1.0 β
0.5	$-0.615 \times 10^{-1}\beta$
0.1	$-4.0 \times 10^{-6}\beta$
0.01	$-4.0 \times 10^{-12}\beta$

We see that the potential differs from zero by only $4 \times 10^{-6}\beta$ at a distance of 10σ . We may therefore truncate the direct summation of ϕ_{ij}^{LJ} over all j at some maximum distance R . That is,

$$\begin{aligned}\phi_{i,\text{total}}^{\text{LJ}} &= 4\beta \sum_{\text{all } j} \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right] \\ &\doteq 4\beta \sum_{\substack{j \\ r_{ij} \leq R}} \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right].\end{aligned}$$

If σ were 4 Å, R 10σ , and the average area per molecule 50 Å², one would compute only some 200 interactions to obtain $\phi_{i,\text{total}}^{\text{LJ}}$. If R were 5σ , only 50 interactions would be needed. Of course, we must perform the same calculation for each force center in the system.

- Calculation of Potential Gradients

Having computed the value of the Coulombic potential at each of the computational-lattice interstices, we calculate the x and y gradients of the potential at the actual location of the I^{th} charge using weighted averages of the potentials at the four nearest lattice points. Thus

$$\left(\frac{\Delta\phi}{\Delta x}\right)_I^C = \frac{1}{h_x} \left[\left(1 - \frac{Y(I)}{h_y}\right)(\phi_{SE} - \phi_{SW}) + \left(\frac{Y(I)}{h_y}\right)(\phi_{NE} - \phi_{NW}) \right]$$

$$\left(\frac{\Delta\phi}{\Delta y}\right)_I^C = \frac{1}{h_y} \left[\left(1 - \frac{X(I)}{h_x}\right)(\phi_{NW} - \phi_{SW}) + \left(\frac{X(I)}{h_x}\right)(\phi_{NE} - \phi_{SE}) \right]$$

where the neighboring potentials are labelled NW, SW, NE, SE, and $(X(I), Y(I))$ is the position of the I^{th} charge relative to the southwestern lattice point. The gradient in z is omitted since the molecular analogues are constrained to lie in the plane of the monolayer.

Gradients of the non-Coulombic potential are found in the same way, requiring the potential to be calculated not at the location of the i^{th} force center, (x_i, y_i) , but at four near-by points: (x_i+s, y_i) , (x_i-s, y_i) , (x_i, y_i+s) , and (x_i, y_i-s) . Thus

$$\left(\frac{\Delta\phi}{\Delta x}\right)_{x_i, y_i}^{\text{LJ}} = \frac{\phi(x_i+s, y_i) - \phi(x_i-s, y_i)}{2s}$$

$$\left(\frac{\Delta\phi}{\Delta y}\right)_{x_i, y_i}^{\text{LJ}} = \frac{\phi(x_i, y_i+s) - \phi(x_i, y_i-s)}{2s}$$

- Application of Forces to the Molecular Analogue

We are now able to write expressions for the resultant forces and the torque on a simple dumbbell-type analogue whose Coulombic and Lennard-Jones force centers lie atop one another at either end of a rigid rod of length L . If we label the rod ends A and B, the force components will be:

$$F_x^A = - \left(\frac{\Delta\phi}{\Delta x} \right)_A^C - \left(\frac{\Delta\phi}{\Delta x} \right)_A^{LJ}$$

$$F_x^B = - \left(\frac{\Delta\phi}{\Delta x} \right)_B^C - \left(\frac{\Delta\phi}{\Delta x} \right)_B^{LJ}$$

$$F_y^A = - \left(\frac{\Delta\phi}{\Delta y} \right)_A^C - \left(\frac{\Delta\phi}{\Delta y} \right)_A^{LJ}$$

$$F_y^B = - \left(\frac{\Delta\phi}{\Delta y} \right)_B^C - \left(\frac{\Delta\phi}{\Delta y} \right)_B^{LJ}$$

The force components at the center-of-mass of the analogue will be:

$$F_x^{CM} = F_x^A + F_x^B$$

$$F_y^{CM} = F_y^A + F_y^B$$

Finally, if we take the center-of-mass to lie at the midpoint of the rod and assume the rod makes an angle θ with the positive x-axis, then the magnitude of the torque about the center-of-mass is:

$$\tau^{CM} = \frac{L}{2} (F_x^A \sin \theta - F_y^A \cos \theta - F_x^B \sin \theta + F_y^B \cos \theta) .$$

II. C. Calculation of Temperature, Surface Pressure, and Molecular Area

- Temperature

When dealing with a system of N particles, each with D degrees of freedom, one may write the total kinetic energy of the system as

$$E_{\text{kin}} = \frac{D}{2} NkT$$

by the equipartition theorem. Since

$$E_{\text{kin}} = \sum_{i=1}^N \left(\frac{M}{2} \dot{x}_i^2 + \frac{M}{2} \dot{y}_i^2 + \frac{I}{2} \dot{\theta}_i^2 \right)$$

for a system of N particles having mass M , moment of inertia I , one rotational and two translational degrees of freedom, the average temperature of the system is

$$\bar{T} = \frac{1}{3kN} \sum_{i=1}^N (M\dot{x}_i^2 + M\dot{y}_i^2 + I\dot{\theta}_i^2) .$$

Each of the velocities in this equation is available during the simulation, so $\bar{T}$ may be calculated at each time step at which the velocities have been computed.

How might one expect the calculated average temperature to behave? At the beginning, just after the analogues have been placed on the central square, they will have a total energy determined by their initial positions and velocities:

$$E_{\text{tot}}(t=0) = \sum_{i=1}^N \left(\frac{M}{2} \dot{x}_{i0}^2 + \frac{M}{2} \dot{y}_{i0}^2 + \frac{I}{2} \dot{\theta}_{i0}^2 \right) + P_i(x_{i0}, y_{i0}, z_{i0})$$

Suppose that we choose to make all initial velocities zero, so that the initial total energy depends only upon initial positions:

$$E_{\text{tot}}(t=0) = \sum_{i=1}^N P_i(x_{i0}, y_{i0}, z_{i0})$$

Unless the initial positions have been chosen so that each of the N particles lies at a local minimum of the potential surface, the particles will be accelerated and some potential energy will be converted to kinetic energy. At $t = n\Delta t$,

$$E_{\text{tot}}(t=n\Delta t) = \sum_{i=1}^N \left(\frac{M}{2} \dot{x}_{in}^2 + \frac{M}{2} \dot{y}_{in}^2 + \frac{I}{2} \dot{\theta}_{in}^2 \right) + P_i(x_{in}, y_{in}, \theta_{in})$$

Since the change in kinetic energy is a complicated function of the shape of the potential surface and the particle positions, it is likely that the amount of energy transformed into kinetic energy will not give the system a temperature very near that of the physical monolayer, about 300° K. We can therefore expect that it will be necessary to modify the energy of the system to bring the temperature into the appropriate range. We can most easily do this by adjusting the magnitudes of the velocity components at various times. This energy modification is

done gradually over a period of several hundred time steps. After this "equilibration period", the system temperature should show fluctuations of only a few percent about the desired average temperature.

- Surface Pressure and Molecular Area

Surface pressure is the force per unit length that a monolayer exerts against the boundaries which confine it to a certain area. In molecular-dynamics simulation, it is possible to compute this pressure from either the collision rate (Alder and Wainwright, 1959) or the virial theorem of Clausius. We shall use the virial method here, but we shall need to derive the theorem in two-dimensional form, rather than the standard three-dimensional form (Mason and Spurling, pp. 18-22).

We begin the derivation with a manipulation of the ordinary expression for the kinetic energy associated with the generalized coordinate q :

$$\begin{aligned}\frac{1}{2}m\dot{q}^2 &= \frac{m}{2} \left(\frac{d}{dt} (q\dot{q}) - q \frac{d}{dt} (\dot{q}) \right) \\ &= \frac{1}{2} \frac{d}{dt} (qp_q) - \frac{1}{2} q \frac{d}{dt} (p_q) \\ &= \frac{1}{2} \frac{d}{dt} (qp_q) - \frac{1}{2} qF_q ,\end{aligned}$$

where p_q is the generalized momentum and F_q the force. We apply this expansion to the expression for the total kinetic energy given above:

$$\begin{aligned}
E_{\text{kin}} &= \sum_{i=1}^N \left(\frac{M}{2} \dot{x}_i^2 + \frac{M}{2} \dot{y}_i^2 + \frac{I}{2} \dot{\theta}_i^2 \right) \\
&= \frac{1}{2} \frac{d}{dt} \sum_{i=1}^N (x_i p_{xi} + y_i p_{yi} + \theta_i p_{\theta i}) \\
&\quad - \frac{1}{2} \sum_{i=1}^N (x_i F_{xi} + y_i F_{yi} + \theta_i F_{\theta i}) .
\end{aligned}$$

If the first term on the right is averaged over the interval τ , we find that it approaches zero as τ is made arbitrarily large, since

$$\begin{aligned}
&\frac{1}{\tau} \int_0^\tau \frac{d}{dt} \sum_{i=1}^N (x_i p_{xi} + y_i p_{yi} + \theta_i p_{\theta i}) dt \\
&= \frac{1}{\tau} \sum_{i=1}^N \left(x_i p_{xi} \Big|_0^\tau + y_i p_{yi} \Big|_0^\tau + \theta_i p_{\theta i} \Big|_0^\tau \right)
\end{aligned}$$

and each term in the last sum is finite (Jeans, pp.

129-30). Using bars to indicate time averages, we now have

$$\bar{E}_{\text{kin}} = - \frac{1}{2} \sum_{i=1}^N \overline{(x_i F_{xi} + y_i F_{yi} + \theta_i F_{\theta i})} .$$

We next separate the expression on the right, called the virial by Clausius, into two parts, one involving the forces exerted on the monolayer molecules by the boundaries and the other involving intermolecular forces. When dealing with the boundary portion, it is convenient to sum over the $2N$ force centers rather than the N centers of mass so that the angular term does not appear explicitly. Hence

$$\begin{aligned} \bar{E}_{kin} = & -\frac{1}{2} \sum_{i=1}^N (\overline{x_i F_{xi} + y_i F_{yi} + z_i F_{zi}})_{im} \\ & -\frac{1}{2} \sum_{j=1}^{2N} (\overline{x_j F_{xj} + y_j F_{yj}})_{bnd} . \end{aligned}$$

We may write the last term as the dot product of force-center positions and forces exerted upon the molecular force centers by the boundaries:

$$-\frac{1}{2} \sum_{j=1}^{2N} (\overline{x_j F_{xj} + y_j F_{yj}})_{bnd} = -\frac{1}{2} \sum_{j=1}^{2N} (\overline{\vec{r}_j \cdot \vec{F}_j}) .$$

These forces are equal and opposite to the force per unit length exerted upon the boundaries by the molecules, so we may transform this sum into an integral over the length of the boundaries using the surface pressure, Π :

$$-\frac{1}{2} \sum_{j=1}^{2N} (\overline{\vec{r}_j \cdot \vec{F}_j}) = \frac{1}{2} \int_b (\vec{r} \cdot \Pi \hat{n}) d\ell ,$$

where $\hat{n}$ is the unit vector outwardly normal to the boundary at $\vec{r}$. If we take β to be the angle between the x unit vector and $\hat{n}$, then this integral can be evaluated as follows:

$$\begin{aligned} \frac{1}{2} \int_b (\vec{r} \cdot \Pi \hat{n}) d\ell &= \frac{1}{2} \Pi \int_b (x\hat{x} + y\hat{y}) \cdot (\cos\beta\hat{x} + \sin\beta\hat{y}) \frac{dy}{\cos\beta} \\ &= \frac{1}{2} \Pi \int (x dy + y \frac{\sin\beta}{\cos\beta} dy) \\ &= \frac{1}{2} \Pi \int x dy + y \left(\frac{-dx}{dy} \right) \left(\frac{dy}{dy} \right) dy \\ &= \frac{1}{2} \Pi \int (x dy - y dx) \\ &= \Pi A, \end{aligned}$$

where A is the area enclosed by the boundaries. Green's theorem in the plane was applied in the final step.

We have now obtained a useful relation between the average kinetic energy, the surface pressure, the area, and the intermolecular forces:

$$\bar{E}_{kin} = \Pi A - \frac{1}{2} \sum_{i=1}^N \overline{(x_i F_{xi} + y_i F_{yi} + \theta_i F_{\theta i})}_{im}.$$

Since $\bar{E}_{kin}$ is $\frac{3}{2}NkT$ by the equipartition theorem, we see that

$$\Pi = \frac{3}{2} \left(\frac{N}{A} \right) kT + \frac{1}{2A} \sum_{i=1}^N \overline{(x_i F_{xi} + y_i F_{yi} + \theta_i F_{\theta i})}_{im}.$$

This equation is used to compute the surface pressure since positions, intermolecular forces, and temperature are all available during the simulation.

Molecular area is the average area occupied by each molecule in the monolayer. For the simulation, it is simply the area of the central square divided by the number of molecules found there.

- Generation of Pressure-area Curves

Using results from the simulator, we want to plot surface pressure (Π) versus molecular area (A_m) for systems at several temperatures. That is, we want to plot isotherms in the Π - A_m plane. Since it is difficult to "equilibrate" the simulator precisely at a given temperature for different values of A_m , we follow a two-step procedure. First, we perform simulations for several values of A_m at various temperatures in the 275°K - 325°K range and plot curves of constant A_m in the Π -T plane. Second, we choose several temperatures to be used for isotherms, find the Π values for these temperatures from the constant- A_m curves, and plot these Π values in the Π - A_m plane. This transformation of data is depicted in Figure II.7.

Figure II.7

Transformation of simulation results from
pressure-versus-temperature form to
pressure-versus-area form

• $\equiv$ simulation result
 ⊙ $\equiv$ interpolated result

SECTION III

COMPARISON OF EXPERIMENTAL AND SIMULATION RESULTS

III.A. Experimental Pressure-area Curves

In order to compare experimental and simulated pressure-area isotherms, we plot, in Figure III.1, experimental results for dipalmitoyl phosphatidylcholine at the air-water interface (Hui, et al, 1975). The films were spread on 10 mM NaCl, pH 7.4, and pressure-area data were obtained by the Langmuir method. Similar results have been obtained by Phillips and Chapman (1968) and Vilallonga (1968).

These isotherms are very similar in shape to the myristic acid isotherm shown in Figure I.2; they exhibit the "expanded", "transition", and "condensed" regions discussed in Section I.C. As expected, condensed behaviour for these double-chain molecules occurs at a molecular area approximately twice as large as for single-chain myristic acid. $3NkT/2A$ is also plotted in Figure III.1 for comparison.

Figure III.1

Experimental pressure-area isotherms
for dipalmitoyl phosphatidylcholine
(after Hui, et al, 1975)

III.B. Simulated Pressure-area Curves

Before discussing simulated pressure-area curves, it is important to consider the effect of various parameters upon the behaviour of the simulator. Of particular importance is the cut-off radius used in the calculation of the Lennard-Jones potential. Experiments were performed at low, intermediate, and high densities with radii of 4, 10, 20, and 40 angstroms. Temperatures and pressures computed at a given time differed by less than one part in ten thousand for radii of 10, 20, and 40 angstroms. Since computation time for this portion of the algorithm increases as the fourth power of the cut-off radius, the smallest of these radii, 10 angstroms, was chosen for general use.

The size of the time step used in numerical integration of the equations of motion is also important. While it should be in the range of 10^{-16} to 10^{-13} seconds, the most appropriate value depends upon the integration algorithm and the size of the intermolecular forces. Steps of 0.1, 0.25, 0.5, 5.0, and 10.0×10^{-15} seconds were tested at the extremes of molecular density. Those above 1.0×10^{-15} seconds caused rapid growth of the system's total kinetic energy. When a value of 0.25×10^{-15} seconds was employed, the kinetic energy remained

within 3 percent of its average value over a period of 1000 time steps.

The dependence of pressure-area results upon the fineness of the computational lattice was also tested. The distance between lattice points was given values of 1.3846, 1.6364, 2.0000, and 2.4444 angstroms; results were compared at constant molecular area. Pressures computed with one of the first three distances differed by only 4 percent, while the last distance led to a 27 percent alteration. A distance of 2.0000 angstroms was chosen as the standard value.

In order to verify that the matrix calculation of the Coulombic interaction based on Poisson's equation gives realistic dynamic results, this interaction was calculated in a direct, pair-wise fashion in several experiments. Cut-off radii of 10, 20, and 40 angstroms (2.5, 5, and 10 times the length of the molecular analogue) were used; pressures deviated from Poisson-based results by approximately 11, 3, and 1 percent. This implies that, when the cut-off radius of the pair-wise calculation is sufficiently large, the Poisson-based and pair-wise results are in close agreement. As was mentioned above, the pair-wise calculation with a relatively large cut-off radius is much more time-consuming than the Poisson-based calculation.

The pressure-area characteristics of the 32-molecule system were relatively insensitive to the precise nature of the non-periodic boundary condition applied to the potential on the xy-planes lying at a fixed distance above or below the plane of charges. The parameter which determines the gradient of the potential at the boundary, w , was given values of 0.95, 0.85, 0.75, and 0.60; the boundary was located at 6, 12, and 24 angstroms from the charges. A boundary condition of zero potential was tested at 12 and 24 angstroms, corresponding to the location of a grounded, conducting plate at these distances. The largest variation in all of these results was 2 percent; in most cases it was substantially less than this. The insensitivity of the results to variation of such parameters is probably due to two factors: the net charge on the plane is zero and the analogues are constrained to lie in the plane.

As one might expect, the simulation results did show substantial dependence upon those factors which affect the size of the intermolecular forces directly. These are the relative permittivity for the electrostatic force, and the well depth and distance parameters of the Lennard-Jones potential for the electrodynamic and overlap forces. The relative permittivity, which is about 80 in bulk water at 298°K, was given values of 80, 60, and 40 in experiments of low, intermediate and high molecular density.

Decreasing the permittivity from 80 to 40 caused an increase in pressure of approximately 11 percent at each density; a decrease from 80 to 60 caused a 4 percent increase. Such increases move the simulated pressure-area curves away from the experimental curves.

The distance parameter of the Lennard-Jones potential was varied from 3.0 to 6.0 angstroms at several molecular densities. Values of 3.0 and 4.0 angstroms result in pressures which differ by less than one percent, while values of 5.0 and 6.0 result in pressures which are 8 to 12 percent higher. This result and theoretical investigation of hydrocarbon-chain interaction energies (Shapiro and Ohki, 1974) suggest that the 4.0 angstrom value is the most appropriate.

The well depth of the Lennard-Jones potential is expected to be approximately 0.26×10^{-22} joules (Shapiro and Ohki 1974). Simulations were run with values of 0.10, 0.26, 0.53, 1.06, and 2.12×10^{-22} joules. The three smallest values gave results which were not significantly different, while the two largest caused pressure results to be about 4 and 7 percent lower at molecular areas of 40.5 and 50.0 square angstroms per molecule. At higher molecular areas, the 1.06 and 2.12×10^{-22} joule values did not change the pressure results appreciably. A value of 0.53×10^{-22} joules was used in most experiments.

The simulation pressures obtained using a cut-off radius of 10 angstroms for the Lennard-Jones 12-5 potential, a time step of 0.25×10^{-15} seconds, a lattice distance of 2.0 angstroms, a relative permittivity of 80, Lennard-Jones parameters of 0.53×10^{-22} joules and 4.0 angstroms, a value of 0.95 for w , and a z-boundary location of 6 angstroms are plotted in Figure III.2 as a function of temperature at fixed molecular area and in Figure III.3 as a function of molecular area at fixed temperature. $3NkT/2A$ is also plotted in Figure III.3 for comparison.

Figure III.2

Simulation results: surface pressure as a
function of temperature at fixed molecular area
(see text for parameter values)

Figure III.3

Simulation results: surface pressure as a
function of molecular area at fixed temperature
(see text for parameter values)

III.C. Comparison of Pressure-area Curves

Figure III.4 combines the experimental and simulation results for ease of comparison. We note the following similarities:

- (i) The slopes of the steepest portions of the curves have approximately the same value.
- (ii) The magnitudes of the simulation pressures lie close to the experimental pressures for molecular areas greater than fifty square angstroms.
- (iii) The variation in simulation pressures with temperature at the same molecular area is of the same order of magnitude as the variation in experimental pressures.

We also note the following dissimilarities:

- (i) The "transition" regions of the experimental isotherms do not appear in the simulation isotherms.
- (ii) Portions of the experimental isotherms lie below $3NkT/2A$; the simulation isotherms lie entirely above this value.

The implications of these similarities and dissimilarities are discussed in Section IV.A.

Figure III.4

Juxtaposition of experimental and
simulation isotherms

SECTION IV

CONCLUSIONS

IV. A. Adequacy of the Model

The simulation results show that:

- (i) Realistic modeling of some monolayer properties is feasible using the molecular-dynamics technique.
- (ii) Qualitative agreement in magnitude, slope and temperature dependence of pressure-area isotherms can be obtained, even when the molecular analogue is a very simple linear dumbbell with coincident Coulombic and Lennard-Jones force centers.
- (iii) The model based on this simple analogue shows no tendency to simulate the highly temperature-dependent transition regions of the pressure-area isotherms, nor are simulated isotherms completely realistic with respect to an ideal-gas isotherm of $3NkT/2A$.

The deviation of simulator isotherms from realistic isotherms is quite likely the consequence of the coarse manner in which the hydrocarbon tails of the phospholipids were modeled: their extension in the z-direction was suppressed and their ability to flail about was ignored. Ways in which these shortcomings might be remedied are

suggested in Section IV.B.

IV. B. Directions for Further Study

Of primary interest in additional work is refinement of the the molecular analogue. Several improvements appear feasible:

- (i) A three-dimensional analogue could be designed.

This might consist of two coupled chains of force centers, perpendicular to the monolayer plane, with an electrostatic center constrained to lie in the plane and electrodynamic centers out of the plane. Initial simulations would compare the behaviour of the linear-dumbbell and coupled-chain analogues with chains constrained to be linear and parallel. Later simulations would allow bending of the chains.

- (ii) The constraint that the analogue have zero net charge could be relaxed. This must be done with care, since it would require that counterbalancing charge in the aqueous phase be modeled .

- (iii) A more complex model for the polar head group could be developed. It might include, in a static or quasi-static arrangement, water molecules which may be associated with the head group.

Attainment of these goals would certainly require algorithms which are extremely efficient in their use of computer memory space.

APPENDIX A

MONOLAYER-SIMULATION PROGRAM

```

( 11) C***** PROGRAM TITLE: MD1.LU *****
(0002) C***** NOTES *****
(0003) C      ITL                      =NUMBER OF TAILS IN THE ARRAY
(0004) C      XCM(I)                 =X-POSITION OF THE CENTER OF MASS OF THE ITH MOLECUL
(0005) C                               (ANGSTROMS), I=2,4,6,...,ITL
(0006) C      VXCM(I)                =X-COMPONENT OF THE VELOCITY OF THE CENTER OF MASS O
(0007) C                               ITH MOLECULE (ANGSTROMS/SECOND), I=2,4,6,...,ITL
(0008) C      AXCM                   =X-COMPONENT OF THE ACCELERATION OF THE CENTER OF MA
(0009) C                               OF THE ITH MOLECULE (JOULES/ANGSTROM), I=2,4,6,...,
(0010) C      YCM(I), VYCM(I), AND AYCM ARE DEFINED SIMILARLY.
(0011) C      VRMS                   =AVERAGE VELOCITY VALUE ASSIGNED TO VXCM AND VYCM
(0012) C                               (ANGSTROMS/SECOND).
(0013) C      KVXSGN(I)              =ALGEBRAIC SIGN OF THE RMS VALUE ASSIGNED TO VXCM(I)
(0014) C      KVYSGN(I)              =ALGEBRAIC SIGN OF THE RMS VALUE ASSIGNED TO VYCM(I)
(0015) C      VXPDI                  =POTENTIAL AT THE POSITION OBTAINED BY ADDING DELTA
(0016) C                               X-COORDINATE OF THE ITH TAIL, I=2,4,6,...,ITL.
(0017) C      VXPDI1                 =AS ABOVE, FOR THE (I-1)TH TAIL.
(0018) C      VXMDI                  =POTENTIAL AT THE POSITION OBTAINED BY SUBTRACTING D
(0019) C                               FROM THE X-COORDINATE OF THE ITH TAIL, I=2,4,6,...,
(0020) C      VYPDI, VYMDI, VYPDI1, VYMDI1, AND VXMDI1 ARE DEFINED SIMILARLY.
(0021) C      DVDXI                  =GRADIENT OF THE POTENTIAL IN THE X-DIRECTION AT THE
(0022) C                               POSITION OF THE ITH TAIL.
(0023) C      THETA(I)               =ANGLE WHICH THE ITH MOLECULE MAKES WITH THE X-AXIS
(0024) C                               (RADIAN)
(0025) C      THEMO(I)               =MODIFIED FORM OF THETA(I) WHICH IS ALWAYS LESS THAN
(0026) C                               RADIAN
(0027) C      VTHETA(I)              =ANGULAR VELOCITY OF THE ITH MOLECULE (RADIAN/SECON
(0028) C      ATHETA                  =ANGULAR ACCELERATION OF THE ITH MOLECULE (RADIAN/S
(0029) C      TEMPX                   =TEMPERATURE CALCULATED FROM THE X-COMPONENTS OF THE
(0030) C                               VELOCITIES (DEGREES KELVIN)
(0031) C      TEMPY                   =TEMPERATURE CALCULATED FROM THE Y-COMPONENTS OF THE
(0032) C                               VELOCITIES (DEGREES KELVIN)
(0033) C      TEMPR                   =TEMPERATURE CALCULATED FROM THE ROTATIONAL VELOCITI
(0034) C                               (DEGREES KELVIN)
(0035) C      TEAR                   =TEMPERATURE CALCULATED BY SETTING 1.5KT EQUAL TO TH
(0036) C                               OF THE X, Y, AND ROTATIONAL KINETIC ENERGIES. TRAR
(0037) C                               ALSO THE ARITHMETIC AVERAGE OF TEMPX, TEMPY, AND TE
(0038) C                               (DEGREES KELVIN)
(0039) C      MCONT                  =SWITCH INDICATOR CONTROLLING INPUT OF XCM, YCM,
(0040) C                               THEMO, XCOMP, YCOMP, AND THETAP FROM TAPE. IF MCONT=
(0041) C                               THEN THESE PARAMETERS ARE NOT ACCEPTED FROM TAPE.
(0042) C      MSAVE                   =SWITCH INDICATOR CONTROLLING STORAGE OF XCM, YCM,
(0043) C                               THEMO, XCOMP, YCOMP, AND THETAP ON TAPE. IF MSAVE=0,
(0044) C                               THEN THESE PARAMETERS ARE NOT STORED.
(0045) C      NREC1                   =INDEX OF THE TAPE RECORD TO BE READ WHEN THE PROGRA
(0046) C                               IS RUN IN CONTINUATION MODE. IF MCONT=0, SET NREC1=
(0047) C      NREC2                   =INDEX OF THE TAPE RECORD PRECEDING THE RECORD TO BE
(0048) C                               WRITTEN UPON WHEN THE PROGRAM IS RUN IN SAVE MODE.
(0049) C      INTEGER*2 KAREAD,KISAMF,KSWRT,KSEXST,K&IUF0,CHRPCS(2),NTL,NMESH
(0050) C      &ESFNTF
(0051) C      LOGICAL STOPSW
(0052) C      PARAMETER (NTL=64,NMESH=1083)
(0053) C      IMPLICIT REAL*8 (A-G,O-Z),INTEGER (H-N)
( 54) C      COMMON//X1(NTL),Y1(NTL),QMESH(NMESH),DHC,DHM,Q,MESH,ITL,NALPHA,N
( 55) C      1BETA,NGAMMA,N,M,NZM,S1,S3,OMEGA,EPSIR,YH,PHIMSH(NMESH),KRSW1,KRSW2
(0056) C      2,KRSW3,KRSW4,KRSW5,KRSW6,KQSW1,KQSW2,KQSW3,KQSW4,KQSW5,KQSW6,S,SCF
(0057) C      3,GAM7,LUSW1,LUSW2,LUSW3,LUSW4,LUSW5,LUSW6,FLJFAC,FMAX,LJSW1,LJSW2,
(0058) C      4LJSW3,LJSW4,LJSW5,LJSW6,QDSAV(8,NTL)

```



```

(0060)      DIMENSION XCM(NTL),VXCM(NTL),YCM(NTL),VYCM(NTL),THETA(NTL),VTHETA(
(0061)      1NTL),AX(NTL),AY(NTL),ATH(NTL),XCMF(NTL),YCMF(NTL),THETAP(NTL),THEM
(0062)      20(NTL),      ANGLE(NTL),FMA(002),FMB(002),FMC(036),ENDFIL(50
(0063)      3),STX(1005),STY(1005),STR(1005),STB(1005),DSTB(1005),EKTOT(1005),
(0064)      4EPTOT(1005),ETOT(1005),SPI(1005),FTXCM(NTL),FTYCM(NTL),XJ(3),YJ(3)
(0065)      5,AXP(NTL),AYP(NTL),ATHP(NTL)
(0066)      DATA K$READ,ASSAMF,K$WRIT,K$RDWR,K$EXST,K$IUFD/1,0,2,3,6,0/
(0067)      DATA ESFNTF,K$CLOS/15,4/
(0068)      DATA CHRPOS/0,13/
(0069)      CALL ATTDEV(12,7,8,8)
(0070)      C      CALL OPENS$(AS$READ+ASSAMF,'MDLUDAT',7,5)
(0071)      C      CALL OPENS$(AS$WRIT+ASSAMF,'MDLUCPT',7,7)
(0072)      C      CALL OPENS$(AS$READ+ASSAMF,'ELU',3,8)
(0073)      CALL SRCH$$ (K$RDWR,'MDLUDATA',8,5,0,0)
(0074)      C      CALL SRCH$$ (K$WRIT,'MDLUOPTA',8,7,0,0)
(0075)      CALL TSRC$$ (K$READ,'MATRIX 42>ELUA',8,CHRPOS,0,0)
(0076)      TCPU1=CTIMSA(CPUTIM)
(0077)      TCPUN=TCPU1
(0078)      C***** SWITCHES *****
(0079)      C      *NAME*      *IF SWITCH=1, THEN*
(0080)      C      MSW1      WRITE J,X1(I),Y1(I),THEM(I)
(0081)      C      MSW2      WRITE FORCES, VTHETA, & ATHMOD
(0082)      C      MSW3      WRITE DTHEV, DTHEVN, DTHEA, & DTHEAN
(0083)      C      MSW4      INCLUDE L-J FORCES
(0084)      C      MSW5      WRITE POSITIONS & VELOCITIES AFTER EACH TIME STEP
(0085)      C      MSW6      REVISE DATA VALUES FOR NEXT RUN FROM TERMINAL
(0086)      C      MSW7      WRITE I,FXI,FYI,FLJXI,FLJYI
(0087)      C      MSW8      USE V(N)=V(N-1)+DT*A(N-1) IN PLACE OF
(0088)      C      V(N)=(X(N)-X(N-1))/DT
(0089)      C      IF MSW8 = -1, USE ALTERNATE DYNAMIC ALGORITHM
(0090)      C      APPLY TRANSLATIONAL AND ROTATIONAL FRICTION
(0091)      C      MSW9      ENABLE FORCES (SEE LOOP 480)
(0092)      C      MSW10     MODIFY VELOCITIES ACCORDING TO TEMPERATURE
(0093)      C      PARTITION (AFTER 370)
(0094)      C      MSW11     MODIFY VELOCITIES TO FORCE TBAR TO LIE
(0095)      C      BETWEEN TBARMX AND TBARMN (AFTER 370)
(0096)      C      MSW12     USE SUBROUTINE INIT32 IN PLACE OF INIT
(0097)      C      IF MSW13 = -1, USE SUBROUTINE INITSP
(0098)      C      MSW13     COMPUTE COULOMBIC FORCES IN LJFCR2
(0099)      C      RATHER THAN LUDEC.
(0100)      C      MCONDS     READ INITIAL DATA FOR NDT.GT.0 FROM DISK (MDLUDAT)
(0101)      C      MCONTPT    READ INITIAL DATA FOR NDT.GT.0 FROM TAPE
(0102)      C      MSAVDS     WRITE FINAL DATA FOR NDT=NDTMAX TO DISK (MDLUDAT)
(0103)      C      MSAVTP     WRITE FINAL DATA FOR NDT=NDTMAX TO TAPE
(0104)      C      I$UFSW     MODIFY DT VIA DT=DT*DTFAC
(0105)      C***** CONTROLS *****
(0106)      C      NCYCLE     NUMBER OF TIMES A CYCLE OF NDTMAX STEPS HAS BEEN
(0107)      C      RUN AND WRITTEN TO DISK (MDLUDAT)
(0108)      C      NDTOFF     TIME STEP AT WHICH VELOCITY CONTROL IS DISABLED
(0109)      C      NDTINC     NUMBER OF TIME STEPS FOR WHICH VELOCITY CONTROL
(0110)      C      IS RE-ENABLED
(0111)      C***** READ DATA FILE *****
(0112)      C      READ(09,0801) NDTED1,NDTED2,NDTED3,MCONTPT,MSAVTP,NREC1,NREC2,RAN
(0113)      1FAX,RANFAY,RANFAT,KXMAX,KYMAX,KTMAX,IPRINT,JPRINT,KPRINT
(0114)      C      READ(09,938) (FMC(I),I=1,36)
(0115)      C      READ(09,0014) (FMA(I),I=68,102,2)
(0116)      C      READ(09,0014) (FMB(I),I=68,102,2)

```

```

( 7) C READ(09,0018) (FMC(I),I=104,142,2) 0
(0118) C READ(09,0018) (FMA(I),I=104,142,2) 0
(0119) C READ(09,0018) (FMB(I),I=104,142,2) 0
(0120) READ(09,0021) A,B,Q,PLMASS,EPSI 0
(0121) READ(09,0023) DEPTH,DIST,DELTA,ZAMBDA,RLJMAX,DT,NDTMAX 0
(0122) READ(09,0025) ITL,IMX,IMY,MESH,DHC,DHM 1
(0123) READ(09,910) NALPHA,NBETA,NGAMMA,S1,S3,OMEGA 1
(0124) READ(09,923) EPSIR,FMAX,SPCFAC,FLJFAC,VSCFAC,FRICT,FRICR 1
(0125) READ(09,927) MSW1,MSW2,MSW3,MSW4,MSW5,MSW6,MSW7,MSW8,MSW9,MSW10,1
(0126) &MSW11,MSW12,MSW13,MSW14
(0127) READ(09,911) KRSW1,KRSW2,KRSW3,KRSW4,KRSW5,KRSW6 1
(0128) READ(09,911) KQSW1,KQSW2,KQSW3,KQSW4,KQSW5,KQSW6 1
(0129) READ(09,911) LUSW1,LUSW2,LUSW3,LUSW4,LUSW5,LUSW6 1
(0130) READ(09,911) LJSW1,LJSW2,LJSW3,LJSW4,LJSW5,LJSW6 1
(0131) READ(09,912) TBARMX,TBARMN,TFACMX,TFACMN,NDTOFF,NDTINC 1
(0132) READ(09,929) DTFAC,BUFFAC,NDTDT1,NDTDT2,NDTLJ 1
(0133) READ(09,911) MCONDS,MSAVDS,NCYCLE 2
(0134) RNDATE=DATE$(DATE)
(0135) ONTIME=TIME$(TIME)
(0136) WRITE(11,802)
(0137) WRITE(11,936) RNDATE,ONTIME
(0138) C***** WRITE DATA FILE *****
(0139) 15 WRITE(11,808)
(0140) WRITE(11,801) NDTE01,NDTE02,NDTE03,MCONTP,MSAVTP,NREC1,NREC2,RAN
(0141) 1FAX,RANFAY,RANFAT,KXMAX,KYMAX,KTMAX,IPRINT,JPRINT,KPRINT 0
(0142) WRITE(11,938) (FMC(I),I=1,36) 0
(0143) C WRITE(11,0014) (FMA(I),I=68,102,2) 0
(0144) C WRITE(11,0014) (FMB(I),I=68,102,2) 0
(0145) C WRITE(11,0018) (FMC(I),I=104,142,2) 0
(0146) C WRITE(11,0018) (FMA(I),I=104,142,2) 0
(0147) C WRITE(11,0018) (FMB(I),I=104,142,2) 0
(0148) WRITE(11,0021) A,B,Q,PLMASS,EPSI 0
(0149) WRITE(11,0023) DEPTH,DIST,DELTA,ZAMBDA,RLJMAX,DT,NDTMAX 0
(0150) WRITE(11,0025) ITL,IMX,IMY,MESH,DHC,DHM 1
(0151) WRITE(11,910) NALPHA,NBETA,NGAMMA,S1,S3,OMEGA 1
(0152) WRITE(11,923) EPSIR,FMAX,SPCFAC,FLJFAC,VSCFAC,FRICT,FRICR 1
(0153) WRITE(11,927) MSW1,MSW2,MSW3,MSW4,MSW5,MSW6,MSW7,MSW8,MSW9,MSW10,1
(0154) &MSW11,MSW12,MSW13,MSW14
(0155) WRITE(11,911) KRSW1,KRSW2,KRSW3,KRSW4,KRSW5,KRSW6 1
(0156) WRITE(11,911) KQSW1,KQSW2,KQSW3,KQSW4,KQSW5,KQSW6 1
(0157) WRITE(11,911) LUSW1,LUSW2,LUSW3,LUSW4,LUSW5,LUSW6 1
(0158) WRITE(11,911) LJSW1,LJSW2,LJSW3,LJSW4,LJSW5,LJSW6 1
(0159) WRITE(11,912) TBARMX,TBARMN,TFACMX,TFACMN,NDTOFF,NDTINC 1
(0160) WRITE(11,929) DTFAC,BUFFAC,NDTDT1,NDTDT2,NDTLJ 1
(0161) WRITE(11,911) MCONDS,MSAVDS,NCYCLE 2
(0162) C***** CONSTANTS *****
(0163) N=NALPHA*NBETA*NGAMMA
(0164) M=NBETA*NGAMMA
(0165) NZM=(NGAMMA+1)/2
(0166) DMOLEC=DBLE(FLQAT(ITL/2))
(0167) DNAVOG=6.020+23
(0168) SIZ*SH=DHM*(MESH-1)
(0169) YH=DHM
( 170) XSPACE=SPCFAC*SIZMSH/(IMX/2)
( 171) YSPACE=SPCFAC*SIZMSH/IMY
(0172) AMESH=SIZMSH*SIZMSH
(0173) RH=A+(IMY-1)*(A+P*0.50+00)
(0174) W=(IMX/2-1)*0.17320510+01*B

```

```

75      DELTAN=-DELTA
76      STB(0)=0.00+00
77      TWOPI=0.62831853070+01
78      AD2=A*0.50+00
79      FCOR=0.23019020-17/(EPSIR*A*A)
80      PLINER=PLMASS*(0.50+00*A)**0.20+01      /*KG.SQ.ANG.*/
81      FRTDM=FRIC/PLMASS                      /*KG PER SEC OVER KG*/
82      FRRDI=FRICR/PLINER                      /*KG.SQ.ANG. PER SEC OVER KG.SQ.ANG.*/
83      DXLJ=0.10-01*DHM
84      DYLJ=0.10-01*DHM
85      TB*XP=TBARMX*0.1250+01
86      BUFFZN=DHM*BUFFAC
87      SAFEZN=SI2MSH-BUFFZN
88      DTORIG=DT
89      DTSQD2=DT*DT*0.50+00
90      KWPIT=0
91      NWRIT=0
92      KSTINI=0
93      JSTINI=0
94      NCTBMN=0
95      NCTBMX=0
96      C***** SWITCH OVERRIDES
97      IF(MSW14.EQ.1) LJSW1=1
98      C***** INITIALIZE APRAYS
99      DO 20 I=1,ITL
100      AX(I)=0.00+00
101      AY(I)=AX(I)
102      ATH(I)=AX(I)
103      AXP(I)=AX(I)
104      AYP(I)=AX(I)
105      20  ATHP(I)=AX(I)
106      C***** READ INITIAL AND PRIOR POSITIONS FROM TAPE (OPTIONAL)
107      IF (MCONTP.EQ.0) GO TO 30
108      DO 27 L=1,NREC1
109      READ (3) XCM,YCM,THETA,THEMO
110      READ (3) XCMP,YCMP,THETAP
111      READ (3,END=27) NDT
112      27  CONTINUE
113      NDT1=NDT+1
114      WRITE (11,0028) NDT
115      WRITE (11,0029) (I,XCM(I),YCM(I),THEMO(I),THETA(I),I=2,142,2)
116      30 IF(MCONDS.EQ.0) GO TO 33
117      C***** READ INITIAL AND PRIOR POSITIONS FROM DISK (OPTIONAL)
118      DO 32 K=1,NCYCLE
119      READ(09,930) CRDDAT,NDT
120      PRINT 931 CRDDAT,NDT
121      DO 31 I=2,ITL,2
122      READ(09,932) XCM(I),YCM(I),THETA(I),THEMO(I)
123      READ(09,932) XCMP(I),YCMP(I),THETAP(I)
124      READ(09,932) AX(I),AY(I),ATH(I)
125      READ(09,932) AXP(I),AYP(I),ATHP(I)
126      31  READ(09,932) VXCM(I),VYCM(I),VTHETA(I)
127      32  READ(09,932) TBAR,TFAC,TFAC,TFAC
128      TEMPX=TFAC*TBAR
129      TEMPY=TFAC*TBAR
130      TEMPR=TFAC*TBAR
131      VCORX1=VSCFAC*DSQRT(TBARMN/TEMPX)
132      VCORY1=VSCFAC*DSQRT(TBARMN/TEMPY)

```

```

023      VCORR1=VSCFAC*DSQRT(TBARMN/TEMPR)
0234     VCORP2=VSCFAC*DSQRT(TBARMX/TEMPX)
0235     VCCRY2=VSCFAC*DSQRT(TBARMX/TEMPY)
0236     VCORR2=VSCFAC*DSQRT(TBARMX/TEMPR)
0237     WRITE(11,930) CRDDAT,NDT
0238     PRINT 931 CRDDAT,NDT
0239     DO 931 I=2,ITL,2
0240     WRITE(11,932) XCM(I),YCM(I),THETA(I),THEMO(I)
0241     WRITE(11,932) XCMP(I),YCMP(I),THETAP(I)
0242     9031 WRITE(11,932) VXCM(I),VYCM(I),AX(I),AY(I)
0243     GO TO 0110
C***** SET INITIAL POSITIONS AND VELOCITIES FOR CENTERS OF MASS
0244     33 IF (MSW13) 34,36,38
0245     34 CALL INITSP(XCM,YCM,THETA,VXCM,VYCM,VTHETA,THEMC,FMC)
0246     GO TO 85
0247     36 CALL INIT(IMX,IMY,A,B,AD2,XSPACE,YSACE,XCM,YCM,THETA,VXCM,VYCM,
0248     &VTHETA,THEMO)
0249     GO TO 85
0250     38 CALL INIT32(XCM,YCM,THETA,VXCM,VYCM,VTHETA,THEMC,A)
0251     C***** RANDOMIZE INITIAL POSITIONS
0252     85 PRINT 919
0253     90 CALL RANLOC (XCM,YCM,A,B,THETA,TWOPI,RANFAX,RANFAY,RANFAT)
0254     95 DO 9096 I=2,ITL,2
0255     96 THEMO(I)=THETA(I)
0256     100 NDTP1=1
0257     101 NDT=0
0258     NDTINI=NDT
0259     C***** CALCULATE XCM(I), YCM(I), AND THETA(I) FOR EACH MOLECULE AFTER A
0260     C TIME INCREMENT OF DT HAS ELAPSED.
0261     C * STATEMENT 110 IS THE RE-ENTRY POINT FOR ADDITIONAL TIME STEPS;
0262     C * STATEMENT 591 DIRECTS THE RETURN TO 110.
0263     C * SET IBUFSW AND DT TO ORIGINAL VALUES.
0264     110 IBUFSW=0
0265     DT=DTORIG
0266     MNDTIP=MOD(NDT,IPRINT)
0267     C * CHECK FOR EXISTENCE OF FILE MDLU.CHK; IF IT EXISTS, WRITE
0268     C THE VALUE OF NDT THERE. THIS ALLOWS ONE TO LEARN THE CURRENT
0269     C STATE OF THE SIMULATION WHEN IT IS RUNNING IN BATCH MODE.
0270     CALL SRCH$(K$EXST+K$IUFD,'MDLU.CHK',8,0,0,ACHECK)
0271     IF(NCHECK.EQ.EBNTF) GO TO 115
0272     CALL SRCH$(K$WRIT,'MDLU.CHK',8,9,0,0)
0273     WRITE(13,921) NDT
0274     CALL SRCH$(K$CLOS,'MDLU.CHK',8,9,0,0)
0275     C * FIND POSITIONS OF FORCE CENTERS FROM C-M AND
0276     C ANGULAR POSITIONS.
0277     115 DO 160 I=2,ITL,2
0278     AD2COS=AD2*DCOS(THEMO(I))
0279     AD2SIN=AD2*DSIN(THEMO(I))
0280     X1(I)=XCM(I)+AD2COS
0281     Y1(I)=YCM(I)+AD2SIN
0282     X1(I-1)=XCM(I)-AD2COS
0283     Y1(I-1)=YCM(I)-AD2SIN
0284     C *SET IBUFSW TO ONE IF ANY FORCE CENTER LIES WITHIN BUFFZN
0285     C OF A BOUNDARY.
0286     IF((X1(I).LE.BUFFZN).OR.(X1(I).GE.SAFEZN)) IBUFSW=1
0287     IF((Y1(I).LE.BUFFZN).OR.(Y1(I).GE.SAFEZN)) IBUFSW=1
0288     IF((X1(I-1).LE.BUFFZN).OR.(X1(I-1).GE.SAFEZN)) IBUFSW=1
0289     IF((Y1(I-1).LE.BUFFZN).OR.(Y1(I-1).GE.SAFEZN)) IBUFSW=1
0290

```

```

91      160      CONTINUE
92)    C      * MODIFY DT IF IBUGSW HAS BEEN SET TO ONE ABOVE.
93)      IF(BUGSW.EQ.0) GO TO 165
94)      DT=DT*DTFAC
95)      WRITE(11,935) DT,NDT
96)    C      * REACTIVATE VELOCITY CONTROL IF IT HAS BEEN DISABLED
97)    C      AND TBAR HAS EXCEEDED TBARMX BY 25 PERCENT.
98)    C 165 IF((NDT.GE.NDTOFF).AND.(TBAR.GT.TBMP)) NDTOFF=NDT+NDTINC
99)      165 CONTINUE
100)   C      * PRINT PHOTO AT SELECTED TIMES.
101)     PRINT 922 NDT
102)     IF(MNDTIP.NE.0) GO TO 168
103)     CALL PHOTO (NDT)
104)     WRITE(11,937) NCTBMN,NCTBMX
105)   C      * CHANGE VALUE OF DT AT NDTDT1 AND NDTDT2
106)   168 IF((NDT.EQ.NDTDT1).OR.(NDT.EQ.NDTDT2)) DT=DT*DTFAC
107)   C      * CALL QDIST AND LUDEC
108)     PRINT 915
109)     CALL QDIST1
110)     PRINT 916
111)     IF(MSW14.EQ.0) CALL LUDEC3
112)     LUSW1=1
113)   C      * BASIC DYNAMIC CALCULATION (DO-LOOP 480)
114)     PCTOT=0.00+00          /* SET PE COUNTERS TO ZERO. */
115)     PLJTOT=0.00+00
116)     PCTOTD=0.00+00
117)   180 DO 480 I=2,ITL,2
118)     IM1=I-1
119)   C      *ZERO ALL FORCES AND POTENTIAL GRADIENTS
120)     FXI=0.00+00
121)     FYI=0.00+00
122)     FLJXI=0.00+00
123)     FLJYI=0.00+00
124)     DVDXI=0.00+00
125)     DVDYI=0.00+00
126)     DVDXI1=0.00+00
127)     DVDYI1=0.00+00
128)     IF(MSW10.EQ.0) GO TO 250
129)     IF ((MSW1.EQ.0).OR.(MOD(NDT,IPRINT).GT.0)) GO TO 200
130)     WRITE (11,9186) IM1,X1(IM1),Y1(IM1),THEMO(IM1)
131)     WRITE (11,9186) I,X1(I),Y1(I),THEMO(I)
132)   C      *CALCULATE DVDX AND DVDY FOR I-1.
133)   200 IF(MSW14.EQ.1) GO TO 210
134)     CALL FORCE1 (IM1,FXI,FYI,SIZMSH,PCAVI,PCTOT)
135)     FXI=FXI-FCOR*DCOS(THEMO(IM1)) /* BROTHOR-CHARGE CORRECTION
136)     FYI=FYI-FCOR*DSIN(THEMO(IM1)) /* BROTHOR-CHARGE CORRECTION
137)   210 IF((MSW4.EQ.1).AND.(NDT.GE.NDTLJ)) CALL LJFCR2(IM1,DXLJ,DYIJ,
138)     &DEPTH,DIST,RLJMAX,FLJXI,FLJYI,FCXI,FCYI,SIZMSH,PLJAVI,PLJTOT,PCAVI
139)     &D,PCTOTD)
140)     IF(MSW14.EQ.0) GO TO 215
141)     FXI=FCXI
142)     FYI=FCYI
143)     PCAVI=PCAVID
144)     PCTOT=PCTOTD
145)   215 IF((MSW7.EQ.1).AND.(MNDTIP.EQ.0)) WRITE(11,995) I,FXI,FYI,FLJ
146)     &XI,FLJYI,FCXI,FCYI
147)     DVDXI1=FXI+FLJXI /*JOULES PER ANGSTROM*/
148)     DVDYI1=FYI+FLJYI
  
```

```

03      C      *CALCULATE DVDX AND DVDY FOR I.
0350)      IF(MSW14.EQ.1) GO TO 220
0351)          CALL FORCE1 (I,FXI,FYI,SIZMSH,PCAVI,PCTOT)
0352)          FXI=FXI-FCOR*DCOS(THEMO(I)) /* BROTHER-CHARGE CORRECTION */
0353)          FYI=FYI-FCCR*DSIN(THEMO(I)) /* BROTHER-CHARGE CORRECTION */
0354)  220      IF((MSW4.EQ.1).AND.(NDT.GE.NDTLJ)) CALL LJFOR2(I,DXLJ,DYIJ,DE
0355)          &PTH,DIST,RLJMAX,FLJXI,FLJYI,FCXI,FCYI,SIZMSH,PLJAVI,PLJTOT,PCAVI,
0356)          &PCTOTD)
0357)          IF(MSW14.EQ.0) GO TO 225
0358)          FXI=FCXI
0359)          FYI=FCYI
0360)          PCAVI=PCAVI
0361)          PCTOT=PCTOTD
0362)  225      IF((MSW7.EQ.1).AND.(MNDTIP.EQ.0)) WRITE(11,995) I,FXI,FYI,FLJ
0363)          &XI,FLJYI,FCXI,FCYI
0364)          DVDXI=FXI+FLJXI /*JOULES PER ANGSTROM*/
0365)          DVDYI=FYI+FLJYI
0366)          FTXCM(IM1)=DVDXI1
0367)          FTXCM(I)=DVDXI
0368)          FTYCM(IM1)=DVDYI1
0369)          FTYCM(I)=DVDYI
0370)  C      *SAVE ACCELERATIONS
0371)          AXP(I)=AX(I)
0372)          AYP(I)=AY(I)
0373)          ATHP(I)=ATH(I)
0374)          AX(I)=AXCM
0375)          AY(I)=AYCM
0376)  C      *CALCULATE ATHETA IN TWO STEPS TO AVOID FLOATING POINT OVERFLOW
0377)  C      *PLINER HAS BEEN EXPRESSED IN KG*SQUARE ANGSTROM UNITS
0378)  C      *POSITIVE TORQUES ARE TAKEN TO BE THOSE WHICH CAUSE CCW ROTATION
0379)          ATHMOD=(0.1D+11*A/(0.2D+01*PLINER))*((DVDYI1-DVDYI)+DCOS(THEM
0380)          10(I))+(DVDXI-DVDXI1)*DSIN(THEMO(I)))
0381)          ATHETA=ATHMOD*0.1D+11
0382)          AXCM=(((-DVDXI-DVDXI1)/PLMASS)*0.1D+21 /*ANG. PER SQ.SEC.*/
0383)          AYCM=(((-DVDYI-DVDYI1)/PLMASS)*0.1D+21
0384)  C      *APPLY TRANSLATIONAL AND ROTATIONAL FRICTION
0385)          IF(MSW9.EQ.0) GO TO 330
0386)          AXCM=AXCM-FRTDM*VXCM(I)
0387)          AYCM=AYCM-FRTDM*VYCM(I)
0388)          ATHETA=ATHETA-FRRDI*VTHETA(I)
0389)  C      *CALCULATE VELOCITIES (ANGSTROMS PER SECOND, RADIANS PER SECOND
0390)  330      IF (NDT.EQ.0) GO TO 412
0391)          IF (MSW8) 340,360,350
0392)  340      VXCM(I)=VXCM(I)+DT*(0.2D+01*AXCM+0.5D+01*AX(I)-AXP(I))
0393)          VYCM(I)=VYCM(I)+DT*(0.2D+01*AYCM+0.5D+01*AY(I)-AYP(I))
0394)          VTHETA(I)=VTHETA(I)+DT*(0.2D+01*ATHETA+0.5D+01*ATH(I)-ATHP(I
0395)          &))
0396)          GO TO 370
0397)  350      VXCM(I)=VXCM(I)+DT*AXCM
0398)          VYCM(I)=VYCM(I)+DT*AYCM
0399)          VTHETA(I)=VTHETA(I)+DT*ATHETA
0400)          GO TO 370
0401)  360      VXCM(I)=(XCM(I)-XCMP(I))/DT
0402)          VYCM(I)=(YCM(I)-YCMP(I))/DT
0403)          VTHETA(I)=(THETA(I)-THETAP(I))/DT
0404)  370      IF((NDT.LE.(NDTE01+2)).OR.(NDT.GE.NDTGFF)) GO TO 385
0405)          IF(MSW11.EQ.0) GO TO 375
0406)  C      *MODIFY VELOCITIES ACCORDING TO TEMPERATURE PARTITION

```

```

(  37)  C      *AT LAST TIME STEP
(0408)      IF ((TXFAC.GT.TFACMX).OR.(TXFAC.LT.TFACMN)) VXCM(I)=VXCM(I)/T
(0409)      1XFAC
(0410)      IF ((TYFAC.GT.TFACMX).OR.(TYFAC.LT.TFACMN)) VYCM(I)=VYCM(I)/T
(0411)      1YFAC
(0412)      IF ((TRFAC.GT.TFACMX).OR.(TRFAC.LT.TFACMN)) VTHETA(I)=VTHETA(
(0413)      1I)/TRFAC
(0414)      375      IF(MSW12.EQ.0) GO TO 385
(0415)  C      *MODIFY VELOCITIES TO FORCE TBAR TO LIE BETWEEN
(0416)  C      *TBARMX AND TBARMN
(0417)      IF(TBAR.GE.TBARMN) GO TO 380
(0418)      VXCM(I)=VXCM(I)*VCORX1
(0419)      VYCM(I)=VYCM(I)*VCORY1
(0420)      VTHETA(I)=VTHETA(I)*VCORR1
(0421)      IF(I.NE.2) GO TO 380
(0422)      NCTBMN=NCTBMN+1
(0423)      WRITE(11,933) NDT,VCORX1,VCORY1,VCORR1
(0424)      380      IF(TBAR.LE.TBARMX) GO TO 385
(0425)      VXCM(I)=VXCM(I)*VCORX2
(0426)      VYCM(I)=VYCM(I)*VCORX2
(0427)      VTHETA(I)=VTHETA(I)*VCORR2
(0428)      IF(I.NE.2) GO TO 385
(0429)      NCTBMX=NCTBMX+1
(0430)      WRITE(11,934) NDT,VCORX2,VCORY2,VCORR2
(0431)  C      *SET VELOCITIES TO ZERO AT CERTAIN TIME STEPS
(0432)      385      NDTPP=NDT+1
(0433)      IF((NDTPP.EQ.NDTE01).OR.(NDTPP.EQ.NDTE02).OR.(NDTPP.EQ.NDTE03
(0434)      1)) GO TO 410
(0435)      GO TO 412
(0436)      410      VXCM(I)=0.00+00
(0437)      VYCM(I)=0.00+00
(0438)      VTHETA(I)=0.00+00
(0439)  C      *STORE VALUES OF THETA(I), XCM(I), AND YCM(I) BEFORE CALCULATING
(0440)      412      THETAP(I)=THETA(I)
(0441)      XCMP(I)=XCM(I)
(0442)      YCMP(I)=YCM(I)
(0443)  C      *CALCULATE NEW C-M POSITION
(0444)      AXDUM=AXCM
(0445)      AYDUM=AYCM
(0446)      ATHDUM=ATHETA
(0447)      IF (*SW8.GE.0) GO TO 460
(0448)      AXDUM=(0.40+01*AXCM-AX(I))/0.30+01
(0449)      AYDUM=(0.40+01*AYCM-AY(I))/0.30+01
(0450)      ATHDUM=(0.40+01*ATHETA-ATH(I))/0.30+01
(0451)      460      XCM(I)=XCM(I)+(DT*VXCM(I))+AXDUM*DTSQD2
(0452)      YCM(I)=YCM(I)+(DT*VYCM(I))+AYDUM*DTSQD2
(0453)      IF(MSW2.EQ.1) WRITE(11,9423) (I,DVDXI,DVDYI,DVDXI1,DVDYI1,VTH
(0454)      RETA(I),ATHMOD)
(0455)      DTHEV=DT*VTHETA(I)
(0456)      DTHEVN=DMOD(DTHEV,TWOPI)
(0457)      DTHEA=ATHDUM*DTSQD2
(0458)      DTHEAN=DMOD(DTHEA,TWOPI)
(0459)      IF(MSW3.EQ.1) WRITE (11,9429) (DTHEV,DTHEVN,DTHEA,DTHEAN)
(  38)      THETA(I)=THETAP(I)+DTHEVN+DTHEAN
(  39)      THEM0(I)=DMOD(THETA(I),TWOPI)
(0462)      ANGLE(I)=THEM0(I)*0.5729577950+02
(0463)  C      *REPOSITION MOLECULES WHICH HAVE C-M OUTSIDE CENTRAL ARRAY
(0464)      IF(XCM(I).GE.0.00+00) GO TO 470

```

```

(   5)      XCM(I)=XCM(I)+SIZMSH
(   6)      XCMP(I)=XCMP(I)+SIZMSH
(0467)      GO TO 472
(0468)      470  IF((XCM(I)-SIZMSH).LE.0.0D+00) GO TO 472
(0469)      XCM(I)=XCM(I)-SIZMSH
(0470)      XCMP(I)=XCMP(I)-SIZMSH
(0471)      472  IF(YCM(I).GE.0.0D+00) GO TO 474
(0472)      YCM(I)=YCM(I)+SIZMSH
(0473)      YCMP(I)=YCMP(I)+SIZMSH
(0474)      GO TO 480
(0475)      474  IF((YCM(I)-SIZMSH).LE.0.0D+00) GO TO 480
(0476)      YCM(I)=YCM(I)-SIZMSH
(0477)      YCMP(I)=YCMP(I)-SIZMSH
(0478)      C 479  PRINT 905 I,XCM(I),YCM(I)
(0479)      480  CONTINUE
(0480)      C***** UPDATE NDT, THE NUMBER OF TIME INCREMENTS WHICH HAVE ELAPSED
(0481)      490 NDT=NDT+1
(0482)      C***** CALCULATE TEMPERATURES, DIFFERENCES & CORRECTION FACTORS
(0483)      PRINT 918
(0484)      TEMPX=0.0D+00
(0485)      TEMPY=0.0D+00
(0486)      SVXSQ=0.0D+00
(0487)      SVYSQ=0.0D+00
(0488)      DO 508 I=2,ITL,2
(0489)          SVXSQ=SVXSQ+(VXCM(I)**2) /*SQ.ANG. PER SQ.SEC.*/
(0490)      508  SVYSQ=SVYSQ+(VYCM(I)**2)
(0491)      TEMPX=(PLMASS*SVXSQ)/(0.138044D-02*ITL/2)
(0492)      TEMPY=(PLMASS*SVYSQ)/(0.138044D-02*ITL/2)
(0493)      SOMSQ=0.0D+00
(0494)      DO 512 I=2,ITL,2
(0495)      512  SOMSQ=SOMSQ+(VTHETA(I)**2)
(0496)      TEMPR=(PLINER*SOMSQ)/(0.69022D-03*ITL)
(0497)      TBAR=(TEMPX+TEMPY+TEMPR)/0.3D+01
(0498)      IF((NDT.LE.(NDT01+1)).OR.(TBAR.LT.0.1D-10)) GO TO 515
(0499)      TXFAC=TEMPX/TBAR
(0500)      TYFAC=TEMPY/TBAR
(0501)      TRFAC=TEMPR/TBAR
(0502)      IF(MSW12.EQ.0) GO TO 515
(0503)          VCORX1=VSCFAC*DSQRT(TBARMN/TEMPX)
(0504)          VCORY1=VSCFAC*DSQRT(TBARMN/TEMPY)
(0505)          VCORR1=VSCFAC*DSQRT(TBARMN/TEMPR)
(0506)          VCORX2=VSCFAC*DSQRT(TBARMX/TEMPX)
(0507)          VCORY2=VSCFAC*DSQRT(TBARMX/TEMPY)
(0508)          VCORR2=VSCFAC*DSQRT(TBARMX/TEMPR)
(0509)      515  NDTDEL=NDT-NDTINI
(0510)      STX(NDTDEL)=TEMPX
(0511)      STY(NDTDEL)=TEMPY
(0512)      STR(NDTDEL)=TEMPR
(0513)      STB(NDTDEL)=TBAR
(0514)      DSTB(NDTDEL)=STB(NDTDEL)-STB(NDTDEL-1)
(0515)      C***** COMPUTE KINETIC, POTENTIAL, AND TOTAL ENERGIES (JOULES)
(0516)      EKTOT(NDTDEL)=0.15D+01*DMOLEC*0.138044D-22*TBAR
(0517)      EPTOT(NDTDEL)=PLJTOT+0.16D-18*PCTOT
(0518)      C * CONVERT ENERGIES TO KILOJOULES PER MOLE
(0519)      EKTOT(NDTDEL)=EKTOT(NDTDEL)/DMOLEC * DNAVOG
(0520)      EPTOT(NDTDEL)=EPTOT(NDTDEL)/DMOLEC * DNAVOG
(0521)      ETOT(NDTDEL)=EKTOT(NDTDEL)+EPTOT(NDTDEL)
(0522)      C***** FIND SYSTEM CENTER-OF-MASS AND CALCULATE P*V IN ERGS

```



```

23)      CALL PRESS1 (XCM,YCM,CMXSYS,CMYSYS,TBAR,PVERG,FTXCM,FTYCM)
(0524) C***** CALCULATE SURFACE PRESSURE AND AREA PER MOLECULE
(0525)      PIPRES=PVERG/AMESH          /* ERGS PER SQ.ANG. */
(0526)      SPI(NDTDEL)=PIPRES*1.0D+16    /* DYNES PER CM. */
(0527)      AMOLEC=AMESH/DMOLEC
(0528) C***** WRITE PARAMETERS AND INTERMEDIATE RESULTS
(0529)      IF (KWRT .GT. 0) GO TO 0533
(0530)      IF (NDT .GT. NDTP1) GO TO 533
(0531)      WRITE (11,808)
(0532)      WRITE (11, 521) A,B,Q,PLMASS,PLINER,EPSI
(0533)      522 WRITE (11, 523) DEPTH,DIST,DELTA,ZAMBDA,RMAX,DT,NDTMAX
(0534)      528 WRITE (11, 529) ITL,IMX,IMY
(0535)      530 WRITE (11, 531) RANFX0,RANFY0,RANFT0,MCONT,MSAVE,NREC1,NREC2
(0536)      533 IF(MSW5.EQ.0) GO TO 591
(0537)      IF(NDT.EQ.1) GO TO 534
(0538)      IF(MOD(NDT,IPRINT).GT.0) GO TO 591
(0539)      534 WRITE (11, 537) NDT,DT
(0540)      WRITE (11, 539)
(0541)      WRITE (11, 541) (I,XCM(I),YCM(I),THETA(I),THEMO(I),ANGLE(I),I=2,IT
(0542)      1L,2)
(0543)      WRITE (11, 545)
(0544)      WRITE (11, 547) (I,VXCM(I),VYCM(I),VTHETA(I),I=2,ITL,2)
(0545)      WRITE (11, 552) TEMPX,TEMPY,TEMPT,TBAR,PVERG,CMXSYS,CMYSYS,PIPRES,
(0546)      1AMOLEC
(0547)      591 IF(MOD(NDT,JPRINT).GT.0) GO TO 593
(0548)      KSTINI=JSTINI+JPRINT+1
(0549)      JSTINI=JSTINI+1
(0550)      WRITE (11,994)
(0551)      DO 592 K=KSTINI,NDTDEL,KPRINT
(0552)      KPNDTI=K+NDTINI
(0553)      592 WRITE(11,995) KPNDTI,EKTOT(K),EPTOT(K),ETOT(K),STB(K),DSTB(K),
(0554)      &SPI(K)
(0555) C***** ALLOW SYSTEM TO EVOLVE FOR ANOTHER TIME INCREMENT IF NDT HAS
(0556) C * NOT REACHED ITS MAXIMUM. STOP IF FILE MDLU.STOP EXISTS.
(0557)      593 CALL SRCH33(KSEXST+K$IUFD,'MDLU.STOP',9,0,0,NSTCP)
(0558)      IF(NSTOP.NE.E$ENTF) GO TO 594
(0559)      IF (NDT.LT.NDTMAX) GO TO 110
(0560) C***** WRITE FINAL RESULTS
(0561) C * PRINT PHOTO OF POSITIONS AT PENULTIMATE TIME-STEP
(0562)      594 CALL PHOTO (NDT-1)
(0563)      WRITE (11,996)
(0564)      WRITE (11,994)
(0565)      DO 596 K=1,NDTDEL,KPRINT
(0566)      KPNDTI=K+NDTINI
(0567)      596 WRITE(11,995) KPNDTI,EKTOT(K),EPTOT(K),ETOT(K),STB(K),DSTB(K),SP
(0568)      &I(K)
(0569) C IF (NWRIT.GT.0) GO TO 598
(0570) C WRITE (11,597)
(0571) C 598 WRITE (11,599) RANFAX,RANFAY,RANFAT,STX(NDT),STY(NDT),STR(NDT),ST
(0572) C 1B(NDT),STXMY(NDT),STYMR(NDT),STXMR(NDT)
(0573) C NWRIT=1
(0574)      600 IF (MSAVTP.EQ.0) GO TO 615
(0575) C***** WRITE FINAL AND PRIOR POSITIONS TO TAPE (OPTIONAL)
(0576)      601 IF (NREC1 .EQ. NREC2) GO TO 610
(0577)      602 DO 0606 L=NREC1,NREC2
(0578)      READ (3) DUM1,DUM2,DUM3,DUM4
(0579)      READ (3) DUM5,DUM6,DUM7
(0580)      READ (3,END= 606) DUM8

```

```

      81)      606      CONTINUE
(0582)      610 WRITE (3) XCM,YCM,THETA,THEMO
(0583)      WRITE (3) XCMP,YCMP,THETAP
(0584)      WRITE (3) NDT
(0585)      615      IF(MSAVDS.EQ.0) GO TO 619
(0586) C***** WRITE INITIAL AND PRIOR POSITIONS TO DISK (OPTIONAL)
(0587)      CRDATE=DATE$(DATE)
(0588)      WRITE(09,930) CRDATE,NDT,JSTINI
(0589)      DO 617 I=2,ITL,2
(0590)          WRITE(09,932) XCM(I),YCM(I),THETA(I),THEMO(I)
(0591)          WRITE(09,932) XCMP(I),YCMP(I),THETAP(I)
(0592)          WRITE(09,932) AX(I),AY(I),ATH(I)
(0593)          WRITE(09,932) AXP(I),AYP(I),ATHP(I)
(0594)      617      WRITE(09,932) VXCM(I),VYCM(I),VTHETA(I)
(0595)      WRITE(09,932) TBAR,TFAC,TFAC,TRFAC
(0596)      619      KWRIT=1
(0597)      TCPUF=CTIMSA(CPUTIM)
(0598)      TCPU=TCPUF-TCPU1
(0599)      PRINT 904 TCPU
(0600)      OFTIME=TIMESA(TIME)
(0601)      WRITE(11,936) RNDATE,ONTIME,OFFTIME
(0602)      WRITE (11,904) TCPU
(0603) C***** REVISE SELECTED DATA FOR NEXT RUN FROM TERMINAL
(0604)      IF(MSW6.EQ.0) GO TO 999
(0605)      700      PRINT 924
(0606)      PRINT 926
(0607)      READ(1,*,ERR=999) MSW6,DEPTH,DIST,DHC,DHM,NDTMAX
(0608)      PRINT 925
(0609)      READ(1,*,ERR=999) DT      ,TBARMX,TBARMN,TFACMX,TFACMN,EPSIR
(0610)      PRINT 928
(0611)      READ(1,*,ERR=999) MSW4,MSW8,MSW9,MSW10,MSW11,MSW12
(0612)      IF(MSW6.EQ.-1) GO TO 999
(0613)      GO TO 15
(0614) C***** FORMAT STATEMENTS
(0615)      14      FORMAT (18F4.1)
(0616)      18      FORMAT (20F4.1)
(0617)      21      FORMAT (2E10.4,4D15.4)
(0618)      23      FORMAT (1D15.4,4D10.4,1D15.4,1I6)
(0619)      25      FORMAT (4I5,2D15.7)
(0620)      28      FORMAT (1H,' PREVIOUS CALCULATIONS COVERING ',I6,' TIME INCREM
(0621)      1ENTS PROVIDE THE FOLLOWING INITIAL VALUES OF XCM, YCM, THEMO, AND
(0622)      2THETA.' //)
(0623)      29      FORMAT (1H,2X,I3,6X,D15.8,6X,D15.8,6X,D15.8,6X,D15.8)
(0624)      186     FORMAT ( 2X,'I=' ,I3,6X,'X1(I)=' ,D15.8,6X,'Y1(I)=' ,D15.8,6X,'THE
(0625)      1MO(I)=' ,D15.8)
(0626)      423     FORMAT (1H,5X,I3,6(5X,D15.8))
(0627)      429     FORMAT (1H,8X,4(5X,D15.8))
(0628)      521     FORMAT (1H,'A      =' ,F10.4/(1H,'B      =' ,F10.4)/(1H,'Q      =' ,D
(0629)      115.4)/(1H,'PLMASS=' ,D15.4)/(1H,'PLINER=' ,D15.4)/(1H,'EPSI      =' ,D
(0630)      115.4))
(0631)      523     FORMAT (1H,'DEPTH =' ,D15.4/(1H,'DIST      =' ,F10.4)/(1H,'DELTA =' ,F
(0632)      110.4)/(1H,'ZAMBDA=' ,F10.4)/(1H,'RMAX      =' ,F10.4)/(1H,'DT      =' ,D
(0633)      115.4)/(1H,'NDTMAX=' ,I6))
(0634)      529     FORMAT (1H,'ITL      =' ,I3/(1H,'IMX      =' ,I3)/(1H,'IMY      =' ,I3))
(0635)      531     FORMAT (1H,'RANFX0=' ,D15.4/(1H,'RANFY0=' ,D15.4)/(1H,'RANFT0=' ,D
(0636)      115.4)/(1H,'MCONT      =' ,I3)/(1H,'MSAVE      =' ,I3)/(1H,'NREC1      =' ,I3)/(
(0637)      2,'NREC2      =' ,I3))
(0638)      537     FORMAT (1H,' AFTER ',I6,' TIME INCREMENTS OF ',E15.4,' SECONDS

```

```

9)      1,THE POSITIONS AND TEMPERATURES ARE AS FOLLOWS:' //)
0) 539   FORMAT (1H0,4X,'I',11X,'XCM(I)',15X,'YCM(I)',14X,'THETA(I)',13X
1)      1,'THEMO(I)',13X,'ANGLE(I)' //)
2) 541   FORMAT (1H ,2X,13,6X,D15.8,6X,D15.8,6X,D15.8,6X,D15.8,6X,D15.8)
3) 545   FORMAT (1H0,4X,'I',10X,'VXCM(I)',14X,'VYCM(I)',12X,'VTHETA(I)' //)
4) 547   FORMAT (1H ,2X,13,6X,D15.8,6X,D15.8,6X,D15.8)
5) 552   FORMAT (1H0,'TEMPX=',D15.8/(1H0,'TEMPY=',D15.8)/(1H0,'TEMPR=',D15.
6)      18)/(1H0,'TBAR =' ,D15.8)/(1H0,'PVERG=',D15.8)/(1H0,'CMXSYS=',D15.8)
7)      2/(1H0,'CMYSYS=',D15.8)/(1H0,'PIPRES=',D15.8)/(1H0,'AMOLEC=',D15.8)
8)      3)
9) 597   FORMAT (1H1,4X,'RANFAX',6X,'RANFAY',6X,'RANFAT',6X,'TEMPX',7X,'TEM
0)      1PY',7X,'TEMPR',8X,'TBAR',7X,'TX-TY',7X,'TY-TR',7X,'TX-TR' //)
1) 599   FORMAT (1H0,10(1X,D11.4))
2) 801   FORMAT (7I3,3D12.5,6I3)
3) 802   FORMAT (1H , '*****')
4) 808   FORMAT (1H0,'***** OUTPUT FROM MAIN *****')
5) 901   FORMAT (10X,'***** END LOOP 620')
6) 902   FORMAT (10X,'***** END LOOP 630')
7) 903   FORMAT (10X,'***** END LOOP 640')
8) 904   FORMAT (1H ,9X,'***** TOTAL CPU TIME = ',D15.7,' SECONDS')
9) 905   FORMAT (15X,15,2D15.8)
0) 906   FORMAT (10X,15,6D10.3)
1) 907   FORMAT (10X,15,4D10.3)
2) 908   FORMAT (15X,15,4D15.8)
3) 910   FORMAT (3I3,2D24.16,D15.7)
4) 911   FORMAT (6I3)
5) 912   FORMAT (4D15.7,2I5)
6) 913   FORMAT (3D15.7)
7) 914   FORMAT (1H1,2X,'M=',15,5X,'N=',15,5X,'S=',F8.5,5X,'SCF=',F8.5,5X
8)      &,'GAMZ=',F8.5)
9) 915   FORMAT(10X,'** ENTER QDIST1')
0) 916   FORMAT(10X,'** ENTER LUDEC3')
1) 917   FORMAT(10X,'** ENTER 480 LOOP')
2) 918   FORMAT(10X,'** CALCULATE TEMPERATURES')
3) 919   FORMAT(10X,'** ENTER RANLOC')
4) 920   FORMAT(10X,'** LX=',15)
5) 921   FORMAT(15)
6) 922   FORMAT(5X,'** NDT=',15)
7) 923   FORMAT(7D11.3)
8) 924   FORMAT(2X,'***** ENTER NEW VALUES FOR MSW6,DEPTH,DIST,DHC,DHM,ND
9)      &TMAX')
0) 925   FORMAT(2X,'***** ENTER NEW VALUES FOR DT ,TBARMX,TBARMN,TFACM
1)      &X,TFACMN,EPSIR')
2) 926   FORMAT(10X,'SET MSW6 TO -1 OR 0 TO EXIT NOW OR AFTER NEXST RUN')
3) 927   FORMAT(14I3)
4) 928   FORMAT(2X,'***** ENTER NEW VALUES FOR MSW4(LJ), MSW9(VEL.EQ.), *
5)      &SW9(MSHCOR), MSW10(FORCES), MSW11(MOD.VEL.TFAC), *SW12(MOD.VEL.TRA
6)      &R)')
7) 929   FORMAT(2D15.7,3I5)
8) 930   FORMAT(A9,2I5)
9) 931   FORMAT(5X,'** DATA DATE AND NDTMAX:',A9,5X,15)
0) 932   FORMAT(4D18.10)
1) 933   FORMAT(1X,'***** TBAR LESS THAN TBARMN AT NDT=',15,'; VCORX1, VC
2)      &CRY1, & VCORD1 WERE: ',3D12.4)
3) 934   FORMAT(1X,'***** TBAR GREATER THAN TBARMX AT NDT=',15,'; VCORX2,
4)      & VCCRY2, & VCCPR2 WERE: ',3D12.4)
5) 935   FORMAT(1X,'***** DT CHANGED TO ',D12.4,' AT NDT=',15)
6) 936   FORMAT(A9,2D15.7)

```

```

069      937  FORMAT(1X,'***** NCTBMN & NCTBMX ARE:',2X,2I6)
069      938  FORMAT(2D10.3,D15.8,3D10.3)
0699)    994  FORMAT (1H1,3X,'NDT',2X,'EKTOT (KJ/MOLE)',2X,'EFTOT (KJ/MOLE)',2
0700)      1X,'ETOT (KJ/MOLE)',3X,'TBAR (DEG.KEL.)',5X,'DELTA TBAR',4X,'PIPRES
0701)      2 (DYN/CM)' //)
0702)    995  FORMAT (      2X,I5,2X,D15.8,2X,D15.8,2X,D15.8,2X,D15.8,2X,D15.8,2
0703)      1X,D15.8)
0704)    996  FORMAT('1***** FINAL TIME-COURSE OF EXPERIMENT *****')
0705)    999  STOP
0706)      END
0707) C*****

```

```

)
) *****
) SUBROUTINE INIT(IMX,IMY,A,B,AD2,XSPACE,YSPACE,XCM,YCM,THETA,VXCM
) R,VYCM,VTHETA,THEMO)
)
) C      * PURPOSE:  ARRANGE ITL/2 MOLECULES IN CHECKERBOARD PATTERN.
) C      IMX        =TWICE THE NUMBER OF MOLECULES ALONG THE X-AXIS
) C      IMY        =NUMBER OF HORIZONTAL ROWS OF MOLECULES (EVEN)
) C      ITWO       =NUMBER OF TAILS IN TWO HORIZONTAL ROWS
) C
) C      * DOCUMENTATION:
) C      INTEGER*2 NTL,NMESH
) C      PARAMETER (NTL=264,NMESH=1083)
) C      IMPLICIT REAL*8 (A-G,O-Z),INTEGER (H-N)
) C      COMMON//X1(NTL),Y1(NTL),QMESH(NMESH),DHC,DHM,Q,MESH,ITL,NALPHA,N
) 1BETA,NGAMMA,N,M,N7M,S1,S3,OMEGA,EPSIR,YH,PHIMSH(NMESH),KRSW1,KRSW2
) 2,KRSW3,KRSW4,KRSW5,KRSW6,KQSW1,KQSW2,KQSW3,KQSW4,KQSW5,KQSW6,S,SCF
) 3,GAMZ,LUSW1,LUSW2,LUSW3,LUSW4,LUSW5,LUSW6,FLJFAC,FMAX,LJSW1,LJSW2,
) 4LJSW3,LJSW4,LJSW5,LJSW6,QDSAV(8,NTL)
) C      DIMENSION XCM(NTL),YCM(NTL),THETA(NTL),VXCM(NTL),VYCM(NTL),VTH
)  ETA(NTL),THEMO(NTL)
) C*** X POSITIONS
) C      33 ITWO=2+IMX-2
) C      0035 DO 0037 J=2,IMX,2
) C          XCM(J)=(J/2-1)*0.1732051D+01*B
) C          XCM(J)=0.5D+00*XSPACE*(J-1)
) C          LIMIT=(IMY/2-1)*ITWO+J
) C          DO 0037 K=J,LIMIT,ITWO
) C      0037      XCM(K)=XCM(J)
) C          IMXP2=IMX+2
) C          DO 0044 J=IMXP2,ITWO,2
) C      C          XCM(J)=(J/2-IMX/2-0.5)*0.1732051D+01*B
) C          XCM(J)=XSPACE*((J-IMX)/2)
) C          LIMIT=(IMY/2-1)*ITWO+J
) C          DO 0044 K=J,LIMIT,ITWO
) C      0044      XCM(K)=XCM(J)
) C*** Y POSITIONS
) C      I1=0
) C      I2=0
) C      I3=1
) C      0054 DO 0067 J=2,ITL,2
) C      C          YCM(J)=AD2+(I3-1)*(A+B/2)
) C          YCM(J)=0.5D+00*YSPACE+YSPACE*(I3-1)+1.00D+00
) C          I1=I1+1
) C          IF (I2.EQ.1) GO TO 0060
) C          IF (I1.EQ.IMX/2) GO TO 0062
) C          GO TO 0067
) C      0060      IF (I1.EQ.IMX/2-1) GO TO 0064
) C      0061      GO TO 0067
) C      0062      I2=1
) C      0063      GO TO 0065
) C      0064      I2=0
) C      0065      I1=0
) C      0066      I3=I3+1
) C      0067      CONTINUE
) C      GO TO 0080
) C      0070 DO 0072 I=68,142,2
) C      0071      XCM(I)=FNC(I)*0.1732051D+01*B
) C      0072      YCM(I)=FVA(I)*A+FVB(I)*B
) C*** VELOCITIES AND ANGULAR POSITIONS

```

```
080 DO 0086 J=2,ITL,2
      VXCM(J)=0.00+00
      VYCM(J)=0.00+00
      THETA(J)=0.157080+01
      VTHETA(J)=0.00+00
      THEM0(J)=THETA(J)
```

```
0086 CONTINUE
```

```
500 RETURN
```

```
END
```

```
C*****
```

```

C *****
SUBROUTINE INIT32 (XCM,YCM,THETA,VXCM,VYCM,VTHETA,THEMO,A)
C   * PURPOSE:  ARRANGE 32 MOLECULES IN TRIANGULAR LATTICE ON
C               SQUARE CENTRAL ARRAY.
C   * DOCUMENTATION:  NOTES, PAGES 520.21 - 520.23.
INTEGER*2 NTL,NMESH
PARAMETER (NTL=064,NMESH=1023)
IMPLICIT REAL*8 (A-G,O-Z),INTEGER (H-N)
COMMON//X1(NTL),Y1(NTL),QMESH(NMESH),DHC,DHM,Q,MESH,ITL,NALPHA,N
1BETA,NGAMMA,N,M,NZM,S1,S3,OMEGA,EPSIR,YH,PHIMSH(NMESH),KRSW1,KRSW2
2,KRSW3,KRSW4,KRSW5,KRSW6,KOSW1,KOSW2,KOSW3,KOSW4,KOSW5,KOSW6,S,SCF
3,GAMZ,LUSW1,LUSW2,LUSW3,LUSW4,LUSW5,LUSW6,FLJFAC,FMAX,LJSW1,LJSW2,
4LJSW3,LJSW4,LJSW5,LJSW6,QDSAV(8,NTL)
DIMENSION XCM(NTL),YCM(NTL),THETA(NTL),VXCM(NTL),VYCM(NTL),VTH
&ETA(NTL),THEMO(NTL)
C***** CONSTANTS *****
SIZMSH=DHM*DBLE(FLOAT(MESH-1))
R=0.1D+00*SIZMSH
DLX=(SIZMSH-2.0D+00*R)/7.5D+00
AD2=0.5D+00*A
SMD5=0.2D+00*SIZMSH
C***** SET UP FOUR ROWS OF EIGHT MOLECULES EACH.
C   Y COORDINATES FOR THE NEGATIVE ENDS OF THE
C   MOLECULES ARE GIVEN BY
C       YNEG = I * 0.2D+00 * SIZMSH (I=0,1,2,3).
C   X COORDINATES FOR NEGATIVE ENDS ARE GIVEN BY
C       XNEG = R + J*DLX (J=0,1,2,...,7).
C   ANGULAR COORDINATES ARE EITHER 60 OR 300 DEGREES
C   (1.0472 OR 5.2360 RADIAN) WRT THE X-AXIS.
DO 20 NROW=1,4
  YNEG=NROW*SMD5
  NCMINI=2+(NROW-1)*16
  NCMFIN=NCMINI+14
  DO 10 NCM=NCMINI,NCMFIN,2
    NCT=(NCM-NCMINI)/2
    NCTP=NCT+2
    THETA(NCM)=1.0472D+00
    IF(MOD(NCTP,2).EQ.1) THETA(NCM)=5.2360D+00
    XNEG=R+NCT*DLX
    XCM(NCM)=XNEG+AD2*DCOS(THETA(NCM))
    YCM(NCM)=YNEG+AD2*DSIN(THETA(NCM))
  10
  20  CONTINUE
C***** SET VELOCITIES AND THEMO ARRAY
DO 30 J=2,ITL,2
  VXCM(J)=0.0D+00
  VYCM(J)=0.0D+00
  VTHETA(J)=0.0D+00
  THEMO(J)=THETA(J)
  30  CONTINUE
500  RETURN
END
C *****

```

```

*****
SUBROUTINE INITSP (XCM,YCM,THETA,VXCM,VYCM,VTHETA,THEMO,FMC)
C   * PURPOSE:  ARRANGE UP TO SIX MOLECULES ON SQUARE CENTRAL ARRAY
C               WITH SPECIFIED POSITIONS AND VELOCITIES.
C   * DOCUMENTATION:  NOTES, PAGES 520.24 - 520.25.
      INTEGER*2 NTL,NMESH
      PARAMETER (NTL=064,NMESH=1083)
      IMPLICIT REAL*8 (A-G,O-Z),INTEGER (H-N)
      COMMON//X1(NTL),Y1(NTL),QMESH(NMESH),DHC,DHM,Q,MESH,ITL,NALPHA,N
1BETA,NGAMMA,N,M,NZM,S1,S3,OMEGA,EPSIR,YH,PHIMSH(NMESH),KRSW1,KRSW2
2,KRSW3,KRSW4,KRSW5,KRSW6,KQSW1,KQSW2,KQSW3,KQSW4,KQSW5,KQSW6,S,SCF
3,GAMZ,LUSW1,LUSW2,LUSW3,LUSW4,LUSW5,LUSW6,FLJFAC,FMAX,LJSW1,LJSW2,
4LJSW3,LJSW4,LJSW5,LJSW6,QDSAV(8,NTL)
      DIMENSION XCM(NTL),YCM(NTL),THETA(NTL),VXCM(NTL),VYCM(NTL),VTH
      &ETA(NTL),THEMO(NTL),FMC(36)
C***** CONSTANTS *****
C***** SET POSITIONS, VELOCITIES AND THEMO ARRAY
      DO 10 J=2,ITL,2
        JP=(J/2-1)*6
        XCM(J)=FMC(JP+1)
        YCM(J)=FMC(JP+2)
        THETA(J)=FMC(JP+3)
        VXCM(J)=FMC(JP+4)
        VYCM(J)=FMC(JP+5)
        VTHETA(J)=FMC(JP+6)
10      THEMO(J)=THETA(J)
500    RETURN
      END
C*****

```



```

)
) *****
) SUBROUTINE RANLOC (XCM,YCM,A,B,THETA,TWOPI,RANFAX,RANFAY,RANFAT)
)   INTEGER*2 NTL,NMESH
)   PARAMETER (NTL=064,NMESH=1083)
)   IMPLICIT REAL*8 (A-G,O-Z),INTEGER (H-N)
)   COMMON//X1(NTL),Y1(NTL),QMFSH(NMESH),DHC,DHM,Q,MESH,ITL,NALPHA,N
)   1BETA,NGAMMA,N,M,NZM,S1,S3,OMEGA,EPSIR,YH,PHIMSH(NMESH),KRSW1,KRSW2
)   2,KRSW3,KRSW4,KRSW5,KRSW6,KQSW1,KQSW2,KQSW3,KQSW4,KQSW5,KQSW6,S,SCF
)   3,GAMZ,LUSW1,LUSW2,LUSW3,LUSW4,LUSW5,LUSW6,FLJFAC,FMAX,LJSW1,LJSW2,
)   4LJSW3,LJSW4,LJSW5,LJSW6,QDSAV(8,NTL)
)   2010 DIMENSION XCM(NTL),YCM(NTL),THETA(NTL)
) C***** SWITCHES (KRSW'S) *****
) C      *INDEX*      *IF SWITCH=1, THEN*
) C      1            WRITE XCM, YCM, AND THETA BEFORE RANDOMIZATON
) C      2            WRITE XCM, YCM, AND THETA AFTER RANDOMIZATON
) C      3            NOT USED
) C      4            NOT USED
) C      5            NOT USED
) C      6            NOT USED
)
)   WRITE (11,400)
)   IF(KRSW1.EQ.0) GO TO 2050
)   WRITE (11,401)
)   DO 2020 I=1,ITL
) 2020   WRITE (11,402) I,XCM(I),YCM(I),THETA(I)
) 2050   RNK=RND(10)
) 2051   DO 2063 I=2,ITL,2
)       RNK=RND(0)
)       RNK=DBLE(RNK)
)       IN=RNK*10
) 2054   MODIN=MOD(IN,2)
) 2055   IF (MODIN.LT.0.1) GO TO 2058
) 2056   IRNK=-1
) 2057   GO TO 2060
) 2058   IRNK=+1
) 2060   XCM(I)=XCM(I)+IRNK*RNK*0.8660255*B*RANFAX
)   GO TO 2063
) 2061   WRITE (11,2062) (I,IRNK,RNK,XCM(I))
) 2062   FORMAT (1H,5X,I3,5X,I3,5X,D15.9,5X,D15.9)
) 2063   CONTINUE
) 2101   DO 2113 I=2,ITL,2
)       RNY=RND(0)
)       RNY=DBLE(RNY)
)       IN=RNY*10
) 2104   MODIN=MOD(IN,2)
) 2105   IF (MODIN.LT.0.1) GO TO 2108
) 2106   IRNY=-1
) 2107   GO TO 2110
) 2108   IRNY=+1
) 2110   YCM(I)=YCM(I)+IRNY*RNY*RANFAY*(A+B)/2
)   GO TO 2113
) 2111   WRITE (11,2112) (I,IRNY,RNY,YCM(I))
) 2112   FORMAT (1H,5X,I3,5X,I3,5X,D15.9,5X,D15.9)
) 2113   CONTINUE
) 2210   DO 2270 I=2,ITL,2
)       RNT=RND(0)
)       RNT=DBLE(RNT)
)       IN=RNT*10
) 2225

```

```

) 2230 MODIN=MOD(IN,2)
) 2235 IF (MODIN .LT. 0.1) GO TO 2250
) 2240 IRNT=-1
) 2245 GO TO 2255
) 2250 IRNT=+1
) 2255 THETA(I)=THETA(I)+IRNT*RNT*TWOPI*ANFAT
) GO TO 2270
) 2260 WRITE (11,2265) (I,IRNT,RNT,THETA(I))
) 2265 FORMAT (1H,5X,I3,5X,I3,5X,D15.9,5X,D15.9)
) 2270 CONTINUE
) IF(KRSW2.EQ.0) GO TO 2500
) WRITE (11,401)
) DO 290 I=1,ITL
) 290 WRITE (11,402) I,XCM(I),YCM(I),THETA(I)
) 400 FORMAT(1H1,'***** OUTPUT FROM RANLOC *****')
) 401 FORMAT(1H1,6X,'*I*',6X,'*XCM(I)*',9X,'*YCM(I)*',8X,'*THETA(I)*')
) 402 FORMAT(1H0,5X,I5,3(2X,D15.7))
) 2500 RETURN
) END
) C*****

```

```

1)
2) C*****
3) SUBROUTINE QDIST1
4) INTEGER*2 NTL,NMESH
5) PARAMETER (NTL=064,NMESH=1083)
6) IMPLICIT REAL*8 (A-G,O-Z),INTEGER (H-N)
7) COMMON//X1(NTL),Y1(NTL),QMESH(NMESH),DHC,DHM,Q,MESH,ITL,NALPHA,N
8) 1BETA,NGAMMA,N,M,NZM,S1,S3,OMEGA,EPSIR,YH,PHIMSH(NMESH),KRSW1,KRSW2
9) 2,KRSW3,KRSW4,KRSW5,KRSW6,KQSW1,KQSW2,KQSW3,KQSW4,KQSW5,KQSW6,S,SCF
0) 3,GAMZ,LUSW1,LUSW2,LUSW3,LUSW4,LUSW5,LUSW6,FLJFAC,FMAX,LJSW1,LJSW2,
1) 4LJSW3,LJSW4,LJSW5,LJSW6,QDSAV(8,NTL)
2) DIMENSION DUM1(10)
3) C TCPU1=CTIME(A(CPUTIM))
4) C READ(09,200) NGAMMA,ZH,OMEGA,TOL
5) C***** SWITCHES (KQSW'S) *****
6) C *INDEX* *IF SWITCH=1, THEN*
7) C 1 WRITE ALL VALUES OF QMESH AND PARTIAL SUMQM'S
8) C 2 WRITE INPUT VALUES OF X1 AND Y1
9) C 3 WRITE INTERNAL PARAMETERS FOR LOOP 100
0) C 4 WRITE SUM OF MESH CHARGES (SUMQM) AT EACH TIME STE
1) C 5 DO NOT ASSOCIATE CLOUDS WITH CHARGES
2) C 6 NOT USED
3) C***** CONSTANTS *****
4) IF((KQSW1.EQ.1).OR.(KQSW2.EQ.1).OR.(KQSW3.EQ.1).OR.(KQSW4.EQ.1)) W
5) 1RITE (11,405)
6) IF(KQSW2.EQ.0) GO TO 50
7) WRITE (11,403)
8) DO 45 I=1,ITL
9) 45 WRITE (11,400) I,X1(I),Y1(I)
0) 50 DO 60 K=1,N
1) 60 QMESH(K)=0.00+00
2) DHC5=0.50+00*DHC
3) DHM5=0.50+00*DHM
4) DHC2IN=0.10+01/(DHC*DHC)
5) DHM2IN=0.10+01/(DHM*DHM)
6) IF(KQSW3.EQ.1) WRITE (11,404)
7) DO 100 I=1,ITL
8) C * ODD-NUMBERED FORCE CENTERS ARE NEGATIVE, EVEN POSITIVE.
9) ISIGN=(-1)**I
0) QISIGN=0*DBLE(FLOAT(ISIGN))
1) IM=IDINT(X1(I)/DHM)+1
2) JM=IDINT(Y1(I)/DHM)+1
3) RX=DMOD(X1(I),DHM)
4) RY=DMOD(Y1(I),DHM)
5) IF(KQSW5.EQ.1) GO TO 70
6) IF (RX .LE. DHM5) GO TO 62
7) LX=1
8) MX=1
9) IMP=IM+1
0) GO TO 63
1) 62 LX=0
2) MX=-1
3) IMP=IM-1
4) 63 IF (RY .LE. DHM5) GO TO 64
5) LY=1
6) MY=1
7) JMP=JM+1
8) GO TO 65
9)

```

```

001      64      LY=0
0060      MY=-1
0061      JMP=JM-1
0062      65      SX=LX*DHM-MX*RX
0063      SY=LY*DHY-MY*RY
0064      FIJ=DHC2IN*(DHC5+SX)*(DHC5+SY)
0065      FIPJ=DHC2IN*(DHC5-SX)*(DHC5+SY)
0066      FIJP=DHC2IN*(DHC5+SX)*(DHC5-SY)
0067      FIPJP=DHC2IN*(DHC5-SX)*(DHC5-SY)
0068      C      * IF CHARGE HAS FALLEN OUTSIDE ARRAY, PLACE IT ON EDGE
0069      IF((IMP-MESH).GT.0) IMP=MESH
0070      IF(IMP.EQ.0) IMP=1
0071      IF((JMP-MESH).GT.0) JMP=MESH
0072      IF(JMP.EQ.0) JMP=1
0073      ILIN00=NZM+(IM-1)*M+(JM-1)*NGAMMA
0074      ILIN10=NZM+(IMP-1)*M+(JM-1)*NGAMMA
0075      ILIN01=NZM+(IM-1)*M+(JMP-1)*NGAMMA
0076      ILIN11=NZM+(IMP-1)*M+(JMP-1)*NGAMMA
0077      GO TO 80
0078      70      DHMMRX=DHM-RX
0079      DHMMRY=DHM-RY
0080      FIJ=DHM2IN*DHMMRX*DHMMRY
0081      FIPJ=DHM2IN*RX*DHMMRY
0082      FIJP=DHM2IN*DHMMRX*RY
0083      FIPJP=DHM2IN*RX*RY
0084      ILIN00=NZM+(IM-1)*M+(JM-1)*NGAMMA
0085      ILIN10=ILIN00+M
0086      ILIN01=ILIN00+NGAMMA
0087      ILIN11=ILIN01+M
0088      C      * PRESERVE THE VERTICES & FRACTIONS FOR EACH CHARGE IN QDSAV
0089      80      QDSAV(1,I)=ILIN00
0090      QDSAV(2,I)=ILIN10
0091      QDSAV(3,I)=ILIN01
0092      QDSAV(4,I)=ILIN11
0093      QDSAV(5,I)=FIJ
0094      QDSAV(6,I)=FIPJ
0095      QDSAV(7,I)=FIJP
0096      QDSAV(8,I)=FIPJP
0097      C      * ASSIGN CHARGE FRACTIONS TO VERTICES
0098      QMESH(ILIN00)=QMESH(ILIN00)+FIJ*QISIGN
0099      QMESH(ILIN10)=QMESH(ILIN10)+FIPJ*QISIGN
0100      QMESH(ILIN01)=QMESH(ILIN01)+FIJP*QISIGN
0101      QMESH(ILIN11)=QMESH(ILIN11)+FIPJP*QISIGN
0102      IF(KQSW3.EQ.0) GO TO 100
0103      WRITE(11,401)I,IM,JM,IMP,JMP,RX,RY,FIJ,FIPJ,FIJP,FIPJP,QMESH(
0104      1ILIN00),QMESH(ILIN10),QMESH(ILIN01),QMESH(ILIN11)
0105      100      CONTINUE
0106      SUMQM=0.0
0107      IF(KQSW1.EQ.1) WRITE (11,406)
0108      DO 200 K=1,NMESH,10
0109      KM1=K-1
0110      SUM1=0.00+00
0111      DO 190 I=1,10
0112      IP=I+KM1
0113      IF(IP.GT.NMESH) GO TO 195
0114      DUM1(I)=QMESH(IP)
0115      190      SUM1=SUM1+DUM1(I)
0116      195      SUMQM=SUMQM+SUM1

```

```

(0117)      IF(KQSW1.EQ.1) WRITE (11,402) KM1,SUMQM,(DUM1(I),I=1,10)
(      8) 200  CONTINUE
(0119)      IF(KOSW4.EQ.1) WRITE(11,407) SUMQM
(0120) 400  FORMAT (1H0,I3,2(2X,D15.7))
(0121) 401  FORMAT (1H0,5I3,10D11.4)
(0122) 402  FORMAT (1H0,I5,2X,D12.5,10D11.4)
(0123) 403  FORMAT (1H0,'*I*',6X,'*X1(I)*',10X,'*Y1(I)*')
(0124) 404  FORMAT (1H0,'*I *IM*JMIMPJMP',4X,'*RX*',7X,'*RY*',6X,'*FIJ*',6X,
(0125) 1'*FIPJ*',5X,'*FIJP*',5X,'*FIPJP*',3X,'*QMESH(00)',1X,'*QMESH(10)',
(0126) 21X,'*QMESH(01)',1X,'*QMESH(11)')
(0127) 405  FORMAT (1H0,'***** OUTPUT FROM QDIST1 *****')
(0128) 406  FORMAT (1H0,1X,'*K*',5X,'*SUMQM*',6X,'*QMESH(K+1) THROUGH QMESH(
(0129) 1K+10)*')
(0130) 407  FORMAT (1H0,6X,'SUMQM=',D15.8)
(0131) 420  FORMAT (10X,'** DHC2IN=',D15.8)
(0132) 421  FORMAT (10X,'** DHC   =',D15.8)
(0133) 422  FORMAT (1H1,'*I* *ILIN00* *ILIN10* *ILIN01* *ILIN11*')
(0134) 423  FORMAT (1H0,81X,4(5X,I6))
(0135) 424  FORMAT (10X,'** I,DHM,X1,&Y1 ARE:',I3,3D15.2)
(0136) 500  RETURN
(0137)      END
(0138) C*****

```

```

( 1)
( 2) C*****
(0003) SUBROUTINE LUDEC3
(0004) INTEGER*2 NROWS,NCOLS,NTL,NMESH
(0005) PARAMETER (NTL=064,NMESH=1083,NROWS=1083,NCOLS=58)
(0006) C * NALPHA, NBETA, AND NGAMMA ARE TAKEN TO BE ODD.
(0007) C * NOTE THAT NROWS=NMESH=NALPHA*NBETA*NGAMMA
(0008) C NCOLS=(NBETA*NGAMMA)+1
(0009) IMPLICIT REAL*8(A-G,O-Z),INTEGER(H-N)
(0010) COMMON//X1(NTL),Y1(NTL),QMESH(NMESH),DHC,DHW,Q,MESH,ITL,NALPHA,N
(0011) 1BETA,NGAMMA,N,M,NZM,S1,S3,OMEGA,EPSIR,YH,PHIMSH(NMESH),KRSW1,KRSW2
(0012) 2,KRSW3,KRSW4,KRSW5,KRSW6,KQSW1,KQSW2,KQSW3,KQSW4,KQSW5,KQSW6,S,SCF
(0013) 3,GAMZ,LUSW1,LUSW2,LUSW3,LUSW4,LUSW5,LUSW6,FLJFAC,FMAX,LJSW1,LJSW2,
(0014) 4LJSW3,LJSW4,LJSW5,LJSW6,QDSAV(8,NTL)
(0015) C COMMON /A/ A(NROWS,NCOLS)
(0016) COMMON /EL/ EL(NROWS,NCOLS)
(0017) COMMON /U/ U(NROWS,NCOLS)
(0018) DIMENSION UINV(NROWS),DUM1(10)
(0019) C DIMENSION APHI(NROWS)
(0020) TCPU1=CTIMSA(CPUTIM)
(0021) C***** SWITCHES (LUSW'S) *****
(0022) C *INDEX* *IF SWITCH=1, THEN*
(0023) C 1 EL AND U ARE NOT READ FROM ELU
(0024) C 2 CALCULATE APHI AND WRITE QMESH, APHI, DIFFI, PHIMS
(0025) C 3 WRITE PHIMSH IN CENTER XY-PLANE
(0026) C 4 WRITE PHIMSH FROM WEST TO EAST
(0027) C 5 WRITE PHIMSH AND UINV AFTER FORWARD SUBSTITUTION
(0028) C 6 NOT USED
(0029) C***** CONSTANTS *****
(0030) NAB=NALPHA*NBETA
(0031) NGP1=NGAMMA+1
(0032) NGM1=NGAMMA-1
(0033) NGTEN=NGAMMA*10
(0034) NP1=N+1
(0035) NMM=N-M
(0036) NM2M=NMM-M
(0037) MP1=M+1
(0038) MMNG=M-NGAMMA
(0039) SIGMA=2.*(1.+S1+S3)
(0040) S1N=-S1
(0041) S3N=-S3
(0042) DNEG1=-1.
(0043) S3NOM=S3N*OMEGA
(0044) EPYH=EPSIR+YH
(0045) RS1S3=DSORT(S1*S3)
(0046) CNP=0.1D+01
(0047) QLUFAC=0.1129943503D+22*RS1S3*CNP/EPYH
(0048) C * VALUE OF Q INSERTED INTO NMESH BY QDIST1.
(0049) IC=NZM+(N-NGAMMA)/2
(0050) C***** SET UP Q VECTOR *****
(0051) DO 15 I=1,N
(0052) QMESH(I)=QMESH(I)*QLUFAC
(0053) (5 PHIMSH(I)=QMESH(I)
(0054) IF(LUSW1.EQ.1) GO TO 105
(0055) C***** READ EL AND U FROM FILE ELU *****
(0056) READ(12) CRDELU,NALELU,NBEELU,NGAELU,NELU,MELU,K8ELU,K10ELU
(0057) READ(12) OMELU,S1ELU,S3ELU,EPSELU,YHELU
(0058) PRINT 311 CRDELU,NALELU,NBEELU,NGAELU,NELU,MELU,K8ELU,K10ELU

```

```

( 9)      PRINT 313 OMELU,S1ELU,S3ELU,EPSELU,YHELU
(0060)      WRITE(11,245)
(0061)      WRITE(11,311) CRDELU,NALELU,NBEELU,NGAELU,NELU,MELU,K8ELU,K10ELU
(0062)      WRITE(11,313) OMELU,S1ELU,S3ELU,EPSELU,YHELU
(0063)      DO 102 J=1,MP1
(0064)          DO 102 I=1,N
(0065)      102      READ(12) EL(I,J),U(I,J)
(0066)          TCPU2=CTIMSA(CPUTIM)
(0067)          TCPU=TCPU2-TCPU1
(0068)          PRINT 302 TCPU
(0069)      C***** FORWARD SUBSTITUTION *****
(0070)          DO 103 I=1,N
(0071)      103      UINV(I)=1.0/U(I,1)
(0072)      105      DO 120 I=1,N
(0073)          IF(I.EQ.1) GO TO 118
(0074)          IF(I.GE.MP1) GO TO 107
(0075)          LMAX=I-1
(0076)          GO TO 109
(0077)      107      LMAX=M
(0078)      109      DO 110 L=1,LMAX
(0079)          IML=I-L
(0080)          LP1=L+1
(0081)      110          PHIMSH(I)=PHIMSH(I)-EL(IML,LP1)*PHIMSH(IML)
(0082)      118      IF(LUSW5.EQ.0) GO TO 120
(0083)          WRITE(11,255)
(0084)      119      WRITE(11,229) I,PHIMSH(I),UINV(I)
(0085)      120      CONTINUE
(0086)      C***** BACKWARD SUBSTITUTION *****
(0087)          DO 140 I=1,N
(0088)          IREV=NP1-I
(0089)          IF(I.EQ.1) GO TO 140
(0090)          IF(I.GE.MP1) GO TO 127
(0091)          LMAX=I-1
(0092)          GO TO 129
(0093)      127      LMAX=M
(0094)      129      DO 130 L=1,LMAX
(0095)          IREVPL=IREV+L
(0096)          LP1=L+1
(0097)      130          PHIMSH(IREV)=PHIMSH(IREV)-U(IREV,LP1)*PHIMSH(IREVPL)
(0098)      140          PHIMSH(IREV)=PHIMSH(IREV)*UINV(IREV)
(0099)          IF(LUSW2.EQ.0) GO TO 170
(0100)      C***** CALCULATE A+POTENTIAL (NOW INACTIVE) *****
(0101)          DO 166 I=1,N
(0102)      C          APHI(I)=0.0
(0103)          IF(I.EQ.1) GO TO 165
(0104)          IM1=I-1
(0105)          DO 164 K=1,IM1
(0106)          IMKP1=I-K+1
(0107)          IF(IMKP1.GT.MP1) GO TO 164
(0108)      C          APHI(I)=APHI(I)+A(K,IMKP1)*PHIMSH(K)
(0109)      164      CONTINUE
(0110)      165      DO 165 K=I,N
(0111)          KMIP1=K-I+1
( 2)          IF(KMIP1.GT.MP1) GO TO 166
( 3)      C          APHI(I)=APHI(I)+A(I,KMIP1)*PHIMSH(K)
(0114)      166      CONTINUE
(0115)      C***** WRITE QMESH, APHI, DIFFI, AND PHIMSH (NOW INACTIVE) *****
(0116)          WRITE(11,290)

```

```

( 17) C      DO 168 I=1,N
(0118) C      DIFFI=QMESH(I)-APHI(I)
(0119) C168      WRITE(11,260) I,QMESH(I),APHI(I),DIFFI,PHIMSH(I)
(0120) 170      IF(LUSW3.EQ.0) GO TO 174
(0121) C***** WRITE POTENTIALS IN CENTER XY-PLANE *****
(0122)      WRITE(11,292)
(0123)      IMAX=NZM+N-NGAMMA
(0124)      DO 173 I=NZM,IMAX,NGTEN
(0125)          DO 172 K=1,10
(0126)              IP=I+(K-1)*NGAMMA
(0127)              IF(IP.GT.IMAX) IP=IMAX
(0128) 172          DUM1(K)=PHIMSH(IP)
(0129) 173          WRITE(11,233) I,(DUM1(K),K=1,10)
(0130) 174          IF(LUSW4.EQ.0) GO TO 178
(0131) C***** WRITE POTENTIALS FOR WEST-TO-EAST PROFILE IN CENTER XY-PLANE *****
(0132)      WRITE(11,293)
(0133)      IMIN=NZM+MMNG/2
(0134)      IMAX=IMIN+MMM
(0135)      DO 176 I=IMIN,IMAX,M
(0136) 176          WRITE(11,233) I,PHIMSH(I)
(0137) C***** WRITE POTENTIALS FOR SOUTH-TO-NORTH PROFILE IN CENTER XY-PLANE ***
(0138)      WRITE(11,294)
(0139)      IMIN=NZM+MMM/2
(0140)      IMAX=IMIN+MMNG
(0141)      DO 177 I=IMIN,IMAX,NGAMMA
(0142) 177          WRITE(11,233) I,PHIMSH(I)
(0143) 178          CONTINUE
(0144) C***** FORMAT STATEMENTS. *****
(0145) 205      FORMAT(1H,2X,'THE CPU TIME USED ON THIS JOB WAS ',D24.16,' SEC
(0146) 2. ')
(0147) 225      FORMAT(1H,2X,'DIAGONAL',I5)
(0148) 229      FORMAT(1H,I5,2D24.16)
(0149) 230      FORMAT(3D24.16)
(0150) 231      FORMAT(1H,15X,1D24.16)
(0151) 232      FORMAT(1H,15X,4D24.16)
(0152) 233      FORMAT(1H,2X,I5,2X,10D12.5)
(0153) 234      FORMAT(1H,2X,3I3,2D24.16,D15.7,2D24.16)
(0154) 240      FORMAT(1H,2X,'SUM.GT.H(I,1) FOR I=',I3)
(0155) 245      FORMAT(1H,'***** OUTPUT FROM LUDEC3 *****')
(0156) 250      FORMAT(1H,2X,'THE VALUES OF NALPHA, NBETA, NGAMMA, S1, S3, OMEG
(0157) 8A, EPSIR, AND YH ARE:')
(0158) 255      FORMAT(1H,1X,'*I*',7X,'*PHIMSH(I)*',14X,'*UINV(I)*')
(0159) 260      FORMAT(1H,I4,2X,D24.16,3(5X,D24.16))
(0160) 290      FORMAT(1H,9X,'***** Q VECTOR *****',9X,'***** A * POTENTIAL *****',
(0161) 88X,'***** DIFFERENCE *****',6X,'***** POTENTIAL VECTOR *****')
(0162) 291      FORMAT(1H,2X,' THE VALUE OF THE CHARGE AT THE CENTER OF THE ARR
(0163) 8AY IS',1D24.16)
(0164) 292      FORMAT(1H,2X,' THE POTENTIALS IN THE CENTER XY-PLANE ARE AS FOL
(0165) 8LOWERS: ')
(0166) 293      FORMAT(1H,2X,' THE POTENTIALS FOR THE CENTRAL WEST-TO-EAST PROF
(0167) 8ILE IN THE CENTER XY-PLANE ARE AS FOLLOWS: ')
(0168) 294      FORMAT(1H,2X,' THE POTENTIALS FOR THE CENTRAL SOUTH-TO-NORTH PR
(0169) 8OFILE IN THE CENTER XY-PLANE ARE AS FOLLOWS: ')
( 170) 301      FORMAT(10X,'A SET UP AND MODIFIED AFTER ',D16.8,' CPU SEC.')
( 171) 302      FORMAT(10X,'EL AND U READ AFTER ',D16.8,' CPU SEC.')
(0172) 303      FORMAT(10X,'PRODUCT OF EL AND U FOUND AFTER ',D16.8,' CPU SEC.
(0173) 8')
(0174) 304      FORMAT(10X,'FORWARD AND BACKWARD SUBSTITUTION COMPLETED AFTER

```



```

75)      &' ,D16.8,' CPU SEC.')
76) 305   FORMAT(10X,'A*POTENTIAL CALCULATED AFTER ' ,D16.8,' CPU SEC.')
(0177) 306   FORMAT(10X,'POTENTIALS WRITTEN AFTER ' ,D16.8,' CPU SEC.')
(0178) 307   FORMAT(15X,'LOOP 80 COMPLETED WITH K = ' ,I4,' AFTER' ,D16.8,' CPU
(0179)      & SEC.')
(0180) 308   FORMAT(20X,'EL(K,KP) = ' ,D24.16)
(0181) 310   FORMAT(A9,7I5)
(0182) 311   FORMAT(1H ,10X,'** ELU PARAMETERS: ' ,A9,7I5)
(0183) 312   FORMAT(5D15.7)
(0184) 313   FORMAT(1H ,29X,5D15.7)
(0185) 500   RETURN
(0186)      END
(0187) C*****

```

```

( 1)
(0002) C*****
(0003) SUBROUTINE FORCE1 (I,FXI,FYI,SIZMSH,PCAVI,PCTOT)
(0004) C *OBJECTIVE: CALCULATE FORCE COMPONENTS AT (X1(I),Y1(I)) FROM
(0005) C MESH POTENTIAL USING WEIGHTED-AVERAGE METHOD.
(0006) C *DOCUMENTATION: NOTES, PAGES 503-505.
(0007) INTEGER*2 NTL,NMESH
(0008) PARAMETER (NTL=064,NMESH=1083)
(0009) IMPLICIT REAL*8 (A-G,O-Z),INTEGER (H-N)
(0010) COMMON//X1(NTL),Y1(NTL),QMESH(NMESH),DHC,DHM,Q,MESH,ITL,NALPHA,N
(0011) 1BETA,NGAMMA,N,M,NZM,S1,S3,OMEGA,EPSIR,YH,PHIMSH(NMESH),KRSW1,KRSW2
(0012) 2,KRSW3,KRSW4,KRSW5,KRSW6,KQSW1,KQSW2,KQSW3,KQSW4,KQSW5,KQSW6,S,SCF
(0013) 3,GAMZ,LUSW1,LUSW2,LUSW3,LUSW4,LUSW5,LUSW6,FLJFAC,FMAX,LJSW1,LJSW2,
(0014) 4LJSW3,LJSW4,LJSW5,LJSW6,QDSAV(8,NTL)
(0015) FRACX=DMOD(X1(I),DHM)/DHM
(0016) FRACY=DMOD(Y1(I),DHM)/DHM
(0017) NPSW=NZM+M*IDINT(X1(I)/DHM)+NGAMMA*IDINT(Y1(I)/DHM)
(0018) NPNW=NPSW+NGAMMA
(0019) NPSE=NPSW+M
(0020) NPNE=NPSE+NGAMMA
(0021) IF((X1(I).LT.0.00+00).OR.(X1(I).GT.SIZMSH)) GO TO 50
(0022) GO TO 60
(0023) 50 NPSE=NPSW
(0024) NPNE=NPNW
(0025) 60 IF((Y1(I).LT.0.00+00).OR.(Y1(I).GT.SIZMSH)) GO TO 70
(0026) GO TO 80
(0027) 70 NPNW=NPSW
(0028) NPNE=NPSE
(0029) 80 FXI=(0.10+01-FRACY)*(PHIMSH(NPSE)-PHIMSH(NPSW))+FRACY*(PHIMSH(NP
(0030) 1NE)-PHIMSH(NPNW))
(0031) FYI=(0.10+01-FRACX)*(PHIMSH(NPNW)-PHIMSH(NPSW))+FRACX*(PHIMSH(NP
(0032) 1NE)-PHIMSH(NPSE))
(0033) FXI=0.16D-18*FXI/DHM /*JOULES PER ANGSTROM*/
(0034) FYI=0.16D-18*FYI/DHM
(0035) PXI=(0.10+01-FRACY)*(PHIMSH(NPSE)+PHIMSH(NPSW))+FRACY*(PHIMSH(NP
(0036) 1NE)+PHIMSH(NPNW))
(0037) PYI=(0.10+01-FRACX)*(PHIMSH(NPNW)+PHIMSH(NPSW))+FRACX*(PHIMSH(NP
(0038) 1NE)+PHIMSH(NPSE))
(0039) PCAVI=0.25D+00*(PXI+PYI) /*VOLTS*/
(0040) PCTOT=PCTOT+PCAVI
(0041) IF((FXI.GT.FMAX).OR.(FYI.GT.FMAX)) PRINT 403 I,FXI,FYI
(0042) C PRINT 400 I,FXI,FYI,X1(I),Y1(I)
(0043) C PRINT 401 NPSW,NPNW,NPNE,NPSE,FRACX,FRACY
(0044) C PRINT 402 PHIMSH(NPSW),PHIMSH(NPNW),PHIMSH(NPNE),PHIMSH(NPSE)
(0045) 400 FORMAT(10X,15,4D15.8)
(0046) 401 FORMAT(15X,4I5,2D15.8)
(0047) 402 FORMAT(15X,4D15.8)
(0048) 403 FORMAT('>>> FORCES EXCEED FMAX IN FORCE1. I=',I3,2X,'FXI=',D11.
(0049) 14,2X,'FYI=',D11.4)
(0050) 500 RETURN
(0051) END
(0052) C*****

```

```

31)
(0002) C*****
(0003) SUBROUTINE FORCE2 (I,XCM,YCM,FXCMI,FYCM I)
(0004) C *OBJECTIVE: CALCULATE FORCE COMPONENTS AT (XCM(I),YCM(I)) FROM
(0005) C MESH POTENTIAL USING WEIGHTED-AVERAGE METHOD.
(0006) INTEGER*2 NTL,NMESH
(0007) PARAMETER (NTL=064,NMESH=1083)
(0008) IMPLICIT REAL*8 (A-G,O-Z),INTEGER (H-N)
(0009) COMMON//X1(NTL),Y1(NTL),QMESH(NMESH),DHC,DHM,Q,MESH,ITL,NALPHA,N
(0010) 1BETA,NGAMMA,N,M,NZM,S1,S3,OMEGA,EPSIR,YH,PHIMSH(NMESH),KRSW1,KRSW2
(0011) 2,KRSW3,KRSW4,KRSW5,KRSW6,KQSW1,KQSW2,KQSW3,KQSW4,KQSW5,KQSW6,S,SCF
(0012) 3,GAMZ,LUSW1,LUSW2,LUSW3,LUSW4,LUSW5,LUSW6,FLJFAC,FMAX,LJSW1,LJSW2,
(0013) 4LJSW3,LJSW4,LJSW5,LJSW6,QDSAV(8,NTL)
(0014) DIMENSION XCM(NTL),YCM(NTL)
(0015) FRACX=DMOD(XCM(I),DHM)/DHM
(0016) FRACY=DMOD(YCM(I),DHM)/DHM
(0017) C PRINT 400 I
(0018) C PRINT 402 XCM(I),YCM(I),DHM
(0019) NPSW=NZM+M*IDINT(XCM(I)/DHM)+NGAMMA*IDINT(YCM(I)/DHM)
(0020) C PRINT 401 I
(0021) NPNW=NPSW+NGAMMA
(0022) NPSE=NPSW+M
(0023) NPNE=NPSE+NGAMMA
(0024) FXI=(0.1D+01-FRACY)*(PHIMSH(NPSE)-PHIMSH(NPSW))+FRACY*(PHIMSH(NP
(0025) 1NE)-PHIMSH(NPNW))
(0026) FYI=(0.1D+01-FRACX)*(PHIMSH(NPNW)-PHIMSH(NPSW))+FRACX*(PHIMSH(NP
(0027) 1NE)-PHIMSH(NPSE))
(0028) FXI=0.16D-18*FXI/DHM
(0029) FYI=0.16D-18*FYI/DHM
(0030) IF((FXI.GT.FMAX).OR.(FYI.GT.FMAX)) PRINT 403 I,FXI,FYI
(0031) 400 FORMAT(15X,'** NPSW REACHED WITH I=',I5)
(0032) 401 FORMAT(15X,'** NPSW COMPLETED WITH I=',I5)
(0033) 402 FORMAT(25X,3D15.8)
(0034) 403 FORMAT('>>> FORCES EXCEED FMAX IN FORCE2. I=',I3,2X,'FXI=',D11.
(0035) 14,2X,'FYI=',D11.4)
(0036) 500 RETURN
(0037) END
(0038) C*****

```

```

J1)
J2) C*****
(0003) SUBROUTINE LJFOR2(I,DX,DY,DEPTH,DIST,RLJMAX,FLJXI,FLJYI,FCXI,FCY
(0004) 1I,SIZMSH,PLJAVI,PLJTOT,PCAVIO,PCTOTD)
(0005) C *OBJECTIVE: CALCULATE PAIR-WISE LJ INTERACTION.
(0006) C COULOMBIC INTERACTIONS MAY BE CALCULATED WHEN
(0007) C LJ AND COULOMBIC FORCE CENTERS COINCIDE.
(0008) C *METHOD: TERMINATE CALCULATIONS AT DISTANCE OF RLJMAX.
(0009) C OPERATE UPON X1 AND Y1 ARRAYS; THESE ARE NOT
(0010) C UPDATED UNTIL NEW C-M POSITIONS FOR ALL MOLECULE
(0011) C HAVE BEEN CALCULATED.
(0012) C *RESTRICTIONS: X AND Y DIMENSIONS OF THE ARRAY ARE TAKEN TO BE
(0013) C EQUAL.
(0014) C *DOCUMENTATION: NOTES, PAGES 518-520.0.3.
(0015) INTEGER*2 NTL,NMESH
(0016) PARAMETER (NTL=064,NMESH=1083)
(0017) IMPLICIT REAL*8 (A-G,O-Z),INTEGER (H-N)
(0018) COMMON//X1(NTL),Y1(NTL),QMESH(NMESH),DHC,DHM,Q,MESH,ITL,NALPHA,N
(0019) 1BETA,NGAMMA,N,M,NZM,S1,S3,OMEGA,EPSIR,YH,PHIMSH(NMESH),KRSW1,KRSW2
(0020) 2,KRSW3,KRSW4,KRSW5,KRSW6,KQSW1,KQSW2,KQSW3,KQSW4,KQSW5,KQSW6,S,SCF
(0021) 3,GAMZ,LUSW1,LUSW2,LUSW3,LUSW4,LUSW5,LUSW6,FLJFAC,FMAX,LJSW1,LJSW2,
(0022) 4LJSW3,LJSW4,LJSW5,LJSW6,QDSAV(8,NTL)
(0023) DIMENSION XJ(3),YJ(3)
(0024) C***** SWITCHES (LJSW'S) *****
(0025) C *INDEX* *IF SWITCH=1, THEN*
(0026) C 1 CALCULATE PAIR-WISE COULOMBIC FORCES
(0027) C 2 NOT USED
(0028) C 3 NOT USED
(0029) C 4 NOT USED
(0030) C 5 NOT USED
(0031) C 6 NOT USED
(0032) VLJPX=0.00+00
(0033) VLJMX=0.00+00
(0034) VLJPY=0.00+00
(0035) VLJMY=0.00+00
(0036) VCPX=0.00+00
(0037) VCMX=0.00+00
(0038) VCPY=0.00+00
(0039) VCMY=0.00+00
(0040) FCXI=0.00+00
(0041) FCYI=0.00+00
(0042) SSMH2P=0.20+01*SIZMSH+DHM
(0043) DHMN=-DHM
(0044) QD4PEP=(Q*0.89918050+20)/EPSIR /*VOLT ANGSTROMS*/
(0045) IMOD2=MOD(I,2)
(0046) IPM1=I+(-1)**(1+IMOD2)
(0047) DO 50 J=1,ITL
(0048) C *CENTERS CONSIDERED ARE THE CENTRAL ONE & ITS 8 NEAREST IMAGES.
(0049) XJ(1)=X1(J)
(0050) XJ(2)=SSMH2P-X1(J)
(0051) XJ(3)=DHMN-X1(J)
(0052) YJ(1)=Y1(J)
(0053) YJ(2)=SSMH2P-Y1(J)
(0054) YJ(3)=DHMN-Y1(J)
(0055) DO 50 K=1,3
(0056) DO 50 L=1,3
(0057) IF((K.EQ.1).AND.(L.EQ.1).AND.((J.EQ.I).OR.(J.EQ.IPM1))) GO
(0058) 1 TO 50

```

```

59)          IF(DABS(XJ(K)-X1(I)).GT.RLJMAX) GO TO 50
...60)          IF(DABS(YJ(L)-Y1(I)).GT.RLJMAX) GO TO 50
(0061)          XIJSQ=(DABS(XJ(K)-X1(I)))*0.2D+01
(0062)          YIJSQ=(DABS(YJ(L)-Y1(I)))*0.2D+01
(0063)          RIJ=DSQRT(XIJSQ+YIJSQ)
(0064)          IF(RIJ.GT.RLJMAX) GO TO 50
(0065)          RIJPX=DSQRT((DABS(XJ(K)-X1(I)-DX))*0.2D+01+YIJSQ)
(0066)          RIJMX=DSQRT((DABS(XJ(K)-X1(I)+DX))*0.2D+01+YIJSQ)
(0067)          RIJPY=DSQRT((DABS(YJ(L)-Y1(I)-DY))*0.2D+01+XIJSQ)
(0068)          RIJMY=DSQRT((DABS(YJ(L)-Y1(I)+DY))*0.2D+01+XIJSQ)
(0069)          VLJPX=VLJPX+DEPTH*((DIST/RIJPX)**0.12D+02-0.2D+01*(DIST/RIJPX)
(0070)          1**0.5D+01)
(0071)          VLJMX=VLJMX+DEPTH*((DIST/RIJMX)**0.12D+02-0.2D+01*(DIST/RIJMX)
(0072)          1**0.5D+01)
(0073)          VLJPY=VLJPY+DEPTH*((DIST/RIJPY)**0.12D+02-0.2D+01*(DIST/RIJPY)
(0074)          1**0.5D+01)
(0075)          VLJMY=VLJMY+DEPTH*((DIST/RIJMY)**0.12D+02-0.2D+01*(DIST/RIJMY)
(0076)          1**0.5D+01)
(0077)          IF(LJSW1.EQ.0) GO TO 50
(0078)  C          *ODD-NUMBERED FORCE CENTERS ARE NEGATIVE, EVEN POSITIVE.
(0079)          JSIGN=(-1)**J
(0080)          QJSIGN=QD4PEP*DRLE(FLOAT(JSIGN))
(0081)          VCPX=VCPX+QJSIGN/RIJPX
(0082)          VCMX=VCMX+QJSIGN/RIJMX
(0083)          VCPY=VCPY+QJSIGN/RIJPY
(0084)          VCMY=VCMY+QJSIGN/RIJMY
(0085)  50          CONTINUE
(0086)          FLJXI=FLJFAC*0.5D+00*(VLJPX-VLJMX)/DX /*JOULES PER ANGSTROM*/
(0087)          FLJYI=FLJFAC*0.5D+00*(VLJPY-VLJMY)/DY
(0088)          PLJAVI=0.25D+00*(VLJPX+VLJMX+VLJPY+VLJMY) /*JOULES*/
(0089)          PLJTOT=PLJTOT+PLJAVI
(0090)          FCXI=0.16D-18*0.5D+00*(VCPX-VCMX)/DX
(0091)          FCYI=0.16D-18*0.5D+00*(VCPY-VCMY)/DY
(0092)          PCAVID=0.25D+00*(VCPX+VCMX+VCPY+VCMY) /*JOULES*/
(0093)          PCTOTD=PCTOTD+PCAVID
(0094)          IF((FLJXI.GT.FMAX).OR.(FLJYI.GT.FMAX)) PRINT 400 I,FLJXI,FLJYI
(0095)  400          FORMAT('>>> FORCES EXCEED FMAX IN LJFOR2. I=',I3,2X,'FLJXI=',D1
(0096)          11.4,2X,'FLJYI=',D11.4)
(0097)  500          RETURN
(0098)          END
(0099)  C*****

```

```

(0031)
(0032) C*****
(0033) SUBROUTINE PRESS1 (XCM,YCM,CMXSYS,CMYSYS,TBAR,PVERG,FTXCM,FTYCM)
(0034) C *OBJECTIVE: CALCULATE SYSTEM PRESSURE USING VIRIAL THEOREM
(0035) INTEGER*2 NTL,NMESH
(0036) PARAMETER (NTL=64,NMESH=1083)
(0037) IMPLICIT REAL*8 (A-G,O-Z),INTEGER (H-N)
(0038) COMMON//X1(NTL),Y1(NTL),QMESH(NMESH),DHC,DHM,Q,MESH,ITL,NALPHA,N
(0039) 1BETA,NGAMMA,N,M,NZM,S1,S3,OMEGA,EPSIR,YH,PHIMSH(NMESH),KRSW1,KRSW2
(0040) 2,KRSW3,KRSW4,KRSW5,KRSW6,KQSW1,KQSW2,KQSW3,KQSW4,KQSW5,KQSW6,S,SCF
(0041) 3,GAMZ,LUSW1,LUSW2,LUSW3,LUSW4,LUSW5,LUSW6,FLJFAC,FMAX,LJSW1,LJSW2,
(0042) 4LJSW3,LJSW4,LJSW5,LJSW6,QDSAV(8,NTL)
(0043) DIMENSION XCM(NTL),YCM(NTL),FTXCM(NTL),FTYCM(NTL)
(0044) C***** CONSTANTS *****
(0045) NMOLEC=ITL/2
(0046) CMXSYS=0.00+00
(0047) CMYSYS=0.00+00
(0048) RDOTF=0.00+00
(0049) C***** FIND SYSTEM CENTER-OF-MASS
(0050) DO 20 I=1,ITL,1
(0051) CMXSYS=CMXSYS+X1(I)
(0052) 20 CMYSYS=CMYSYS+Y1(I)
(0053) CMXSYS=CMXSYS/ITL
(0054) CMYSYS=CMYSYS/ITL
(0055) C***** CALCULATE DOT PRODUCT OF POSITION AND FORCE VECTORS
(0056) DO 30 I=2,ITL,2
(0057) IM1=I-1
(0058) RDOTF=RDOTF+FTXCM(IM1)*(X1(IM1)-CMXSYS)+FTYCM(IM1)*(Y1(IM1)-CM
(0059) 8YSYS)
(0060) RDOTF=RDOTF+FTXCM(I)*(X1(I)-CMXSYS)+FTYCM(I)*(Y1(I)-CMYSYS)
(0061) C WRITE(11,400) RDOTF,FTXCM(I),FTYCM(I),CMXSYS,CMYSYS
(0062) 30 CONTINUE
(0063) C***** CALCULATE P*V IN ERGS (PI*A = 1.5*NKT + 0.5*RDOTF)
(0064) PVERG=1.50+00*NMOLEC*0.1380-15*TBAR+0.50+07*RDOTF
(0065) PRINT 400 RDOTF,FTXCM(ITL),FTYCM(ITL),CMXSYS,CMYSYS
(0066) 400 FORMAT(2X,5D11.3)
(0067) 500 RETURN
(0068) END
(0069) C*****

```

```

( 11)
(0002) C*****
(0003) SUBROUTINE PHOTO (NDT)
(0004) C * PURPOSE: PLOT POSITIONS OF PARTICLES GIVEN X AND
(0005) C * Y COORDINATES.
(0006) INTEGER*2 NTL,NMESH
(0007) PARAMETER (NTL=064,NMESH=1083)
(0008) IMPLICIT REAL*8 (A-G,O-Z),INTEGER (H-N)
(0009) COMMON//X1(NTL),Y1(NTL),QMESH(NMESH),DHC,DHM,Q,MESH,ITL,NALPHA,N
(0010) 1BETA,NGAMMA,N,M,NZM,S1,S3,OMEGA,EPSIR,YH,PHIMSH(NMESH),KRSW1,KRSW2
(0011) 2,KRSW3,KRSW4,KRSW5,KRSW6,KQSW1,KQSW2,KQSW3,KQSW4,KQSW5,KQSW6,S,SCF
(0012) 3,GAMZ,LUSW1,LUSW2,LUSW3,LUSW4,LUSW5,LUSW6,FLJFAC,FMAX,LJSW1,LJSW2,
(0013) 4LJSW3,LJSW4,LJSW5,LJSW6,QDSAV(8,NTL)
(0014) DIMENSION KCELL(32,62),LIT(72)
(0015) DATA KSTAR/'*'/,KSPACE/' '/,KDOT/'.'/,KPOS/'+'/,KNEG/'-'/
(0016) DATA LIT(01),LIT(03),LIT(05),LIT(07),LIT(09)/'0','1','2','3','4'/
(0017) DATA LIT(11),LIT(13),LIT(15),LIT(17),LIT(19)/'5','6','7','8','9'/
(0018) DATA LIT(21),LIT(23),LIT(25),LIT(27),LIT(29)/'A','B','C','D','E'/
(0019) DATA LIT(31),LIT(33),LIT(35),LIT(37),LIT(39)/'F','G','H','I','J'/
(0020) DATA LIT(41),LIT(43),LIT(45),LIT(47),LIT(49)/'K','L','M','N','O'/
(0021) DATA LIT(51),LIT(53),LIT(55),LIT(57),LIT(59)/'P','Q','R','S','T'/
(0022) DATA LIT(61),LIT(63),LIT(65),LIT(67),LIT(69)/'U','V','W','X','Y'/
(0023) DATA LIT(71)/'Z'/
(0024) TCPU1=CTIMSA(CPUTIM)
(0025) C***** CONSTANTS *****
(0026) CELLX=DHM*(NALPHA-1)
(0027) CELLY=DHM*(NBETA-1)
(0028) SCALEX=0.3D+02/CELLX
(0029) SCALEY=0.3D+02/CELLY
(0030) C***** CLEAR KCELL AND SET BOUNDARIES *****
(0031) DO 10 J=2,61
(0032) DO 10 I=2,31
(0033) 10 KCELL(I,J)=KSPACE
(0034) DO 20 I=1,32
(0035) KCELL(I,1)=KSTAR
(0036) 20 KCELL(I,62)=KSTAR
(0037) DO 30 J=2,61
(0038) KCELL(1,J)=KSTAR
(0039) 30 KCELL(32,J)=KSTAR
(0040) C***** MODIFY KCELL ACCORDING TO POSITIONS *****
(0041) DO 40 I=2,ITL,2
(0042) IM=I-1
(0043) NROWI=31-IDINT(Y1(I)*SCALEY)
(0044) NCOLI=2-IDINT(X1(I)*SCALEX)+2
(0045) NCOLIP=NCOLI+1
(0046) KCELL(NROWI,NCOLI)=KPOS
(0047) KCELL(NROWI,NCOLIP)=LIT(IM)
(0048) NROWI=31-IDINT(Y1(I-1)*SCALEY)
(0049) NCOLI=2-IDINT(X1(I-1)*SCALEX)+2
(0050) NCOLIP=NCOLI+1
(0051) KCELL(NROWI,NCOLI)=KNEG
(0052) 40 KCELL(NROWI,NCOLIP)=LIT(IM)
(0053) C***** WRITE KCELL *****
(0054) WRITE(11,202) NDT
(0055) DO 50 I=1,32
(0056) 50 WRITE(11,201) (KCELL(I,J),J=1,62)
(0057) C***** FORMAT STATEMENTS *****
(0058) 201 FORMAT(1H,62A1)

```

```
(  19)  202  FORMAT('1***** OUTPUT FROM PHOTO AT NDT= ',I5,' *****')
(0000)  500  RETURN
(0061)      END
(0062)  C*****
```


APPENDIX B

PROGRAM TO SOLVE MATRIX EQUATION

```

( 1) C***** PROGRAM TITLE: LUDEC3 *****
(0002) C * LUDEC3 DIFFERS FROM LUDEC2 IN THAT
(0003) C EL AND U ARE WRITTEN TO FILE ELU.
(0004) C PROC DISABLED TO SAVE SPACE.
(0005) C * BOUNDARY CONDITIONS ON Z: (SWITCH=KSW8; SEE PAGES 350.1 - 350.16)
(0006) C PHI(U)=OMEGA*PHI(P)
(0007) C OR PHI(U)=(1+OMEGA)*PHI(P)-OMEGA*PHI(D) (BACKWARD DERIVATIVE)
(0008) C NOTE THAT OMEGA MUST BE SET TO 1.0 IN BACKWARD DERIVATIVE FORM.
(0009) C * SYMMETRY IN Z: IF THE Z=C PLANE SEES THE SAME ENVIRONMENT ABOVE
(0010) C AND BELOW, THEN COMPUTATIONAL EFFORT MAY BE HALVED
(0011) C BY DEALING ONLY WITH THE UPPER PORTION OF THE
(0012) C PRISM. SET KSW10 EQUAL TO ONE IN THIS CASE.
(0013) C SEE NOTES, PAGES 350.17 - 350.25.
(0014) C * PARAMETERS SHOWN ASSUME THAT NALPHA=21, NBETA=21, & NGAMMA=03;
(0015) C * NOTE THAT NROWS=NALPHA*NBETA*NGAMMA
(0016) C * NCOLS=(NBETA*NGAMMA)+1
(0017) C * NR1=NALPHA
(0018) C * NC1=NBETA
(0019) C * NOTE THAT CHARGE COORDINATES MUST BE CONSISTENT WITH VALUES
(0020) C * OF NALPHA, NBETA, NGAMMA, AND KSW10.
(0021) C INTEGER*2 NROWS,NCOLS,NR1,NC1
(0022) C PARAMETER (NROWS=1323,NCOLS=64)
(0023) C PARAMETER (NR1=21,NC1=21)
(0024) C IMPLICIT REAL*8(A-G,O-Z),INTEGER(H-N)
(0025) C INTEGER*2 A$READ,A$SAME,A$WRIT,A$RDWR
(0026) C COMMON /A/ A(NROWS,NCOLS)
(0027) C COMMON /EL/ EL(NROWS,NCOLS)
(0028) C COMMON /U/ U(NROWS,NCOLS)
(0029) C COMMON /PROC/ PROC(NROWS,NCOLS)
(0030) C DIMENSION UINV(NROWS),Q(NROWS)
(0031) C DIMENSION QIN(NROWS),APHI(NROWS)
(0032) C DIMENSION IQ(NC1,NR1),VWE(NC1),VSN(NR1)
(0033) C DATA A$READ,A$SAME,A$WRIT,A$RDWR/1,0,2,3/
(0034) C CALL ATTDEV(12,7,3,8)
(0035) C CALL OPENSA(A$READ+A$SAME,'LUDATR',6,5)
(0036) C CALL OPENSA(A$WRIT+A$SAME,'LU3OPTB',7,7)
(0037) C CALL OPENSA(A$RDWR+A$SAME,'ELUB',4,8)
(0038) C TCPU1=CTIMEA(CPUTIM)
(0039) C***** NALPHA, NBETA, AND NGAMMA ARE TAKEN TO BE ODD *****
(0040) 1 READ(02,220) NALPHA,NBETA,NGAMMA,S1,S3,OMEGA
(0041) READ(09,221) KSW1,KSW2,KSW3,KSW4,KSW5,KSW6,KSW7,KSW8,KSW9,KSW10
(0042) READ(09,222) EPSIR,YH
(0043) C***** SWITCHES (KSW'S) *****
(0044) C *INDEX* *IF SWITCH=1, THEN*
(0045) C 1 CALCULATE PROFILES ON 1/R BASIS
(0046) C 2 CALCULATE A PHI AND WRITE QIN, A PHI, DIFFT, Q
(0047) C 3 WRITE Q IN CENTER XY-PLANE
(0048) C 4 WRITE Q FROM WEST TO EAST
(0049) C 5 WRITE Q AND UINV AFTER FORWARD SUBSTITUTION
(0050) C 6 WRITE EL AND U TO FILE ELU
(0051) C 7 WRITE EL AND U TO FILE ELU; THEN STOP (ALSO SET KS
(0052) C 8 USE BACKWARD-DERIVATIVE BC ON Z
(0053) C 9 FIND PRODUCT OF EL AND U; WRITE A,PROD,EL,U
( 4) C 10 TREAT PROBLEM AS SYMMETRIC IN Z
( 5) C***** CONSTANTS *****
(0056) N=NALPHA*NBETA*NGAMMA
(0057) M=NBETA*NGAMMA
(0058) NAB=NALPHA*NBETA

```

```

( 19)      NGP1=NGAMMA+1
(0060)      NGM1=NGAMMA-1
(0061)      NAF1D2=(NALPHA+1)/2
(0062)      NBP1D2=(NBETA+1)/2
(0063)      NP1=N+1
(0064)      NMM=N-M
(0065)      NM2M=NM-M
(0066)      MP1=M+1
(0067)      MMNG=M-NGAMMA
(0068)      SIGMA=0.2D+01*(0.1D+01+S1+S3)
(0069)      S1N=-S1
(0070)      S3N=-S3
(0071)      DNEG1=-0.1D+01
(0072)      S3NOM=S3N*OMEGA
(0073)      EPYH=EPSIR*YH
(0074)      RS1S3=DSQRT(S1*S3)
(0075)      C=14.32688751D+00/EPYH
(0076)      NZM=NGP1/2
(0077)      IC=NP1/2
(0078)      ICZEQ1=IC-NGM1/2
(0079)      BDFAC=0.5D+00
(0080)      IF(KSW8.EQ.1) BDFAC=(0.1D+01-OMEGA)*0.5D+00
(0081)      WRITE(11,249)
(0082)      WRITE(11,250)
(0083)      WRITE(11,234) NALPHA,NBETA,NGAMMA,S1,S3,OMEGA,EPYH
(0084)      WRITE(11,251) KSW1,KSW2,KSW3,KSW4,KSW5,KSW6,KSW7,KSW8,KSW9,KSW10
(0085) C***** SET UP Q VECTOR *****
(0086)      CNP=0.1D+01
(0087)      DO 05 I=1,N
(0088)          QIN(I)=0.0D+00
(0089) 05      Q(I)=QIN(I)
(0090) C      * THE VALUE OF QUNIT IS DERIVED ON PAGES 408-409 OF THESIS NOTES.
(0091) C      * THE VALUE DEPENDENT UPON S1, S3, EPSIR, AND YH IS USED ORDINARILY
(0092)      QUNIT=180.7909605D+00*RS1S3*CNP/EPYH
(0093) C      QUNIT=90.39549D+00/NGAMMA
(0094) C      QIN(004)=QUNIT*0.64D+00
(0095) C      QIN(011)=QUNIT*0.16D+00
(0096) C      QIN(081)=QUNIT*0.16D+00
(0097) C      *CHARGES AT (5,5),(20,30),(25,5),(40,30),(5,45),(35,45),(20,40),
(0098) C          (40,15)
(0099) C      QIN(109)=QUNIT
(0100) C      QIN(451)=-QUNIT
(0101) C      QIN(505)=QUNIT
(0102) C      QIN(847)=-QUNIT
(0103) C      QIN(181)=-QUNIT
(0104) C      QIN(775)=QUNIT
(0105) C      QIN(469)=QUNIT
(0106) C      QIN(820)=-QUNIT
(0107) C      Q(109)=QIN(109)
(0108) C      Q(451)=QIN(451)
(0109) C      Q(505)=QIN(505)
(0110) C      Q(847)=QIN(847)
(0111) C      Q(181)=QIN(181)
( 2) C      Q(775)=QIN(775)
( 3) C      Q(469)=QIN(469)
(0114) C      Q(820)=QIN(820)
(0115) C      *CHARGE AT CENTER
(0116) C      IC9=IC

```

```

( 17) C      IF(KSW10.EQ.1) ICQ=ICZER1
( 18) C      QIN(ICQ)=QUNIT
( 19) C      Q(ICQ)=QIN(ICQ)
( 20) C      *CHARGES AT (2,2) AND (6,6) FOR (5,5,5) Z-SYMMETRIC CASE
( 21) C      QIN(91)=QUNIT
( 22) C      QIN(31)=-QUNIT
( 23) C      Q(91)=QIN(91)
( 24) C      Q(31)=QIN(31)
( 25) C***** SET UP NORMAL ELEMENTS OF POISSON A *****
( 26) C      DO 10 J=1,MP1
( 27) C      DO 10 I=1,N
( 28) 10      A(I,J)=0.0D+00
( 29) C      DO 20 I=1,N
( 30) 20      A(I,1)=SIGMA
( 31) C      DO 30 K=1,NAB
( 32) C      KG=NGAMMA*(K-1)
( 33) C      INI=KG+1
( 34) C      IMAX=KG+NGM1
( 35) C      DO 30 I=INI,IMAX
( 36) 30      A(I,2)=S3N
( 37) C      IF(KSW8.EQ.0) GO TO 37
( 38) C      IMAX=NP1-NGAMMA
( 39) C      DO 34 J=1,IMAX,NGAMMA
( 40) 34      A(I,2)=A(I,2)-S3NQM
( 41) 37      DO 40 K=1,NALPHA
( 42) C      KG=M*(K-1)
( 43) C      INI=KG+1
( 44) C      IMAX=KG+MMNG
( 45) C      DO 40 I=INI,IMAX
( 46) 40      A(I,NGP1)=DNEG1
( 47) C      DO 50 I=1,NMM
( 48) 50      A(I,MP1)=S1N
( 49) C***** MODIFY ELEMENTS USING FLIPPING REPLICATION *****
( 50) C      IMAX=N-NGM1
( 51) C      IF(KSW8.EQ.1) S3NOM=S3NOM+S3N
( 52) C      DO 52 I=1,IMAX,NGAMMA
( 53) C      IP=I+NGM1
( 54) C      IF(KSW10.EQ.0) A(I,1)=A(I,1)+S3NOM
( 55) C      IF(KSW10.EQ.1) A(I,2)=A(I,2)/BDFAC
( 56) 52      A(IP,1)=A(IP,1)+S3NOM
( 57) C      DO 57 J=1,NGAMMA
( 58) C      IMAX=MMNG+J
( 59) C      DO 53 I=J,IMAX,NGAMMA
( 60) C      IP=I+NMM
( 61) C      A(I,1)=A(I,1)+S1N
( 62) 53      A(IP,1)=A(IP,1)+S1N
( 63) C      IMIN=J
( 64) C      IMAX=NMM +J
( 65) C      DO 56 I=IMIN,IMAX,M
( 66) C      IP=I+MMNG
( 67) 55      A(I,1)=A(I,1)+DNEG1
( 68) 56      A(IP,1)=A(IP,1)+DNEG1
( 69) 57      CONTINUE
( 70) C      TCPU2=CTIMEA(CPUTIM)
( 71) C      TCPU=TCPU2-TCPU1
( 72) C      PRINT 301 TCPU
( 73) C      WRITE(11,301) TCPU
( 74) C***** CALCULATE EL AND U *****

```

```

( 5)      DO 60 J=1,N
(0176)      60      EL(J,1)=0.10+01
(0177)      DO 80 K=1,N
(0178)          KPM=K+M
(0179)          KM1=K-1
(0180)          KMM=K-M+1
(0181)      DO 64 J=K,KPM
(0182)          JP=J-K+1
(0183)          SUMLU=0.00+00
(0184)          IF(K.EQ.1) GO TO 64
(0185)      DO 62 I=1,KM1
(0186)          JQ=J-I+1
(0187)          IF(JQ.GT.MP1) GO TO 62
(0188)          KQ=K-I+1
(0189)          SUMLU=SUMLU+EL(I,KQ)*U(I,JQ)
(0190)      62      CONTINUE
(0191)      64      U(K,JP)=A(K,JP)-SUMLU
(0192)          IF(K.EQ.N) GO TO 80
(0193)          INI=1
(0194)      66      IF(K.GT.M) INI=KMM
(0195)      68      DO 75 J=INI,K
(0196)          KP=K-J+2
(0197)          JM1=J-1
(0198)          SUMLU=0.00+00
(0199)          IF(J.EQ.1) GO TO 71
(0200)      DO 70 I=1,JM1
(0201)          KQ=K-I+2
(0202)          IF(KQ.GT.MP1) GO TO 70
(0203)          JQ=J-I+1
(0204)          SUMLU=SUMLU+EL(I,KQ)*U(I,JQ)
(0205)      70      CONTINUE
(0206)      71      JD=J
(0207)          FEFAC=0.10+01
(0208)      C      IF((KSWR.EQ.1).AND.(KP.EQ.2)) GO TO 72
(0209)      C      GO TO 75
(0210)          IF(KP-2) 75,72,75
(0211)      72      JD=N-J
(0212)          JDM1=JD-1
(0213)          IF((KSW10.EQ.1).AND.(MOD(JDM1,NGANMA).EQ.0)) FEFAC=BDFAC
(0214)      75      EL(J,KP)=(FEFAC*A(JD,KP)-SUMLU)/U(J,1)
(0215)          NTEST=MOD(K,100)
(0216)          IF(NTEST.GT.0) GO TO 80
(0217)      78      TCPU2=CTIMSA(CPUTIM)
(0218)          TCPU=TCPU2-TCPU1
(0219)          PRINT 307 K,TCPU
(0220)          WRITE(11,307) K,TCPU
(0221)          PRINT 308 EL(K,KP)
(0222)          WRITE(11,308) EL(K,KP)
(0223)      80      CONTINUE
(0224)          TCPU2=CTIMSA(CPUTIM)
(0225)          TCPU=TCPU2-TCPU1
(0226)          PRINT 302 TCPU
(0227)          WRITE(11,302) TCPU
( 8)      DO 92 I=1,N
(0229)      92      UINV(I)=0.10+01/U(I,1)
(0230)          IF(KSWR.EQ.0) GO TO 100
(0231)      C***** FIND THE PRODUCT OF EL AND U (NOW DISABLED) *****
(0232)      C 93      WRITE (11,202)

```

```

33)          DO 96 I=1,N
0234)          DO 96 J=1,MP1
0235)      C          PROD(I,J)=0.0
0236)          DO 96 K=1,I
0237)              IMK=I-K
0238)              IPRM=IMK+1
0239)              IF(IPRM.GT.MP1) GO TO 96
0240)              JPRM=IMK+J
0241)              IF(JPRM.GT.MP1) GO TO 96
0242)      C          PROD(I,J)=PROD(I,J)+EL(K,IPRM)*U(K,JPRM)
0243)          96          CONTINUE
0244)      C***** WRITE A, PROD, EL, AND U (NOW DISABLED) *****
0245)      C          WRITE(11,204)
0246)      C          DO 98 J=1,MP1
0247)      C          WRITE(11,225) J
0248)      C          DO 98 I=1,N
0249)      C 98          WRITE(11,232) A(I,J),PROD(I,J),EL(I,J),U(I,J)
0250)      100          IF(KSW6.EQ.0) GO TO 105
0251)      C***** WRITE EL AND U TO FILE ELU *****
0252)          CRDATE=DATE$(DATE)
0253)          WRITE(12) CRDATE,NALPHA,NBETA,NGAMMA,N,M,KSW8,KSW10
0254)          WRITE(12) OMEGA,S1,S3,EPSIR,YH
0255)          DO 102 J=1,MP1
0256)              DO 102 I=1,N
0257)      102          WRITE(12) EL(I,J),U(I,J)
0258)      C          CALL RWNSA(8)
0259)      C          READ(12,310) CRDELU,NALELU,NBEELU,NGAELU,NELU,MELU
0260)      C          READ(12,311) OMELU,S1ELU,S3ELU,EPSELU,YHELU
0261)      C          PRINT 310 CRDELU,NALELU,NBEELU,NGAELU,NELU,MELU
0262)      C          PRINT 311 OMELU,S1ELU,S3ELU,EPSELU,YHELU
0263)          TCPU2=CTIM$(CPUTIM)
0264)          TCPU=TCPU2-TCPU1
0265)          PRINT 303 TCPU
0266)          WRITE(11,303) TCPU
0267)          IF(KSW7.EQ.1) GO TO 218
0268)      C***** FORWARD SUBSTITUTION *****
0269)      105          DO 120 I=1,N
0270)              IF(I.EQ.1) GO TO 118
0271)              IF(I.GE.MP1) GO TO 107
0272)              LMAX=I-1
0273)              GO TO 109
0274)      107          LMAX=M
0275)      109          DO 110 L=1,LMAX
0276)              IML=I-L
0277)              LP1=L+1
0278)      110          Q(I)=Q(I)-EL(IML,LP1)*Q(IML)
0279)      118          IF(KSW5.EQ.0) GO TO 120
0280)      119          WRITE(11,230) Q(I),UINV(I)
0281)      120          CONTINUE
0282)      C***** BACKWARD SUBSTITUTION *****
0283)          DO 140 I=1,N
0284)              IREV=MP1-I
0285)              IF(I.EQ.1) GO TO 140
0286)              IF(I.GE.MP1) GO TO 127
0287)              LMAX=I-1
0288)              GO TO 120
0289)      127          LMAX=M
0290)      129          DO 130 L=1,LMAX

```

```

(  1)      IREVPL=IREV+L
(0292)      LP1=L+1
(0293)      130      Q(IREV)=Q(IREV)-U(IREV,LP1)*Q(IREVPL)
(0294)      140      Q(IREV)=Q(IREV)*UINV(IREV)
(0295)      TCPU2=CTIM3A(CPUTIM)
(0296)      TCPU=TCPU2-TCPU1
(0297)      PRINT 304 TCPU
(0298)      WRITE(11,304) TCPU
(0299)      IF(KSW2.EQ.0) GO TO 170
(0300)  C***** CALCULATE A*POTENTIAL *****
(0301)      DO 166 I=1,N
(0302)          APHI(I)=0.0D+00
(0303)          IF(I.EQ.1) GO TO 165
(0304)          IM1=I-1
(0305)  C      * FOR ELEMENTS OF A LYING BELOW THE MAIN DIAGONAL:
(0306)      DO 164 K=1,IM1
(0307)          IMKP1=I-K+1
(0308)          IF(IMKP1.GT.MP1) GO TO 164
(0309)          KD=K
(0310)          FEFAC=0.1D+01
(0311)  C      IF((KSW8.EQ.1).AND.(IMKP1.EQ.2)) KD=N-K
(0312)          IF(IMKP1-2) 162,160,162
(0313)      160      KD=N-K
(0314)          KDM1=KD-1
(0315)          IF((KSW10.EQ.1).AND.(MOD(KDM1,NGAMMA).EQ.0)) FEFAC=BDFAC
(0316)      162      APHI(I)=APHI(I)+FEFAC*A(KD,IMKP1)*Q(K)
(0317)      164      CONTINUE
(0318)  C      * FOR ELEMENTS OF A LYING IN OR ABOVE THE MAIN DIAGONAL:
(0319)      165      DO 166 K=I,N
(0320)          KMIP1=K-I+1
(0321)          IF(KMIP1.GT.MP1) GO TO 166
(0322)          APHI(I)=APHI(I)+A(I,KMIP1)*Q(K)
(0323)      166      CONTINUE
(0324)      TCPU2=CTIM3A(CPUTIM)
(0325)      TCPU=TCPU2-TCPU1
(0326)      PRINT 305 TCPU
(0327)      WRITE(11,305) TCPU
(0328)  C***** WRITE QIN, APHI, DIFFI, AND Q *****
(0329)      WRITE(11,290)
(0330)      DO 168 I=1,N
(0331)          DIFFI=QIN(I)-APHI(I)
(0332)      168      WRITE(11,260) I,QIN(I),APHI(I),DIFFI,Q(I)
(0333)      170      IF(KSW3.EQ.0) GO TO 174
(0334)  C***** WRITE POTENTIALS IN CENTER XY-PLANE *****
(0335)      WRITE(11,292)
(0336)      IMAX=N2M+N-NGAMMA
(0337)      DO 172 I=N2M,IMAX,NGAMMA
(0338)      172      WRITE(11,233) I,Q(I)
(0339)      174      IF(KSW4.EQ.0) GO TO 178
(0340)  C***** WRITE POTENTIALS FOR WEST-TO-EAST PROFILE IN CENTER XY-PLANE *****
(0341)      WRITE(11,293)
(0342)      IMIN=N2M+MMNG/2
(0343)      IMAX=IMIN+N2M
(0344)      DO 176 I=IMIN,IMAX,M
(0345)          DIFFQ=0.0D+00
(0346)          IMM=I-M
(0347)          IF(I.GT.IMIN) DIFFQ=Q(I)-Q(IMM)
(0348)      176      WRITE(11,235) I,Q(I),DIFFQ

```

```

(  9) C***** WRITE POTENTIALS FOR SOUTH-TO-NORTH PROFILE IN CENTER XY-PLANE ***
(0350)      WRITE(11,294)
(0351)      IMIN=NZM+NMM/2
(0352)      IMAX=IMIN+MMNG
(0353)      DO 177 I=IMIN,IMAX,NGAMMA
(0354)          DIFFQ=0.0D+00
(0355)          IMNG=I-NGAMMA
(0356)          IF(I.GT.IMIN) DIFFQ=Q(I)-Q(IMNG)
(0357) 177      WRITE(11,235) I,Q(I),DIFFQ
(0358)      TCPU2=CTIMSA(CPUTIM)
(0359)      TCPU=TCPU2-TCPU1
(0360)      PRINT 306 TCPU
(0361)      WRITE(11,306) TCPU
(0362) 178      CONTINUE
(0363)          IF(KSW1.EQ.0) GO TO 218
(0364) C***** CALCULATE POTENTIAL PROFILES ON CHARGE-BY-CHARGE, 1/R BASIS *****
(0365) C      * IXPRO DETERMINES THE X-LOCATION OF THE SOUTH-TO-NORTH PROFILE
(0366) C      * JYPRO DETERMINES THE Y-LOCATION OF THE WEST-TO-EAST PROFILE
(0367)          IXPRO=NAP1D2
(0368)          JYPRO=NBP1D2
(0369)          DO 180 I=1,NALPHA
(0370)              DO 180 J=1,NBETA
(0371) 180          IQ(I,J)=0
(0372)          IQ(4,5)=1
(0373)          IQ(4,9)=1
(0374)          IQ(10,5)=1
(0375)          IQ(10,9)=1
(0376)          DO 186 K=1,NALPHA
(0377)              DO 186 I=1,NALPHA
(0378)                  DX=I-K
(0379)                  DO 186 J=1,NBETA
(0380)                      IF(IQ(I,J).EQ.0) GO TO 186
(0381)                      DY=J-JYPRO
(0382)                      IF(J.EQ.JYPRO) GO TO 182
(0383)                      GO TO 183
(0384) 182          VINC=0.1D+15
(0385)                      IF(I.EQ.K) GO TO 184
(0386) 183          R=DSQRT(DX**2+DY**2)
(0387)                      VINC=(C*IQ(I,J))/R
(0388) 184          VWE(K)=VWE(K)+VINC
(0389) 186          CONTINUE
(0390)          DO 192 L=1,NBETA
(0391)              DO 192 J=1,NBETA
(0392)                  DY=J-L
(0393)                  DO 192 I=1,NALPHA
(0394)                      IF(IQ(I,J).EQ.0) GO TO 192
(0395)                      DX=I-IXPRO
(0396)                      IF(I.EQ.IXPRO) GO TO 188
(0397)                      GO TO 189
(0398) 188          VINC=0.1D+15
(0399)                      IF(J.EQ.L) GO TO 190
(0400) 189          R=DSQRT(DX**2+DY**2)
(0401)                      VINC=(C*IQ(I,J))/R
(  12) 190          VSN(L)=VSN(L)+VINC
(  13) 192          CONTINUE
(0404)      WRITE(11,295)
(0405)      DO 194 K=1,NALPHA
(0406)          DIFFV=0.0D+00

```



```

(   7)      IF(K.GT.1) DIFFV=VWE(K)-VWE(K-1)
(0408) 194    WRITE(11,235) K,VWE(K),DIFFV
(0409)      WRITE(11,296)
(0410)      DO 196 L=1,NBETA
(0411)      DIFFV=0.00+00
(0412)      IF(L.GT.1) DIFFV=VSN(L)-VSN(L-1)
(0413) 196    WRITE(11,235) L,VSN(L),DIFFV
(0414) C***** CALCULATE POTENTIALS AT POINTS NEAR CHARGES ON 1/R PROFILES *****
(0415)      WRITE(11,297)
(0416)      DO 204 I=1,NALPHA
(0417)      DO 204 J=1,NBETA
(0418)      IF(IQ(I,J).EQ.0) GO TO 204
(0419)      IF(J.EQ.JYPRO) GO TO 200
(0420)      GO TO 204
(0421) 200    IP=I
(0422)      PX1=IP-0.50+00
(0423)      PX2=IP+0.50+00
(0424)      V1=0.00+00
(0425)      V2=0.00+00
(0426)      DO 202 K=1,NALPHA
(0427)      DO 202 L=1,NBETA
(0428)      IF(IQ(K,L).EQ.0) GO TO 202
(0429)      DX1=K-PX1
(0430)      DX2=K-PX2
(0431)      DY=L-JYPRO
(0432)      R1=DSQRT(DX1**2+DY**2)
(0433)      VINC1=(C*IQ(K,L))/P1
(0434)      V1=V1+VINC1
(0435)      R2=DSQRT(DX2**2+DY**2)
(0436)      VINC2=(C*IQ(K,L))/P2
(0437)      V2=V2+VINC2
(0438) 202    CONTINUE
(0439)      DIFF1=0.00+00
(0440)      IF(IP.GT.1) DIFF1=V1-VWE(IP-1)
(0441)      DIFF2=0.00+00
(0442)      IF(IP.LT.NALPHA) DIFF2=VWE(IP+1)-V2
(0443)      WRITE(11,236) IP,V1,DIFF1,V2,DIFF2
(0444) 204    CONTINUE
(0445)      WRITE(11,298)
(0446)      DO 210 J=1,NBETA
(0447)      DO 210 I=1,NALPHA
(0448)      IF(IQ(I,J).EQ.0) GO TO 210
(0449)      IF(I.EQ.IXPRO) GO TO 206
(0450)      GO TO 210
(0451) 206    JP=J
(0452)      PY1=JP-0.50+00
(0453)      PY2=JP+0.50+00
(0454)      V1=0.00+00
(0455)      V2=0.00+00
(0456)      DO 208 K=1,NALPHA
(0457)      DO 208 L=1,NBETA
(0458)      IF(IQ(K,L).EQ.0) GO TO 208
(0459)      DY1=L-PY1
(   0)      DY2=L-PY2
(   1)      DX=K-IXPRO
(0462)      R1=DSQRT(DX**2+DY1**2)
(0463)      VINC1=(C*IQ(K,L))/R1
(0464)      V1=V1+VINC1

```

```

(  55)          R2=DSQRT(DX**2+DY2**2)
(0466)          VINC2=(C*IQ(K,L))/R2
(0467)          V2=V2+VINC2
(0468)  208      CONTINUE
(0469)          DIFF1=0.00+00
(0470)          IF(JP.GT.1) DIFF1=V1-VSN(JP-1)
(0471)          DIFF2=0.00+00
(0472)          IF(JP.LT.NBETA) DIFF2=VSN(JP+1)-V2
(0473)          WRITE(11,236) JP,V1,DIFF1,V2,DIFF2
(0474)  210      CONTINUE
(0475)          TCPU2=CTIMSA(CPUTIM)
(0476)          TCPU=TCPU2-TCPU1
(0477)          PRINT 309 TCPU
(0478)          WRITE(11,309) TCPU
(0479)  218      TCPU2=CTIMSA(CPUTIM)
(0480)          TCPU=TCPU2-TCPU1
(0481)          WRITE(11,223) TCPU
(0482)  220      FORMAT(3I3,2D24.16,D15.7)
(0483)  221      FORMAT(10I3)
(0484)  222      FORMAT(2D24.16)
(0485)  223      FORMAT(1H,2X,'**THE CPU TIME USED ON THIS JOB WAS ',D24.16,' S
(0486)  REC.')
```

```

(0487)  225      FORMAT(1H,2X,'DIAGONAL',I5)
(0488)  230      FORMAT(3D24.16)
(0489)  231      FORMAT(1H,15X,1D24.16)
(0490)  232      FORMAT(1H,15X,4D24.16)
(0491)  233      FORMAT(1H,2X,I5,10X,1D24.16)
(0492)  234      FORMAT(1H,2X,3I3,2D24.16,D15.7,2D24.16)
(0493)  235      FORMAT(1H,2X,I5,10X,2D24.16)
(0494)  236      FORMAT(1H,2X,I5,10X,4D24.16)
(0495)  240      FORMAT(1H,2X,'SUM.FT.U(I,1) FOR I=',I3)
(0496)  249      FORMAT(1H,2X,'***** LUDEC3 OUTPUT *****')
```

```

(0497)  250      FORMAT(1H,2X,'THE VALUES OF NALPHA, NBETA, NGAMMA, S1, S3, OMEG
(0498)  &A, EPSIR, AND YH ARE:')
```

```

(0499)  251      FORMAT(1H,2X,'SWITCH VALUES: ',10I3)
(0500)  260      FORMAT(1H,14,2X,D24.16,3(5X,D24.16))
(0501)  290      FORMAT(1H,9X,'**** Q VECTOR ****',9X,'**** A * POTENTIAL ****',
(0502)  &8X,'**** DIFFERENCE ****',6X,'**** POTENTIAL VECTOR ****')
```

```

(0503)  291      FORMAT(1H,2X,'THE VALUE OF THE CHARGE AT THE CENTER OF THE ARR
(0504)  &AY IS',1D24.16)
(0505)  292      FORMAT(1H,2X,'THE POTENTIALS IN THE CENTER XY-PLANE ARE AS FOL
(0506)  &LCWS: ')
```

```

(0507)  293      FORMAT(1H,2X,'THE POTENTIALS FOR THE CENTRAL WEST-TO-EAST PROF
(0508)  &ILE IN THE CENTER XY-PLANE ARE AS FOLLOWS (WITH DIFFERENCES): ')
```

```

(0509)  294      FORMAT(1H,2X,'THE POTENTIALS FOR THE CENTRAL SOUTH-TO-NORTH PR
(0510)  &OFILE IN THE CENTER XY-PLANE ARE AS FOLLOWS (WITH DIFFERENCES): ')
```

```

(0511)  295      FORMAT(1H,2X,'THE WEST-TO-EAST PROFILE CALCULATED ON A CHARGE-B
(0512)  &Y-CHARGE, 1/P BASIS IS GIVEN BELOW (WITH SEQUENTIAL DIFFERENCES): '
(0513)  &)
```

```

(0514)  296      FORMAT(1H,2X,'THE SOUTH-TO-NORTH PROFILE CALCULATED ON A CHARGE
(0515)  &-BY-CHARGE, 1/R BASIS IS GIVEN BELOW (WITH SEQUENTIAL DIFFERENCES)
(0516)  &: ')
```

```

(0517)  297      FORMAT(1H,2X,'THE POTENTIALS AT POINTS ONE-HALF UNIT W & E OF C
(  8)  &HARGES ON THE W-E PROFILE ARE: (WITH NEXT-POINT DIFFERENCES)')
```

```

(0519)  298      FORMAT(1H,2X,'THE POTENTIALS AT POINTS ONE-HALF UNIT S & N OF C
(0520)  &HARGES ON THE S-N PROFILE ARE: (WITH NEXT-POINT DIFFERENCES)')
```

```

(0521)  301      FORMAT(10X,'**A SET UP AND MODIFIED AFTER ',D16.8,' CPU SEC.')
```

```

(0522)  302      FORMAT(10X,'**EL AND U CALCULATED AFTER ',D16.8,' CPU SEC.')
```

```
( 3) 303 FORMAT(10X,'**EL AND U WRITTEN TO ELU AFTER ',D16.8,' CPU SEC.  
(0524) &')  
(0525) 304 FORMAT(10X,'**FORWARD AND BACKWARD SUBSTITUTION COMPLETED AFTER  
(0526) &',D16.8,' CPU SEC.')(0527) 305 FORMAT(10X,'**A*POTENTIAL CALCULATED AFTER ',D16.8,' CPU SEC.')(0528) 306 FORMAT(10X,'**POTENTIALS WRITTEN AFTER ',D16.8,' CPU SEC.')(0529) 307 FORMAT(15X,'**LOOP 80 COMPLETED WITH K = ',I4,' AFTER',D16.8,' C  
(0530) &PU SEC.')(0531) 308 FORMAT(20X,'**EL(K,KP) = ',D24.16)  
(0532) 309 FORMAT(10X,'**1/R PROFILES WRITTEN AFTER ',D16.8,' CPU SEC.')(0533) 310 FORMAT(A9,7I5)  
(0534) 311 FORMAT(SD15.7)  
(0535) 500 STOP  
(0536) END
```

APPENDIX C

PROGRAM TO COMPUTE POTENTIAL DUE TO
AN INFINITE ARRAY OF DISCRETE CHARGES

```

( 1) C***** PROGRAM TITLE: PWSUM *****
(0002) C          * CREATED 23 OCTOBER 1978
(0003)          INTEGER*2 ASREAD,ASSAMF,ASWRIT,NSOL
(0004)          PARAMETER (NSOL=400)
(0005)          IMPLICIT REAL*8(A-G,O-Z),INTEGER(H-N)
(0006)          DIMENSION SIFRAC(100),XC(100),YC(100),ZC(100),XF(20),YP(20),ZP(2
(0007)          &0)
(0008)          DATA ASREAD,ASSAMF,ASWRIT/1,0,2/
(0009)          CALL OPNSA(ASREAD+ASSAMF,'PWSDAT3',7,5)
(0010)          CALL OPNSA(ASWRIT+ASSAMF,'PWSOPT3',7,7)
(0011)          TCPU1=CTIMSA(CPUTIM)
(0012) C***** SWITCHES (KSW'S) *****
(0013) C          *INDEX*          *IF SWITCH=1, THEN*
(0014) C              1              NOT USED
(0015) C              2              NOT USED
(0016) C              3              NOT USED
(0017) C              4              NOT USED
(0018) C              5              NOT USED
(0019) C              6              NOT USED
(0020) C***** READ DATA FROM PWSDAT & WRITE IT TO PWSOPT *****
(0021)          READ(09,201)NPTMAX,NCHGMX,NSQMAX,NSQWR1,NSQWR2,S,EPSIR
(0022)          READ(09,202)(XP(I),I=1,NPTMAX)
(0023)          READ(09,202)(YP(I),I=1,NPTMAX)
(0024)          READ(09,202)(ZP(I),I=1,NPTMAX)
(0025)          READ(09,202)(SIFRAC(I),I=1,8)
(0026)          READ(09,202)(XC(I),I=1,8)
(0027)          READ(09,202)(YC(I),I=1,8)
(0028)          READ(09,202)(ZC(I),I=1,8)
(0029)          READ(09,202)(SIFRAC(I),I=9,16)
(0030)          READ(09,202)(XC(I),I=9,16)
(0031)          READ(09,202)(YC(I),I=9,16)
(0032)          READ(09,202)(ZC(I),I=9,16)
(0033)          READ(09,202)(SIFRAC(I),I=17,24)
(0034)          READ(09,202)(XC(I),I=17,24)
(0035)          READ(09,202)(YC(I),I=17,24)
(0036)          READ(09,202)(ZC(I),I=17,24)
(0037)          READ(09,202)(SIFRAC(I),I=25,32)
(0038)          READ(09,202)(XC(I),I=25,32)
(0039)          READ(09,202)(YC(I),I=25,32)
(0040)          READ(09,202)(ZC(I),I=25,32)
(0041)          WRITE(11,201)NPTMAX,NCHGMX,NSQMAX,NSQWR1,NSQWR2,S,EPSIR
(0042)          WRITE(11,202)(XP(I),I=1,NPTMAX)
(0043)          WRITE(11,202)(YP(I),I=1,NPTMAX)
(0044)          WRITE(11,202)(ZP(I),I=1,NPTMAX)
(0045)          WRITE(11,202)(SIFRAC(I),I=1,8)
(0046)          WRITE(11,202)(XC(I),I=1,8)
(0047)          WRITE(11,202)(YC(I),I=1,8)
(0048)          WRITE(11,202)(ZC(I),I=1,8)
(0049)          WRITE(11,202)(SIFRAC(I),I=9,16)
(0050)          WRITE(11,202)(XC(I),I=9,16)
(0051)          WRITE(11,202)(YC(I),I=9,16)
(0052)          WRITE(11,202)(ZC(I),I=9,16)
(0053)          WRITE(11,202)(SIFRAC(I),I=17,24)
( 4)          WRITE(11,202)(XC(I),I=17,24)
(0055)          WRITE(11,202)(YC(I),I=17,24)
(0056)          WRITE(11,202)(ZC(I),I=17,24)
(0057)          WRITE(11,202)(SIFRAC(I),I=25,32)
(0058)          WRITE(11,202)(XC(I),I=25,32)

```

```

( 19)      WRITE(11,202)(YC(I),I=25,32)
(0060)      WRITE(11,202)(ZC(I),I=25,32)
(0061)      C***** CONSTANTS *****
(0062)      FAC=0.14387D+02/(EPSIR*S)
(0063)      C***** NORMALIZE COORDINATES WRT S *****
(0064)      DO 10 I=1,NPTMAX
(0065)          XP(I)=XP(I)/S
(0066)          YP(I)=YP(I)/S
(0067)      10      ZP(I)=ZP(I)/S
(0068)      DO 15 I=1,NCHGMX
(0069)          XC(I)=XC(I)/S
(0070)          YC(I)=YC(I)/S
(0071)      15      ZC(I)=ZC(I)/S
(0072)      C***** PRINCIPAL CALCULATION *****
(0073)      DO 30 NPT=1,NPTMAX
(0074)          POT=0.0D+00
(0075)          DO 30 NCHG=1,NCHGMX
(0076)      C          *RELATIVIZE COORDINATES
(0077)              XR=XP(NPT)-XC(NCHG)
(0078)              YR=YP(NPT)-YC(NCHG)
(0079)              ZR=ZP(NPT)-ZC(NCHG)
(0080)              ZRSQ=ZR*ZR
(0081)              FACN=SIFRAC(NCHG)*FAC
(0082)      C          *COMPUTE POTENTIAL DUE TO CHARGE AT ORIGIN
(0083)              DISTR=((XR*XR)+(YR*YR)+ZRSQ)**0.5D+00
(0084)              IF(DISTR.GT.0.1D-05) POT=POT+FACN/DISTR
(0085)      C          *COMPUTE POTENTIAL DUE TO SURROUNDING SQUARES
(0086)              DO 30 NSQ=1,NSQMAX
(0087)                  NSQP1=NSQ+1
(0088)                  IJMAX=2*NSQ+1
(0089)                  INCR=IJMAX-1
(0090)                  DELPOT=0.0D+00
(0091)                  DO 20 I=1,IJMAX
(0092)                      IP=I-NSQP1
(0093)                      XRMIP=XR-IP
(0094)                      XRZRSQ=XRMIP*XRMIP+ZRSQ
(0095)                      INC=INCR
(0096)                      IF(ABS(IP).EQ.NSQ) INC=1
(0097)                      DO 20 J=1,IJMAX,INC
(0098)                          JP=J-NSQP1
(0099)                          YRMJP=YR-JP
(0100)                          DISTR=(XRZRSQ+YRMJP*YRMJP)**0.5D+00
(0101)      20          DELPOT=DELPOT+FACN/DISTR
(0102)                  POT=POT+DELPOT
(0103)                  IF(MOD(NSQ,NSQWR1).EQ.0) WRITE(11,203) NPT,NCHG,NSQ,DELPOT
(0104)      8,POT
(0105)                  IF(MOD(NSQ,NSQWR2).EQ.0) WRITE(01,203) NPT,NCHG,NSQ,DELPOT
(0106)      8,POT
(0107)      30      CONTINUE
(0108)      C***** FORMAT STATEMENTS *****
(0109)      201      FORMAT(5I5,2D15.8)
(0110)      202      FORMAT(8F8.3)
(0111)      203      FORMAT(3I5,2D15.8)
(  2)      204      FORMAT(2D8.1)
(0113)      300      FORMAT(6D15.8)
(0114)      301      FORMAT('OK2')
(0115)      302      FORMAT(2D15.8,2I5)
(0116)      500      STOP

```

C***** PROGRAM TITLE: PWSUM *****

151

(17)

END

APPENDIX D

PRESSURE-DENSITY ISOTHERMS FOR IMPERFECT GAS OF STOCKMAYER PARTICLES

It may be of interest to investigate the behavior of the pressure-density isotherms of an imperfect gas of Stockmayer particles: that is, point-particles whose total interaction energy is the sum of the Lennard-Jones 6-12 interaction energy and the interaction energy of two point-dipoles. Writing the virial expansion in terms of the number of moles per unit volume (z), we have

$$(1) \quad \frac{P}{NKT} = z + B(T)z^2 + C(T)z^3 + \dots,$$

where N is Avogadro's number. We shall truncate this expression after the third term, so that we may use published analytical results for the Stockmayer energy (Hirschfelder, et al., 1954, pp. 209-222, 1147-1149, 1154). The second virial coefficient, $B(T)$, is obtained from integrals involving the interaction energy of two particles, while the third virial coefficient, $C(T)$, is obtained from integrals involving the interaction energy of three particles (Hirschfelder, et al., 1954, pp. 148-153). The two-particle Stockmayer interaction energy has the form (Hirschfelder, et al., 1954, pp. 210-111)

$$(2) \quad U = 4E \left[\left(\frac{s}{r} \right)^{12} - \left(\frac{s}{r} \right)^6 \right] - \frac{d^2}{r^3} [2\cos x_1 \cos x_2 - \sin x_1 \sin x_2 \cos(y_2 - y_1)],$$

where d is the dipole moment; x_1 , x_2 , y_1 , and y_2 describe the orientation of the dipole; E and s determine the depth and position of the Lennard-Jones energy minimum. Hence, the virial coefficients are dependent not only upon T , but also upon E , s , and d .

Equation (1) is a cubic relation between P and z , so we may expect, for certain values of $B(T,E,s,d)$ and $C(T,E,s,d)$, that as many as three distinct values of z will correspond to a single value of P . Since $P(z)$ is a thermodynamic state-function, the P -to- z correspondence must be one-to-one. This means that the portion of the P - z curve which shows more than one value of z corresponding to the same value of P must be modified. A straight line parallel to the z -axis is ordinarily used to connect the two single-valued parts of the P - z curve; this segment marks a phase transition.

From the algebra of cubic equations, one can show that the transition region will not appear unless B^2 is greater than $3C$. For purposes of illustration, we may satisfy this condition by giving T, E, s , and d the following values: 403.27°K , 0.3975×10^{-13} ergs, $4.0\text{\AA}$, and 2.685 debyes. In this case $B(T,E,s,d)$ is found to be $-1.1281 \times 10^2 \text{cm}^3$ and $C(T,E,s,d)$ is found to be $3.8417 \times 10^3 \text{cm}^6$; equation (1) then becomes

$$P = z(3.3502 \times 10^{10} \text{ergs})(1.0 - 112.81z \text{cm}^3 + 3841.7 z^2 \text{cm}^6).$$

This relation is plotted in figure D.1. In contrast, when E, s and d remain the same, but T is 460.88°K , we find that B^2 is not greater than $3C$. In this case, $B(T,E,s,d)$ is found to be $-0.86188 \times 10^2 \text{cm}^3$ and $C(T,E,s,d)$ is $3.48422 \times 10^3 \text{cm}^6$; equation

(1) becomes

$$P = z(3.8288 \times 10^{10} \text{ ergs})(1.0 - 86.188 z \text{ cm}^3 + 3484.22 z^2 \text{ cm}^6).$$

This relation is also plotted in figure D.1.

The values of E , s , and d used here are 300, 1.0, and 0.1413 times the values used in the molecular-dynamics simulation. However, the simulation did not use the spherically-symmetric particles with embedded point dipoles which were used in this simple, virial calculation. Furthermore, if one assumes a film thickness equal to the length of one fully-extended phospholipid molecule, the molecular areas in the simulation correspond to values of $z(0.5 \times 10^{-3} \text{ cm}^{-3}$ to $1.25 \times 10^{-3} \text{ cm}^{-3})$ which are quite far from the transition region shown in figure D.1.

Figure D.1

Pressure-density isotherms of an imperfect
gas of Stockmayer particles

BIBLIOGRAPHY

BIBLIOGRAPHY

- Adam, N.K. (1968). "The Physics and Chemistry of Surfaces", 3rd edition, New York: Dover.
- Adam, N.K., and Jessop, G. (1926). The structure of thin films: part VIII - expanded films, Proc. Roy. Soc. Lon. Ser. A 112:362.
- Adamson, A.W. (1976). "Physical Chemistry of Surfaces", 3rd edition, New York: Wiley.
- Alder, B.J., and Wainwright, T.E. (1959). Studies in molecular dynamics. I. General method, J. Chem. Phys. 31(2):459.
- Alder, B.J., and Wainwright, T.E. (1960). Studies in molecular dynamics. II. Behavior of a small number of elastic spheres, J. Chem. Phys. 33(5):1439.
- Atkins, P.W. (1974). "Quanta: A Handbook of Concepts", London: Oxford University Press.
- Barnes, G.T. (1979). Insoluble monolayers - dynamic aspects, Colloid Science 3:150.
- Barojas, J., Levesque, D., and Quentrec, B. (1973). Simulation of diatomic homonuclear liquids, Phys. Rev. A 7(3):1092.
- Berne, B.J., and Forster, D. (1971). Topics in time-dependent statistical mechanics, Ann. Rev. Phys. Chem. 22:563.
- Bothorel, P., Belle, J., and Lemaire, B. (1974). Theoretical study of aliphatic chain structure in mono- and bilayers, Chem. Phys. Lipids 12:96.
- Brenner, S.L., and McQuarrie, D.A. (1973). Force balances in systems of cylindrical polyelectrolytes, Biophys. J. 13(4):301.
- Buneman, O. (1967). Time-reversible difference procedures, J. Comp. Phys. 1:517.
- Buzbee, B.L., Golub, G.H., and Nielson, C.W. (1970). On direct methods for solving Poisson's equations, SIAM J. Numer. Anal. 7:627.
- Cadenhead, D.A. (1971). Monomolecular films at the air-water interface, in "2nd Symposium on the Physics and Chemistry of Interfaces", Amer. Chem. Society, Wash., D.C., pp. 28-34.

- Carnie, S., Israelachvili, J.N., and Pailthorpe, B.A. (1979). Lipid packing and transbilayer asymmetries of mixed lipid vesicles, *Biochim. Biophys. Acta* 554:340.
- Chapman, D. (1975). Phase transitions and fluidity characteristics of lipids and cell membranes, *Qtr. Rev. Biophys.* 8:185.
- Cherry, R.J. (1976). Protein and lipid mobility in biological and model membranes, in "Biological Membranes" (D. Chapman and D. Wallach, eds.) New York: Academic Press, vol. 3, pp. 47-102.
- Cooper, E.R., and Mann, J.A. (1973). A kinetic theory model for insoluble monolayer transport properties: The dilute gas case, *J. Phys. Chem.* 77:3024.
- Croxtan, C.A. (1974). "Liquid State Physics - A Statistical Mechanical Introduction", Cambridge: Cambridge University Press.
- Demel, R.A., Van Deenen, L.L.M., and Pethica, B.A. (1967). Monolayer interactions of phospholipids and cholesterol, *Biochem. Biophys. Acta* 135:11.
- Denbigh, K. (1971). "The Principles of Chemical Equilibrium", London: Cambridge University Press.
- Dzyaloshinskii, I.E., Lifshitz, E.M., Pitaevskii, L.P. (1961). The general theory of van der Waals forces, *Adv. Physics* 10:165.
- Eastwood, J.W. (1975). Optimal particle-mesh algorithms, *J. Comp. Phys.* 8(1):1.
- Eastwood, J.W., and Hockney, R.W. (1974). Shaping the force law in two-dimensional particle-mesh models, *J. Comp. Phys.* 16:342.
- Erpenbeck, J.J., and Wood, W.W. (1977). Molecular dynamics techniques for hard-core systems, in "Statistical Mechanics, Part B: Time-dependent Processes" (B.J. Berne, ed.), New York: Plenum, pp. 1-40.
- Evans, D.J., and Murad, S. (1977). Singularity-free algorithm for molecular dynamics simulation of rigid polyatomics, *Molec. Phys.* 34(2):327.
- Fischer, P.C., and Probert, R.L. (1979). Storage reorganization techniques for matrix computation in a paging environment, *Communications of the ACM*, 22(7):405.

- Frank, H.S., and Evans, M.W. (1945). Free volume and entropy in condensed systems. III. Entropy in binary liquid mixtures molal entropy in dilute solutions, structure and thermodynamics in aqueous electrolytes, *J. Chem. Phys.* 13:507.
- Franklin, B. (1773). Letter to William Brownrigg dated 7 November 1773, in "The Papers of Benjamin Franklin", (W.B. Willcox, ed.) New Haven: Yale University Press, volume 20, pp. 463-474.
- Gaines, G.L., Jr. (1972). Insoluble monolayers, in "MTP International Review of Science, Surface Chemistry and Colloids" (M. Kerker, ed.), Baltimore: University Park Press, vol. 7, pp. 1-24.
- Gaines, G.L., Jr. (1966). "Insoluble Monolayers at Liquid-Gas Interfaces", New York: Wiley Interscience.
- Geiger, A., Rahman, A., and Stillinger, F.H. (1979). Molecular dynamics study of the hydration of Lennard-Jones solutes, *J. Chem. Phys.* 70(1):263.
- Gershfeld, N.L. (1974). Thermodynamics and experimental methods for equilibrium studies with lipid monolayers, in "Methods in Membrane Biology" (E.D. Korn, ed.) New York: Plenum Press, volume 1, pp. 69-104.
- Gershfeld, N.L. (1974). Physical chemistry of lipid films at fluid interfaces, *Ann. Rev. Phys. Chem.* 27:349.
- Gottlieb, D., and Turkel, E. (1978). Boundary conditions for multistep finite-difference methods for time-dependent equations, *J. Comp. Phys.* 26:181.
- Hagler, H.T., and Moult, J. (1978). Computer simulation of the solvent structure around biological macromolecules, *Nature* 272:222.
- Hamaker, H.C. (1937). The London-van der Waals attraction between spherical particles, *Physica* 4:1058.
- Hanai, T., Haydon, D.A., and Taylor, J. (1965). Polar group orientation and the electrical properties of lecithin biomolecular leaflets, *J. Theor. Biol.* 9:278.
- Heizinger, K., Reide, W.O., Schaefer, L., and Szasz, G.I. (1978). Molecular dynamics simulations of liquids with ionic interactions, in "Computer Modeling of Matter" (P. Lykos, ed.) Washington, D.C.: Am. Chem. Soc., pp. 1-28.

- Hirschfelder, J.O., Curtiss, C.F., and Bird, R.B. (1954). "Molecular Theory of Gases and Liquids", New York: Wiley.
- Hitchcock, P.B., Mason, R., Thomas, K.M., and Shipley, G.G. (1974). Structural chemistry of 1,2 dilauroyl-DL-phosphatidylethanolamine: Molecular conformation and intermolecular packing of phospholipids, *Proc. Nat. Acad. Sci. USA* 71(8):3036.
- Hockney, R.W. (1970). The potential calculation and some applications, *Meth. Comp. Phys.* 9:135.
- Hockney, R.W., and Goel, S.P. (1975). Phase transitions of two-dimensional potassium chloride, *Chem. Phys. Letters* 35(4):500.
- Hohl, F., and Hockney, R.W. (1969). A computer model of disks of stars, *J. Comp. Phys.* 4:306.
- Hoover, W.G., and Ashurst, W.T. (1975). Nonequilibrium molecular dynamics, in "Theoretical Chemistry, Advances and Perspectives" (H. Eyring and D. Henderson, eds.) New York: Academic Press, vol. 1, pp. 1-51.
- Horn, L.W. and Gershfeld, N.L. (1977). Equilibrium and metastable states in lecithin films, *Biophys. J.* 18:301.
- Jacobs, J.E., Hudson, B., and Andersen, H.C. (1975). A theory of the chain melting phase transition of aqueous phospholipid dispersions, *Proc. Nat. Acad. Sci. USA* 72:3993.
- Jain, M.K. (1972). "The Bimolecular Lipid Membrane: A System", New York: Van Nostrand Reinhold.
- Jeans, J. (1954). "The Dynamical Theory of Gases", 4th ed. New York: Dover.
- Kushick, J., and Berne, B.J. (1977). Molecular dynamics methods: continuous potentials, in "Statistical Mechanics, Part B: Time-dependent Processes" (B.J. Berne, ed.) New York: Plenum, pp. 41-63.
- Langmuir, I. (1917). The constitution and fundamental properties of solids and liquids. II. Liquids, *J. Am. Chem. Soc.* 39:1848.
- Langmuir, I. (1933). Oil lenses on water and the nature of monomolecular expanded films, *J. Chem. Phys.* 1:756.
- Lee, A.G. (1975). Functional properties of biological membranes: A physical approach, *Prog. Biophys. Molec. Biol.* 29(1):3.

- Lee, A.G. (1977). Lipid phase transitions and phase diagrams.: I. Lipid phase transitions, *Biochim. Biophys. Acta* 472:237.
- Lee, A.G., Birdsall, N.J.M., Levine, Y.K., and Metcalfe, J.C. (1972). High resolution proton relaxation studies of lecithins, *Biochim. Biophys. Acta* 255:43.
- Levine, Y.K., Birdsall, N.J.M., Lee, A.G., and Metcalfe, J.C. (1972). C-13 nuclear magnetic resonance relaxation measurements of synthetic lecithins and the effect of spin-labelled lipids, *Biochemistry* 11:1416.
- Lifshitz, E.M. (1956). The theory of molecular attractive forces between solids, *Soviet Physics* 2:73.
- London, F. (1930a). Zur Theorie und Systematik der Molekularkrafte, *Z.f. Physik* 63:245.
- London, F. (1930b). Ueber einige Eigenschaften und Anwendungen der Molekularkrafte, *Z.f. Phys. Chemie B* 11:222.
- Mason, E.A., and Spurling, T.H. (1969). "The Virial Equation of State", Oxford: Pergamon.
- McCammon, J.A., and Deutch, J.M. (1975). Semiempirical models for biomembrane phase transitions and phase separations, *J. Am. Chem. Soc.* 97:6675.
- Melchior, D.L., and Steim, J.M. (1976). Thermotropic transitions in biomembranes, *Ann. Rev. Biophys. and Bioeng.* 5:205.
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., and Teller, A.H. (1953). Equation of state calculations by fast computing machines, *J. Chem. Phys.* 21(6):1087.
- Miller, R.H., and Prendergast, K.H. (1968). Stellar dynamics in a discrete phase space, *Astrophysical J.* 151:699.
- Nagle, J.F. (1973). Theory of biomembrane phase transitions, *J. Chem. Phys.* 58(1):252.
- Nagle, J.F. (1976). Theory of lipid monolayer and bilayer phase transitions: effect of headgroup interactions, *J. Memb. Biol.* 27:233.
- Nutting, G.C., and Harkins, W.D. (1939). The pressure-area and pressure-temperature relations of expanded monolayers of myristic and pentadecylic acids, *J. Amer. Chem. Soc.* 61:2040.

- Ohki, S. (1970). Dispersion forces and stability of lipid bilayers, in "Physical Principles of Biological Membranes" (Snell, et al, ed.) Gordon and Breach, pp. 175-225.
- Ohki, S., and Fukuda, N. (1967). Interlayer-interaction for a lipid bilayer model, *J. Theoret. Biol.* 15:362.
- Phillips, M.C., and Chapman, D. (1968). Monolayer characteristics of saturated 1,2-diacyl phosphatidylcholines (lecithins) and phosphatidylethanolamines at the air-water interface, *Biochem. Biophys. Acta* 163:301.
- Phillips, M.C., and Hauser, H. (1974). Spreading of solid glycerides and phospholipids at the air-water interface, *J. of Colloid and Interface Science* 49(1):31.
- Potter, D. (1973). "Computational Physics", London: Wiley.
- Quentrec, B., and Brot, C. (1973). New methods for searching for neighbors in molecular dynamics computations, *J. Comp. Phys.* 13:430.
- Rabenhold, D.A., and Linder, B. (1977). Many-body analysis of long-chain interactions. Inclusion of quadrupoles, *J. Chem. Phys.* 67(3):1108.
- Rahman, A., and Stillinger, F.H. (1971). Molecular dynamics study of liquid water, *J. Chem. Phys.* 55(7):3336.
- Rayleigh, J.W.S. (1910). Capillary action, in "Encyclopaedia Britannica", 11th edition, Cambridge: Cambridge University Press. This is an extensive revision of J.C. Maxwell's article in the 9th edition.
- Rodrigue, G.H., Madison, N.K., and Karush, J.I. (1979). Odd-even reduction for banded linear equations, *J. Assoc. Comp. Mach.* 26(1):72.
- Rothfield, L.I., and Fried, V.A. (1975). Use of monolayer techniques in reconstitution of biological activities, in "Methods in Membrane Biology" (E.D. Korn, ed.) New York: Plenum Press, volume 4, pp. 277-292.
- Rothman, J.E. (1973). The molecular basis of mesomorphic phase transitions in phospholipid systems, *J. Theor. Biol.* 38:1.
- Ryckaert, J., Ciccotti, G., and Berendsen, H.J.C. (1977). Numerical integration of the Cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes, *J. Comp. Phys.* 23:327.

- Salem, L. (1960). The calculation of dispersion forces, *Molec. Physics* 3:441.
- Salem, L. (1962). Attractive forces between long saturated chains at short distances, *J. Chem Phys.* 37:2100.
- Salsbury, N.J., Darke, A., and Chapman, D. (1972). Deuteron magnetic resonance studies of water associated with phospholipids, *Chem. Phys. Lipids* 8:142.
- Schofield, P. (1973). Computer simulation studies of the liquid state, *Comp. Phys. Comm.* 5:17.
- Scott, H.L. (1977). Monte Carlo studies of the hydrocarbon region of lipid bilayers, *Biochim. Biophys. Acta* 469:264.
- Sehgal, K.C., Pickard, W.F. and Jackson, C.M. (1979). Measurement of phospholipid monolayer surface potentials at a hydrocarbon-electrolyte interface, *Biochim. Biophys. Acta* 552:1.
- Sehgal, K.C., Pickard, W.F. and Jackson, C.M. (1979). Phospholipid monolayers at the hydrocarbon-electrolyte interface, *Biochim. Biophys. Acta* 552:11.
- Shamoo, A.E., and MacLennan, D.H. (1974). The Ca^{++} -dependent and -selective ionophore as part of the Ca^{++} + Mg^{++} -dependent adenosinetriphosphatase of sarcoplasmic reticulum, *Proc. Nat. Acad. Sci. USA* 71:3522.
- Shapiro, E., and Ohki, S. (1974). The interaction energy between hydrocarbon chains, *J. Colloid and Interface Sci.* 47:38.
- Singer, S.J. (1971). The molecular organization of biological membranes, in "Structure and Function of Biological Membranes" (L.I. Rothfield, ed.) New York: Academic.
- Singer, S.J., and Nicolson, G.L. (1972). The fluid mosaic model of the structure of cell membranes, *Science* 175:720.
- Smith, R., and Tanford, C. (1973). Hydrophobicity of long chain n-alkyl carboxylic acids, as measured by their distribution between heptane and aqueous solutions, *Proc. Nat. Acad. Sci. USA* 70:289.
- Smythe, W.R. (1968). "Static and Dynamic Electricity", McGraw-Hill, New York.
- Stillinger, F.H., and Rahman, A. (1974). Improved simulation of liquid water by molecular dynamics, *J. Chem. Phys.* 60(4): 1545.

- Stoeckenius, W., and Engelman, D.M. (1969). Current models for the structure of biological membranes, *J. Cell Biology* 42:613.
- Tanford, C. (1973). "The Hydrophobic Effect: Formation of Micelles and Biological Membranes", Wiley, New York.
- Taylor, J., and Haydon, D.A. (1966). Stabilization of thin films of liquid hydrocarbon by alkyl chain interaction, *Disc. Faraday Soc.* 42:51.
- Tinker, D.O. (1975). A model of lateral diffusion in phosphatide bilayers and natural membranes, *Chem. and Phys. of Liquids* 14:33.
- Toxvaerd, S., and Praestgaard, E. (1977). Molecular dynamics calculation of the liquid structure up to a solid surface, *J. Chem. Phys.* 67(11):5291.
- Trauble, H., and Sackman, E. (1972). Studies of the crystalline-liquid crystalline phase transition of lipid model membranes. III. Structure of a steroid-lecithin system below and above the lipid phase transition, *J. Am. Chem. Soc.* 94:4499.
- Vaidhyanathan, V.S. (1968). Stability of lipid films in aqueous electrolyte media: Electrostatic interactions, *J. Theoret. Biol.* 21:331.
- Van Deenen, L.L.M., Houtsmuller, U.M.T., De Haas, G.H., and Mulder, E. (1962). Monomolecular layers of synthetic phosphatides, *J. Pharmacy and Pharmacology* 14(7):429.
- Verlet, L. (1967). Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules, *Phys. Rev.* 159(1):98.
- Vesley, F.J. (1977). N-particle dynamics of polarizable Stockmayer-type molecules, *J. Comp. Phys.* 24:361.
- Vilallonga, F. (1968). Surface chemistry of L-alpha-dipalmitoyl lecithin at the air-water interface, *Biochim. Biophys. Acta* 163:290.
- Wijeyesekera, S.D. and Kushick, J.N. (1979). Velocity correlations in a family of simple soft core liquids, *J. Chem. Phys.* 71:1397.
- Wilkinson, J.H. (1965). "The Algebraic Eigenvalue Problem" Oxford University Press, London.