1986

Structural and Topological Contributions to the Electronic Properties of Inhomogeneous Media (Fractal, Electrode, Impedance)

Michael Kramer
The Graduate Center, City University of New York

Kramer, Michael

STRUCTURAL AND TOPOLOGICAL CONTRIBUTIONS TO THE ELECTRONIC PROPERTIES OF INHOMOGENEOUS MEDIA

City University of New York

PH.D. 1986

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark __✓__.

1. Glossy photographs or pages _____
2. Colored illustrations, paper or print _____
3. Photographs with dark background ✓
4. Illustrations are poor copy _____
5. Pages with black marks, not original copy _____
6. Print shows through as there is text on both sides of page _____
7. Indistinct, broken or small print on several pages ✓
8. Print exceeds margin requirements _____
9. Tightly bound copy with print lost in spine _____
10. Computer printout pages with indistinct print ✓
11. Page(s) _____ lacking when material received, and not available from school or author.
12. Page(s) _____ seem to be missing in numbering only as text follows.
13. Two pages numbered _____. Text follows.
14. Curling and wrinkled pages _____
15. Dissertation contains pages with print at a slant, filmed as received ✓
16. Other _____

_____

_____

University
Microfilms
International

STRUCTURAL AND TOPOLOGICAL CONTRIBUTIONS TO THE ELECTRONIC PROPERTIES OF INHOMOGENEOUS MEDIA

by

MICHAEL KRAMER

A dissertation submitted to the Graduate Faculty in Physics in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

1986

This manuscript has been read and accepted for the Graduate Faculty in Physics in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

[Signature]

12/2/85
Date

Prof. Micha Tomkiewicz
Chairman of Examining Committee

[Signature]

1-29-86
Date

Executive Officer

[Signature]

Dr. Morrel Cohen. Exxon Research & Eng
[Signature]

Prof. Myriam Sarachik. City College
[Signature]

Prof. Fred Pollak, Brooklyn College
[Signature]

Prof. Yedidyah Langsam, Brooklyn College

Supervisory Committee

The City University of New York

ABSTRACT

STRUCTURAL AND TOPOLOGICAL CONTRIBUTIONS

TO THE ELECTRONIC PROPERTIES

OF INHOMOGENEOUS MEDIA

by

MICHAEL KRAMER

Adviser: Professor Micha Tomkiewicz

Recent theoretical developments in the theory of inhomogeneous media have enabled Physicists to explore various properties of random composites. We demonstrate how these techniques can be applied to the treatment of electrochemical systems, specifically porous structures. A Random Network Model is introduced to represent the porous electrode immersed in electrolyte, and using numerical simulation the a.c. impedance behavior of the system is explored. We explore how various compositional and structural parameters contribute to the electrical properties of the system. The topology of our model is analyzed, and the effect of fractal structures on impedance behavior is considered. To examine the electrode topology during electrode operation, another model is introduced, the dissolution-precipitation model. We show how the model correctly predicts the concentration profile of dissolved particles in solution and how the reaction produces fractal structures on the surface of the model electrode.

Dedicated to my dear wife, Debby

and to

Yehoshua, Yair and Shoshana

ACKNOWLEDGEMENTS

I wish to express my greatest appreciation to my adviser, Professor Micha Tomkiewicz. Without his guidance, support, prodding and encouragement, this work would never have seen the light of day. His valuable friendship and his constant interest in my progress has given me more than one could hope for in a mentor. My thanks to Professor Fred Pollak for sharing his vast knowledge and insight from the days of my undergraduate studies and throughout my graduate career, including his participation in my doctoral committee. To the other members of the committee, Dr. Morrel Cohen of Exxon Research and Engineering, Dr. Myriam Sarachek of City College, and to Dr. Yedidyah Langsam of Brooklyn College my sincerest thanks for agreeing to serve on my committee and for your helpful advice and encouragement. I particularly want to thank Dr. Cohen for numerous discussions during which he patiently took the time to illuminate some obscure points. Special thanks also to Dr. Langsam for always being available to assist me in getting my computer to jump through new loops. I am deeply indebted to Dr. Phil Seiden of IBM Thomas J. Watson Research Center and to Dr. Itzhak Webman of Exxon Research and Engineering for their valuable guidance in the initial phases of this work, and to the Office of Naval Research for providing the funding. I have been most fortunate to have had the opportunity to work closely with Dr. Joseph Lyden and Dr. Benedict Aurien-Blajeni. Dr. Lyden helped inaugurate me into the world of experimental

Physics, and it was his experimental work that helped serve as the inspiration for the topic of this thesis. Dr. Aurien-Blajeni has helped me in numerous discussions in which he has enhanced my grasp of electrochemistry, and the work of chapter IV on the dissolution-precipitation model is a result of our collaboration. I also appreciate the help of colleagues in graduate school past and present; Dr. Pinchas Laufer, Dr. Orest Glembocki, and Dr. Padman Paryanthal stand out in this regard. Many members of the Physics Department faculty provided me with much knowledge and support. I would particularly like to thank Professor Joe Krieger for his constant encouragement and his wonderful lectures.

This research involved very CPU-intensive computer processing, and my special thanks go to Bill Gruber, Director of Consulting and Education in the University Computer Center (UCC) of the City University of New York who, from the outset, took a special interest in insuring that I was able to obtain the computer resources that I needed to do this work, and who always seemed to have the answers to questions of "what went wrong?" or "how do I do this?". Without his help in moving mountains of bureaucracy these computations could not have been done. Thanks are also due to Ben Klein, Director of UCC for his full cooperation as well. Thank you to Julio Berger, and Larry Schweitzer from the Brooklyn College Computer Center (BCCC) who were instrumental in insuring that I obtained full support in the local computing facilities. Joey Sussman of BCCC provided me with much technical support and assistance for which I am grateful.

In general, all staff members of both UCC and BCCC were wonderful, and too numerous to mention.

To paraphrase a hebrew idiom "last mentions are dearest to the heart" ("acharon acharon chaviv"). Words cannot express my appreciation to and admiration for my wife, Debby for her support and sacrifice during the sometimes grueling years of my graduate career. This work is as much of a reflection of that as of my efforts. Thanks also to my parents and in-laws as well as all our brothers and sisters for their help and support. And lastly to my children Yehoshua, Yair and Shoshana my thanks for livening up our lives, always being ready to play (whether I needed a break or not) and putting up with the demands of my work.

TABLE OF CONTENTS

LIST OF FIGURES

Introduction

The focus of Physics in the 19th century was on the world of the visible and experiential - gravity, mechanics, electricity and magnetism. The Newtonian and Maxwellian theories are extraordinarily successful in describing the world of our senses. With the advent of the twentieth century and the birth of quantum mechanics much of the physicist's attention shifted to the opposite extreme - the microscopic world of atoms and subatomic particles. The quantum theory of solids has brought with it a deep understanding of the behavior of materials from first principles. The behavior of electrons in the solid's crystal potential, the interaction of electromagnetic radiation with the solid, the identification of phonons and their effect on elastic, thermal and electrical properties are all relatively well understood and have sparked our technological explosion.

The existence of multi-phase granular systems presents new challenges, due to the fact that neither of the above approaches are suitable for an accurate treatment of this problem. To approach the electronic properties of the system- on the one hand, each phase of the material can be well characterized by local electronic properties (the inhomogeneity is manifest on a length scale larger than the electronic mean free path). On the other hand, the system is macroscopically inhomogeneous, and horrendous boundary conditions confront anyone who would attempt to solve Maxwell's equations

across the different phases. Recent studies into phenomena based on random aggregation and the clustering of small particles have demonstrated that by utilizing statistical techniques, the aggregate properties of composite materials may be derived from the local behavior of the system's constituents.

This work is an attempt to apply the study of inhomogeneous media to electrochemical systems, particularly porous structures, and to explore the contributions of the electrode morphology to the electronic properties of the system. The basis of the research is a computer model of the electrode system that captures the composite media character of the electrode. Chapter I presents an overview of the theory of conduction in heterogeneous media, and Chapter II describes the porous electrode-electrolyte system and explains the applicability of the composite media description. Chapter III introduces the Random Network Model- the structure of the system, the impedance characteristics and the relationship between them. Chapter IV details an alternative simulation for the construction of structures on the surface of porous electrodes and sketches how this may be used to further explore the system.

Electronic Conduction in Composite Media: Theoretical Tools

A. Introduction

Electronic conduction in randomly disordered systems has been explored using the tools of percolation theory. If the system under study is composed of an insulating phase and a conducting phase, then by varying the composition of the system, a sharp change in the conductivity of the system is observed. The critical point (of volume fraction) at which this occurs is termed the percolation threshold and is determined by the dimensionality of the system. Percolation is actually a general process that applies to many systems in which a random media can exhibit long range connectivity. If the composition of the system is varied, the point at which this long range connectivity disappears is the percolation threshold. In this manner, percolation has been used as a model to describe the flow of liquid in a porous media, stochastic star formations in spiral galaxies, and dilute magnetic systems, to name just a few areas of application[1].

Percolation can be illustrated by considering a lattice of N sites (which we will consider in the limit as $N \to \infty$), in which some of the sites are randomly selected as being "allowed", and the remaining sites are vacant. In addition to this site percolation, one can formulate the problem in terms of allowed and missing bonds to arrive at bond percolation. If c is defined as the concentration of

allowed sites, then when c is very small, allowed sites occur as scattered small islands in the lattice. As c increases, the clusters grow larger and larger, until we reach c^*, where the mean cluster size approaches the size of an infinite cluster (i.e. it provides a path of connectivity through the whole space). If P(c) is the ratio of the number of sites in the infinite cluster to the number of sites on the lattice, P(c) becomes the probability that a given site will be part of the infinite cluster. A plot of P(c) as a function of c for a 3D lattice is shown in Figure 1.1, and the existence of a critical point is seen from the shape of the curve as $c \to c^*$. Near threshold (from above), the behavior of P(c) may be characterized by a power law,

$$P(c) \propto (c - c^*)^s, \tag{1.1}$$

where s, for a cubic lattice, has been found to be approximately

$$0.3 \leq s \leq 0.4,^{2}$$

and $c^* \approx .3$ (site percolation)[66] and .25 (bond percolation)[67].

An alternate quantity of interest is the correlation length, ξ, which measures the average cluster size. As the concentration approaches the critical concentration, the formation of the infinite cluster described above is represented by the divergence of ξ, such that

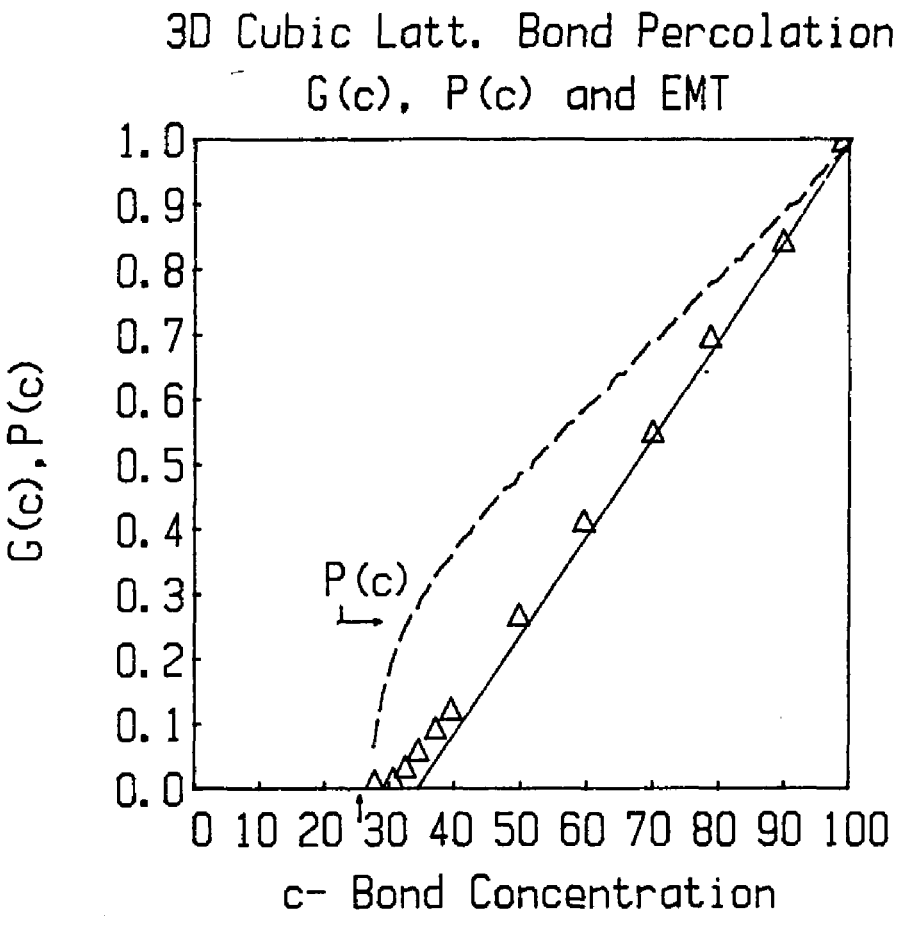


Figure 1.1 - 3D Bond Percolation, after Kirkpatrik (Ref. 3). c is the % bond concentration of conducting bonds. The data for P(c) (dotted line) and G(c) (triangles) are obtained by a computer simulation on a cubic lattice of size 15x15x15. P(c) is the probability of a given site being part of the infinite cluster, G(c) is the conductance of the system, and the solid line is the Effective Medium Prediction for G(c). The arrow under the x axis corresponds to the location of c*, the percolation threshold.

$$\xi \sim \xi_0 (c-c^*)^{-\nu} \qquad (1.2)$$

where ξ_0 is the size of the granules of which the composite material is made. As one approaches the percolation threshold, the correlation length becomes the dominant length scale of the system. The exponent ν will prove to be of some theoretical importance, as we shall see later.

It has been shown, however, that P(c) is not suitable for representing the dependance of the conductivity on the composition of the system[3]. Although an allowed site may be part of a cluster, it is not necessarily a participant in the transport properties of the system. Many cluster "dead ends" will increase the probability of an infinite cluster, but will not themselves be participating in conduction. In order to formulate the percolation problem in terms of conductivity, $\sigma(c)$, Kirkpatrick[5] introduced the Random Resistor Network Model.

B. The Random Resistor Network Model

We assume that the system is microscopically inhomogeneous but that the dimensions of the conducting regions are large with respect to the electronic mean free paths. We may therefore define a local conductivity, $\sigma(\vec{r})$, at a position $\vec{r}$ in the material. The system is considered to have two phases of microscopic conductances $\sigma_1(r)=0$, and $\sigma_2(r)=1$, and we suppose that we have a detailed knowledge of how

the two phases are arranged. We can calculate the macroscopic conductivity of the system using the Ohm's law equation

$$j(r) = - \sigma(r) \nabla V(r) \tag{1.3}$$

and the continuity equation

$$\nabla \cdot j(r) = 0. \tag{1.4}$$

By introducing a cubic mesh of points $\{r_i\}$ with spacing Δr, these continuum equations may be solved using a finite difference approximation:

$$\sum_j g_{ij} (V_i - V_j) = 0 \tag{1.5}$$

with $g_{ij} = \Delta r \sigma[(r_i + r_j)/2]$, and i,j being neighboring sites on the mesh. Kirkpatrick noted[5] that Equation 1.5 is identical to the Kirchoff's law equations for a 3D network of resistors, and Equations 1.3-1.5 may be solved by numerical simulation. We define a cubic mesh of resistors, and using a random number generator a specified fraction, c, of these resistors are assigned conductivity 1 while the remaining resistors are assigned conductivity 0. $\sigma(c)$ as a function of the relative concentration c, is also shown in Figure 1.1. Again it is determined that near the threshold (as approached from above),

$$\sigma(c) \propto (c - c^*)^t \tag{1.6}$$

where $t \approx 1.94 \pm .1$ for both bond and site percolation[68]. (The value of c^* remains $\approx .25$ (bond percolation) and .3 (site percolation)). Thus far the numerical simulations considered only discrete lattice percolation; by introducing correlations between neighboring bonds, Webman, Jortner and Cohen[6] transformed the problem into a continuous percolation model, and showed that $c^* \approx .15$ (bond percolation).

A generalization of the above result[7,8] is for $\sigma_2 = b$ with a probability c and $\sigma_1 = a$ with a probability (1-c), where a and b are both greater than zero. If we assume that $a \ll b$, then we have:

$$\sigma \propto b\,(c - c^*)^t \qquad \text{for } c > c^* \qquad (1.7a)$$

$$\sigma \propto a\,(c^* - c)^{-s} \qquad \text{for } c < c^* \qquad (1.7b)$$

Equation 1.7b may be explained by observing that although a is the low conductance phase of material, at very low concentrations of "b" the current is forced to traverse a path through "a". As the concentration of "b" increases, the current shorts through clusters of "b", resulting in an apparent divergence in the conductivity as one approaches threshold (c^*). At $c = c^*$, however, there is no divergence of the conductivity due to the fact that the current does not follow all the twists and turns of the conducting phase and travels through the more numerous paths of the "a" phase, even though it is more poorly conducting. (There is also no divergence

in the correlation length). If we assume that at c^* the power laws in Equations 1.7 are no longer applicable, and σ is almost constant in a small region $\delta=|c-c^*|$, we may equate Equations 1.7 at $c = c^* \pm \delta$:

$$\sigma = a\delta^{-s} = b\delta^{t}$$

we find that the "crossover region" is defined by

$$\delta = \left(\frac{a}{b}\right)^{1/(t+s)}$$

and that near c^*, the conductivity is defined as:

$$\sigma \propto a^{u} b^{1-u}, \quad u = t/(s+t). \tag{1.7c}$$

Thus far we have considered only systems containing a mixtures of pure conductors and non-conductors. Bergman and Imry[9] considered the case of a heterogeneous mixture of a conducting phase and an insulating dielectric phase. The bulk effective dielectric constant, κ_e is

$$\kappa_e = \varepsilon_e + \frac{4\pi\sigma_e}{i\omega} \tag{1.8}$$

where ε_e and σ_e are the bulk static dielectric constant and conductivity respectively, and ω is the frequency of the excitation signal. For $\kappa_1 = \varepsilon_1$ and $\kappa_2 = 4\pi\sigma_2/i\omega$, with $|\kappa_1| << |\kappa_2|$ and near

the percolation threshold $c_2 \to c^*$, σ_e scales as $(c - c^*)^{\alpha}$ while ε_e increases as $(c - c^*)^{-\gamma(1-\beta)}$. They postulated that the large enhancement of the dielectric constant as one approaches c^* from below is due to the presence of long paths of conducting material separated by thin barriers, whose effective capacitance is enormous. In addition, at $c=c^*$,

$$\sigma_e \sim \omega^{\beta} \quad \text{and} \quad \varepsilon_e \sim \omega^{\beta-1}, \text{ where } \beta=0.73\pm.05 \ . \tag{1.9}$$

Equations 1.7 present a scaling behavior for the conductivity, and exponents s,t and u are the critical exponents that characterize this scaling relation. Values for these exponents have been obtained from numerical simulation, and for 3D bond percolation are:

$$s=.75 \quad t=1.94 \quad u=.67\pm.08 \ .^{69}$$

A correlation length for the resistor network has been defined by Bass and Stephen[11] as follows: apply a potential V_0 at vertex 0, and measure the average potential $\langle V_n \rangle$ at vertices n links away. Then

$$\xi^{-1} = -\lim_{n\to\infty} n^{-1} \ln\langle V_n/V_0 \rangle \ . \tag{1.10}$$

C. Finite Size Scaling and Real Space Renormalization

The existence of the scaling relationships of Equations 1.7 has prompted the application of scaling theories[12] which have been successfully applied to critical phenomena, such as magnetic spin systems (the simplest of them being the Ising Model). Real space renormalization is applied to map the system into regions of physical parameters where the correlations are easy to compute. Following this, one matches these transformed values to the original correlations, and calculates the correlation function. This allows us to identify the correlation length ξ (which is the scale on which, for a particular concentration, the system first appears homogeneous[13]), and to calculate the critical exponent ν from Equation 1.2. (A detailed example calculation is presented in Stauffer, p. 15). This renormalization is allowed due to the divergence of ξ near c^*, so that the system's properties are invariant no matter which finite length scales we use to investigate the system. "Thus it should not matter at c^* on what length scale we are investigating the system; apart from simple scale factors the system looks similar whether we look at it with the eye, with a magnifying glass, or with an optical microscope"[14].

Systems that exhibit such properties of self-similarity under transformations of scale are prime candidates for classification as fractal systems. In fact, a major theoretical contribution to the study of critical behavior in heterogeneous systems due to Mandelbrot[15] is the concept that the scaling property of the

clusters result from the analytic properties of the cluster's geometry. Although the clusters constitute a random disordered system, when the clusters are treated as fractals their geometric properties become simply describable through their fractal dimension, which is related to the exponent ν of Equation 1.2.

D.Fractals

The traditional Euclidean geometry defines geometric constructs as occurring in a Euclidean space E (the dimensions of the space in which it is embedded), and having a topological dimension, D_T (e.g. a line or a circle have D_T=1). In connection with some systems, however, mathematicians have found it useful to define other dimensions, one such being the Hausdorff dimension, D. The Hausdorff dimension describes how the measure of an object changes under the application of different scales of measurement. In general,

$$D_T \leq D \leq E \qquad (1.11)$$

and specifically, Mandelbrot advances the following definition:

> A fractal object is one whose Hausdorff dimension is strictly greater than it's topological dimension. D is termed the object's fractal dimension, which is in general not an integer.

The classic example of a fractal dimension is the Richardson Effect (Figure 1.2). If we wish to measure the length of an island coastline, we can take a ruler and start to measure the length of the land-water border. It quickly becomes apparent though, that the resulting length measure is totally dependent on the size of the ruler with which we measured the coast (the resolution of the measurement). As we decrease the size of our measuring stick we are able to measure the coast with finer and finer detail. The result is the Richardson curves of Figure 1.2, where the relationship between the length scale used and the total length is linear when viewed on a log-log plot. If ε is the resolution of the measurement, and $L(\varepsilon)$ is the total length measured, then

$$L(\varepsilon) \propto \varepsilon^{1-D} \tag{1.12}$$

where D is the fractal dimension. Thus, coastlines are representative of a fractal curve. Mandelbrot argues that D is the only measure of convincing physical significance, since it is the only scale-invariant quantity in the problem. In the case of the coast of Britain, the Richardson slope is approximately -.2, which means that D=1.2, which is indeed greater than the topological dimension of the coastline curve (D_T=1), thus satisfying the condition for a fractal.

An alternate definition of fractals is the relationship between measure of contained mass and radius, (which implies a relationship

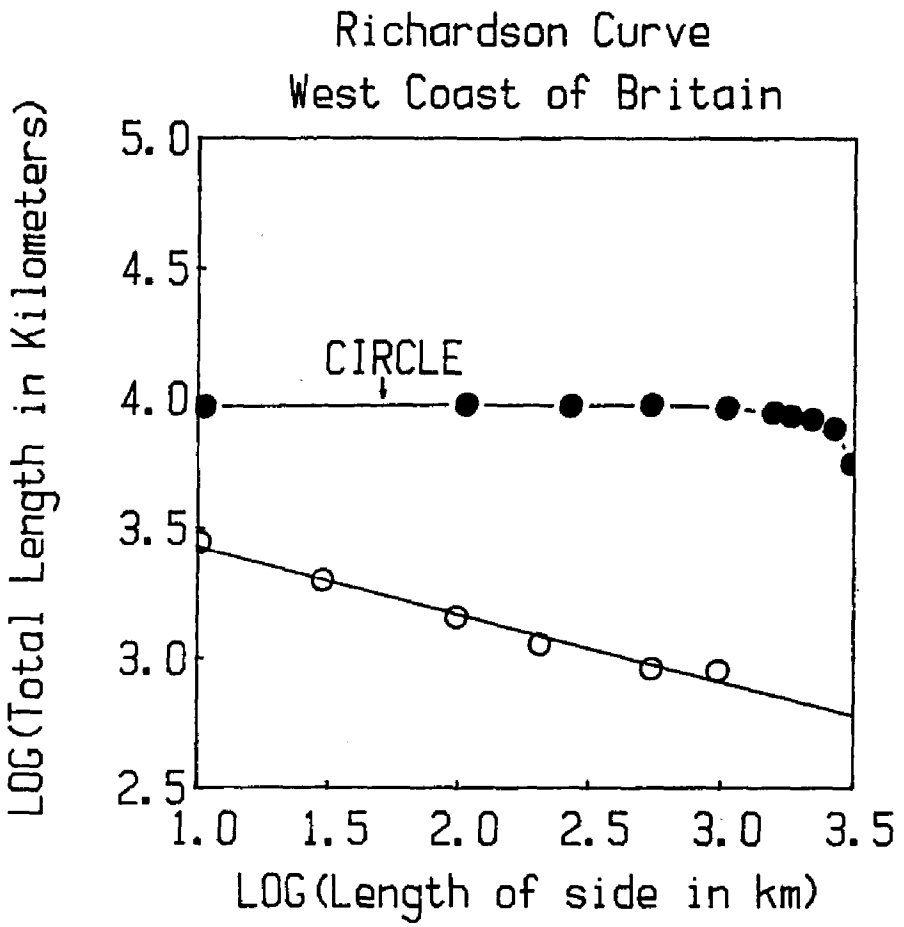


Figure 1.2 - The Richardson curve illustrating that the coast of Britain (outlined points) is a fractal curve. The Richardson Curve for a circle (solid points) has a slope of 0.0, indicating that its effective dimension is equal to its topological dimension. (After Mandelbrot, Ref. 15)

between length and volume, or length and area). The amount of mass, M, contained in an object of radius R can be expressed as:

$$M(R) \propto R^{D} . \qquad (1.13)$$

(We are well acquainted with the classic cases for which D=3 (e.g. a homogeneous sphere) or D=2 (e.g. a homogeneous 2-D square plate). It is this relationship which may be utilized to calculate the fractal dimension of a percolating cluster. The straightforward procedure[14] consists of taking circles or spheres of increasing size and measuring their contents or mass (i.e. how much of the percolating cluster do they contain at different sizes), and using Equation 1.11 to directly calculate D. Kirkpatrick[16] employs a variation of this method for calculating a fractal dimension of the "backbone" of the infinite cluster.

An alternate expression of Equation 1.11 is the scaling property of the density-density correlation function[19,20]:

$$C(r) \equiv N^{-1} \sum_{r'} \rho(r')\rho(r'+r) \propto r^{D-d} \equiv r^{-A} \qquad (1.14)$$

where d is the spatial (Euclidean) dimension (E), and D is the fractal dimension. The density $\rho(r)$ is defined to be 1 for occupied sites and 0 for empty sites. The calculation is performed by computing C(r) for a given r in a given direction over all the particles in the lattice (or in the cluster), and then C(r) is

averaged over different directions. This is repeated for a number of discrete lengths r, and the exponent A is evaluated from the slope of log(C(r)) plotted as a function of log(r). The utility of this calculation is that it is easy to perform in computer aggregation simulations. This relationship is also directly verifiable using scattering experiments such as neutron or X-ray scattering. The structure factor, S(k) which is measured in a scattering experiment, is none other than the Fourier transform of C(r).[20,21] Thus, we may relate the scattering intensity, I, and the wavenumber, k, by

$$I(k) \propto k^{-x} . \qquad (1.15)$$

The interpretation of this exponent, x, is dependant on the scattering system. For "volume fractals" (i.e. polymer-like structures), x is none other than D from Equation 1.13; for scattering from surfaces, x=6-D, where D is the fractal dimension of the surface (D=2 corresponds to a flat surface). For fractally porous materials[22,23,24], x=7-γ, where γ is the exponent describing the distribution of pores of radius r,

$$P(r) = r^{-\gamma} . \qquad (1.16)$$

Real life objects do not exhibit fractal behavior over all length scales. In practice, there is an inner and outer cutoff for ϵ and R, within which Equations 1.12 and 1.13 hold - corresponding to the

range where these systems exhibit self-similarity. In the percolation problem the bounds are from the lattice size, a, to the correlation length, ξ.[16]

E. Scaling of the Conductance on Percolating Fractal

By combining Equation 1.2 and the conductance scaling relationship of Equation 1.7b, we find that the conductivity may be expressed as

$$\sigma \propto \xi^{-\mu/\nu} \qquad (1.17)$$

Using finite-size scaling arguments due to the self-similarity of the system, Palevski and Deutscher[25] show that for a system of size $L \ll \xi$:

$$\sigma \propto L^{-\mu/\nu} \qquad (1.18)$$

Thus, effective dimensions for the impedance properties of the system may be calculated using methods similar to those for the classic fractal relationships reviewed above. In fact, Palevski and Deutscher measure the dependance of R(L) as a function of L (where R(L) is the resistance in a segment of the percolating gold clusters and L is length of the side of a square within which they measure the resistance). As they increase the area of the square within which they measure R, they observe a linear relationship between log

R(L) and log L. Computation of the slope of that line yields the ratio μ/ν.

F. Effective Medium Theory (EMT)

While the Random Resistor Network approach treats the electrical properties of the system as explicitly arising from the system's microscopic constituents, EMT takes almost the opposite approach. We consider a composite system made up of two materials having conductivity σ_1 and σ_2. Our objective is to calculate σ_e, an effective conductivity of the entire composite. We begin[28] by considering a small (spherical) grain inside the material, and treat that grain as if it is embedded in a homogeneous effective medium, of conductivity σ_e, to be computed self-consistently (see Figure 1.3). Let the field and current density far from the central grain be E_0 and $J_0=\sigma_e E_0$. We can therefore calculate the fields and current density within the grain to be

$$E_{in} = [3\sigma_e/(\sigma_i+2\sigma_e)]\ E_0 \qquad (1.19)$$

$$J_{in} = \sigma_i\ E_{in} \qquad (1.20)$$

where σ_i is either σ_1 or σ_2. (the microscopic assumptions and geometrical considerations necessary to produce Equation 1.19 are somewhat controversial, although all derivations produce this result).[26,27,28] The self consistency for computing σ_e comes from the assumption that

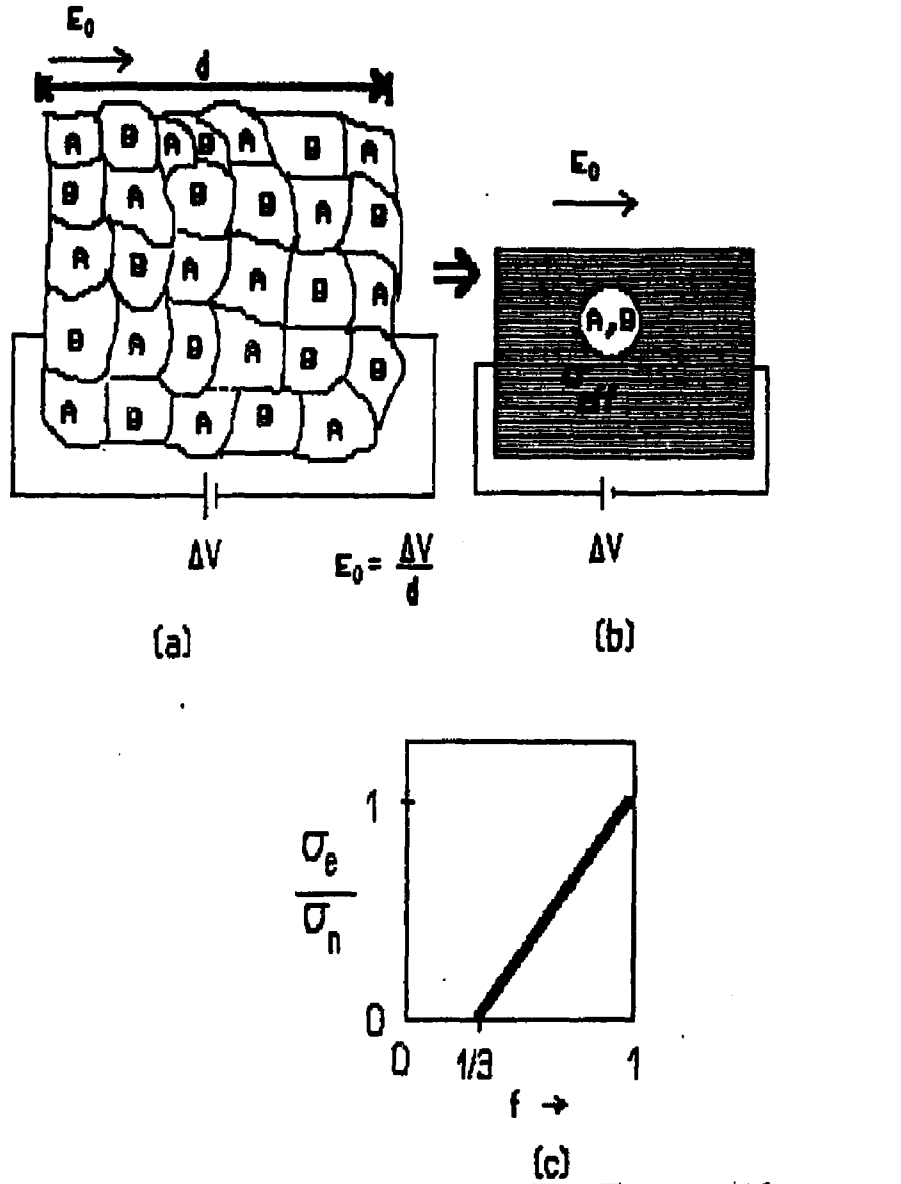


Figure 1.3 - Illustration of Effective Medium Theory. (After Stroud, Ref. 28).
(a) Schematic of a two-component composite in an applied electric field, E0.
(b) An illustration of the self-consistent embedding procedure which defines the effective conductivity, σe.
(c) Schematic behavior of σe as a function of concentration, f (from the application of Equation 1.23 .

$$\langle J_{in} \rangle = \sigma_e \langle E_{in} \rangle . \qquad (1.21)$$

Assuming that our composite is made up of the fraction f of component 1 and (1-f) of component 2, we can substitute Equations 1.19 and 1.20 into 1.21 to yield,

$$\frac{3f\sigma_1}{\sigma_1+2\sigma_e} + \frac{3(1-f)\sigma_2}{\sigma_2+2\sigma_e} = \sigma_e \left(\frac{3f}{\sigma_1+2\sigma_e} + \frac{3(1-f)}{\sigma_2+2\sigma_e} \right) \qquad (1.22)$$

which simplifies to:

$$f \frac{\sigma_1 - \sigma_e}{\sigma_1 + 2\sigma_e} + (1-f) \frac{\sigma_2 - \sigma_e}{\sigma_2 + 2\sigma_e} = 0 . \qquad (1.23)$$

Although EMT has proved to be a useful approximation for dealing with systems about which we have very little structural information, the theory fails when a percolating system approaches the percolation threshold. EMT predicts a much higher critical concentration, c^*, then is observed experimentally or by numerical simulation[3]. This may be explained[29] by noting our bias for spherical conducting geometries. If in fact our system consists of flattened spheroids or elongated conducting components, the probability of connectivity through the system is enhanced at lower conductor concentrations. In a random system, there is no apriori reason to assume the favoring of spherical structures over elongated ones. This illustrates that some knowledge of the internal geometry of a particular system is necessary for the application of EMT such a system.

G. Summary

We have presented a multiplicity of techniques for the theoretical treatment of conduction in heterogeneous media. The current presentation has centered on a simple application, the impedance characterization of a two-phase random mixture of conductor and non-conductor. It should be pointed out however, that these theories may be adapted to treat more complex systems, not only systems with multiple component phases, but also systems with unique non-random correlations, and systems with specialized geometries. Moreover, although this work focuses on conduction in composite media, these are general tools equally applicable for the treatment of a wide range of transport and static problems in stochastic systems, all revolving around the properties of clustering and aggregation in these systems. A full discussion of all the ramifications of these theories are beyond the scope of this work (and me). In the coming chapters we will employ these concepts and techniques to our problem at hand - the stochastic morphology of the porous electrode and it's contribution to the electronic conduction.

II. The Chemistry and Physics of Porous Electrodes

A. Solid-Liquid Interfaces

When two phases of material are brought into contact with each other, the difference in their free energy develops a potential difference across the interface. When these two phases are metal (electrode) and electrolyte, the potential difference induces a surface charge on the metal and region of net charge of opposite sign in the electrolyte. In the Gouy-Chapman (GC) model of the interface, there is a diffuse region of local net charge that extends a distance into the electrolyte. If we imagine a microscopically thin lamina of electrolyte at a distance x from the interface, utilizing Boltzmann statistics the charge density in that region can be written as

$$\rho(x) = nze \exp[-ze\phi(x)/k_B T] \qquad (2.1)$$

where n is the ion concentration, z is the net charge on each ion, $\phi(x)$ is the electrostatic potential at position x in the solution with respect to the interface, k_B is the Boltzmann constant, and T is the absolute temperature. Combining this with the Poisson equation:

$$\nabla^2\phi = \frac{4\pi}{\varepsilon} \rho \qquad (2.2)$$

the approximate solution is[30]

$$\varphi = \varphi_0 \, e^{-\kappa x} \qquad (2.3)$$

κ has units of inverse length and κ^{-1} is the characteristic expanse of the charge distribution. Thus, we can consider all the charge in solution to be located on a plane parallel to the interface and located at $d_{GC}=\kappa^{-1}$. The behavior of this double layer is therefore identical to that of a parallel plate capacitor of capacitance

$$C_{GC} = \frac{\varepsilon_0 A}{GC} \qquad (2.4)$$

d

where ε_0 is the permittivity of free space and A is the cross sectional area of the interface. It is the electric field at the interface, represented by the capacitor, that constitutes the driving force for charge transfer across the interface and which represents the electrode potential (in comparison with other electrodes in solution). Another field at the interface is generated by adsorbed water molecules and unsolvated ions on the interface and it is also associated with a capacitance C_H, the Helmholtz Capacitance. The total capacitance of the interface is due to two capacitors in series, and the total capacitance is:

$$\frac{1}{C} = \frac{1}{C_H} + \frac{1}{C_{GC}} . \qquad (2.5)$$

Figure 2.1 is a diagram of the charge distribution at the interface and the location of the two planes of effective charge. At high

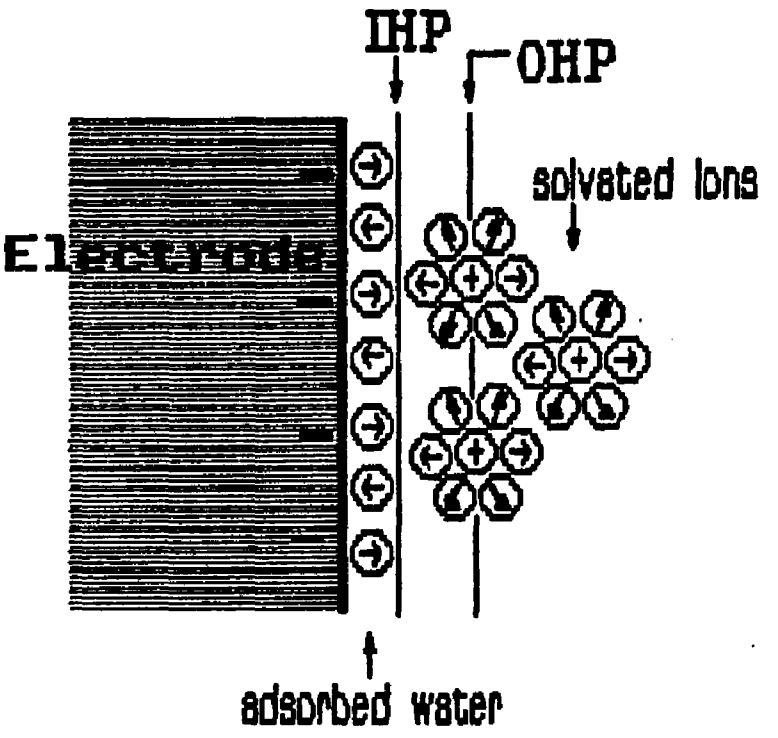


Figure 2.1

Schematic of the electrified interface. The IHP (inner Helmholtz plane)is due to the adsorbed surface species, and the OHP (outer Helmholtz plane) is due to the diffuse charge distribution in solution.

solution concentrations the GC layer is very narrow, and as a result of the capacitance behavior of Equation 2.4, C_{GC} becomes large and may be ignored in comparison with C_H. At low electrolyte concentrations C_{GC} dominates (the Helmholtz capacitance is independent of concentration).

Thus far we have considered no faradaic current flow across the interface, an assumption of an Ideally Polarizable Electrode (IPE). More generally (depending on the catalytic properties of the metal and on the solution), the metal electrolyte interface is represented by the equivalent circuit model of Figure 2.2, where R_p represents a leakage current (a deviation from IPE behavior). In Figure 2.2 R_s is the resistance of the metal - which is usually negligible.

If there is an insulator or semiconductor build up on the electrode due to some chemical reaction at the surface, then the metal-solution capacitance remains, but the value will be altered by the dielectric properties of the new layer. In this case, the R_s of Figure 2.2 is the bulk resistance of the insulating or semiconducting layer on the interface.

The bulk electrolyte solution has an ohmic conductivity due to the movement of ions, and for a given volume of solution we can represent the current carrying capacity of the solution using an equivalent resistance,

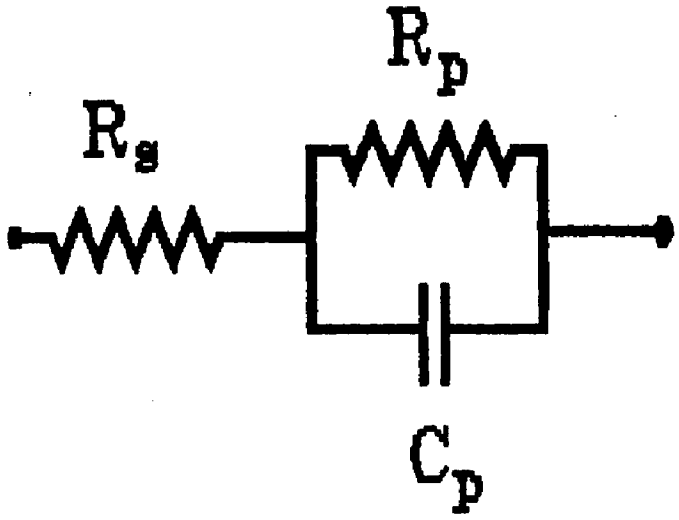


Figure 2.2

Equivalent circuit model of the solid-liquid interface.

$$R_e = \frac{\rho_e \ell}{A} , \qquad (2.6)$$

where the electrolyte volume is defined by the length ℓ and cross-sectional area A, and the electrolyte resistivity is determined from the relationship[31]

$$\rho_e = \frac{1}{nz\mu} \qquad (2.7)$$

n is the number of ions in the given volume, z is the charge per ion, and μ is the mobility determined by the characteristic species in solution.

A useful experimental technique for the study of electrochemical systems is the use of a.c. impedance measurements[32]. These measurements allow for nondestructive in-situ characterization of the system. Impedance measurements are a natural choice given the theoretical behavior of these systems as simple passive circuit components. Using very low signal a.c. excitations, we can measure both static and kinetic electrochemical processes.

B. Typical Application: A Secondary Battery Cell

Perhaps the most important technological application of these electrochemical principles is in battery electrodes. A typical example is the Silver-Zinc cell of Figure 2.3[33]. A silver oxide

electrode and a Zinc electrode are placed in a solution of KOH. The overall reaction is a two step one[34]:

$$Ag_2O_2 + H_2O + Zn \underset{\longleftarrow\text{charge}}{\overset{\text{discharge}\longrightarrow}{}} Ag_2O + Zn(OH)_2$$

$$Ag_2O + H_2O + Zn \underset{\longleftarrow\text{charge}}{\overset{\text{discharge}\longrightarrow}{}} 2Ag + Zn(OH)_2 \qquad (2.8)$$

In the discharge process the Zn electrode is oxidized, liberating electrons to flow through the external circuit, and forming a layer of ZnO on the electrode. The half-cell reaction is:

$$Zn \longrightarrow Zn^{2+} + 2e^- \qquad (2.9)$$

$$Zn^{2+} + 2OH^- \longrightarrow Zn(OH)_2 \quad .$$

On the silver electrode the AgO is reduced to produce Ag[35]:

$$Ag_2O_2 + H_2O + 2e^- \longrightarrow Ag_2O + 2OH^- \qquad (2.10)$$

$$Ag_2O + H_2O + 2e^- \longrightarrow 2Ag + 2OH^-$$

or schematically:

$$2Ag^+ + 2e^- \longrightarrow 2Ag.$$

Focusing on the Zn electrode, the net result of the discharge process is the buildup of ZnO on the surface of the electrode. The state of charge of the electrode can be determined from the amount of ZnO on the surface, and as previously mentioned this changes the capacitance of the interface.

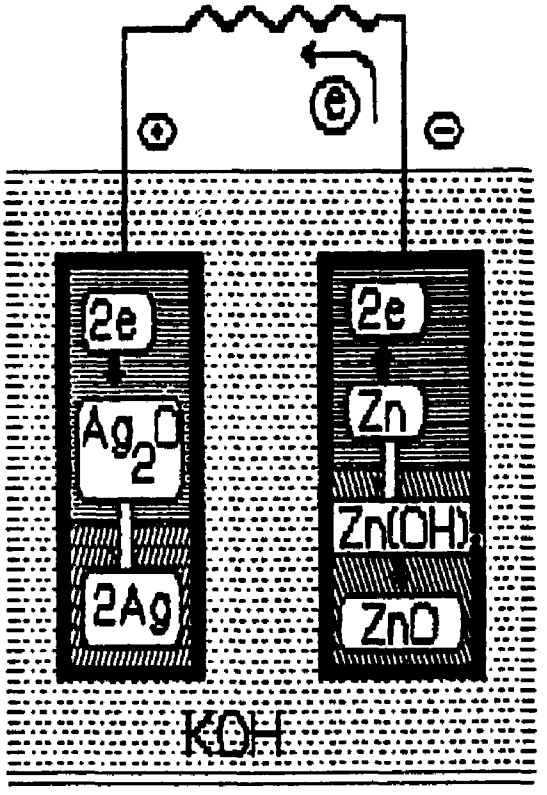


Figure 2.3

A typical secondary Battery cell during discharge.
During discharge, a layer of ZnO is forming on the anode.

The most important characteristics of any battery are its storage capacity, its energy density and its power density. The energy density can be increased by the choice of two electrodes with large differences in their electrode potential, and by producing a very high current density on discharge. To achieve this, a large real surface area is needed between the metal and the electrolyte which can be obtained by using a porous electrode structure. Indeed, all commercial batteries are based on porous morphology. Although the electrochemical theory discussed thus far assumes flat surfaces, it is a logical starting point for the treatment of rough surfaces associated with porous electrodes. It is clear though, that some corrections and modifications are needed to account for the new morphological effects that accompany the increased current density.

C. Porous Electrodes - Classical Approach

DeLevie[36] introduced the first coherent treatment of the effects of surface roughness. He considered a single pore with a uniform cross section, homogeneously filled with electrolyte, and assumed that the resistance of the electrode material is negligible. The actual potentials in the pore are replaced by their average values in planes perpendicular to the pore axis. This may be represented by the transmission line equivalent circuit model of Figure 2.4, which transforms the problem into a one-dimensional one. R is the solution resistance per unit pore length (R has dimensions of ohm/cm), and Z is the impedance of the electrode-electrolyte interface per unit pore length [Z has dimensions of ohm-cm and Z/dx is the

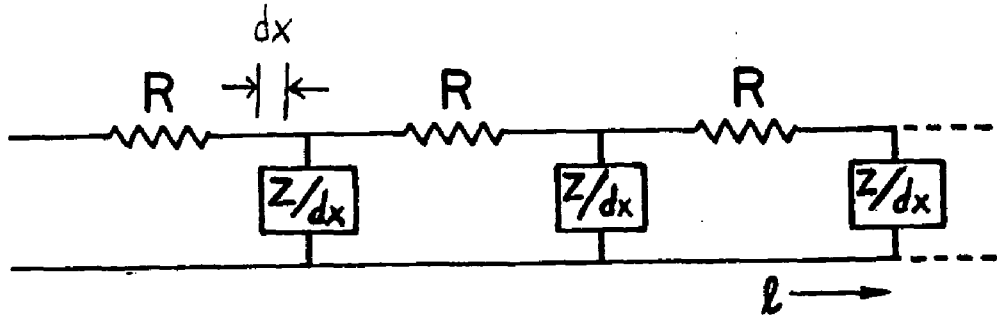


Figure 2.4- DeLevie's single pore transmission line model, for a pore of length l. R is the electrolyte resistance per unit length, and Z/dx is the impedance of the interface (a microscopic section of length dx).

interfacial impedance of a tiny expanse of the pore of length dx - this makes sense if you consider an example of a pure capacitance on the interface of value (C per unit pore length) ; the impedance of a length dx of that will be $1/(i\omega Cdx)$]. Solving the current-voltage equations for the transmission line yields the impedance of the entire pore of length ℓ:

$$Z_0 = (RZ)^{1/2} \coth(\rho\ell) \qquad (2.11)$$

where $\rho = (R/Z)^{1/2}$.

This leads to the definition of a characteristic penetration depth for the current in the pore, $\lambda=1/\rho$. When $\ell>>\lambda$ $\coth(\rho\ell)\approx 1$ which DeLevie refers to as the "semi-infinite pore" where there are effectively no "edge effects" due to the pore end. Another important result of this model is the fact that although Z represents the interfacial impedance of a unit flat surface, the presence of the pore produces a $Z^{1/2}$ dependance for the entire rough surface. For an arbitrary a.c. signal, this translates into a frequency dependance of

$$Z \propto \omega^{-1/2} \qquad (2.12)$$

and in the literature this has been taken to be the "signature" of a rough surface[37], as opposed to an ω^{-1} dependance for a flat surface. (This signature, however is not unique since one can show, for example, that the Warburg impedance which is a pure diffusion phenomenon unrelated to morphology, is also characterized by $\omega^{-1/2}$).

The definition of penetration depth in Equation 2.11 also illustrates the fact that at high frequencies the a.c. current penetrates very little into the pores. The ratio of Z/R controls the penetration due to the fact that at Z >> R, the current tends to flow through the "top" of the transmission line in Figure 2.4 (through R exclusively). On the other hand, when Z << R, then the current tends to flow in Figure 2.4, right through the first "down" path that it finds (through Z). At high frequencies, Z is low and the current short circuits "down" early in its flow through the pore.

D.Critique of Classical Model- A Search for Alternatives...

There are numerous problems with this approach to rough surface impedance behavior. The most striking difficulty is that there is a large body of experimental evidence to the fact that Equation 2.12 is simply not a unique characterization of surface roughness. Some manifestly rough surfaces have impedance behavior of a smooth surface[38]. In addition, experimental studies have found many different frequency dependances not at all limited to exponents of -.5 .[39,40] A general dependance of

$$Z \propto \omega^{-\beta} \qquad (2.13)$$

where $0 \leq \beta \leq 1.0$, emerges from the literature, with the deviation from $\beta=1$ attributed to surface roughness[41,42]. In various studies this fractional frequency scaling has been called a Constant Phase Angle (CPA)[41], Constant Phase Element (CPE)[43], or Fractional Power

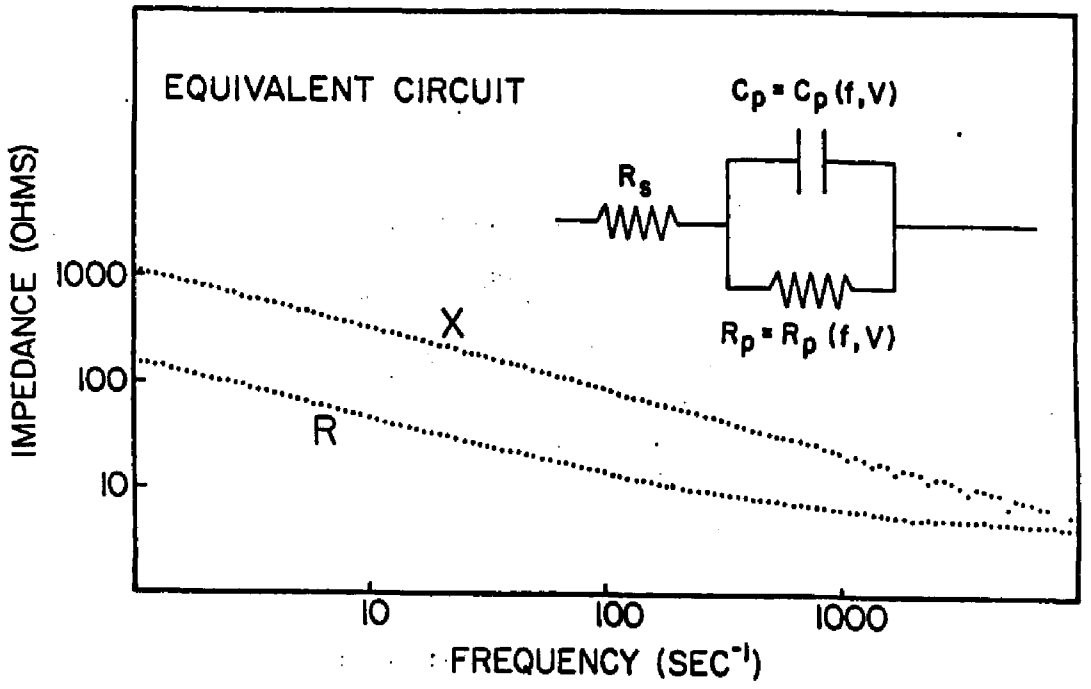


Figure 2.5 - A.C. impedance measurement of polycrystalline CdSe showing that both the real (R) and the imaginary (X) components of the impedance present a power-law frequency dependence. The CPA behavior is evident at low frequencies where R and X have identical frequency scaling exponents.

Frequency Dependence (FPFD)[44], with little new theoretical light shed on the subject. (We will hereafter refer to this behavior as CPA behavior; the constant phase angle refers to the fact that both the real and the imaginary parts of the impedance scale with the same exponent.)

Lyden, Cohen and Tomkiewicz, in an experimental study on the impedance characterization of polycrystalline CdSe, demonstrated a characteristic scaling (CPA) behavior of the impedance as a function of frequency[45] (Figure 2.5). The authors identified this as a percolation-type behavior which they traced directly to the polycrystalline semiconductor electrolyte interface, with the critical exponents within the range of the Imry-Bergman treatment (see Equation 1.9). The percolation behavior may be associated with the morphology of the system, an analysis which arises from the identification of the interface region as a composite medium. However, the model that was employed[46] to explain the percolation mechanism is very specific to that system and has little general applicability. Despite this fact, the work is most noteworthy because it is a first attempt at applying the ideas of the study of composite media to the porous electrode and rough surface problem.

E. ...Which Brings Us Back to Fractals

Very recent studies have shown that CPA behavior arises as a natural consequence of a fractal interface structure.

Liu[47] has constructed a model electrode-electrolyte interface out of a "textbook" fractal structure - a Cantor bar[48] (Figure 2.6a). He assumes that the interface is ideally polarizable and that each "prong" in the bar can be represented by a series R-C element that scales with the "prong" width (Figure 2.6b). By solving for the total impedance of this equivalent circuit he obtains a CPA behavior. The exponent β that he measures (Equation 2.13) is related to the fractal dimension of the bar by β=1-D. It should be noted that the shape of the impedance curve that he obtains is as follows (In chapter 3 we will show very similar results from the Random Network model): On a plot of Log(Z) vs. LOG(ω), at low frequency the real impedance is a constant value, and the imaginary impedance is of slope -1. In mid-frequency both the real and imaginary slopes are -β. Finally at high frequency the real impedance is again constant, and the imaginary impedance has a slope of -1. Liu gives a very intuitive explanation of the scaling in the mid frequency range. If we view the a.c. signal in light of it's penetrability (see the description of penetration depth above) then the lower the frequency the finer is the length scale that is spanned by the current. In the limit of high frequency, the current is unable to penetrate the pores at all due to the large length scale that the current averages over. At low frequencies the current is able to penetrate everywhere due to the fact that it scales very microscopic lengths. It is in the mid-frequency regime in which the length scale of the "current yard stick" is varying, and that is when it is measuring the interface length with varying length yardsticks and the

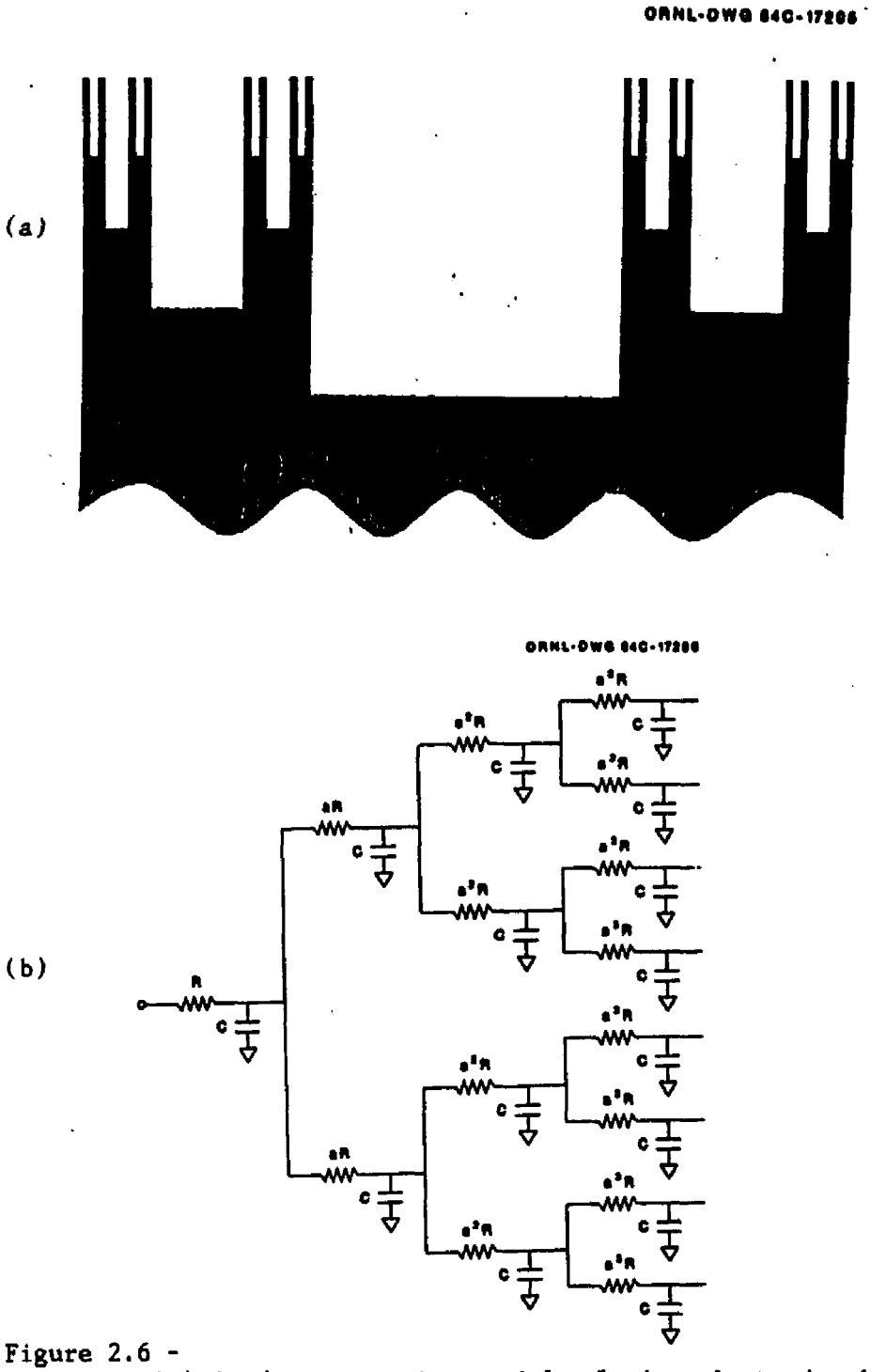


Figure 2.6 -
(a) Liu's cantor bar model of the electrode-electrolyte interface with a fractal surface. The electrolyte is the dark area.
(b) Equivalent circuit representation of the cantor bar geometry.
(From Ref. 47, reprinted by permission of the author.)

self-similarity ratio emerges directly, as in the Richardson curve of Figure 1.3.

More generally, Nyikos and Pajkossy[49] have shown how resistance and capacitance scale with length (for a self-similar distribution of R-C networks), so that when the total admittance is calculated from the combination of R and C it scales as ω^{β}, where β is related to the fractal dimension by

$$\beta=1/(D-1) \quad . \qquad (2.14)$$

This treatment builds on many of the assumptions in our Random Network Model except that no specific model structure is chosen; the authors simply argue that one can choose a fractal surface that may be represented by the self-similar R-C network, and calculate from there. We will consider one specific possible implementation of this idea in chapter IV.

The work of LeMehaute[50] is also noteworthy; he assumes that most real life electrochemical systems have a fractal morphology on the interface, and derives the contributions of a fractal interface using a highly mathematical TEISI model. In this treatment he computes fractal time derivatives of kinetic phenomenon and illustrates scaling relationships.

The DeLevie model is a convenient one dimensional description of trends in the a.c. impedance behavior of porous electrodes. What the model lacks is the morphological sophistication of real-life systems. Similarly, the treatment of Liu has built in to it a realistic model of a rough interface, but it is still two dimensional and not connected with real structures. In the next section, we present our Random Network Model, a theoretical treatment of porous electrodes that mimics the three dimensional heterogeneous distribution of metal and electrolyte, while still providing a means of exhaustive calculations of its a.c. impedance behavior.

III. Porous Electrode Simulation

A. The Random Network Model

The Random Network model is designed to merge the techniques of handling conduction in inhomogeneous media (chapter I) with the theory of the porous electrode-electrolyte system (chapter II). The result is effectively a four phase inhomogeneous system composed of air, electrolyte, metal, and semiconductor. (In this chapter we refer to the phase that is present on the interface as a semiconductor, although following the theory presented in Chapter II the same electrical properties of the interface would arise from the existence of the metal-electrolyte interface.) The electrode problem may actually be divided up into two subcategories: a) The rough electrode is a description of the entire system consisting of the bulk electrolyte and the bulk metal of the electrode, and everything in between (the whole of Figure 3.1). b) The porous electrode is a description of the system from the point of view of a simple heterogeneous distribution of metal and electrolyte, (area (B) in Figure 3.1), and is the focus of this study.

Following the methodology of Equations 1.3-1.5, we assume that our electrode system is composed of a heterogeneous mixture of metal, electrolyte and semiconductor microscopic components. These components are assumed to be large enough to be represented by an average resistivity or complex dielectric constant, the same values as for bulk quantities of the materials. We use the equivalent circuit representations introduced in Chapter II to represent the electronic

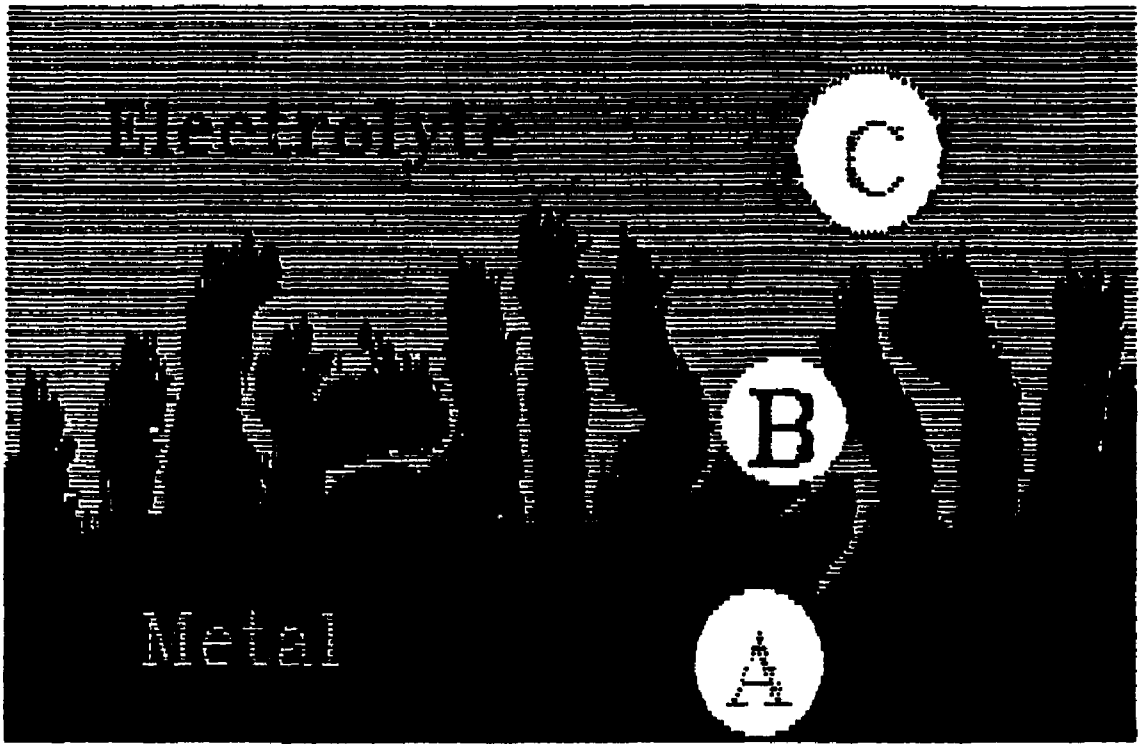


Figure 3.1

Schematic of a rough electrode surface. Regions A and C are comprised of bulk metal and electrolyte respectively, while region B may be viewed as a composite inhomogeneous system.

properties of these phases. Our objective is to obtain the total macroscopic impedance of the system. With a detailed knowledge of the system's composition (since after all, we built it ourselves), we set out to correlate the impedance information with the system's microscopic characteristics. Due to the charge storage behavior of the insulating phase (semiconductor) on the metal-electrolyte interface, the system's impedance is frequency dependent, and much useful information is hypothesized to lie in the impedance vs. frequency relation. This theoretical study mimics the a.c. impedance characterization of electrochemical systems that is widely utilized experimentally[32]. It is important to note that this simulation is not meant to model the d.c. operation of the electrode and to follow in detail the mechanisms of Equations 2.8-2.10, but merely to simulate the electronic behavior of the porous electrode system under small signal a.c. excitations at any given in moment in time.

B. The Algorithm

We define a three dimensional cubic lattice, A(i,j,k), where i,j and k range from 1 to N, N being the total number of nodes. N is chosen so that it is sufficiently large to approximate an infinite system. Construction of the electrode is simulated using a random number generator to place "metal" at various points in the lattice, until the desired porosity is reached (plane k=N is all metal as in the back electrode contact). A check is made of the resulting electrode to insure that it is continuous and that no pieces of

metal are "hanging" in mid-air. Any disjointed pieces of metal are removed, and are randomly replaced on the remaining available lattice points. This process is repeated until a continuous electrode of the desired porosity is obtained. Lattice plane k=1 is defined as electrolyte, to represent the solution; the electrode itself begins at lattice plane k=2. The electrolyte is allowed to "seep into" the pores in the electrode until all accessible pores are filled, by searching for continuous electrolyte paths from the k=1 plane throughout the electrode. Any point on the lattice which has been left empty (i.e. no metal was placed there, and electrolyte was precluded from penetrating to that site) is considered to be "air".

The geometric assumptions in this model are that the various components consist of volume filling cubes centered on the lattice points A(i,j,k). Two adjoining cubes of metal (or even two on the diagonal with each other) prevent the electrolyte from seeping past them. The result of this constraint is that the fluid flow through the system shows classic percolation behavior. If the electrolyte is introduced at k=1, then at low porosities it flows only a small part of the way through to k=N. At porosities above the percolation threshold the electrolyte does indeed flow throughout the system. There is, in fact, a whole body of literature that includes this complication of pore connectivity in an analysis of electrode structure.[54] We have minimized these effects by limiting our consideration of the system to be above the fluid flow percolation threshold. An alternative to this approach would be to consider the

system components as closely packed spheres that permit fluid flow in the interstices, which effectively eliminates all "air" in the system (electrolyte flows everywhere that there is an absence of metal). This introduces discontinuities in the electrolyte connectivity as far as the cubic lattice is concerned, and great pains must be taken to insure that the computer will recognize electrical connectivity betweens islands of electrolyte. This can be done using a more laborious algorithm for calculating the electrical conductances, but we have not done so in this phase of the work.

Each lattice point represents a node in a three dimensional circuit network (See Figure 3.2), and each pair of nodes defines a circuit branch in the network. The impedance of each branch is determined by the characteristics of the two surrounding nodes. Thus, for example, if A(1,1,1) is electrolyte and A(1,2,1) is also electrolyte, the circuit branch defined by those nodes will consist of two series resistors of resistance R_e (the resistance of a microscopic section of the electrolyte). If A(1,1,2) is metal and A(1,2,2) is metal, then the circuit branch defined by those two nodes will be two series resistors of value R_m (the resistance of a microscopic section of the metal). Finally, the branch defined by A(1,1,1) - electrolyte, and A(1,1,2) - metal, is represented by an R_m and an R_e resistor in series, with the addition of a parallel R-C element in series with them to represent the semiconductor-electrolyte interface that results (see Figure 3.2). We refer to "semiconductor" on the interface, although the interface could more generally have an insulating layer or a simple Gouy-Chapman layer on

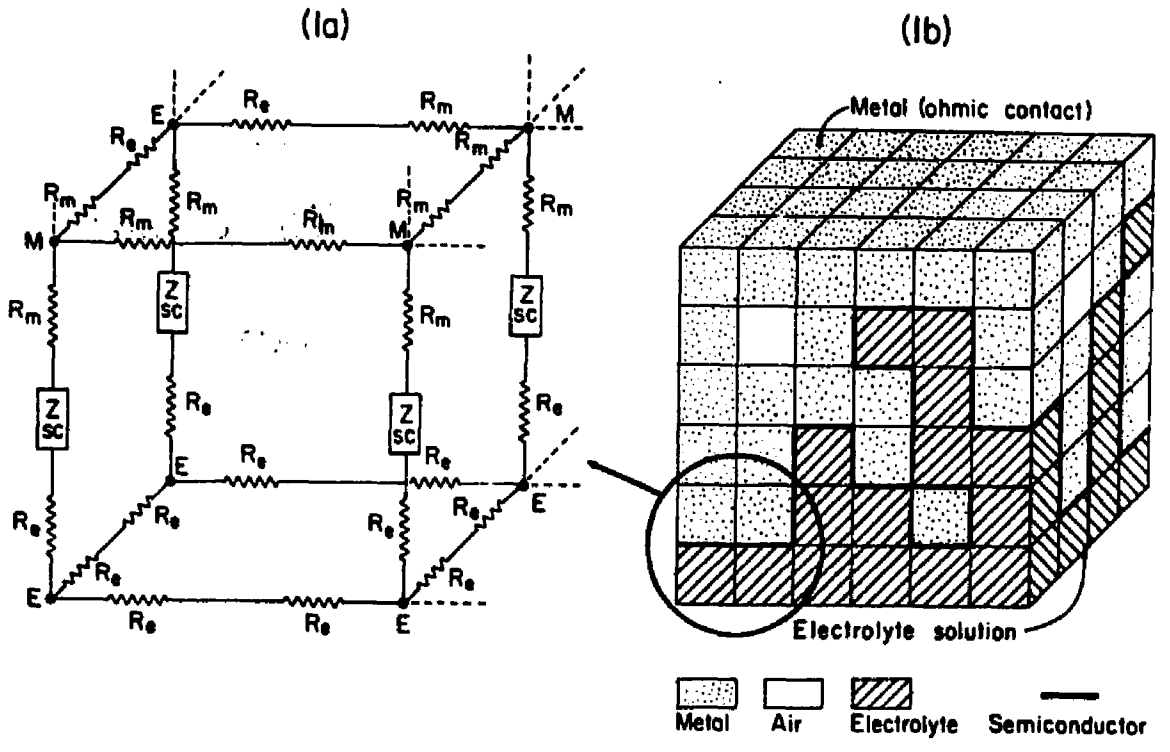


Figure 3.2 - The Random Network Model of a porous electrode. Figure 3.2b shows our idealized picture of the electrode's structure (the metal was distributed using a random number generator), and 3.2a shows a portion of this structure represented by a network of circuit elements.

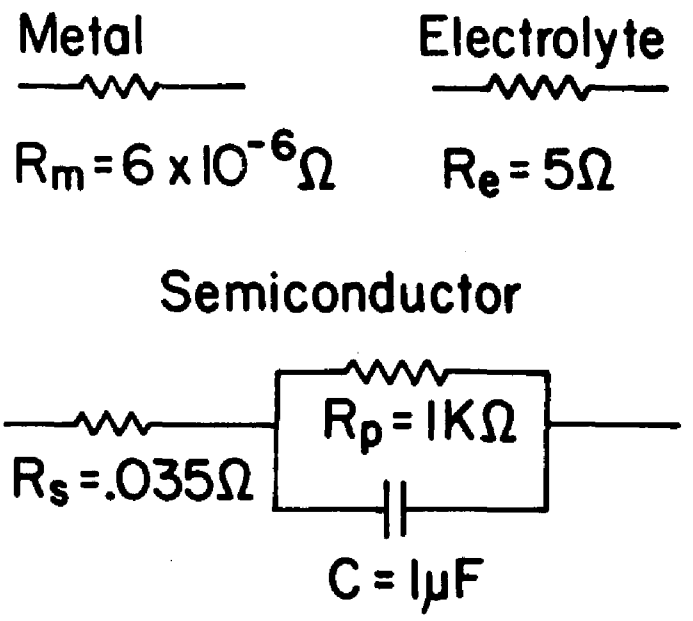


Figure 3.3 - Equivalent circuit representation of the three different phases in our electrode system, and typical values chosen to represent physical processes in realistic systems. Note that the interface is describes as semiconducting, and appropriate circuit element values were chosen to reflect this. To represent an insulating phase on the interface these values may be suitably altered.

it. Any circuit branches leading into nodes that are defined as "air" are taken to be of infinite resistance.

A typical set of values that were chosen as unit impedances for the individual components of metal, electrolyte, and semiconductor are presented in Figure 3.3. These values were obtained by taking the magnitudes of bulk impedance for 1 cm^3 sized samples, (we chose numbers characteristic of Zn and ZnO). In the interest of generating physically meaningful numbers, we defined a unit pore of arbitrary diameter 10μm and scaled the impedance values to represent the dimensions of the unit pore. In effect we consider each of the microscopic components (metal or electrolyte) as being comprised of a cube whose side is of length $10\mu m=10^{-3}cm$. When we normalize the 1 cm^3 values given in Figure 3.3 to this shrunken size, we have to multiply the impedance of the 1 cm^3 "resistor chunks" by a factor of 10^3. This is due to the fact that resistors have a dimensional dependance of ℓ/A where ℓ is the longitudinal length and A is the cross-sectional area. The determination of the scaling factors for the input parameters that represents the semiconductor is not so straightforward. As far as the geometry of the model goes, the semiconductor-electrolyte interface is "pure" surface area with no thickness (i.e. we do not devote any lattice coordinates to the interface- the computer just determines where it is and calculates an enhanced impedance corresponding to the semiconductor on the interface at that point). The thickness of the semiconductor deposit is implicit in the values that were chosen for R_p, R_s, and C. (The numbers chosen in Figure 3.3 approximate a 100 Angstrom

layer of moderately doped semiconductor of area 1 cm^2). Thus, when we normalize the components from 1 cm to 10^{-3} cm, we multiply R_s and R_p by 10^6 and divide the capacitance C by 10^6 because of the reduced surface area. These components are then assembled into a system, as described above (see Figure 3.2) and the total impedance of the system is calculated. The total impedance is then rescaled by an appropriate factor that normalizes the whole system to a 1 cm^3 size. Specifically, for an, 11×11×11 system there is 10×10^{-3}cm per side. Thus, we scale the total calculated impedance by 10^4. This final normalization assures that no matter what size we choose for our lattice size, the numbers will be comparable.

Once the electrical network has been defined, the computer sets up the Kirchoff's law equations,

$$G \cdot V = I \qquad (3.1)$$

where G is the conductance matrix (see Equation (1.3); here: $G_{ii}=\Sigma_j g_{ij}$ and $G_{ij}=-g_{ij}$), V is a vector containing the voltages on each node, and I is a similar vector containing the net current inputs to each node. Due to the fact that our model includes reactive circuit elements, the quantities G, V , and I in Equation 3.1 are complex, and may be represented as 2N-1 equations of the form:

$$\begin{pmatrix} G_r & -G_i \\ G_i & G_r \end{pmatrix} \begin{pmatrix} V_r \\ V_i \end{pmatrix} = \begin{pmatrix} I_r \\ I_i \end{pmatrix} \qquad (3.2)$$

where the subscripts r and i stand for the real and imaginary

components respectively.

A known current is sent uniformly into all the bottom nodes, which is designed to insure an equipotential surface on the bottom plane. We have verified that the resulting voltage fluctuations on the bottom plane are in fact very small. The impedance of the system is calculated between one node on the electrolyte plane (1,1,1) and one node of the top electrode plane (N,N,N) by solving for the V's on all the nodes. For a system of N=15 (a lattice of size 15x15x15 with 3375 nodes) there are up to 6748 equations to be solved. This is a formidable computer problem. Indeed, the task would be prohibitive without taking advantage of the fact that the conductance matrix, G, is a sparse-symmetric matrix.

A number of techniques exist for the solution of this class of problems. A widely used method for solving the simultaneous equations in the Random Resistor lattice is the Gauss Seidel iteration procedure with over-relaxation[6]. This is a very economical method, in both execution time and storage space. A serious drawback of this technique is the relatively strict requirements that it places on the coefficient matrix in order for convergence to be achieved, namely that the matrix be either diagonally dominant, or at least positive definite[51]. For Kirchoff's law problems with purely resistive components, diagonal dominance is assured, since the diagonal elements of the conductance matrix are simply the sum of the off diagonal elements. In our problem, with complex impedances, we find that at frequencies for which the real conductance

approaches the same order of magnitude as the imaginary conductance, the Gauss-Seidel procedure does not converge. This is due to the form of the Equations 3.2 where there are many additional off-diagonal elements due to the G_i values. When these imaginary components are large, the matrix is no longer diagonally dominant, and becomes indefinite. For most of our computations, we have resorted to the use of a Gaussian elimination routine which takes advantage of both the sparseness and the symmetry of the G coefficient matrix[13]. Only the nonzero elements in the upper half triangle of the matrix are stored, and an efficient pivoting strategy is chosen to minimize non-zero matrix fill during the pivoting and to minimize the number of multiplications required in the solution. In practice, the technique is approximately an order of magnitude more expensive to use than Gauss-Seidel in both speed and storage requirements. The advantage of using the Gaussian elimination technique is that a solution is guaranteed for all but the most ill-conditioned problem. The solutions obtained from the Gauss-Seidel procedure (in the range in which convergence is reached) are in complete agreement with the results obtained using the modified Gaussian elimination technique.

The final step in "fine tuning" the model is to choose a minimum lattice size that provides some degree of statistical significance but is still within reasonable limits of available computer time. If the random network model is a true statistical model then the calculation should not be sensitive to the lattice size of the system. Thus the minimum lattice size was chosen by solving a

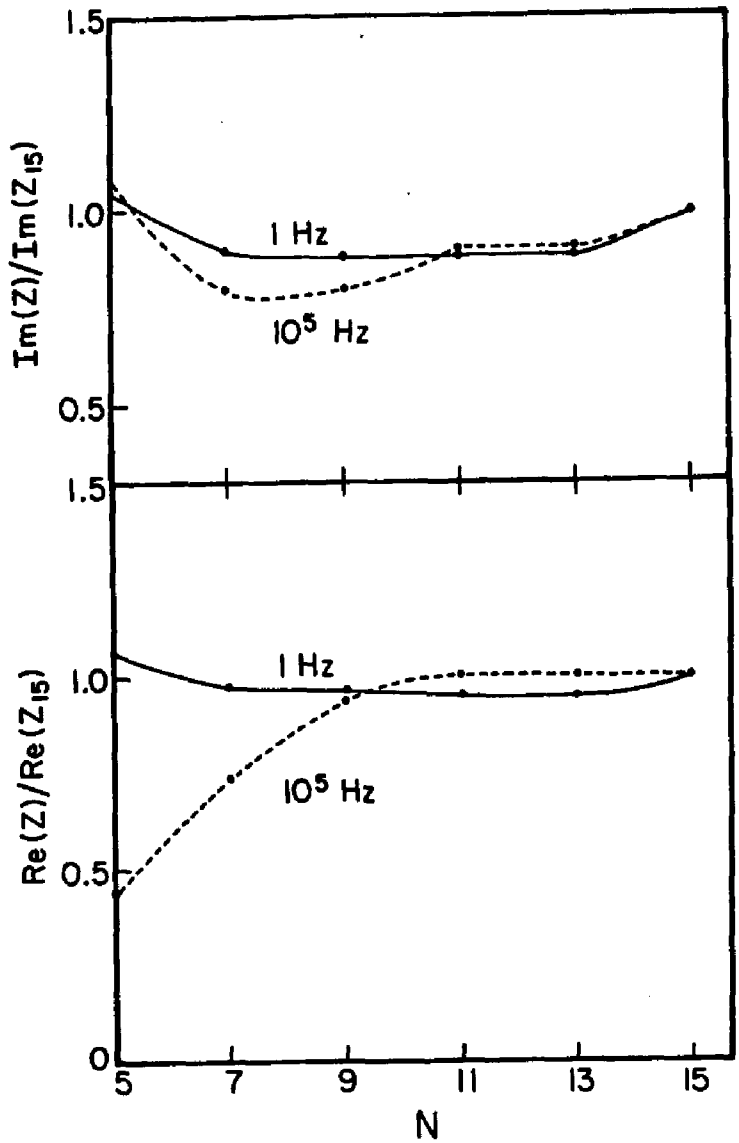


Figure 3.4 - The impedance of the system in the RNM as a function of lattice size. Although the solution is converging, numerical limitations prevented us from solving for larger systems, and some statistical deviation is assumed to result.

single problem on increasingly large lattice sizes until the answer converges to a fixed value. The results of this study is presented in Figure 3.4. There are two significant difficulties that we encountered in trying to solve for the impedance of our system. Firstly, we were not able to eliminate all "dead end" conductive paths as Kirkpatrik did in his composite media simulations. He was able to do this because all he was concerned about was the system's degree of connectivity. Thus , he was able to choose only the cluster's "backbone" and measure the resistance of that to decide on the conductivity of his system. In our system, the capacitance of all surfaces contributes to the impedance, and it is not clear which current paths may be discarded, if any. Secondly, as already mentioned, the complex impedance doubled the size of the network, and turned the problem from a well-behaved calculation into an ill-conditioned one. For most of the results in this study a system of size 11×11×11 was used. Computational limitations forced us to choose this minimum size, although continuous boundary conditions were employed to minimize finite-size effects.

C.Homogeneities & Systematic Correlations

From the foregoing description, it is obvious that our model differs in a number of important ways from the purely random conductor-insulator distributions of Chapter I. For one thing, although the distribution of metal is governed by a random number generator, we have modified the random statistics to require a connectivity in the metal for a realistic representation of an electrode (although some metal powder - electrolyte electrode

systems have been experimentally explored). One effect that this has on the electronic properties of the system is that metal connectivity is assured across the system - it always lies above the percolation threshold. Another constraint introduced is the fact that the three phases electrolyte, semiconductor and metal are not mixed homogeneously, but are distinctly separated by an orderly juxtaposition to each other (see Figure (3.3) for a schematic representation of this). Even in the pores where the paths are contorted and the surface rough and complicated, there is always a continuous distribution of electrolye and metal respectively on either side of an interface. It is thus impossible to formulate an effective medium theory for this arrangement. Any treatment of local inhomogeneity in an average medium such as in chapter I cannot be made to preserve this structured distribution around an interface.

The RNM simulation is a valid approximation of conduction in a porous electrode system, and superior to previous treatments of electrodes precisely because it preserves the stochastic nature of the system to the extent that it is present. Thus far, that aspect has not been properly treated in the literature.

It may be more precise to say that, unlike the conductor-insulator problem which presents a bulk inhomogeneity in two or three dimensions, the porous electrode system is a problem of <u>a quasi two dimensional inhomogeneity</u> of the interface. Although the system's bulk component composition will also exhibit bulk properties and its

behavior will be governed by these bulk-related phenomena, it is the properties due to the random morphology of the rough surface that is of greatest interest to us, and we will attempt to show when these interfacial contributions dominates the system's properties. We define the surface (or interface) in the model as being those metal sites that are adjoined by one or more electrolyte sites.

D. The Model Parameters

The input parameters of the model consist of the porosity, p, the electrolyte resistance, R_e, and the metal electrode resistance, R_m (all bulk impedance values are for a typical 1 cm^3). By varying the system input parameters, we can simulate various physical conditions in our electrode system. Variations in R_e correspond to different electrolyte concentrations. Z_{sc}, the impedance of the semiconductor is represented by a resistor, R_p, in parallel with a capacitor, C_p, which are both in series with a resistor, R_s (Figure 3.3). Varying these parameters correspond to variations in the state of charge of the electrode (i.e. the thickness of the semiconductor buildup), and the doping level of the semiconductor buildup (determined by the reaction mechanisms at the electrode surface during electrode discharge).The resulting (frequency dependant) impedance of the interface is:

$$Z_{sc} = R_s + \frac{R_p}{1 + (\omega\tau_p)^2} - j \frac{\omega R_p^2 C_p}{1 + (\omega\tau_p)^2} \qquad (3.3)$$

where $\tau_p = R_p C_p$. R_s corresponds to the resistivity of the bulk semiconductor, R_p to the faradaic current leakage through the

interface (a space charge region if it is a semiconductor), and C_p to the capacitance of the interfacial charge storage region. We obtained the C_p value by assuming a semiconductor buildup of approximate thickness ℓ'=100 A, an area of a=1 cm^2 and the relationship

$$C = \frac{\varepsilon_0 \varepsilon a}{\ell'} , \qquad (3.4)$$

Where ε_0 is the permittivity of free space and ε is the dielectric constant of the material on the interface (we chose ε=10 representative of a ZnO deposit). As previously mentioned, the primitive size of our lattice was taken to be 10μ, such that for a lattice of size 11×11×11 we have a cube of side 100μ.

E.Model Output- Bulk Contributions

Figure 3.5 is a typical result from the electrode RNM at a given p=.4, using the input values of Figure 3.3. We may loosely categorize the imaginary impedance vs. frequency behavior as being composed of three regimes: At low frequency (f<100 Hz) we have a slope 1 behavior on the log-log plot, at high frequency (f>10^5 Hz) we see a slope (-1) dependance, and in the middle frequency regime is some sort of transition region or regions. The low and high frequency behavior is the same as would be observed from measuring the impedance of an R-C circuit, however, when an attempt is made to calculate the capacitance, it is observed that the capacitance has different values at low and high frequency. The measure of capacitance is usually taken as an indication of a measure of some active interfacial area, and thus it would appear that there are two

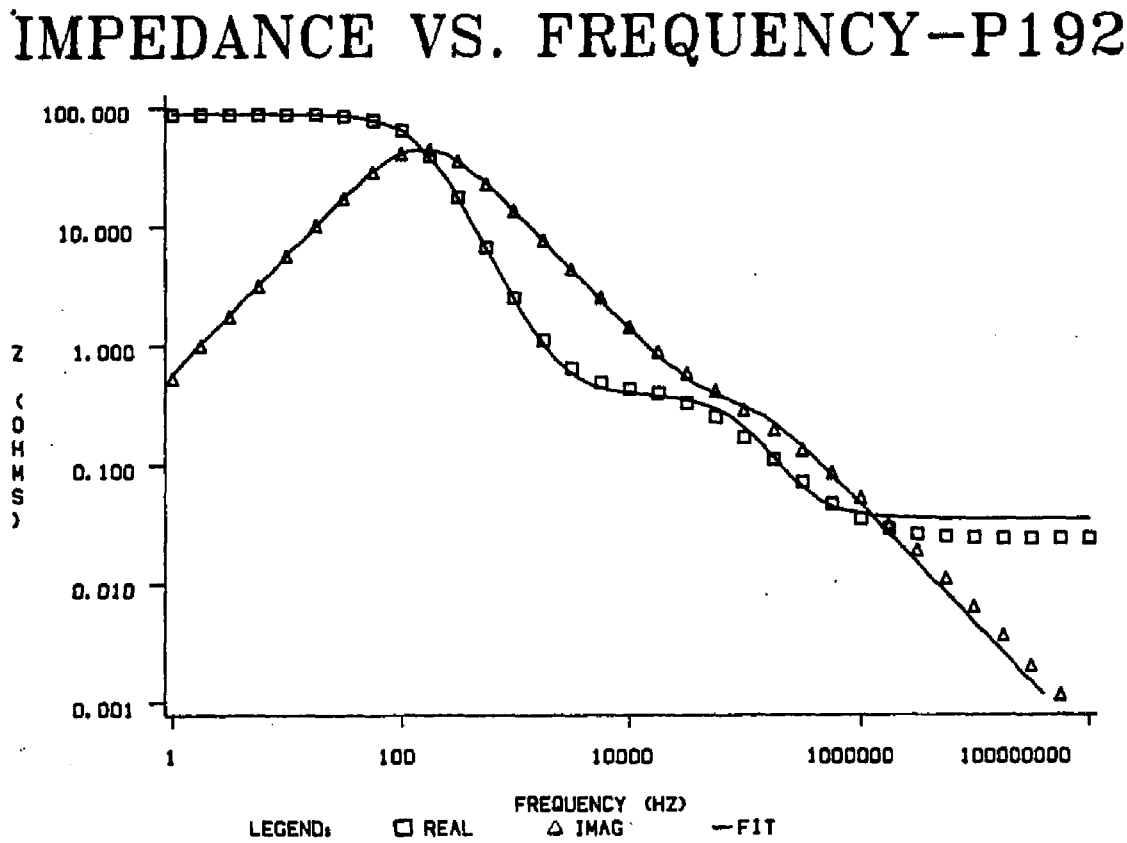


Figure 3.5 - A typical data set from the RNM computation. The data points are from the simulation, and the solid line drawn through the points is a fit to the data (see section III.G).

characteristic areas being measured in this system; one at high frequencies and another at low frequencies.

DeLevie[36] had suggested that for cylindrical pore electrodes the penetration depth of the ac signal decreases with increasing frequency (see Chapter II). If we accept this as applicable to our electrode model, this would account for the lower capacitance being measured at higher frequencies- the current is not penetrating far into the pores and hence the current is scanning less area. It is easy to verify that in fact that is what is occurring in our RNM electrode. Figure 3.6 depicts the current distribution as it flows through the semiconductor (labeled S), electrolyte (E) and metal (M). These values were obtained by actually calculating the current flow throughout the nodes in our model. At low frequency (Figure 3.6a) all the semiconductor interfaces are passing an appreciable amount of current; all of the electrolyte paths are contributing to the current flow as well (in order for the current to reach all the interfaces it must go through all the electrolyte). In contrast to this, at high frequency (Figure 3.6b) we note that only a small minority of all the semiconductor interfaces are passing any current at all.

Figure 3.1 schematically depicts the process. The current flows through the bulk electrolyte (labeled C) and tries to get through to (A), the metal of the electrode. Before it can do that, it must pass through the interface (labeled B). How (and where) it passes through the interface is the determining factor in the ac

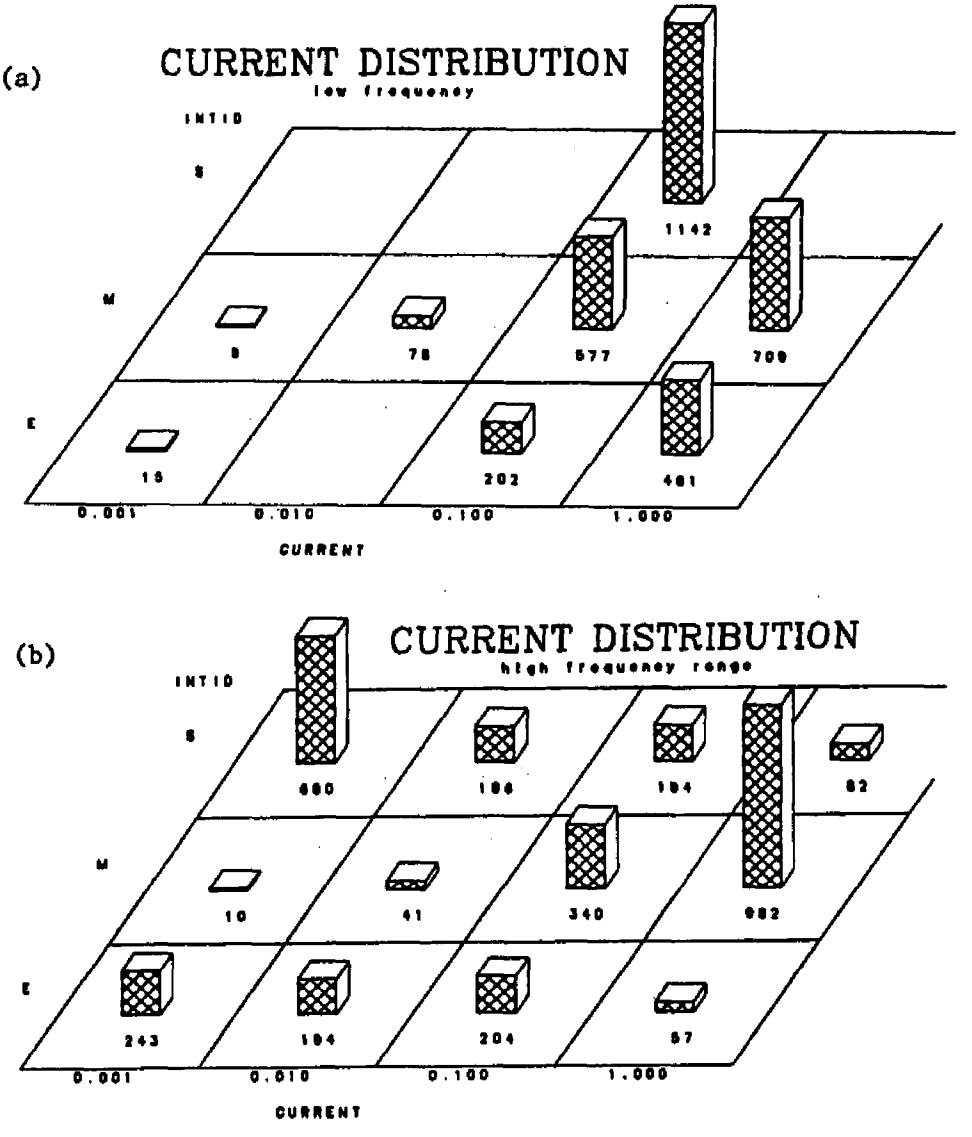


Figure 3.6

The bar charts depict the current distribution among the three internal phases in the electrode. The left axis corresponds to the three phases electrolyte, metal, and semiconductor. The bottom axis is the current magnitude (in % of input current), and the Z axis corresponds to an enumeration of how many of the microscopic components are carrying the stated amount of current. Thus, at low frequency (3.6a) all of the semiconductor interfaces are carrying a significant percentage (.1%) of the current because there is a large current flow deep into the electrolyte. At high frequency (3.6b), most interfaces and electrolyte are carrying almost no current, since the current flow shorts into the metal before traveling deeply into the pores through many electrolyte components.

capacitance. If the interface is an easier path for the current to go through than continuing through the electrolyte in the pores (since the pores may be narrow the resistance could be high), then the current will "short" right through (B) into (A) at it's earliest opportunity (near k=1 in our model). If on the other hand, the journey through R_e is the least-resistive path, the current will travel through the electrolyte deep into the pores before being forced to enter the interface into (A).

The low frequency capacitance value would then be a measure of the total surface area of the electrode (at least as far as the current got), and the low frequency impedance data may be taken to be a measurement of the electrode surface area. Further verification of this may be obtained by increasing the electrolyte resistivity and watching if the value of C_{lf} changes as the electrolyte becomes more and more unsuited for performing this area measurement. Sure enough, Figure 3.7 shows how we can make the value of C_{lf} approach C_{hf} simply by raising the R_e. Conversely, we can raise the value of C_{hf} simply by lowering R_e which improves the penetrability of the current even at high frequency. Figure 3.7 shows this as well, but we were prevented from making $C_{hf} = C_{lf}$ due to numerical difficulties.

To further test the current's area scanning capability, we varied the porosity and measured the corresponding C_{lf}. Figure 3.7 shows how closely the low frequency capacitance is able to follow trends in the roughness factor. (The roughness factor, ρ, is a measure of

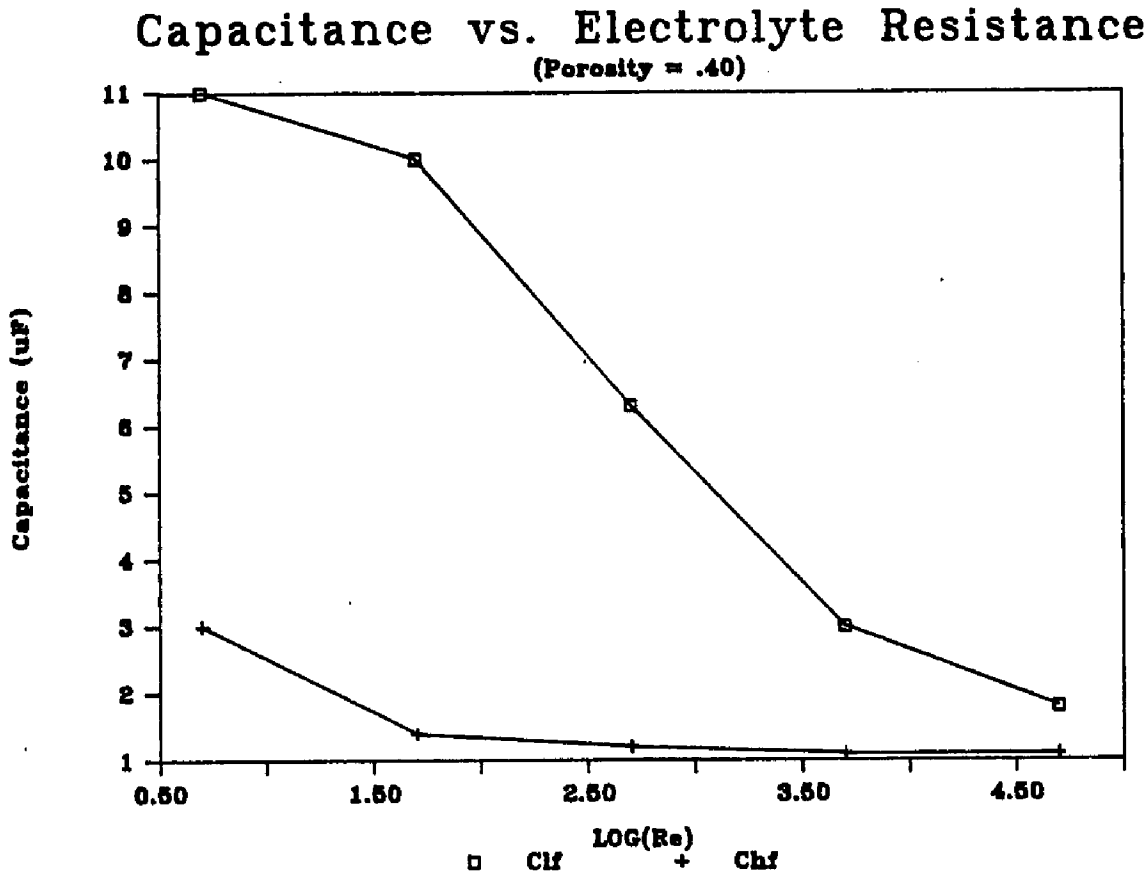


Figure 3.7 - The change in the measured low frequency and high frequency capacitance as Re is varied. At high Re, even the low frequency capacitance fails to measure the apparent surface area of the pores.

total interfacial area of a rough electrode to the geometric surface area- the area of a flat 1 cm^3 electrode. The capacitance in our system is also normalized to a 1 cm^3 value by our simulation so we may compare ρ and C_{lf} directly.) The value of ρ in Figure 3.8 was computed by simply counting up all the metal-electrolyte unit interfaces in the computer model for each porosity. Figure 3.8 is a conclusive demonstration of the utility of the value of the low frequency capacitance from the impedance measurement. At the same time it also demonstrates that C_{hf} is a measurement corresponding to the geometric surface area. Another interesting feature of the ρ vs. porosity plot is that ρ is not a unique measure of porosity. A roughness factor of .9 may be obtained by a porosity of 0.36 or 0.62 . The bell shaped curve is due to a competition of two opposing factors for the increased surface area. On the one hand, as the volume fraction of metal increases from zero, the more available metal surfaces there are for metal-electrolyte interfaces. On the other hand, as the volume fraction of metal increases beyond .5 or so, the probability of having a metal site bordered by another metal (and hence no metal-electrolyte interface) also increases!

To quantify the penetrability of the current with decreasing frequency, DeLevie had introduced the penetration depth, $\lambda=(|Z|/R)^{\frac{1}{2}}$ (see chapter II;this R is related to our parameter R_e). The concept should be applicable here as well, except that a single-valued penetration depth is of little meaning in this 3-D system, as opposed to DeLevie's one-dimensional pore. The current may penetrate deeply into wide necks of the pores, but get choked off in a

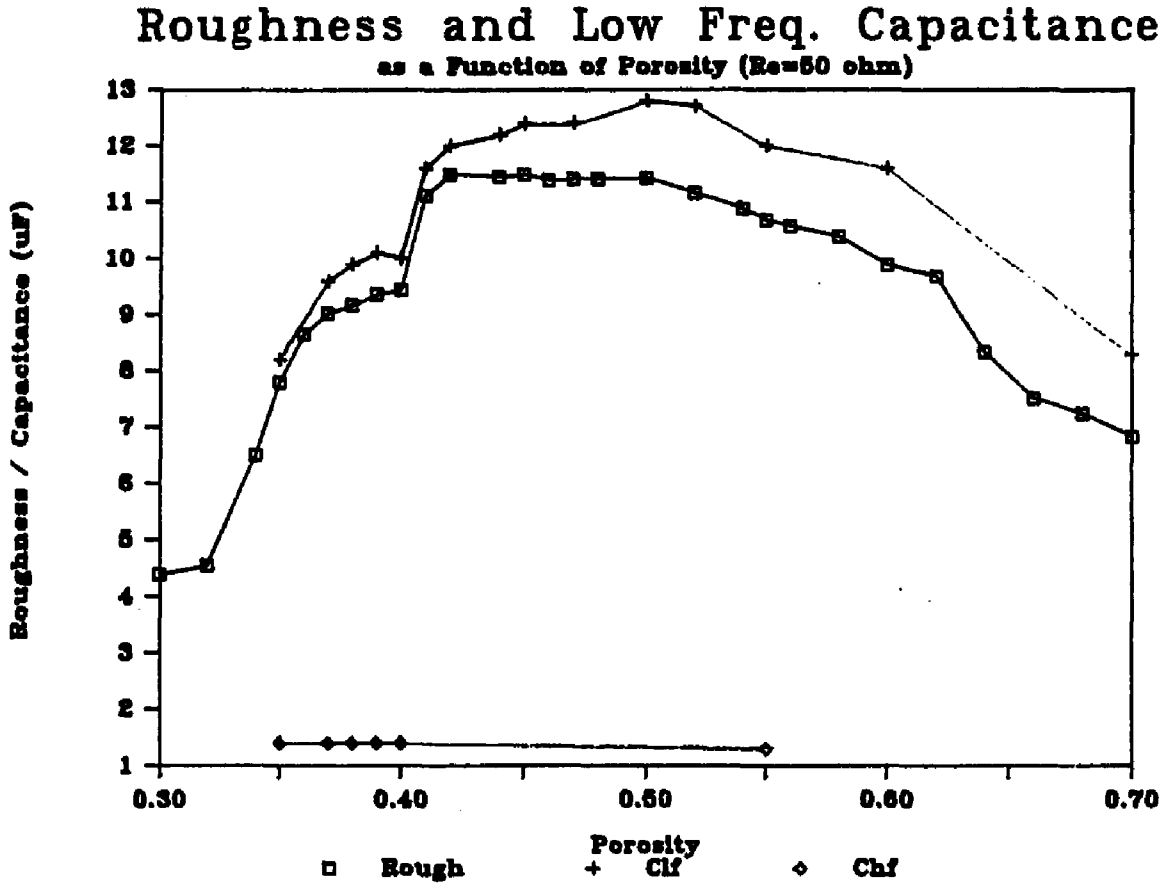


Figure 3.8 - A comparison of the actual surface roughness and the measured low frequency capacitance illustrates that the low frequency capacitance may be used to measure the apparent surface area of the pores. For comparison, the high frequency capacitance is shown, and it is not sensitive to changes in electrode surface.

narrow horizontal cross link. Nevertheless, the value of lambda is useful if thought of as a measure of the current "penetrability". In Figure 3.9, $\sqrt{Z_{sc}}$ is plotted as a function of frequency. We may draw a horizontal line across the figure corresponding to any value of $\sqrt{R_e}$ that we choose to introduce into our system. At the frequencies that the R_e line falls below Z_{sc}, then the current will flow through the electrolyte. At frequencies for which Z_{sc} falls below R_e, then the current will short through the interface.

F.Duplicating the DeLevie results as a Special Case of RNM:

To demonstrate that our RNM is a more general case of the DeLevie model, we can numerically simulate DeLevie's cylindrical pore electrode by constructing uniform pores with our random resistors. The DeLevie parameters R and Z differ from our Z_{sc} and R_e due to the fact that the former quantities are normalized to the unit pore length. We may consider a resistance and impedance per unit length as (see Figure 3.10):

$$R = \frac{\rho}{A} \qquad \text{and} \quad Z = \frac{\rho' \ell'}{4\sqrt{A}} \tag{3.5}$$

where A is the geometric area of the pore, ℓ' is the thickness of the semiconductor layer, and ρ' is an effective (complex) resistivity of the semiconductor. Consider an ideally polarized interface where

$$|Z_{sc}| = 1/(\omega C)$$

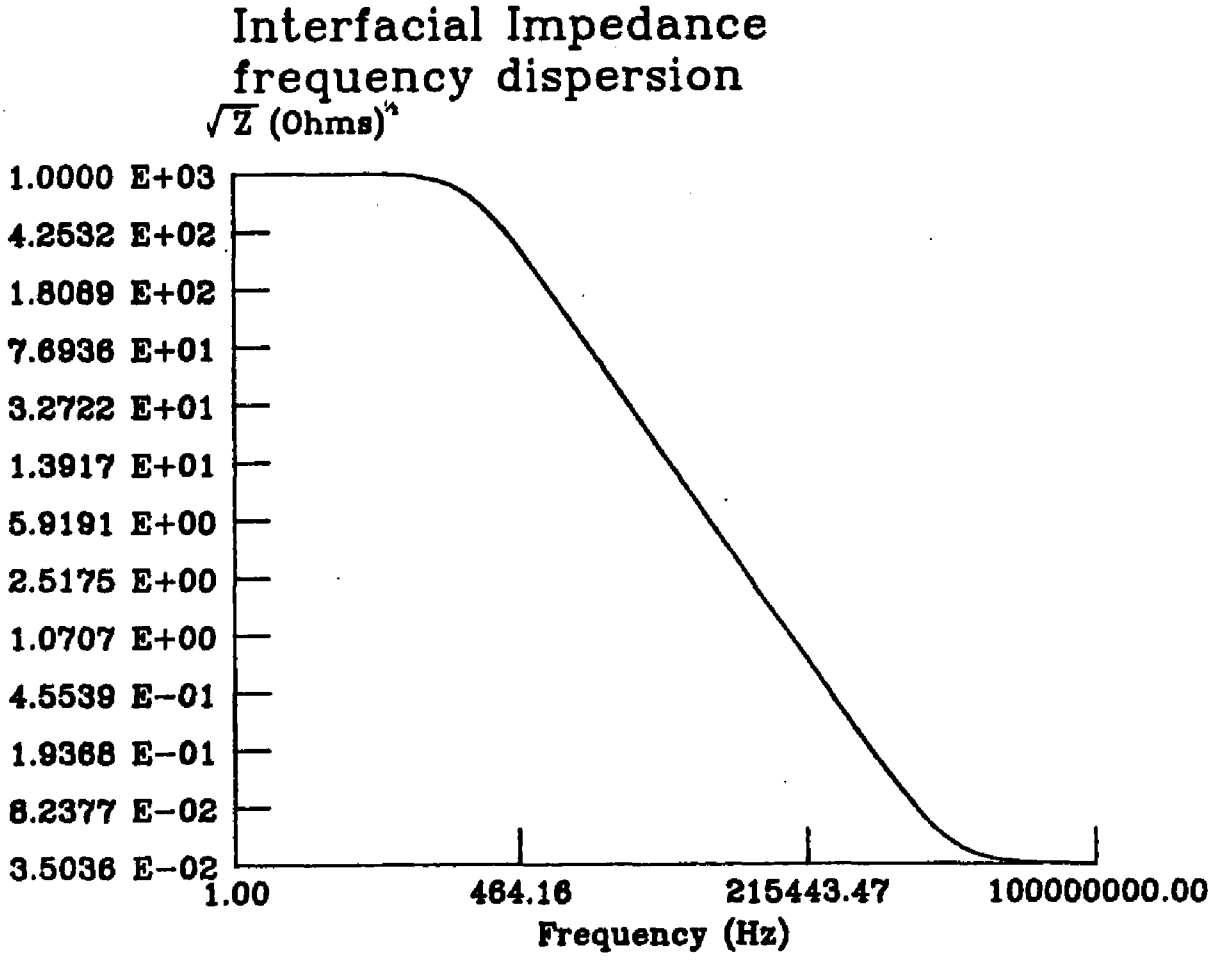


Figure 3.9- The impedance of the interface alone as a function of frequency.

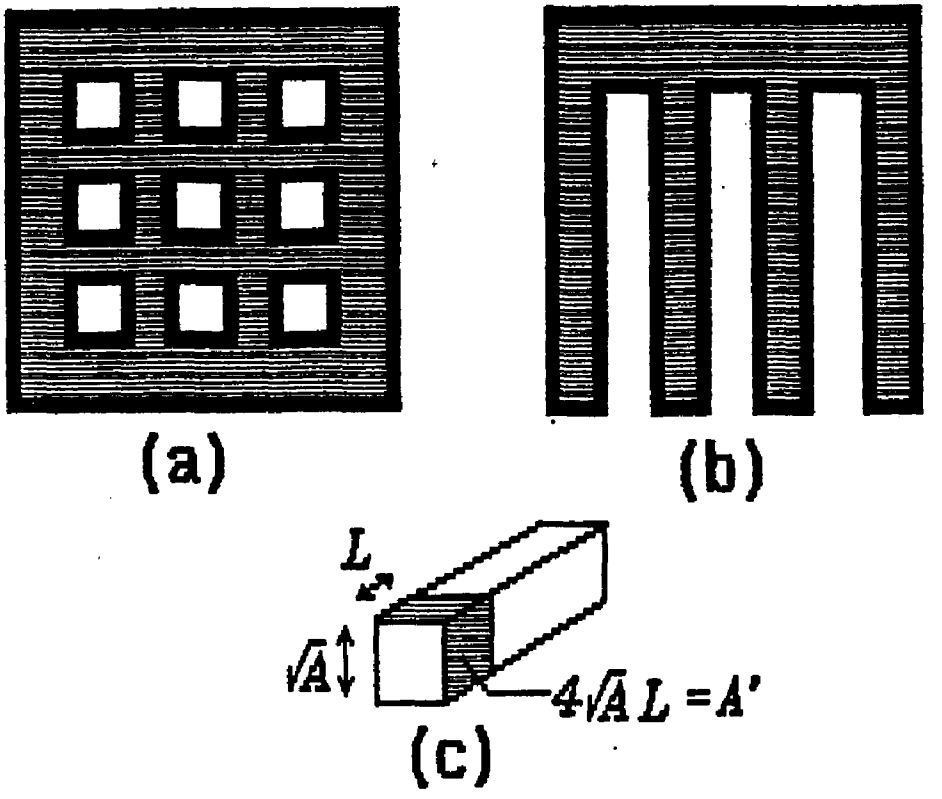


Figure 3.10

Rectangular pores set up to explore simple pore behavior. (a) is a view of the 9 pores in a "top view" cross section, and (b) is a "side view" cross section. (c) illustrates the dimensions of each pore.

and a dielectric solid, for which we may define

$$C = \frac{\varepsilon_0 \varepsilon a}{d}$$

where ε is the static dielectric constant, ε_0 is the permitivity of free space. From Figure 3.9 $a=4\sqrt{A}\ell$, $d=\ell'$ so that

$$Z_{sc} = \frac{\ell'}{\omega \varepsilon_0 \varepsilon 4\sqrt{A}\omega} = \frac{\rho' \ell'}{4\sqrt{A}} \qquad (3.6)$$

Therefore the impedance per unit length is Z_{sc}/ℓ and

$$\lambda = \left(\frac{Z}{R}\right)^{\frac{1}{2}} = \left(\frac{\ell'\sqrt{A}}{4\varepsilon_0 \varepsilon \rho \omega}\right)^{\frac{1}{2}} \qquad (3.7)$$

taking the values (as before), $\ell'=100A=10^{-8}$m, $A=10^{-10}$m, and $\varepsilon=10$, we get a penetration depth of

$$\lambda = \begin{cases} 2.8 \times 10^{-3} & \text{at 1 Hz} \\ 9.0 \times 10^{-5}\ \text{m} & \text{at 1 KHz} \\ 2.8 \times 10^{-6} & \text{at 1 MHz.} \end{cases}$$

For our system of (as previously mentioned) size 100μ on a side, this corresponds to a current flow throughout the electrode at under 1 KHz, which is in agreement with the location of the low frequency regime in Figure 3.5.

To construct a simple cylindrical pore system with the RNM, we arrange our resistors so that they are distributed as in Figure 3.10, with the shaded area denoting the metal and the white area being the pores where the electrolyte flows. We solve for the current distribution in the system, and plot I, the current flow in

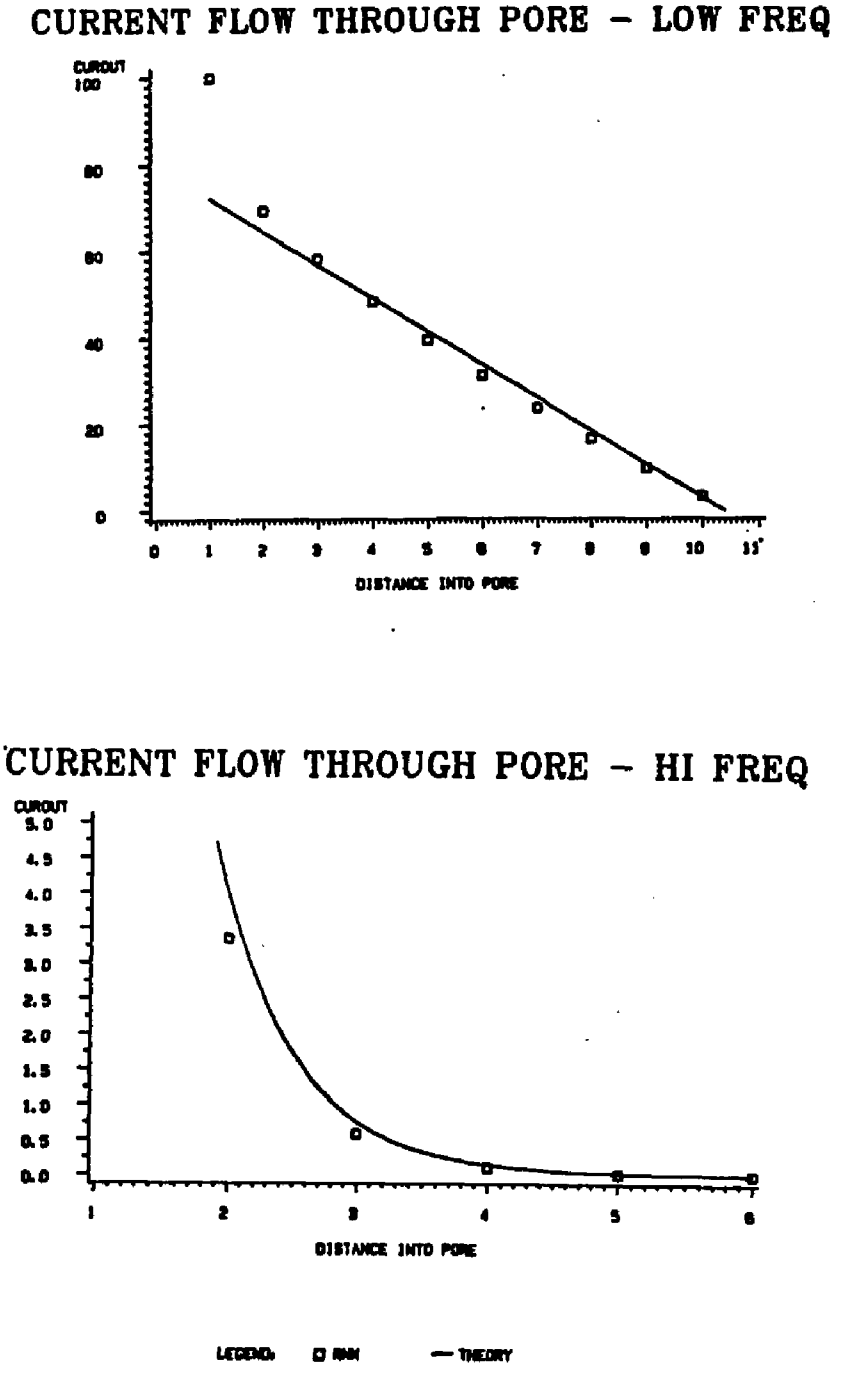


Figure 3.11- Magnitude of current flow into the pores as a function of distance, using the pore structure of Figure 3.10. The data points are the results of the calculation, and the line is a fit from the DeLevie model.

the Z direction, as a function of z, the distance into the electrolyte (Figure 3.11). Figure 3.11a is the result for high frequencies (1 MHz with an R_e of 1 KΩ). The shape of the curve is exponential, and following DeLevie[52] we look for a result of the form

$$I = I_0 \exp^{-z/\lambda} \qquad (3.8)$$

Upon fitting this function to the data of 3.11a, we arrive at λ=.6 which indicates that at high frequency the penetration depth is not even through to the first lattice point. Figure 3.11b shows the results at low frequency, and we may also try to fit that curve to Equation 3.8. However, if $\lambda >> z$ then the exponential may be expanded as

$$I_0 = I\ (1-z/\lambda) \qquad (3.9)$$

We perform a linear fit of this function to the data in Figure 3.11b and we obtain $\lambda = 10.4$. This verifies the frequency behavior is as DeLevie postulated, and the exponential behavior of the current in the pores.

G.Other Features of the Impedance Curve

Returning to the impedance curve of Figure 3.4, we identify more features that may be correlated with the input parameters. As an alternative to the high frequency and low frequency capacitance that we have been using thus far to model the frequency response of the system, we can construct an equivalent circuit model for the system

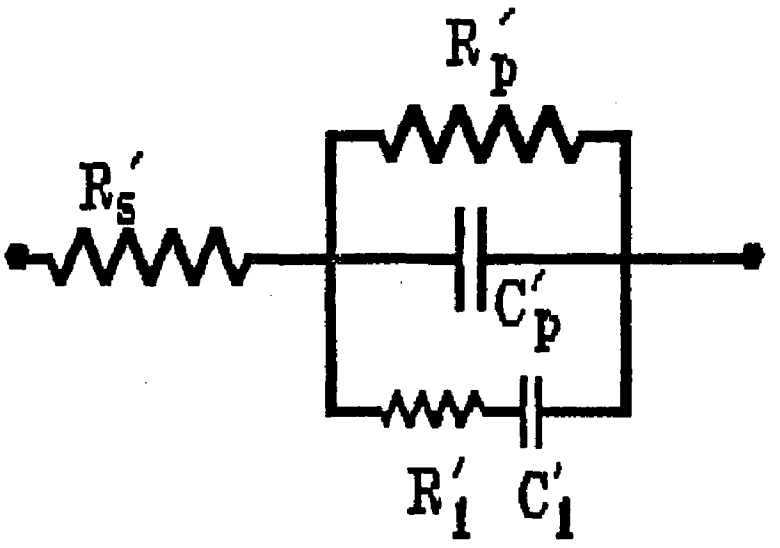


Figure 3.12

Equivalent circuit model of the total electrode impedance.

that is valid at all frequencies. Figure 3.12 introduces some more circuit elements, but this time they correspond to the output parameters of the system (the total impedance calculated) and hence are denoted by the ' (prime) notation. The curve drawn through the data in Figure 3.5 is a theoretical fit to the impedance dispersion curve, using the circuit of Figure 3.12. By fitting the theoretical impedance curve, we evaluate the output parameters (the circuit element magnitudes).

We have varied the input parameters, and correlated that with changes in the equivalent circuit output parameters. R_s' is determined from the values of the real impedance at high frequency. Its value is usually governed (dominated) by the value of the input electrolyte resistivity, R_e. R_p' is determined by examining the system's real impedance at low frequency. The space charge resistance, input parameter R_p, of the semiconductor on the interface usually decides the value of R_p'. The peak in the imaginary component of the impedance occurs at $\omega\tau_p=1$, where $\tau_p=R_pC_p$. These dependencies can be altered. For example, if the value of R_e is made very small, then even at high frequencies it is the R_s of the interface, the bulk impedance of the insulator, that will dominate the high frequency R_s'. If R_e is made very large, then at low frequencies it is R_e that will dominate the real impedance, and not the interface's R_p.

H.Summary - Bulk Properties of the System

We have shown how the system's bulk properties may be related to

it's microscopic composition. The prominance of the electronic properties of the interface is largely due to the semiconductor on the interface being the sole contributor to the charge storage in the system. There is an interplay between the deliverance of the current to the interface by the electrolyte and the interface's ability to pass current (and we have shown that this interplay is highlighted by the frequency dependance of the impedance). The penetrability of the current is solely determined by the complex impedance of the interface (Figure 3.8) and the electrolyte's concentration. The increased electrode surface area can be determined from the low frequency impedance data, whereas the high frequency impedance typifies flat electrode behavior of the area outside the pores. Thus far, however, we have not focused on the ramifications of the porous electrode's surface inhomogeneities and irregularities other than as a means of increasing the total surface area available for the electrochemical reaction. This information is contained in the middle frequency impedance data (see section II.E), and it remains for us to attempt to extract some unique interpretation from it.

I.Model Output-Surface Contributions

In an attempt to assess the surface roughness contributions to the impedance, we address two questions:

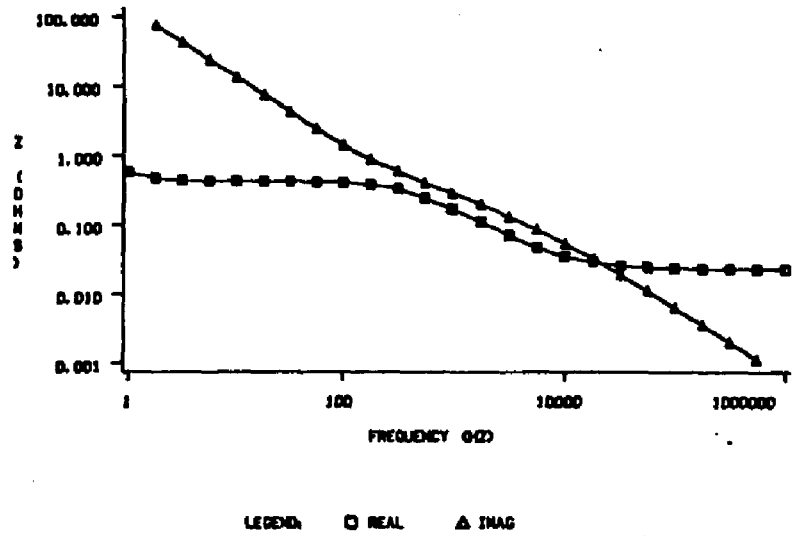
1) Can a characteristic topology and morphology of the RNM porous electrode surface be identified? Can the features of the surface be quantified?

2) Can a characteristic morphology so identified be correlated with the electrode's impedance characterization?

To answer the second question first, recall that in section I.E we cited the work of Deutscher et. al. in showing a fractal impedance behavior of metal film fractal structures, and in section II.E we cited the work of Liu who used a similar approach in calculating the impedance behavior of a fractal electrode interface. These studies both utilized systems whose impedance was linearly dependent on area, so that a self-similar area-length relationship translated into a self-similar impedance-length relationship. Thus, Liu chose an ideally polarized electrode interface so that the interfacial impedance was that of a pure capacitor, to obtain the simple area dependent impedance. Following suit, we change the RNM interfacial impedance from the general non-IPE treatment of Figure 3.3 to that of an IPE by eliminating R_p. It follows that if we identify the electrode surface as fractal, we can directly correlate it with a CPA appearing in the impedance spectrum.

Solving the RNM for the ideally polarized interface presents some additional numerical and computational difficulties. This is due to the fact that at low frequencies the impedance of the interface is so high. This prevents us from examining the results of all input parameters, including the configuration of figures 3.3 and 3.5. By reducing the capacitance from 10^{-6} to 10^{-4} Farads, we succesfully obtain the results of Figure 3.13, which indeed shows a CPA behavior in the intermediate frequency range. From Figure 3.13b we evaluate

IMPEDANCE - IPE - P220



MID FREQUENCY CPA - P220

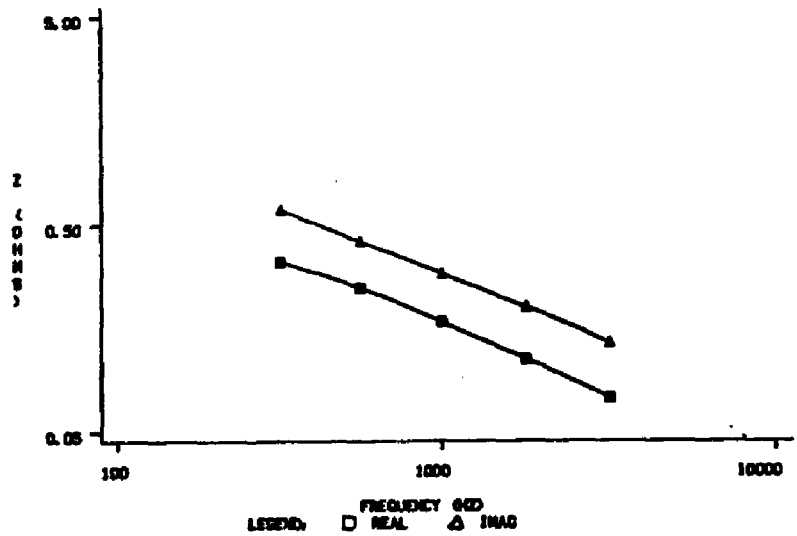


Figure 3.13- CPA behavior of the impedance in the RNM when we assume an ideally polarized electrode.

the slope of the log-log plot as -.67. We find, however, that the value of the CPA slope may be altered simply by changing the impedance values of the constituents of the system. Changing the values of C_p or of R_m (the metal resistance) varies the CPA slope from .6 to .8 . This strongly supports the notion that the appearance of a CPA dependance is not uniquely determined by the purely structural morphology of the electrode.

To further explore the possible fractal nature of our surface, we utilize Equation 1.14, to calculate the density-density correlation function for the surface metal sites. In our calculation we averaged the two particle correlations over the six nearest neighbor directions, and we did not employ continuous boundary conditions. The correlation function as a function of radius does not show a linear relationship on a log-log plot (Figure 3.14). In fact, the behavior is not unlike that of a Richardson curve for a circle (Figure 1.3). We verified that this was not due to finite-size effects by showing that the same behavior was obtained irrespective of system size. The same non-fractal correlation is obtained irrespective of porosity (Figure 3.15) which suggests that our RNM porous electrode surface is not fractal. The fact that our electrode does not exhibit fractal properties further undermines the Liu hypothesis which attempts to uniquely tie CPA behavior to fractal morphologies.

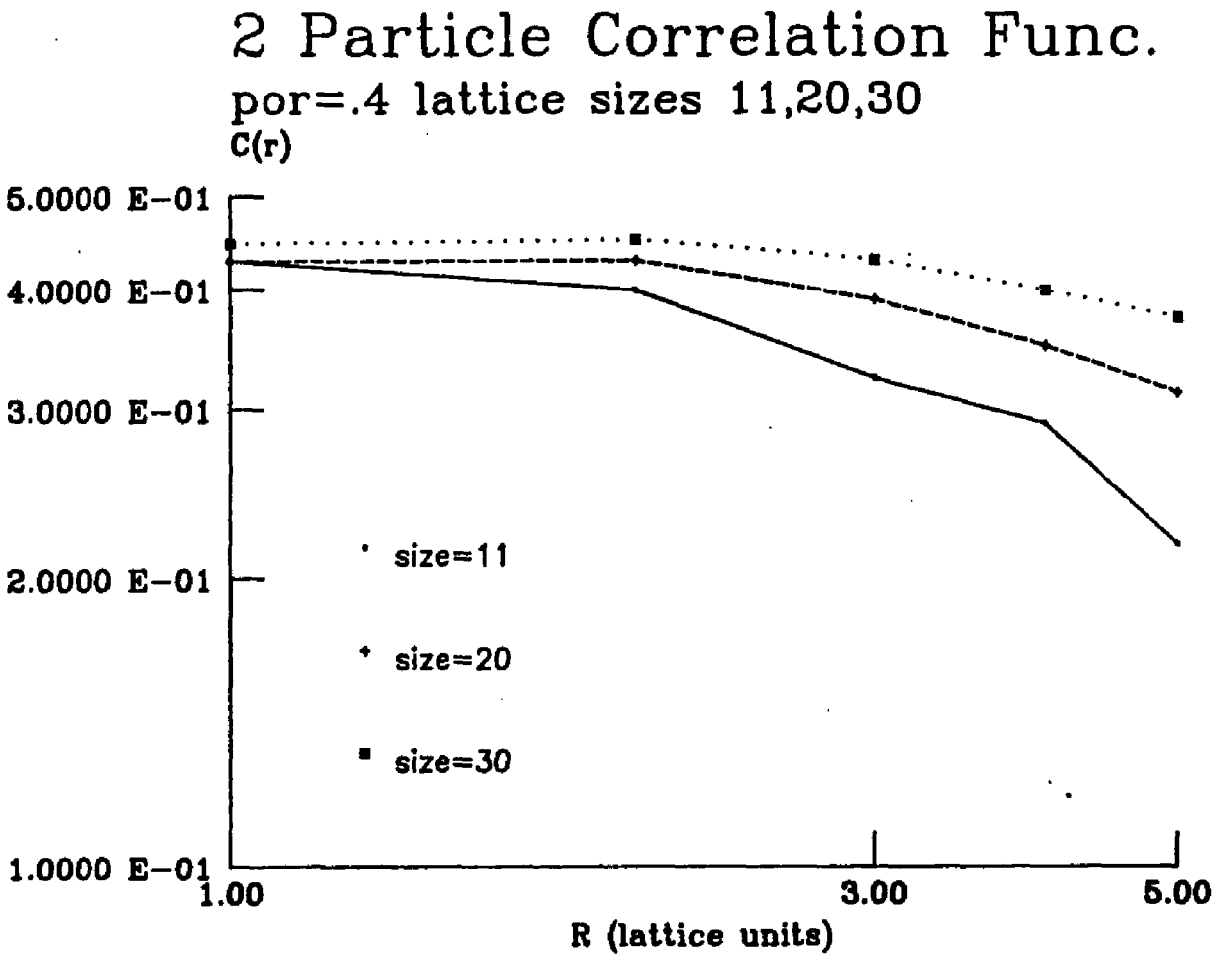


Figure 3.14- Correlation function of our RNM lattice.

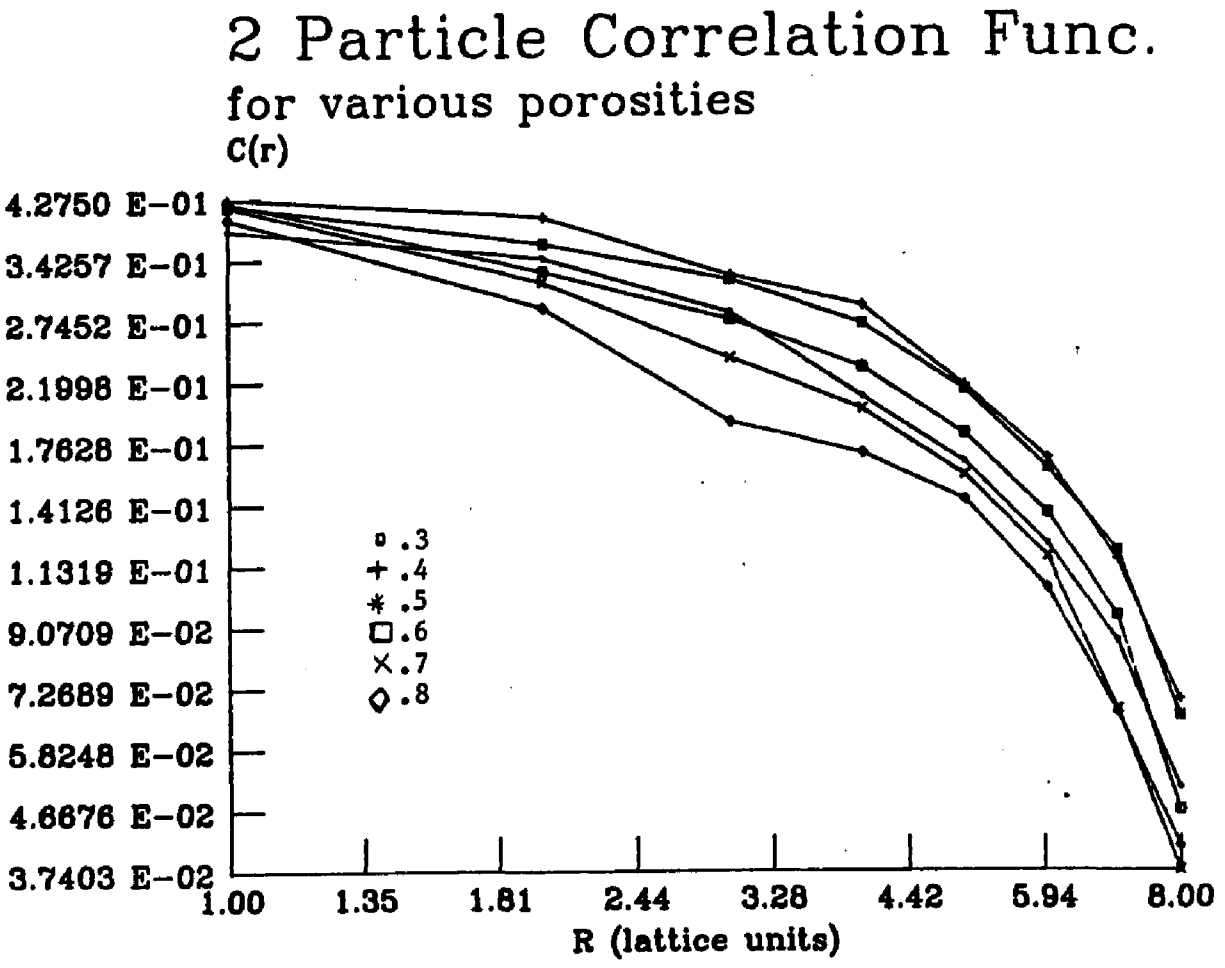


Figure 3.15- All porosities are measured at lattice size 11.

IV. Dissolution-Precipitation Reactions

Many electrode processes involve the formation of deposits on an electrode from solution and the dissolution of material back into solution. Corrosion, deposition, dendritic growth are all complex processes which involve a mass transfer between the solution and the electrode surface and back[58]. The reactions involved are diverse and complex, and we present an idealized computer simulation of the process in order to explore the physical structures formed on the electrode and to present methods for characterizing this surface. The simulation does not concern itself with the specifics of the chemical mechanisms involved, but rather treats the dissolution and precipitation as random processes on a large number of particles which are governed by average reaction and diffusion rates.

The use of computer simulation to study this class of phenomena is not new. Witten and Sander[19] were the first to apply this approach to the study of diffusion-limited aggregation, in which particles were allowed to form in spherical clusters growing from one initial seed site. They measured a correlation exponent (equation 1.14) of .34 on a square lattice. Meakin[55] has extended the original study by considering varying sticking probabilities, and clusters in three and four dimensional space. In two dimensions, Meakin lists the correlation exponent as .29 for a sticking probability of 0.25, and .26 for a sticking probability of 0.1. There has also been numerous experimental verification of these ideas, that indeed aggregation phenomena build fractal structures[58-62]. More recently, Voss and

Tomkiewicz[56,57] have applied the same methodology to the study of dendritic growth in electrodeposition, and were able to correlate their theoretical calculations of dendritic growth rates with experiment. Thus far, however, the simulations have considered only the aggregation of particles onto a substrate. The more general situation in which particles are allowed to dissociate back into solution has not yet been treated. It is precisely dissolution precipitation that has been identified[63] as a reaction mode in porous electrodes (as in the anodic formation of ZnO on Zn electrodes in KOH), and in order to achieve a better understanding of the type of surface structures that result from this system, we have extended the previous models. This simulation is an alternative approach to the simple random metal distribution that we utilized in the previous chapter for constructing our electrode, and it also serves as a more realistic picture of the surface as the electrode's deposit of insulating material builds during discharge.

A. The Simulation

A two dimensional NxN lattice is defined on which two types of particles reside- solid particles are fixed (=solid phase), and dissolved particles are free to move through the lattice (=liquid phase) under Brownian motion. We define a dissolution probability, p_d, and a precipitation probability, p_p, (corresponding to the rate constants of those respective reactions) and a moving probability, p_m (corresponding to the diffusion of dissolved particles). Dissolved particles precipitate only if they are part of a cluster of

minimum size N_m. Each time step in the simulation every particle in the system is examined in a random order, and the following occurs:

1) If the particle is in the bulk of the solid then it remains stationary.
2) If the particle is part of the "solid", and it resides on the surface, it is given a chance to dissolve into solution with probability p_d.
3) If the particle is dissolved, it moves one step in a random direction with probability p_m. If the lattice site that it tries to move to is occupied, the particle remains in place.
4) If the particles is dissolved in solution but adjoins the solid surface, and is part of a cluster bigger than N_m, it is given a chance to precipitate with probability p_p. When the particle precipitates, the adjoining cluster also precipitates.

These checks are made sequentially, so that a particle may be dissolved in step (2) and reattached in step (4). The simulation is begun with rows 1 to N/2 populated with a flat-surfaced solid and with the top half of the lattice empty.

The simulation proceeds for t time steps, and a schematic representation of what the system might look like after time t is shown in Figure 4.1. The particles in solution are represented by the stars, the particles of the bulk are represented by squares, and the particles on the interface are highlighted by the thick continuous line. As illustrated in Figure 4.1, quantitative measures of possible interest for the precipitated particles are the maximum,

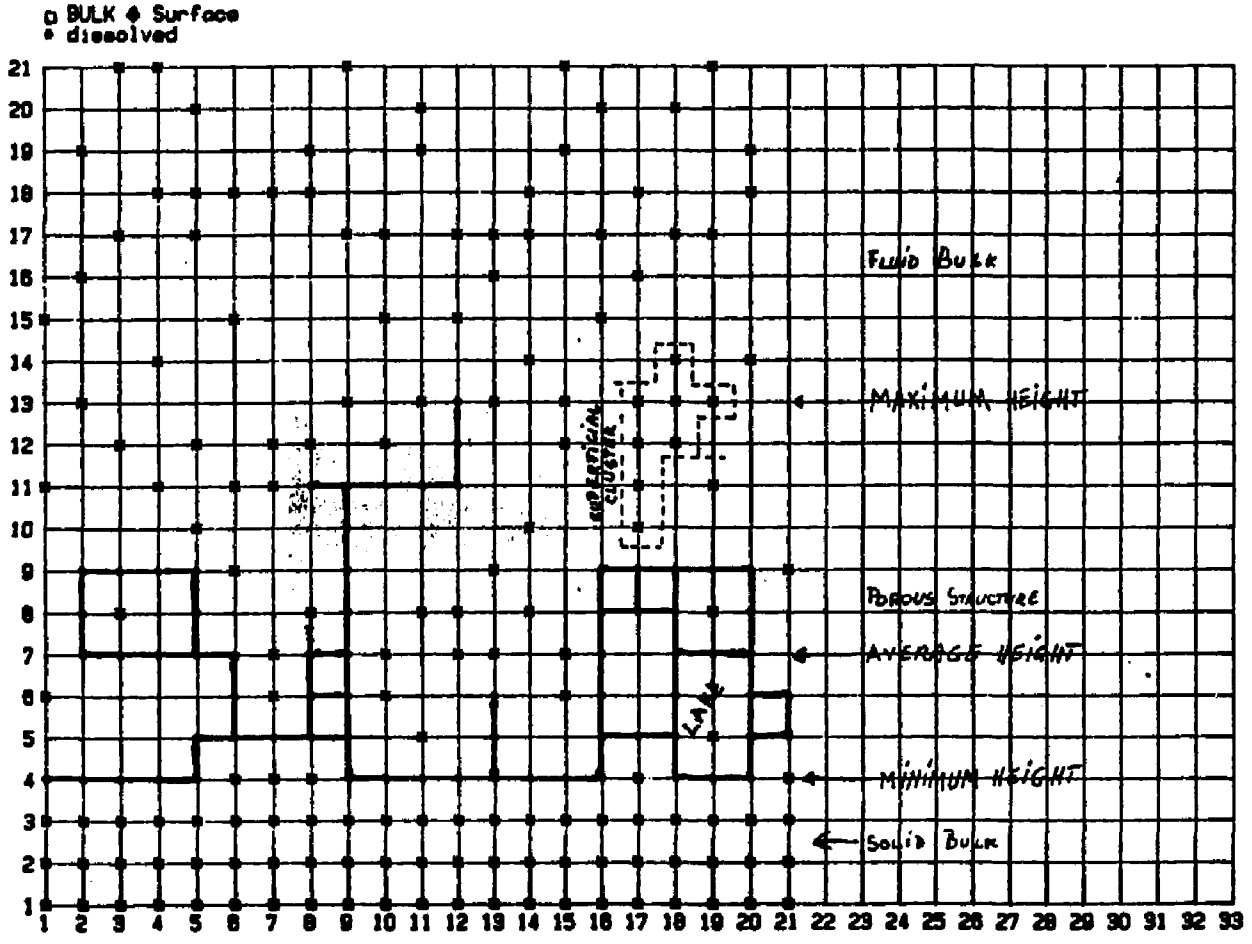


Figure 4.1- Schematic of the DP model at a given time t. Some quantities of possible theoretical interest are labeled. The surface of the solid is indicated by the heavy line.

Figure 4.2- A typical system configuration of the DP model. This result is for a dissolution probability of .5, a precipitation probability of .5 after 1000 time steps.

minimum and average growth heights, as well as the surface roughness and the possible fractal dimension of the interface. The solution may be characterized by the diffusion behavior of the dissolved particles. Figure 4.2 shows an actual simulation result at t=1000 for $p_d=p_p=.5$, and for $p_m=1.0$. It remains for us to attempt to quantify the system and ultimately to correlate the solute behavior and the grown structures with the input parameters of the simulation.

B.Solution characterization-Diffusion Behavior

It is to be expected that if the simulation is realistic, the particles in solution should follow the diffusion equation:

$$\frac{\partial C(x,t)}{\partial t} = D\,\frac{\partial^2 C(x,t)}{\partial x^2} \qquad (4.1)$$

with the following boundary conditions:

$$\begin{aligned} c(x,0) &= c^{*}_{x} \quad , x>0 \\ c(\infty,t) &= c^{*} \\ D\left(\frac{\partial C}{\partial x}\right)_{x=0} &= k_1 C(0,t) - k \end{aligned} \qquad (4.2)$$

where D is the diffusion coefficient, C(x,t) is the time and position dependent concentration, c^{*} is the bulk solution concentration, k is the dissolution constant and k_1 is the precipitation constant. The third boundary condition establishes a fictitious flat plane at x=0 at which all the dissolution-precipitation process is said to occur at all times, t. As the simulation progresses and the surface gets to be rough, this is no longer true. With this approximation, we assume that one can always find a plane at $x=x_0$ where the dissolution precipitation process effectively occurs. The

value of x_0 is a parameter to be determined in fitting our data to these calculations.

The differential equation 4.1 may then be solved to yield:

$$C(x,t) = c^* + \left(\frac{k-k_1c^*}{k_1}\right)\left\{ \operatorname{erfc}\left(\frac{x}{2(Dt)^{\frac{1}{2}}}\right) - \exp\left(\frac{k_1(x+k_1t)}{D}\right) \operatorname{erfc}\left(\frac{x}{2(Dt)^{\frac{1}{2}}} + k_1\sqrt{\frac{t}{D}}\right)\right\} \quad (4.3)$$

Solving for the total number of particles in solution, from Fick's first law,

$$\frac{\partial N}{\partial t} = -D\,\frac{\partial C}{\partial x} \quad (4.4)$$

$$\left.\frac{\partial N}{\partial t}\right|_{x=0} = k - k_1C(0,t) \quad . \quad (4.5)$$

These diffusion equation have been successfully utilized to predict the behavior of experimental dissolution-precipitation systems[64]. In our simulation $c^*=0$, and equation 4.4 may be integrated to give the total number of particles in solution at time t as:

$$N(t) = \frac{kD}{k_1^2}\left\{ \exp\left(\frac{k_1^2t}{D}\right)\operatorname{erfc}\left(k_1\sqrt{\frac{t}{D}}\right) + 2k_1\left(\frac{t}{\pi D}\right)^{\frac{1}{2}} - 1 \right\} \quad (4.6)$$

Figure 4.3 is a plot of concentration as a function of position from the surface into the the solution at t=1000 time steps. The dotted line separates the x axis into two regions. In region I is the

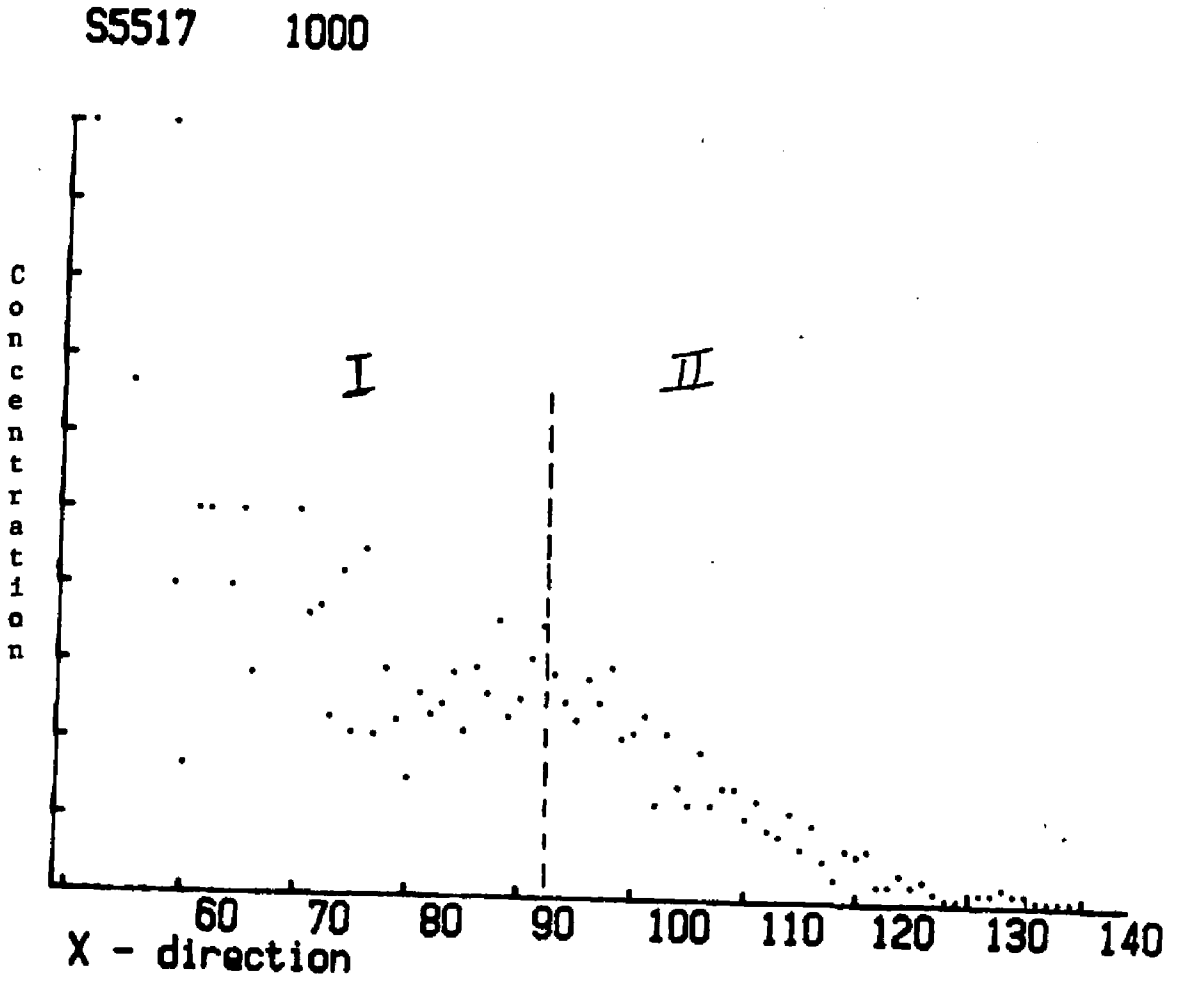


Figure 4.3- The concentration in solution as a function of the distance away from the surface. Region I is still within the heterogeneous mixture of solid and solute. Region II is above the height of the tallest solid peak.

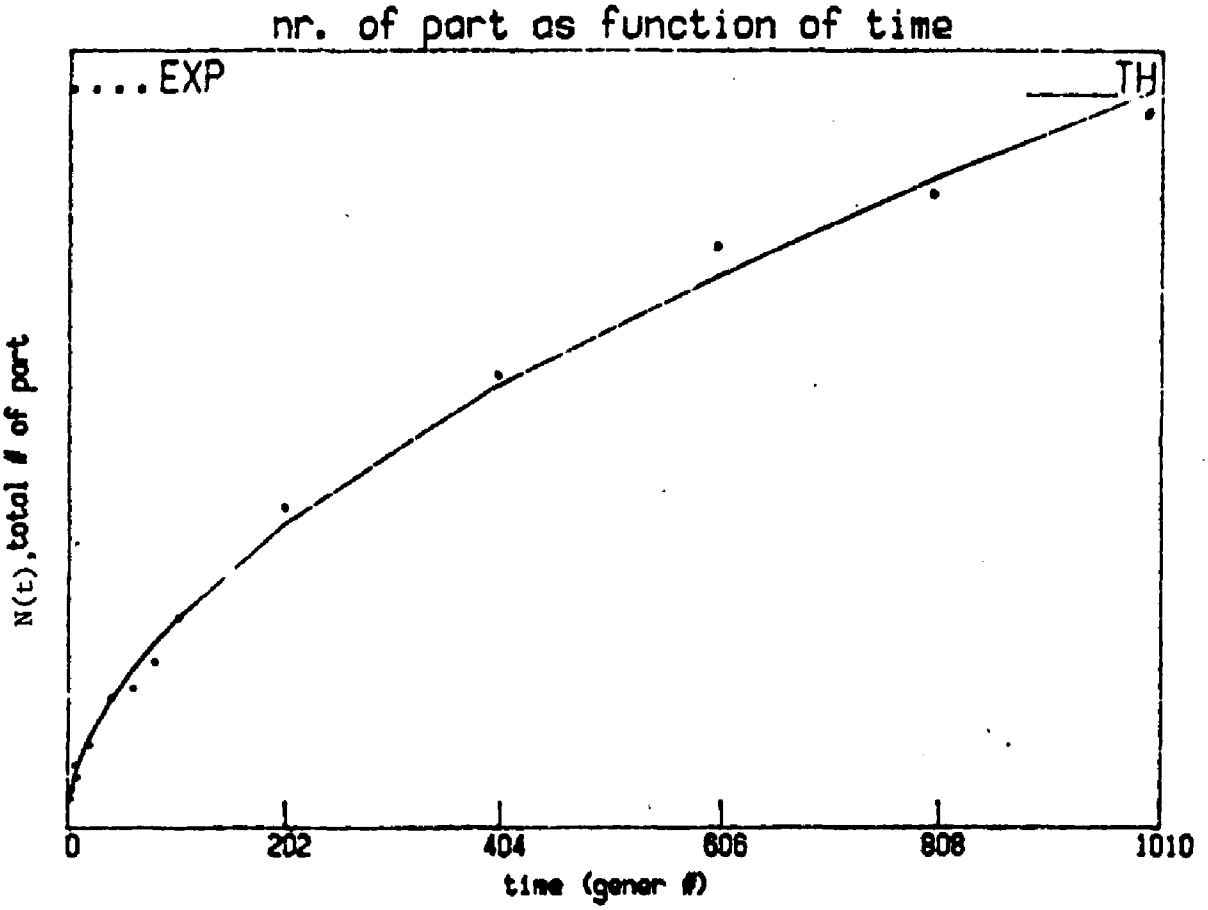


Figure 4.4- The total number of dissolved particles in solution as a function of time. The straight line is a fit of the data to equation 4.6 .

rough interface where there is an inhomogeneous mixture of solid and dissolved particles, which leads to large fluctuations in the dissolved particle concentration. In region II is the bulk of the solution, and we see the particle concentration begin to behave similar to what we would expect from equation 4.3. The total number of particles in solution as a function of time is shown in figure 4.4, and the solid line is a fit from equation 4.6. This is a confirmation of the validity of the simulation, and from the fit parameters we obtain the theoretical k, k_1, and D. Using this methodology, it is possible to correlate the simulation input parameters p_p, p_d, p_m with the more realistic diffusion equation parameters.

C.Solid Characterization - Rough Surface

We can also explore how surface topology is altered by variations in the previously mentioned input parameters. The surface properties can be quantified by calculating the density-density correlation function as in chapter III, or the more straightforward Richardson curve for the surface length under different measuring scales. (The Richardson curve measurement on a discrete lattice is easily implemented on a two dimensional surface as opposed to the 3D Random Network Model of chapter III.) Figure 4.5 is the result of such a calculation for the system of figure 4.2. The calculation is done by considering pair correlations only for the surface particles. First, the computer identifies which particles are on the interface (the solid line in Figure 4.1) and then discards all the other points. The pair correlation for each radius r is computed, and

this is averaged over the four nearest-neighbor directions. Figure 4.5 shows a power law behavior in the relationship between C(r) and r, with a correlation exponent of 0.22 . C(r) is shown at t=600,800,1000,2000 time steps, and the power law behavior is stable for r<9.0 .Finite size effects seem to dominate for r>9.0 . From equation 1.14 the fractal dimension of this two-dimensional system is D=1.78 . This value is considerably higher than that calculated by Witten, Sander and Meakin, but it is worthwhile to note that the geometry of our system is very different from the spherical cluster aggregation that they considered. It is hoped that a further understanding of the significance of the values in this system may be realized by examining the fractal dimension under different dissolution and precipitation probabilities, as well as at longer times.

The surface constructed using the dissolution-precipitation model is an alternative starting point for our Random Network Model. Instead of taking the electrode configuration as constructed using a simple random number generator, we can utilize the dissolution-precipitation lattice as a template for assembling our system of resistors and capacitors. This is a more realistic system than the Cantor Bar model, and is an example of exactly the type of self-similar R-C network that Nyikos and Pajkossky proposed to construct for the examination of fractal interfaces (see section II.E).

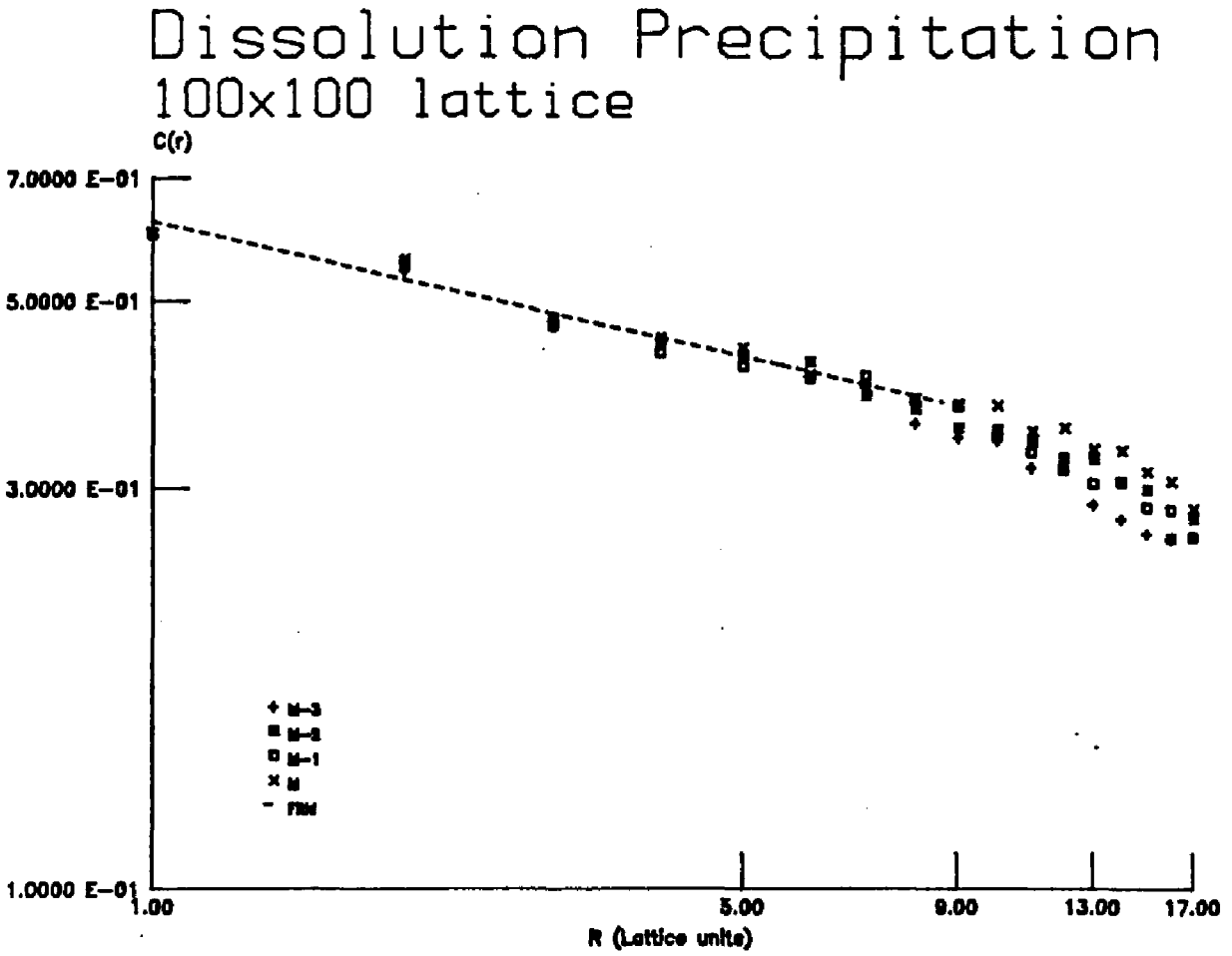


Figure 4.5- Correlation function calculation for the surface of the DP model for the system shown in Figure 4.2 .

V. Summary and Conclusions

Recent developments in the field of composite media and disordered systems have availed us with powerful techniques for characterizing stochastic multi-phase systems. Analytic methods such as Effective Medium Theory offer a simplicity of concept and execution, but are more limited in their ability to model some of the salient features of real life systems. Computer simulations offer more flexibility in this regard, and provide a powerful means of exploring the behavior of systems with detailed constraints and/or complicated geometries. The drawback to numerical simulations is that one is limited by a finite accuracy that may not be sufficient in some cases (the "round off error"), and by the fact that as fast as today's computer's are, sophisticated models and large systems require huge amounts of computer time.

The simulations presented in this work are physical simulations involving the construction of a model structure to mimic the topological features of a system. The process is really a two step one:

Firstly, the system must be constructed in the computer using a realistic set of rules and input parameters. We have presented two examples of this - the random network model (RNM) and the dissoloution-precipitation (DP) model. These representations are very different from each other and complementary; the RNM model is an attempt to depict a typical porous metal structure in an electrolyte with the major topological contribution arising from the

metal's porosity, while the DP model is an attempt to depict the topology of an electrode surface as it is roughened by chemical deposition and dendritic growth. An "ideal" examination of porous electrodes might utilize both of these configurations simultaneously. To be sure, both of these models introduce simplifying assumptions, and additional variations can be considered to explore how different pictures of the geometry and different constraints would affect the results.

Secondly, once the system has been constructed, the computer may be utilized to analyze and perhaps quantify the structure of the model. In addition, other properties of the system may be calculated and an attempt can be made to correlate these with the system's topology. In this work we examined the impedance behavior of the system and it's dependance on morphology. The electrical impedance is very well suited for this type of exploration, due to the fact that impedance in addition to being dependent on the internal structure of the material is highly dependent on it's macroscopic shapes and lengths. The frequency dependance of the impedance is especially useful in this respect as well, for it allows us a consistent way of weighting various surface and bulk effects with respect to each other and seeing the result. It is also a measurement that is well open to experimental exploration and verification.

In addition to providing sample implementations of the above methodology, we have provided some new insight into the porous electrode system. We have demonstrated the utility of the low frequency

capacitance values in measuring the apparent area of the rough electrode, and provided a caveat concerning "what is called low frequency?"- showing that it is dependent on the relative magnitudes of the interfacial impedance and the electrolyte resistivity in each electrode system. In this regard, for all it's simplicity, the DeLevie model is an excellent representation of the current flow in a porous structure. We have found that in an ideally polarized rough electrode there is a fractional power law relationship between Z and ω with an identical scaling exponent for the real and the imaginary components (the CPA behavior), but we have shown that it is not necessarily related to fractal structures, as suggested by Liu and others. All in all, the RNM seems to be a useful addition to an electrochemist's toolbox. The DP model of Chapter IV is an attempt to explore the morphology of the electrode which results from the electrochemical reactions on the interface. We have shown that the model is valid by verifying the predicted diffusion behavior of the dissolved particles. The DP model also produces a fractal surface which is fertile ground for additional research.

Future Work

We have treated a single geometry in the RNM, and an extension of that to alternate geometries is warranted. One example is allowing the fluid to flow through the cracks between adjoining metal sites. The DP model is as yet largely unexplored - we have not yet examined how the fractal geometry of the system changes under varying sticking and dissolution probabilities. An important further consideration is to attempt to construct the system in different geometries

(for example on a triangular lattice) to verify that the results are independent of the numerical implementation of the model. The merging of the RNM and the DP model is an important further step to be explored. We can thus construct a realistic picture of the electrode under operating conditions, and solve for the impedance of such a structure. New techniques are being utilized for the calculation of network impedances utilizing a transfer matrix algorithm[68,69]. The advantage of this method is greater accuracy as well as greater efficiency in utilizing computer storage. This method has already been applied to 2-D complex impedances[70], and extension to three dimensions should be trivial. Any further work in network simulations would benefit by utilizing these transfer matrix techniques instead of the simultaneous equation solutions.

APPENDIX

PROGRAM LISTINGS

```
C                                                                       POR00010
C   ***************************************************************     POR00020
C   *                                                             *     POR00030
C   *                PROGRAM POROUS_METAL                         *     POR00040
C   *    (C) COPYRIGHT 1985    BY  MICHAEL KRAMER                 *     POR00050
C   *    VS-FORTRAN VERSION ---------- MAIN PROG ONLY            *     POR00060
C   *        STORED ON FILE : VSPORMET REV 09/05/84              *     POR00070
C   *        USES SL-MATH      ---         IBM SR'S              *     POR00080
C   *       TO DIAGONALIZE A SPARSE SYMMETRIC MATRIX             *     POR00090
C   *  THIS VERSION USES 1 SEED FOR WHOLE FREQ RANGE (MSSO&MNSO)  *     POR00100
C   *    ALSO ADAPTED TO EVENLY DISTRIBUTE CURRENT IN BOT PLANE   *     POR00110
C   *              CALCULATES  CURRENT OUT FOR CHECK              *     POR00120
C   *   NORMALIZES COMPONENTS TO 10 MICRON PORE SIZE             *     POR00130
C   *      EXCEPT FOR DIFFERENT NORM FOR RELECTROLYTE (NEW)      *     POR00140
C   *       COMPLEX EQU:    G*V=I   (G=GR+J*GI)  (V=VR+J*VI)     *     POR00150
C   *       REAL EQU:  GR*VR-GI*VI=IR                           *     POR00160
C   *       IMAG EQU:  GI*VR+GR*VI=II                           *     POR00170
C   *                                                           *     POR00180
C   *     MATRIX IS SYMMETRIZED BY MULT IMAG EQU BY -1 :        *     POR00190
C   *                                                           *     POR00200
C   *      (  GR       -GI )  (  VR  )       (  IR  )           *     POR00210
C   *     (                 ) (       )  =  (        )          *     POR00220
C   *      (  -GI      -GR )  (  VI  )       (  -II )           *     POR00230
C   *                                                           *     POR00240
C   *                                                           *     POR00250
C   ***************************************************************     POR00260
*     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%     POR00270
*     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    POR00280
*     %  I/O UNIT NUMBERS:                                          %    POR00290
*     %  5 - INPUT - DISK - INPUT PARAMETERS                        %    POR00300
*     %  6 - OUTPUT - PRT - REGULAR DOCUMENTATION                   %    POR00310
*     %  7 - OUTPUT - TERM - IMMEDIATE DOCUMENTATION/CONSOLE LOG    %    POR00320
*     %  8 - OUTPUT - DISK - SHORT DATABASE SUMMARY OF CALCULATION  %    POR00330
*     %  9 - OUTPUT - PUN  - LONG SUMMARY OF CALCULATION            %    POR00340
*     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    POR00350
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&    POR00360
#TOGGLE IDEALLY POLARIZED ELECTRODE USING THE FLAG IPOLAR IN LINE 337&    POR00370
*&    DATA FILE STRUCTURE (INPUT PARAMETERS) (UNIT 5)                 &    POR00380
*&                                                                    &    POR00390
*&    IM,JM,KM,POR,POREND,PORINC  - LATTICE SIZE & POROSITY           &    POR00400
*&    MODEL                       - WHICH LATTICE GEOMETRY IS USED    &    POR00410
*&    KMSTOP                      - TRUNCATE LATTICE IN K DIRECTION?  &    POR00420
*&    RELIN,RMETIN,RSIN,RPIN,CPIN - INPUT PARAMETERS FOR EQUIV.       &    POR00430
*&                                  CIRCUIT COMPONENTS                &    POR00440
*&    CUR(1),CUR(2)               - INPUT CURRENT (COMPLEX)           &    POR00450
*&    FRBEG,FREND,IPPD            - FREQ RANGE AND POINTS/DECADE      &    POR00460
*&    IRPT                        - HOW MANY REPEATS WITH RNDM SEEDS  &    POR00470
*&    DSEED                       - RANDOM NUMBER SEED(S)             &    POR00480
*&      .                           (ENOUGH FOR THE IRPT PARAMETER)   &    POR00490
*&      .                                                             &    POR00500
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&    POR00510
***********************************************************************    POR00520
```

```
*            LIST OF CALLED SUBROUTINES:  (-- DENOTES A S.R. NOT LISTED POR00530
*    --CALL ERRSET (208,0,5,1)                    HERE)                 POR00540
*    --CALL DATTIM (EVCPU,ETIME,ETCPU)                                  POR00550
*      CALL NODES(A,LBLA,POR,DSEED,MODEL)                               POR00560
*      CALL DISPLA(A,ISHOW,MODEL)                                       POR00570
*      CALL CORLAT (A,PORVOL,VOLNOR,1)                                  POR00580
*      CALL CHOP (A)                                                    POR00590
*      CALL CONDUC(A,G,JA,IA,B,N,NTCALC,MODEL)                          POR00600
*    --CALL MOOO(N,NTCALC,IBUF1,IA,JA,IPM,IER,IB,IU,JU)                 POR00610
*    --CALL MSSO(N,IBUF1,IBUF2,IA,JA,IPM,IU,JU,IK,IER,IUP,IAT,JAT,IAP,  POR00620
*    --CALL MNSO (N,ISW,IA,JA,G,IAT,JAT,IAP,IPM,IU,JU,U,DI,IER,IUP,X)   POR00630
*    --CALL MBSO(N,ISL,IPM,IU,JU,U,DI,B,G,X)                            POR00640
*      CALL CURCAL (A,G,MODEL)                                          POR00650
*      CALL CURPRF (A,LBLA,G)                                           POR00660
*      CALL LATSET(A,POR,MODEL)                                         POR00670
*      CALL PLANE (A,1,5)                                               POR00680
*      CALL RNDM(A,MB,DSEED,KMIN,KMAX,1)                                POR00690
*      CALL CONECT(A,K,5,1,IFND,KFND)                                   POR00700
*      CALL RANPT(A,DSEED,KMIN,KMAX,IT,JT,KT)                           POR00710
*      CALL NEARN(A,IT,JT,KT,0,1,IFND,KFND)                             POR00720
*      CALL FILABL(A,LBLA)                                              POR00730
*      CALL GGUBS(DSEED,3,R)                                            POR00740
*      CALL DISPLA (A,ISHOW,MODEL)                                      POR00750
*      CALL NOD146(A,1,IAT,NEIGH,IGND,Z1,Z2)                            POR00760
*      CALL NOD235(A,2,IAT,NEIGH,IGND,Z1,Z2)                            POR00770
*********************************************************************** POR00780
C                                                                        POR00790
C     SIZE OF A IS (IM,JM,KM)                                            POR00800
      PARAMETER (MAXSIZ = 11,  INODES = MAXSIZ**3)                       POR00810
      PARAMETER (ISMAX = 2*INODES,  ISMIN = ISMAX-2)                     POR00820
      PARAMETER (NT = 10 * (4*INODES - 3*MAXSIZ*MAXSIZ + 3))            POR00830
      PARAMETER (NT1 = 15*NT, IBOT = 46*NT, NRD = NT - ISMIN)            POR00840
      PARAMETER (IBO1 = 2*NRD)                                           POR00850
                                                                         POR00860
      INTEGER*4 A(MAXSIZ,MAXSIZ,MAXSIZ),LBLA(MAXSIZ,MAXSIZ,MAXSIZ)      POR00870
      REAL *8 DSEED,DHOLD,X                                              POR00880
      LOGICAL*1 IPOLAR                                                   POR00890
C                                                                        POR00900
C                                                                        POR00910
      COMMON/DIMS/IDIM1,IDIM2,IDIM3,IDIM4                                POR00920
      COMMON/NUMB/PI                                                     POR00930
      COMMON/LATICE/ IM,JM,KM,KMSTOP                                     POR00940
      COMMON/ELEC/Z(6,2),CUR(2),NCUR,NGND,REALI,RMAGI                    POR00950
      COMMON/GTEMP/ NG1,NG2,NG3,NG4,NG5,ICOL(6),DUMMY(12)                POR00960
      DIMENSION IDIM(7),CURNOR(2)                                        POR00970
C                                                                        POR00980
C     **************  VARIABLE  SIZE PROBLEM ********************        POR00990
C                                                                        POR01000
C   THE BASIC SYSTEM DIMENSIONS ARE IM,JM,KM                             POR01010
C      ISM = IM*JM*KM-1                                                  POR01020
```

```
C     N= # OF SIMUL EQU (N.LE.2*S)                                          PORO1030
C                                                                           PORO1040
C    1)   ISMIN=N=2*(IM*JM*KM-1)                                            PORO1050
C                                                                           PORO1060
C    ------------------                                                     PORO1070
C    2) DIMENSION OF ISMAX IS ISMIN+2                                       PORO1080
       DIMENSION IA(ISMAX),IB(ISMAX),IU(ISMAX),IPMI(ISMAX),                 PORO1090
      1IPM(ISMAX),DI(ISMAX),IX(ISMAX),B(ISMAX),X(ISMAX),                    PORO1100
      1IUP(ISMAX),IAT(ISMAX)                                                PORO1110
C    ------------------                                                     PORO1120
C                                                                           PORO1130
C    3) NT IS # OF PACKED ELEMENTS IN MATRIX G--THE SPARSE                  PORO1140
C                        STORAGE COEFF. MATRIX                              PORO1150
C               NT.LE.4*(4*S-3*IM*JM+3)                                     PORO1160
C         NT=4* (4*S-3*IM*JM+3)                                             PORO1170
       DIMENSION JA(NT),G(NT)                                               PORO1180
C    -------------------                                                    PORO1190
C                                                                           PORO1200
C    4)    NT1 IS BETWEEN 2*NT+N AND 3*NT AND HERE WE SET IT TO 14*NT       PORO1210
C       NT1=14*IDIM(3)                                                      PORO1220
        DIMENSION U(NT1)                                                    PORO1230
C                                                                           PORO1240
C    ------------------------------------                                   PORO1250
C                                                                           PORO1260
C     5) IBOT SHOULD BE BETWEEN 2*NT+N AND 3*NT BUT HERE WE USE 44*NT       PORO1270
C      IBOT=44*IDIM(3)                                                      PORO1280
       DIMENSION JU(IBOT)                                                   PORO1290
C                                                                           PORO1300
C   ---------------------------------------------                           PORO1310
C                                                                           PORO1320
C      6) NRD = NT - N                                                      PORO1330
C                                                                           PORO1340
       DIMENSION JAT(NRD),IAP(NRD)                                          PORO1350
C                                                                           PORO1360
C                                                                           PORO1370
C   ------------------------------------------                              PORO1380
C                                                                           PORO1390
C      7) IBO1 SHOULD BE 2*NRD                                              PORO1400
C                                                                           PORO1410
       DIMENSION JUA (IBO1)                                                 PORO1420
C                                                                           PORO1430
C   -----------------------------------------------                         PORO1440
C                                                                           PORO1450
        DIMENSION IER(7),ERIND(4)                                           PORO1460
        DATA ERIND/'MOOO','MSSO','MNSO','MBSO'/                             PORO1470
C  ****************        ******************                               PORO1480
       IDIM1=ISMIN                                                          PORO1490
       IDIM2=ISMAX                                                          PORO1500
       IDIM3=NT                                                             PORO1510
```

```
      IDIM4=NT1                                                          PORO1520
      IDIM5=IBOT                                                         PORO1530
      IDIM6 = NRD                                                        PORO1540
      IDIM7 = IBO1                                                       PORO1550
C                                                                        PORO1560
C     TOTAL STORAGE REQUIREMENT IS :                                     PORO1570
C    12*N + 2*NT + NT1 + IBOT + 2*NRD + IBO1                             PORO1580
      NSTOR = 12*IDIM2 + 2*IDIM3 + IDIM4 + IDIM5 + 2*IDIM6 + IDIM7       PORO1590
      NBYTES = 4 * NSTOR                                                 PORO1600
      WRITE (6,1) MAXSIZ,ISMIN,ISMAX,NT,NT1,IBOT,NRD,IBO1,NSTOR,NBYTES   PORO1610
    1 FORMAT (' STORAGE ALLOCATIONS- IDIM1-->IDIM7:  MAXSIZE= ',I2,//,   PORO1620
     C' N= ',I10,' ISMAX= ',I10,' NT= ',I10,' NT1= ',I10,/,              PORO1630
     C' IBOT= ',I10,' NRD= ',I10,' IBO1= ',I10,                          PORO1640
     C//,' MIN TOTAL STORAGE REQUIRED= ',I12,'=',I12,'  BYTES')          PORO1650
C ********************************************************************** PORO1660
      PI=3.14159                                                         PORO1670
C  ERROR SETTING ROUTINE-- UNDERFLOW ERROR                               PORO1680
C                                                                        PORO1690
      CALL ERRSET (208,0,5,1)                                            PORO1700
*    INITIALIZE  -- IDUMP=1 MEANS GENERATE DETAILED NUMERICAL SUMMARY    PORO1710
*   ONLY ON FREQ'S WITH MULTIPLES OF 10.                                 PORO1720
      IDUMP = 1                                                          PORO1730
C                                                                        PORO1740
C   INITIALIZE FLAG TO INDICATE NUMERICAL DIFFICULTIES                   PORO1750
      IBOMB = 0                                                          PORO1760
C                                                                        PORO1770
********************************                                       PORO1780
* PROGRAM MAY BE TERMINATED BY THE COMMAND '#CP STORE 464 1'            PORO1790
* INITIALIZE NUSERFWD IN NUCON (LOC X'464 ) TO 0.                       PORO1800
      IZERO = ITYBIT (-1)                                                PORO1810
********************************                                       PORO1820
C                                                                        PORO1830
C    SYSTEM TIME READING ROUTINE -- FORTRAN UTILITIES TXTLIB             PORO1840
C                                                                        PORO1850
C    VARIABLES USED:                                                     PORO1860
C   ETCPU = ELAPSED TOTAL CPU                                            PORO1870
C   ETIME = ELAPSED CONNECT TIME                                         PORO1880
C   STIME = STARTING TIME THAT PROBLEM WAS RUN  (CPU)                    PORO1890
C   TIMNOW = CURRENT TIME (CPU)                                          PORO1900
C   TIMLAS = LAST VALUE OF CURRENT TIME   (CPU)                          PORO1910
C   RUNTIM = TIMNOW - TIMLAS                                             PORO1920
C                                                                        PORO1930
C                                                                        PORO1940
C     CONNOW = CURRENT CONNECT TIME                                      PORO1950
C     CONLAS = LAST CONNECT TIME                                         PORO1960
C     CONTIM = CONNOW - CONLAST                                          PORO1970
C                                                                        PORO1980
C      ALL TIMES ARE STORED IN MINUTES.                                  PORO1990
C                                                                        PORO2000
C                                                                        PORO2010
```

```
  5   CALL DATTIM (EVCPU,ETIME,ETCPU)                                    POR02020
C   WE'RE ONLY INTERESTED IN TOTAL CPU TIME                              POR02030
C   INITIALIZE TIME COUNTERS AND CONVERT TO MINUTES                      POR02040
      STIME = ETCPU/60.0                                                 POR02050
      CONLAS = ETIME/60.0                                                POR02060
      TIMLAS = STIME                                                     POR02070
C                                                                        POR02080
C   ***********************************************                      POR02090
C                                                                        POR02100
      NRPT=0                                                             POR02110
C FIRST SET UP LATTICE PARAMETERS                                        POR02120
      READ (5,10,END=900)IM,JM,KM,POR,POREND,PORINC                      POR02130
  10  FORMAT (3I2,3F3.2)                                                 POR02140
      READ (5,19) MODEL                                                  POR02150
  19  FORMAT (I3)                                                        POR02160
      IF (MODEL.LT.0) THEN                                               POR02170
         ISOLVE = 0                                                      POR02180
         MODEL = IABS(MODEL)                                             POR02190
      ELSE                                                               POR02200
      ISOLVE = 1                                                         POR02210
      ENDIF                                                              POR02220
      IF (IM.GT.MAXSIZ) THEN                                             POR02230
           WRITE (7,11) IM, MAXSIZ                                       POR02240
           GOTO  910                                                     POR02250
           ENDIF                                                         POR02260
  11  FORMAT (' ERROR-- IM= ',I3,' IS GT MAXSIZ= ',I3)                   POR02270
      IF (PORINC.EQ.0) PORINC = .1                                       POR02280
C--------- EXTRA PARAMETER DEFINES HOW HIGH UP IN THE                    POR02290
C          LATTICE THE ELECTROLYTE IS EXPECTED TO FLOW                   POR02300
C                                                                        POR02310
      READ (5,20) KMSTOP                                                 POR02320
  20  FORMAT (I2)                                                        POR02330
      IF (KMSTOP.GT.KM) KMSTOP = KM                                      POR02340
C                                                                        POR02350
      IS = IM*JM*KM                                                      POR02360
      ISM = IS - 1                                                       POR02370
C NOW READ IN ELECTRICAL PARAMETERS                                      POR02380
      READ (5,50) RELIN,RMETIN,RSIN,RPIN,CPIN                           POR02390
  50  FORMAT (5E9.1)                                                     POR02400
C  *****NORMALIZE THE ELEMENTS TO THE LATTICE SIZE                       POR02410
C                                                                        POR02420
C    PORSIZE OF 10 MICRON= 1E-3 CM; WE USE UNIT IN CM                    POR02430
C                                                                        POR02440
      PORSIZ = 1.0E-3                                                    POR02450
      AREA = (IM-1) * (JM-1)                                             POR02460
C        SYSIZE IS THE SIZE OF SYSTEM IN CM**2=PORSIZ**2 * AREA          POR02470
C              1 / SYSIZE IS THE SCALING FACTOR TO SCALE                 POR02480
C              THE SYSTEM  UP TO THE 1 CM**2 STANDARD VAL.               POR02490
C                                                                        POR02500
```

```
      SYSIZE = PORSIZ**2 * AREA                                          POR02510
      SCALE = 1.0/SYSIZE                                                 POR02520
C   CAPACITORS ARE TO BE MULT BY SCALE; RESISTORS DIVIDED                POR02530
C                                                                        POR02540
      WRITE (6,60) PORSIZ,AREA,SYSIZE,SCALE                              POR02550
   60 FORMAT (' PORE SIZE-IN CM-FOR THE UNIT PORE ',1P,E9.3,             POR02560
     1' LATTICE AREA = (IM-1)*(JM-1) = ',0PF6.0,//,                      POR02570
     1' SIZE OF WHOLE SYSTEM= ',1P,E12.3,/,                              POR02580
     1' AREA MUST BE SCALED BY ',0PF10.0,' TO APROXIMATE A LARGER',      POR02590
     1' 1 CM**2 SIZE SYSTEM')                                            POR02600
*  NOW NORMALIZE THE CIRCUIT ELEMENTS FROM 1 CM**2 TO PORSIZE**2         POR02610
*                                                                       POR02620
      ENORM = 1.0 / (PORSIZ**2)                                          POR02630
*                                                                       POR02640
*   THE ELECTROLYTE & METAL RESISTOR IS NORMALIZED BY L/A = 1/L ONLY     POR02650
*                                                                       POR02660
      RNORM = 1.0 / PORSIZ                                               POR02670
                                                                         POR02680
                                                                         POR02690
      REL = RELIN * RNORM                                                POR02700
      RMET = RMETIN * RNORM                                              POR02710
      RS = RSIN * ENORM                                                  POR02720
      RP = RPIN * ENORM                                                  POR02730
      CP = CPIN / ENORM                                                  POR02740
      READ (5,100) CUR(1),CUR(2)                                         POR02750
  100 FORMAT (2E7.0)                                                     POR02760
      READ (5,110) FRBEG,FREND,IPPD                                      POR02770
  110 FORMAT (2E7.0,I2)                                                  POR02780
C    HOW MANY TIMES DO U REPEAT WITH A DIFFERENT RNDM # ?                POR02790
C    IRPT=0 MEANS THAT ONLY ONE RNDM SEED SHOULD BE USED                 POR02800
*    IRPT LT 0 MEANS JUST CONSTRUCT LATTICE -- DONT SET UP MATRICES      POR02810
*    AND DONT SOLVE EQUATIONS                                            POR02820
      READ (5,20) IRPT                                                   POR02830
* THERE ARE TWO NESTED LOOPS HERE: 1-CHANGE POR  2- CHANGE FREQ         POR02840
***==========>>>>>  START LOOP FOR DATA  <<<<<==========***             POR02850
  120 FREQ=FRBEG                                                         POR02860
      IF (FREQ.LT.0.0) FREQ=1.0                                          POR02870
      NCUR = 1                                                           POR02880
C                                                                        POR02890
C SET FLAG TO INDICATE DISTRIBUTION OF CURRENT EQUALLY                   POR02900
C THROUGH BOTTOM (ELECTROLYTE) PLANE                                     POR02910
C                                                                        POR02920
      IDIST = 1                                                          POR02930
C                                                                        POR02940
      NGND = IS                                                          POR02950
  122 FRCNT = 0.0                                                        POR02960
C                                                                        POR02970
C                                                                        POR02980
C  FRCNT IS FREQ LOOP COUNTER- OUTERMOST LOOP                            POR02990
C                                                                        POR03000
```

```
C  NOW SET UP AND DISPLAY PHYSICAL SYSTEM                                  PORO3010
C                                                                          PORO3020
      WRITE(6,124) IM,JM,KM,POR                                            PORO3030
      WRITE(7,124) IM,JM,KM,POR                                            PORO3040
  124 FORMAT ('OLATTICE SIZE = ',3I4,'    POROSITY = ',F3.2)               PORO3050
      READ (5,125) DSEED                                                   PORO3060
  125 FORMAT(D13.0)                                                        PORO3070
C  INITIALIZE 'NEW LATTICE' FLAG                                           PORO3080
C                                                                          PORO3090
      IWENT = 0                                                            PORO3100
C                                                                          PORO3110
C                                                                          PORO3120
C    STORE INITIAL DSEED VALUE FOR THIS LATTICE                            PORO3130
      DHOLD=DSEED                                                          PORO3140
C  --------------- WRITE SUMMARY TO DATABASE FILE ----------------         PORO3150
      WRITE (8,141) IM,JM,KM,MODEL                                         PORO3160
  141 FORMAT (' 01 ',3I2,I4 )                                              PORO3170
      WRITE (8,142) POR,KMSTOP,NRPT                                        PORO3180
  142 FORMAT (' 02 ',F3.2,2I3)                                             PORO3190
      WRITE (8,143) RELIN,RMETIN,RSIN,RPIN,CPIN                            PORO3200
  143 FORMAT (' 03',1P,5E9.1)                                              PORO3210
      WRITE (8,144) FRBEG,FREND,IPPD                                       PORO3220
  144 FORMAT (' 04',1P,2E7.0,I3)                                           PORO3230
      WRITE (8,121) DHOLD                                                  PORO3240
  121 FORMAT (' 05',F13.0)                                                 PORO3250
C  --------------- END OF PRELIMINARY DATABASE ENTRY ---------------      PORO3260
      CALL NODES(A,LBLA,POR,DSEED,MODEL)                                   PORO3270
      IF (IDUMP.EQ.1) THEN                                                 PORO3280
           WRITE (9,133) (((A(II,JJ,KK),II=1,IM),JJ=1,JM),KK=1,KM)         PORO3290
  133      FORMAT (' ',3312)                                               PORO3300
      ENDIF                                                                PORO3310
      ISHOW = 1                                                            PORO3320
      CALL DISPLA(A,ISHOW,MODEL)                                           PORO3330
*   LOOK FOR LATTICE CORRELATIONS TO TRY AND FIGURE OUT GEOMETRY           PORO3340
      WRITE (6,*) 'METAL CORRELATIONS:'                                    PORO3350
      WRITE (7,*) 'METAL CORRELATIONS:'                                    PORO3360
      CALL CORLAT (A,PORVOL,VOLNOR,1)                                      PORO3370
      WRITE (6,*) 'ELECTROLYTE CORRELATIONS:'                              PORO3380
      WRITE (7,*) 'ELECTROLYTE CORRELATIONS:'                              PORO3390
      CALL CORLAT (A,PORVOL,VOLNOR,2)                                      PORO3400
C                                                                          PORO3410
C   ------- STORE CURRENT VALUE OF KM                                      PORO3420
      KHOLD = KM                                                           PORO3430
      IF (MODEL.EQ.3) GOTO 130                                             PORO3440
      IF (KMSTOP.EQ.KM) GOTO 130                                           PORO3450
      CALL CHOP (A)                                                        PORO3460
C                                                                          PORO3470
C                                                                          PORO3480
C                                                                          PORO3490
C                                                                          PORO3500
*******==========>>>>>  NEXT FREQUENCY   <<<<<==========*******           PORO3510
```

```
  130 IF (IDUMP.EQ.1) THEN                                              PORO3520
*   ONLY LIST EVERY MULTIPLE OF 10                                      PORO3530
      IF (MOD(INT(FRCNT),IPPD).EQ.0)    WRITE (9,131) FREQ              PORO3540
  131    FORMAT (' FREQ',1P,E10.3)                                      PORO3550
      ENDIF                                                             PORO3560
      WRITE (6,135)                                                     PORO3570
  135 FORMAT (' ',///,'##################################################')PORO3580
      WRITE (6,505) FREQ                                                PORO3590
C NOW SET UP IMPEDANCES OF DIFFERENT CIRCUIT ELEMENTS AT THIS FREQ       PORO3600
      OMEGA = 2 * PI * FREQ                                             PORO3610
      DENOM = 1 + (OMEGA * CP * RP)**2                                  PORO3620
*HERE WE HAVE THE OPTION OF AN IDEALLY POLARIZED ELECTRODE               PORO3630
* IF IPOLAR = TRUE THEN THE ELECTRODE IS IDEALLY POLARIZED               PORO3640
      IPOLAR = .TRUE.                                                   PORO3650
      ZREAL = RS + RP/DENOM                                             PORO3660
      ZIMAG =    -OMEGA * CP * RP * RP /DENOM                           PORO3670
*.....                                                                  PORO3680
      IF (IPOLAR) THEN                                                  PORO3690
           ZREAL = RS                                                   PORO3700
          ZIMAG = - 1.0 / (OMEGA*CP)                                    PORO3710
      ENDIF                                                             PORO3720
**********************************************************************  PORO3730
C                                                                       PORO3740
C    Z(1,N) IS METAL RESISTANCE;  Z(2,N) IS ELECTROLYTE RESIST          PORO3750
C    Z(3,N) IS METAL+S.C. RESIST; Z(3,N) IS ELEC +S.C. RES              PORO3760
C      **** N  REFERS TO: 1-REAL   2-IMAG  ********                     PORO3770
C                                                                       PORO3780
      Z(1,1)=RMET                                                       PORO3790
      Z(1,2)=0.0                                                        PORO3800
      Z(2,1)=REL                                                        PORO3810
      Z(2,2)=0.0                                                        PORO3820
      Z(3,1)=ZREAL+RMET                                                 PORO3830
      Z(3,2)=ZIMAG                                                      PORO3840
      Z(4,1)=ZREAL+REL                                                  PORO3850
      Z(4,2)=ZIMAG                                                      PORO3860
C   FLAG #6 DENOTES METAL ON BOTTOM WHICH IS NOT TO HAVE SC INTFC       PORO3870
C   ABOVE IT--- IT'S PURPOSE IS JUST TO DISTRIBUTE THE CURRENT          PORO3880
C   INTO THE PLANE ABOVE IT                                             PORO3890
      Z(6,1) = RMET                                                     PORO3900
      Z(6,2) = 0.0                                                      PORO3910
      WRITE(6,132)REL,RMET,RS,RP,CP                                     PORO3920
  132 FORMAT (' REL,RMET,RS,RP,CP= ',1P,5E13.3)                         PORO3930
      WRITE(6,138) CUR(1),CUR(2)                                        PORO3940
  138 FORMAT(' INPUT CURRENT= ',1P,E12.2,' +(J*) ',E12.2)               PORO3950
      WRITE (6,150)                                                     PORO3960
  150 FORMAT (' IMPEDANCES , REAL & IMAG. :')                           PORO3970
      WRITE (6,200)((Z(I,J) ,I=1,4),J=1,2)                              PORO3980
  200 FORMAT (' Z= ',4E13.4)                                            PORO3990
      IF (IDUMP.EQ.1) THEN                                              PORO4000
```

```
      IF (MOD(INT(FRCNT),IPPD).EQ.0) WRITE (9,200)((Z(I,J),I=1,4),J=1,2)PORO4010
      ENDIF                                                           PORO4020
C                                                                     PORO4030
C   NOW CONSTRUCT CONDUCTIVITY MATRIX                                 PORO4040
      CALL CONDUC(A,G,JA,IA,B,N,NTCALC,MODEL)                         PORO4050
C                                                                     PORO4060
      IF (ISOLVE.EQ.0) THEN                                           PORO4070
          WRITE (7,301)                                               PORO4080
  301     FORMAT (' LATTICE CONSTRUCTED -- NO SOLVE OPTION IN EFFECT') PORO4090
          GOTO 760                                                    PORO4100
      ENDIF                                                           PORO4110
C                                                                     PORO4120
C IF THIS IS THE SAME LATTICE AS BEFORE, GO RIGHT TO SR MNSO          PORO4130
C  SINCE IPM AND SYMBOLIC FACTORIZATION IS UNCHANGED!                 PORO4140
C                                                                     PORO4150
      IF (IWENT.NE.0) GOTO 380                                        PORO4160
C                                                                     PORO4170
       IBUF1 = IBOT                                                   PORO4180
       IERIND=1                                                       PORO4190
       CALL MOOO(N,NTCALC,IBUF1,IA,JA,IPM,IER,IB,IU,JU)               PORO4200
      IF (IER(1).NE.0) WRITE (6,350) ERIND(IERIND),(IER(I),I=1,7)     PORO4210
  350 FORMAT (' ERROR IN ',A4,4X,'IER=',7I10)                         PORO4220
      IF (IER(1).EQ.0) WRITE (6,351) ERIND(IERIND),(IER(I),I=1,7)     PORO4230
  351 FORMAT (' NORMAL TERMINATION CODES FROM ',A4,4X,7I8)            PORO4240
      IBUF1 = IBOT                                                    PORO4250
      IBUF2 = IBO1                                                    PORO4260
      IERIND=2                                                        PORO4270
      CALL MSSO(N,IBUF1,IBUF2,IA,JA,IPM,IU,JU,IK,IER,IUP,IAT,JAT,IAP, PORO4280
     1IPMI,IX,IB,JUA)                                                 PORO4290
       IF (IER(1).EQ.0) GOTO 360                                      PORO4300
C  ----------ERROR DOCUMENTATION                                      PORO4310
       WRITE (6,350) ERIND(IERIND),(IER(I),I=1,7)                     PORO4320
      WRITE (6,353) IB(1),IB(2)                                       PORO4330
  353 FORMAT (' FIRST AND LAST ELEMENT IN ROW I OF G= ',2I11)         PORO4340
      WRITE (6,354) N,IK                                              PORO4350
  354 FORMAT (' NUMBER OF ROWS PROCESSED = ',I8,' TOTAL NO. OF',      PORO4360
     C' OPERATIONS = ',I20)                                           PORO4370
C   ----------------------------                                     PORO4380
  360  WRITE (6,352) IBUF1,IBUF2                                      PORO4390
  352  FORMAT (' SR MSSO: NUMBER OF BUFFER LOCATIONS USED  IN JU',    PORO4400
     1' AND JUA = ',2I20)                                             PORO4410
      WRITE (6,370) IK                                                PORO4420
  370 FORMAT (' NUMBER OF MULTIPLICATIONS REQUIRED = ',I20)           PORO4430
      IBUF3 = IU(N+1) - 1                                             PORO4440
      WRITE (6,371) IBUF3                                             PORO4450
  371 FORMAT (' MINIMUM BUFFER SPACE FOR JU AND U IN MNSO = ',I20)    PORO4460
C                                                                     PORO4470
C       ABNORMAL TERMINATION                                          PORO4480
      IF (IER(1).NE.0) GOTO 820                                       PORO4490
C                                                                     PORO4500
```

```
C  SET FLAG TO INDICATE THAT THIS LATTICE HAS MOOO AND MSSO SOLVED       PORO4510
C                                                                       PORO4520
      IWENT = 1                                                         PORO4530
      ISW = IB(1)                                                       PORO4540
  380 IERIND = 3                                                        PORO4550
      CALL MNSO (N,ISW,IA,JA,G,IAT,JAT,IAP,IPM,IU,JU,U,DI,IER,IUP,X)    PORO4560
      IF (IER(1).NE.0) WRITE (6,350) ERIND(IERIND),(IER(I),I=1,7)       PORO4570
       ISL=0                                                            PORO4580
C   IF FLAG 'IDIST' IS SET, THEN CURRENT IS TO BE UNIFORMLY DISTRIBUTED PORO4590
C         OVER ALL OF BOTTOM PLANE (ELECTROLYTE).                       PORO4600
C                                                                       PORO4610
      IF (IDIST.NE.1) GOTO 410                                          PORO4620
C                                                                       PORO4630
C                                                                       PORO4640
      WAREA = IM * JM                                                   PORO4650
      WRITE (6,395) WAREA                                               PORO4660
  395 FORMAT(' CURRENT DISTRIBUTED EVENLY THROUGH BOTTOM',F4.0,' NODES')PORO4670
      CURNOR(1) = CUR(1) / WAREA                                        PORO4680
      CURNOR(2) = CUR(2) / WAREA                                        PORO4690
      IEND = 2 * IM * JM                                                PORO4700
      DO 400 II=1,IEND,2                                                PORO4710
      B(II) = CURNOR(1)                                                 PORO4720
      B(II+1) = - CURNOR(2)                                             PORO4730
  400 CONTINUE                                                          PORO4740
  410  CALL MBSO(N,ISL,IPM,IU,JU,U,DI,B,G,X)                            PORO4750
      DIV1=CUR(1)**2 + CUR(2)**2                                        PORO4760
      ZREAL = (G(1)*CUR(1) + G(2)*CUR(2))/DIV1                          PORO4770
      ZIMAG = (G(2)*CUR(1) - G(1)*CUR(2))/DIV1                          PORO4780
      WRITE (6,503) DHOLD                                               PORO4790
      WRITE(6,172) ZREAL,ZIMAG                                          PORO4800
  172 FORMAT (' COMPL IMPED.= ',1P,E12.4,' (+J*) ',E12.4)               PORO4810
      IF (IDUMP.EQ.1) THEN                                              PORO4820
*        DUMP V OF ALL NODES                                            PORO4830
         IF (MOD(INT(FRCNT),IPPD).EQ.0) WRITE (9,182) (G(II),II=1,N)    PORO4840
  182    FORMAT (1P,6E13.4)                                             PORO4850
      ENDIF                                                             PORO4860
                                                                        PORO4870
      CALL CURCAL (A,G,MODEL)                                           PORO4880
      WRITE (6,420)REALI,RMAGI                                          PORO4890
  420 FORMAT (' CALCULATED CURRENT = ',1P,E12.2,' +(J*) ',E12.2)        PORO4900
C   NORMALIZE IMPEDANCE TO GIVE 1 CM**2 READING:                        PORO4910
C                                                                       PORO4920
      ZRLNOR = ZREAL / SCALE                                            PORO4930
      ZIMNOR = ZIMAG / SCALE                                            PORO4940
      WRITE (6,190) SCALE                                               PORO4950
  190 FORMAT (' NORMALIZED 1 SQ CM IMPEDANCE BY ',F8.0,' NORM FACTOR:') PORO4960
      WRITE (6,172) ZRLNOR,ZIMNOR                                       PORO4970
C                                                                       PORO4980
C                                                                       PORO4990
C   DOCUMENTATION PAGE                                                  PORO5000
C                                                                       PORO5010
```

```
      II=1                                                                  PORO5020
      WRITE(7,500)                                                          PORO5030
  500 FORMAT(' ',/,'  SUMMARY :')                                           PORO5040
      WRITE(7,124) IM,JM,KHOLD,POR                                          PORO5050
      IF (KMSTOP.NE.KHOLD) WRITE (6,502) KMSTOP                             PORO5060
  502 FORMAT (' ',/,' PROBLEM TRUNCATED AT PLANE # ',I2,//)                 PORO5070
      WRITE(7,505) FREQ                                                     PORO5080
  503 FORMAT (' DSEED = ',F13.0,'D0')                                       PORO5090
  505 FORMAT(' FREQ  = ',1P,E12.3)                                          PORO5100
      WRITE (7,420)REALI,RMAGI                                              PORO5110
C                                                                           PORO5120
C ------------- CHECK FOR NUMERICAL ERROR -------------                    PORO5130
      TOL = .03                                                             PORO5140
      DEVRL = (ABS(CUR(1)-REALI)) / CUR(1)                                  PORO5150
      DEVIM = (ABS(CUR(2)-RMAGI)) / CUR(2)                                  PORO5160
C IS THE DEVIATION BETWEEN CURR IN AND CURR OUT LT ALLOWED TOLERANCE?       PORO5170
      IF (DEVRL.LT.TOL.AND.DEVIM.LT.TOL) GOTO 610                           PORO5180
      WRITE (6,602) DEVRL,DEVIM                                             PORO5190
      WRITE (7,602) DEVRL,DEVIM                                             PORO5200
  602 FORMAT (' ********** DEVIATION OF CURRENT - REAL + IMAG = ',          PORO5210
     1   2P,2F6.1,'  PERCENT')                                              PORO5220
      IBOMB = 1                                                             PORO5230
C -------------- END OF ERROR CHECK -----------------                       PORO5240
  610 WRITE (7,190) SCALE                                                   PORO5250
      WRITE (7,172) ZRLNOR,ZIMNOR                                           PORO5260
C ---- MAINTAIN DATABASE --------------                                     PORO5270
      WRITE (8,650) FREQ,ZRLNOR,ZIMNOR                                      PORO5280
  650 FORMAT (' 10',1P,E12.3,2E12.4)                                        PORO5290
C  --------- END DATABASE ENTRY ----------                                  PORO5300
C                                                                           PORO5310
C     CALL CURPRF (A,LBLA,G)                                                PORO5320
  700 CONTINUE                                                              PORO5330
C                                                                           PORO5340
C   FINISHED A GROUP OF TRIALS AT A GIVEN POROSITY AND A GIVEN FREQ.        PORO5350
C                                                                           PORO5360
C  CALCULATE TIMES:                                                         PORO5370
C                                                                           PORO5380
      CALL DATTIM (EVCPU,ETIME,ETCPU)                                       PORO5390
C                                                                           PORO5400
      CONNOW = ETIME/60.0                                                   PORO5410
      TIMNOW = ETCPU/60.0                                                   PORO5420
      CONTIM = CONNOW - CONLAS                                              PORO5430
      RUNTIM = TIMNOW - TIMLAS                                              PORO5440
C                                                                           PORO5450
      IF (CONTIM.GT.0) GOTO 720                                             PORO5460
      WRITE (7,710)                                                         PORO5470
  710 FORMAT (' ACCOUNTING BOUNDARY CROSSED-- INVALID TIME DATA')           PORO5480
      STIME = 0.0                                                           PORO5490
      GOTO 750                                                              PORO5500
```

```
  720 WRITE (7,725) CONTIM,RUNTIM                                          PORO5510
  725 FORMAT (' ELAPSED TIME = ',F7.2,'  CPU TIME = ',F7.2,' MINUTES')     PORO5520
C                                                                          PORO5530
C                                                                          PORO5540
C   NOW STORE THESE VALUES FOR NEXT TIME AROUND                            PORO5550
  750 CONLAS = CONNOW                                                      PORO5560
      TIMLAS = TIMNOW                                                      PORO5570
C                                                                          PORO5580
*****************************                                              PORO5590
*  USER REQUESTED ABORT?? -MAY BE DONE BY SETTING LOC X"464 NONZERO        PORO5600
  760  IF (ITYBIT(0).NE.0) THEN                                            PORO5610
           WRITE (7,751)                                                   PORO5620
           WRITE (6,751)                                                   PORO5630
  751    FORMAT (' USER INTERRUPT IN HEXLOC X"464. PROG TERMINATING')      PORO5640
           GOTO 900                                                        PORO5650
           ENDIF                                                           PORO5660
*********************************                                        PORO5670
                                                                           PORO5680
C   NEXT POR & NEXT FREQ                                                   PORO5690
      FRCNT=FRCNT + 1                                                      PORO5700
      FREQ = FRBEG * 10 ** (FRCNT/IPPD)                                    PORO5710
      IF (FREQ.LE.FREND) GOTO 130                                          PORO5720
      WRITE(6,799)                                                         PORO5730
  799 FORMAT('1NEXT PROBLEM:')                                             PORO5740
  820 NRPT = NRPT+1                                                        PORO5750
C    RESET KM FOR NEXT PROBLEM                                             PORO5760
C                                                                          PORO5770
      KM = KHOLD                                                           PORO5780
      IF (NRPT.LE.IRPT) GOTO 120                                           PORO5790
      POR = POR + PORINC                                                   PORO5800
      IF (POR.LE.POREND) GOTO 120                                          PORO5810
C  END OF THIS DATA SET                                                    PORO5820
      TT = TIMNOW - STIME                                                  PORO5830
      WRITE (7,850) TT                                                     PORO5840
  850 FORMAT ('  TOTAL CPU TIME FOR THIS PROBLEM WAS: ',F7.2,' MINUTES')   PORO5850
C                                                                          PORO5860
C  PROGRAM IS TERMINATED BY EOF CONDITION ON INPUT DATASET                 PORO5870
C  AT EOF IN THE NEXT READ, CONTROL IS TRANSFERRED TO 900                  PORO5880
      GOTO 5                                                               PORO5890
  900 IF (IBOMB.EQ.0) GOTO 910                                             PORO5900
      WRITE (6,905)                                                        PORO5910
      WRITE (7,905)                                                        PORO5920
  905 FORMAT (' CHECK INPUT DATA... CALCULATION INCORRECT!!!')             PORO5930
  910 CONTINUE                                                             PORO5940
      STOP                                                                 PORO5950
      END                                                                  PORO5960
*-------------------------------                                          PORO5970
*              04/02/85  -- IMPORTANT BUG FIX:                             PORO5980
*     RANDOM PTS I,J,K ARE CORRECTLY CHOSEN IN RANPT NOW.                  PORO5990
*                                                                          PORO6000
```

```
      SUBROUTINE NODES (A,LBLA,POR,DSEED,MMODEL)                        POR06010
C     SIZE OF A IS (IM,JM,KM)                                           POR06020
C     REVISED 2/1/83                                                    POR06030
C                                                                       POR06040
C    --THIS SUBROUTINE SETS UP THE PHYSICAL MODEL                       POR06050
C   THERE ARE A FEW CHOSABLE OPTIONS AS TO WHAT MODEL IS DESIRED:       POR06060
C         MODEL 1  ===>   STANDARD MODEL.  MAKE SURE THAT               POR06070
C                         ELECTRODE IS CONTINUOUS AND THAT ELECTROLYTE   POR06080
C                         FLOW IS CONTINUOUS ("CUBES" MODEL)            POR06090
C                                                                       POR06100
C         MODEL 2  ===>   METAL ELECTRODE IS CONSTRUCTED CONTINUOUS     POR06110
C                         BUT ELECTROLYTE PLACED EVERYWHERE ELSE        POR06120
C                         ("SPHERES MODEL")                             POR06130
C                                                                       POR06140
C         MODEL 3  ====>  ALTERNATE MODEL, CONSTRUCTED IN SUBROUTINE    POR06150
C                         "LATSET".                                     POR06160
C                                                                       POR06170
C         MODEL 4  ====>  METAL ELECTRODE NOT CONTINUOUS.               POR06180
C                         ELECTROLYTE PLACED IN ALL NONMETAL SITES      POR06190
C                         (SPHERES MODEL AGAIN).                        POR06200
*                          (FOR USE WITH RESISTORS AND CAPACITORS)      POR06210
C                                                                       POR06220
C     ---- IF PERIODIC BOUND CONDITIONS ARE REQUESTED, THEN 50 HAS BEEN POR06230
C          ADDED TO THE VALUE OF THE FLAG 'MODEL'                       POR06240
C                                                                       POR06250
      INTEGER*4 A(IM,JM,KM),LBLA(IM,JM,KM)                              POR06260
      REAL *8    DSEED                                                  POR06270
      COMMON/LATICE/IM,JM,KM,KMSTOP                                     POR06280
C                                                                       POR06290
      MODEL = MMODEL                                                    POR06300
      IF (MODEL.GT.50) MODEL = MMODEL-50                                POR06310
C                                                                       POR06320
                                                                        POR06330
      KMAX = KM-1                                                       POR06340
      KMIN = 2                                                          POR06350
      DO 5  I= 1,IM                                                     POR06360
      DO 5  J= 1,JM                                                     POR06370
      DO 5  K= 1,KM                                                     POR06380
      A(I,J,K)=0                                                        POR06390
    5 CONTINUE                                                          POR06400
      ISP=IM*JM*(KM-2)                                                  POR06410
      MB=(1-POR)*ISP                                                    POR06420
      WRITE (6,10) MMODEL                                               POR06430
      WRITE (7,10) MMODEL                                               POR06440
   10 FORMAT (' USING MODEL NUMBER ',I2)                                POR06450
C  WHICH MODEL ARE WE USING?                                            POR06460
      GOTO (20,20,15,17            ),MODEL                              POR06470
C     ========================================                          POR06480
C     MODEL 3- USE ALTERNATE LATTICE SETUP ROUTINE (NON RANDOM)         POR06490
```

```
   15  CALL LATSET(A,POR,MODEL)                                          POR06500
       GOTO 505                                                          POR06510
C    ==========================================                          POR06520
C    MODEL 4 - NO CONTIN. METAL                                          POR06530
   17  KMIN = 2                                                          POR06540
       KMAX = KM - 1                                                     POR06550
       CALL PLANE (A,1,5)                                                POR06560
       CALL PLANE (A,KM,5)                                               POR06570
       CALL RNDM(A,MB,DSEED,KMIN,KMAX,1)                                 POR06580
       GOTO 505                                                          POR06590
C    ==========================================                          POR06600
C   MODELS 1 & 2 -- OLD FASHIONED WAY                                    POR06610
   20  CALL RNDM(A,MB,DSEED,KMIN,KMAX,5)                                 POR06620
C   SET BOTTOM PLANE = ELECTROLYTE ; TOP PLANE = METAL                   POR06630
       CALL PLANE(A,1,2)                                                 POR06640
       CALL PLANE (A,KM,1)                                               POR06650
C                                                                        POR06660
C   ******************                                                   POR06670
C                                                                        POR06680
C   NOW CHECK FOR CONTINUATION OF METAL --NO ISLANDS ALLOWED             POR06690
C                                                                        POR06700
C THE (KM-1) PLANE IS ALL CONNECTED TO THE TOP METAL                     POR06710
       DO 50 I=1,IM                                                      POR06720
       DO 50 J=1,JM                                                      POR06730
       IF (A(I,J,KM-1).EQ.5) A(I,J,KM-1)=1                               POR06740
   50  CONTINUE                                                          POR06750
C                                                                        POR06760
   60  KFND=0                                                            POR06770
C    THIS LOOP LOOPS DOWN FROM K=(KM-2) TO K=2                           POR06780
       DO 400 KT=4,KM                                                    POR06790
       K=KM-KT+2                                                         POR06800
C                                                                        POR06810
C   FIRST CONNECT FROM ABOVE                                             POR06820
       DO 75 I=1,IM                                                      POR06830
       DO 75 J=1,JM                                                      POR06840
       IF (A(I,J,K).NE.5) GOTO 75                                        POR06850
       IF (A(I,J,K+1).EQ.1) A(I,J,K)=1                                   POR06860
   75  CONTINUE                                                          POR06870
C   NOW GO CONNECT METAL TOGETHER ON THE REST OF KTH PLANE               POR06880
  100  CALL CONECT(A,K,5,1,IFND,KFND)                                    POR06890
C   FINISHED PLANE YET ?                                                 POR06900
       IF (IFND.NE.0) GOTO 100                                           POR06910
C   IF FINISHED, GO TO NEXT PLANE                                        POR06920
  400  CONTINUE                                                          POR06930
C                                                                        POR06940
C     FINISHED GOING THRU ALL PLANES ...GO AAIN?                         POR06950
       IF (KFND.NE.0) GOTO 60                                            POR06960
C                                                                        POR06970
C                                                                        POR06980
C FINISHED CONNECTIN ALL METAL; NOW DEAL W/ ISOLATED PTS                 POR06990
C   ICTR COUNTS # OF POINTS TO BE REPLACED                               POR07000
```

```
      ICTR=0                                                            PORO7010
      KK=KM-2                                                           PORO7020
      DO 500 K=2,KK                                                     PORO7030
      DO 500 I=1,IM                                                     PORO7040
      DO 500 J=1,JM                                                     PORO7050
      IF (A(I,J,K).NE.5) GOTO 500                                       PORO7060
C     THIS IS AN ISLAND OF METAL                                        PORO7070
      A(I,J,K)=0                                                        PORO7080
  450 CALL RANPT(A,DSEED,KMIN,KMAX,IT,JT,KT)                            PORO7090
C  IS THIS REPLACEMENT PT OK?                                           PORO7100
      IFND=0                                                            PORO7110
      CALL NEARN(A,IT,JT,KT,0,1,IFND,KFND)                              PORO7120
      IF (IFND.NE.1) GOTO 450                                           PORO7130
      ICTR=ICTR+1                                                       PORO7140
      WRITE (6,460) I,J,K,IT,JT,KT                                      PORO7150
  460 FORMAT (' PT # ',3I3,'  REPLACED BY ',3I3)                        PORO7160
  500 CONTINUE                                                          PORO7170
      WRITE (6,501) ICTR                                                PORO7180
  501 FORMAT ('  # OF PTS REPLACED= ',I6)                               PORO7190
C                                                                       PORO7200
C  ====================================                                 PORO7210
C  NOW FIND ELECTROLYTE CONNECTIONS                                     PORO7220
  505 CONTINUE                                                          PORO7230
C    WE HAVE TWO METHODS--- 1) FILL IN ONLY CONNECTING ELEC PATHS       PORO7240
C                           2) FILL IN ELEC WHEREVER THERE IS NO METAL  PORO7250
      GOTO (540,515,540,515          ),MODEL                            PORO7260
C   ========================================                            PORO7270
C  WE HAVE CHOSEN MODEL #2 OR #4 -- FILL IN ALL NON-METAL               PORO7280
C    SPACE WITH ELECTROLYTE                                             PORO7290
  515 DO 520 K =  KMIN,KM                                               PORO7300
  520 CALL PLANE (A,K,2)                                                PORO7310
      CALL FILABL(A,LBLA)                                               PORO7320
      RETURN                                                            PORO7330
C   ========================================                            PORO7340
C   MODEL #1 -  WATCH FOR ELECTROLYTE FLOW                              PORO7350
  540 CALL PLANE (A,2,2)                                                PORO7360
C NOW GO THROUGH ALL PLANES BOTTOM TO TOP                               PORO7370
  550 KFND=0                                                            PORO7380
      KT=KM-1                                                           PORO7390
      DO 700 K=3,KT                                                     PORO7400
C   FIRST MAKE CONNECTIONS FROM BELOW                                   PORO7410
      DO 600 I= 1,IM                                                    PORO7420
      DO 600 J= 1,JM                                                    PORO7430
      IF (A(I,J,K).NE.0) GOTO 600                                       PORO7440
      IF (A(I,J,K-1).NE.2) GOTO 600                                     PORO7450
      A(I,J,K) = 2                                                      PORO7460
  600 CONTINUE                                                          PORO7470
C                                                                       PORO7480
C  NOW DEAL WITH NN ON PLANE                                            PORO7490
```

```
  610 CALL CONECT(A,K,0,2,IFND,KFND)                                    PORO7500
      IF (IFND.NE.0) GOTO 610                                           PORO7510
C   NOW GO TO NEXT PLANE                                                PORO7520
  700 CONTINUE                                                          PORO7530
C   FINISHED GOING THROUGH ALL PLANES.   DO IT AGAIN?                   PORO7540
      IF (KFND.NE.0) GOTO 550                                           PORO7550
      CALL FILABL(A,LBLA)                                               PORO7560
      RETURN                                                            PORO7570
      END                                                               PORO7580
*------------------------------                                        PORO7590
C                                                                       PORO7600
      SUBROUTINE CONECT(A,K,ISTAY,ILOOK,IFND,KFND)                      PORO7610
C    SIZE OF A IS (IM,JM,KM)                                            PORO7620
C    SIZE OF G IS (ISM,7,2)  WHERE ISM=IM*JM*KM-1                       PORO7630
C                                                                       PORO7640
C   SR TO SCAN A PLANE AT K FROM THE 4 DIFFERENT CORNERS OF THE PLANE   PORO7650
      INTEGER*4 A(IM,JM,KM)                                             PORO7660
      REAL *8    DSEED                                                  PORO7670
      COMMON/LATICE/IM,JM,KM,KMSTOP                                     PORO7680
C    IF A PREVIOUS CALL TO NEARN FOUND NO NEW VALUES THEN WE'RE FINISHEDPORO7690
C   IFND IS THE FLAG TO INDICATE WHETHER A NEW VALUE HAS BEEN FOUND     PORO7700
      IFND=0                                                            PORO7710
      DO 325 I= 1,IM                                                    PORO7720
      DO 325 J= 1,JM                                                    PORO7730
      CALL NEARN(A,I,J,K,ISTAY,ILOOK,IFND,KFND)                         PORO7740
  325 CONTINUE                                                          PORO7750
      IF (IFND.NE.1) GOTO 400                                           PORO7760
      IFND=0                                                            PORO7770
      DO 335 J= 1,JM                                                    PORO7780
      DO 335 I= 1,IM                                                    PORO7790
      IT=IM-I+1                                                         PORO7800
      JT=JM-J+1                                                         PORO7810
      CALL NEARN(A,IT,JT,K,ISTAY,ILOOK,IFND,KFND)                       PORO7820
  335 CONTINUE                                                          PORO7830
      IF (IFND.NE.1) GOTO 400                                           PORO7840
      IFND=0                                                            PORO7850
      DO 345 I= 1,IM                                                    PORO7860
      DO 345 J= 1,JM                                                    PORO7870
      JT=JM-J+1                                                         PORO7880
      CALL NEARN(A,I,JT,K,ISTAY,ILOOK,IFND,KFND)                        PORO7890
  345 CONTINUE                                                          PORO7900
      IF (IFND.NE.1) GOTO 400                                           PORO7910
      IFND=0                                                            PORO7920
      DO 355 J =1,JM                                                    PORO7930
      DO 355 I =1,IM                                                    PORO7940
      IT=IM-I+1                                                         PORO7950
      CALL NEARN(A,IT,J,K,ISTAY,ILOOK,IFND,KFND)                        PORO7960
  355 CONTINUE                                                          PORO7970
  400 CONTINUE                                                          PORO7980
```

```
      RETURN                                                            POR07990
      END                                                               POR08000
*-------------------------------                                        POR08010
C                                                                       POR08020
      SUBROUTINE RNDM(A,MB,DSEED,KMIN,KMAX,IVAL)                        POR08030
C                                                                       POR08040
C   SUBROUTINE TO GENERATE RANDOM LATTICE                               POR08050
C                                                                       POR08060
      REAL *8   DSEED                                                   POR08070
      INTEGER*4 A(IM,JM,KM)                                             POR08080
      COMMON/LATICE/IM,JM,KM,KMSTOP                                     POR08090
C                                                                       POR08100
C                                                                       POR08110
      DO 100 N=1,MB                                                     POR08120
 10   CALL RANPT(A,DSEED,KMIN,KMAX,I,J,K)                               POR08130
      IF (A(I,J,K).NE.0) GOTO 10                                        POR08140
C  PLACE METAL ON NODE (PLACE THE VALUE IVAL ON A(I,J,K))               POR08150
      A(I,J,K) = IVAL                                                   POR08160
 100  CONTINUE                                                          POR08170
      RETURN                                                            POR08180
      END                                                               POR08190
*-------------------------------                                        POR08200
C                                                                       POR08210
      SUBROUTINE PLANE(A,IZ,ICONST)                                     POR08220
C  S.R. TO SET A WHOLE PLANE AT SOME VALUE K=IZ TO A CONSTANT           POR08230
C                                                                       POR08240
      INTEGER*4 A(IM,JM,KM)                                             POR08250
      COMMON/LATICE/IM,JM,KM,KMSTOP                                     POR08260
C                                                                       POR08270
      DO 50 I =1,IM                                                     POR08280
      DO 50 J =1,JM                                                     POR08290
      IF (A(I,J,IZ).NE.0) GOTO 50                                       POR08300
      A(I,J,IZ) = ICONST                                                POR08310
 50   CONTINUE                                                          POR08320
      RETURN                                                            POR08330
      END                                                               POR08340
*-------------------------------                                        POR08350
C                                                                       POR08360
      SUBROUTINE NEARN(A,I,J,K,ISTAY,ILOOK,IFND,KFND)                   POR08370
C                                                                       POR08380
C  S.R. TO LOOK AT NEAREST NEIGHBORS FOR VALUE 'ILOOK'                  POR08390
C   IF 'ILOOK' IS FOUND IN ONE OF NN THEN A(I,J,K)=ILOOK TOO            POR08400
      INTEGER*4 A(IM,JM,KM)                                             POR08410
      COMMON/LATICE/IM,JM,KM,KMSTOP                                     POR08420
C                                                                       POR08430
      IF (A(I,J,K).NE.ISTAY) GOTO 400                                   POR08440
      IF (I-1.LT.1) GOTO 20                                             POR08450
      IF (A(I-1,J,K).EQ.ILOOK) GOTO 300                                 POR08460
 20   IF (I+1.GT.IM) GOTO 30                                            POR08470
```

```
      IF (A(I+1,J,K).EQ.ILOOK) GOTO 300                                 PORO8480
   30 IF (J-1.LT.1) GOTO 40                                             PORO8490
      IF (A(I,J-1,K).EQ.ILOOK) GOTO 300                                 PORO8500
   40 IF (J+1.GT.JM) GOTO 50                                            PORO8510
      IF (A(I,J+1,K).EQ.ILOOK) GOTO 300                                 PORO8520
   50 IF (K-1.LT.1) GOTO 60                                             PORO8530
      IF (A(I,J,K-1).EQ.ILOOK) GOTO 300                                 PORO8540
   60 IF (K+1.GT.KM) GOTO 100                                           PORO8550
      IF (A(I,J,K+1).EQ.ILOOK) GOTO 300                                 PORO8560
C NO NN VALUE OF ILOOK FOUND:                                           PORO8570
  100 GOTO 400                                                          PORO8580
C     ONE OF ADJACENT NODES = ILOOK                                     PORO8590
  300 A(I,J,K) =ILOOK                                                   PORO8600
      IFND=1                                                            PORO8610
      KFND=1                                                            PORO8620
  400 RETURN                                                            PORO8630
      END                                                               PORO8640
*-------------------------------                                        PORO8650
C                                                                       PORO8660
      SUBROUTINE RANPT(A,DSEED,KMIN,KMAX,I,J,K)                         PORO8670
C                                                                       PORO8680
C    SUBROUTINE TO GENERATE RANDOM LATTICE  PT (I,J,K)                  PORO8690
*     NUMBER BETWEEN [KMIN,KMAX] = INT(R(1)*(KMAX-KMIN+1)+KMIN)        PORO8700
      REAL *8    DSEED                                                  PORO8710
      INTEGER*4 A(IM,JM,KM)                                             PORO8720
      DIMENSION R(3)                                                    PORO8730
      COMMON/LATICE/IM,JM,KM,KMSTOP                                     PORO8740
C                                                                       PORO8750
      CALL GGUBS(DSEED,3,R)                                             PORO8760
      I=INT(R(1)*IM + 1.0)                                              PORO8770
      J=INT(R(2)*JM + 1.0)                                              PORO8780
      K=INT(R(3)*(KMAX-KMIN+1) + KMIN)                                  PORO8790
      RETURN                                                            PORO8800
      END                                                               PORO8810
*-------------------------------                                        PORO8820
C                                                                       PORO8830
      SUBROUTINE CHOP (A)                                               PORO8840
C                                                                       PORO8850
C           SUBROUTINE TO EFFECTIVELY CHOP OFF THE UPPER PLANES         PORO8860
C            AND ELIMINATE THEM FROM OUR PROBLEM                        PORO8870
C            IF THERE IS LITTLE ELECTROLYTE PENETRATION THERE.          PORO8880
C                                                                       PORO8890
      INTEGER*4 A(IM,JM,KM)                                             PORO8900
C                                                                       PORO8910
      COMMON/LATICE/IM,JM,KM,KMSTOP                                     PORO8920
      COMMON/ELEC/Z(6,2),CUR(2),NCUR,NGND,REALI,RMAGI                   PORO8930
C                                                                       PORO8940
C    CHANGE KM TO KMSTOP                                                PORO8950
      KM = KMSTOP                                                       PORO8960
C    SET GND NODE TO NEW LOCATION                                       PORO8970
C    IN LOWER RT HAND CORNER OF PLANE # KMSTOP                          PORO8980
```

```
      NGND = IM*JM*KM                                                    POR08990
C                                                                        POR09000
C  CHECK TO SEE THAT CORNER NODE IS CONNECTED TO METAL                   POR09010
C                                                                        POR09020
      IF (A(IM,JM,KM).EQ.1) GOTO 100                                     POR09030
      IF (A(IM,JM,KM).EQ.2) WRITE (6,20)                                 POR09040
  20  FORMAT (' ELECTROLYTE AT CORNER TOP PLANE DISPLACED FOR GND')      POR09050
      DO 50 I=1,IM                                                       POR09060
      DO 50 J=1,JM                                                       POR09070
      II = IM-I+1                                                        POR09080
      JJ = JM-J+1                                                        POR09090
      A(II,JJ,KM) = 1                                                    POR09100
      IFND = 0                                                           POR09110
      CALL NEARN(A,II,JJ,KM,1,1,IFND,KFND)                               POR09120
C  IFND IS SET (=1) IF THIS PT IS CONNECTED TO                           POR09130
C   ANOTHER METAL.                                                       POR09140
      IF (IFND.NE.0) GOTO 100                                            POR09150
  50  CONTINUE                                                           POR09160
C                                                                        POR09170
  100 WRITE (6,175) KMSTOP                                               POR09180
  175 FORMAT (' ',//,' LATTICE TRUNCATED ABOVE PLANE # ',I3)             POR09190
C                                                                        POR09200
C COMPUTE NEW STATISTICS; NO NEED TO RE-DISPLAY LATTICE                  POR09210
C                                                                        POR09220
      ISHOW = 0                                                          POR09230
C                                                                        POR09240
      CALL DISPLA (A,ISHOW,MODEL)                                        POR09250
      RETURN                                                             POR09260
      END                                                                POR09270
*------------------------------                                          POR09280
C                                                                        POR09290
C                                                                        POR09300
      SUBROUTINE DISPLA (A,ISHOW,MODEL)                                  POR09310
      INTEGER*4 A(IM,JM,KM)                                              POR09320
      DIMENSION LBL(7),LAB1(20)                                          POR09330
      COMMON/LATICE/IM,JM,KM,KMSTOP                                      POR09340
      DATA LBL/' ','M','E','X','X','X','B'/                              POR09350
C                                                                        POR09360
C                                                                        POR09370
C    ISHOW = 1 IS A FLAG THAT PRINTS OUT A PICTURE OF                    POR09380
C              THE LATTICE                                               POR09390
C    ISHOW = 0 IS A FLAG THAT JUST CAUSES PRINT OUT OF STATISTICS        POR09400
      WRITE (6,4)                                                        POR09410
    4 FORMAT(' KEY :  " "-AIR   M-METAL   E-ELECTROLYTE')                POR09420
C  INITIALIZE COUNTERS FOR DISTRIBUTION TOTALS                           POR09430
      NMET = 0                                                           POR09440
      NELEC = 0                                                          POR09450
      NAIR = 0                                                           POR09460
      NSC = 0                                                            POR09470
C                                                                        POR09480
C  NOW GO THROUGH EACH PLANE :                                           POR09490
```

```
      DO 600 K=KM,1,-1                                                  POR09500
C  ****************************                                         POR09510
C  NOW COMPUTE THE DISTRIBUTIONS ON THIS PLANE :                        POR09520
C                                                                       POR09530
      ICONT = 0                                                         POR09540
      IF (MODEL.GT.50) ICONT = 1                                        POR09550
      KMET=0                                                            POR09560
      KELEC=0                                                           POR09570
      KSC=0                                                             POR09580
      DO 250 I=1,IM                                                     POR09590
      DO 250 J=1,JM                                                     POR09600
      IAT = A(I,J,K)                                                    POR09610
      IF (IAT.EQ.0) GOTO 250                                            POR09620
      IF (IAT.EQ.1) KMET=KMET+1                                         POR09630
      IF (IAT.EQ.2) KELEC=KELEC+1                                       POR09640
C   NOW CHECK IN THE 3 FORWARD DIRECTIONS FOR SC INTFC                  POR09650
      IF (I+1.GT.IM) THEN                                               POR09660
           IF (ICONT.EQ.1) THEN                                         POR09670
                NEXT = A(1,J,K)                                         POR09680
                IF (IAT+NEXT.EQ.3) KSC = KSC + 1                        POR09690
           ENDIF                                                        POR09700
       ELSE                                                             POR09710
           NEXT = A(I+1,J,K)                                            POR09720
           IF (IAT+NEXT.EQ.3) KSC = KSC + 1                             POR09730
       ENDIF                                                            POR09740
      IF (J+1.GT.JM) THEN                                               POR09750
           IF (ICONT.EQ.1) THEN                                         POR09760
                NEXT = A(I,1,K)                                         POR09770
                IF (IAT+NEXT.EQ.3) KSC = KSC + 1                        POR09780
           ENDIF                                                        POR09790
       ELSE                                                             POR09800
           NEXT = A(I,J+1,K)                                            POR09810
           IF (IAT+NEXT.EQ.3) KSC = KSC + 1                             POR09820
       ENDIF                                                            POR09830
  140 IF (K+1.GT.KM) GOTO 250                                           POR09840
      IF (IAT+A(I,J,K+1).EQ.3) KSC=KSC+1                                POR09850
C                                                                       POR09860
C                                                                       POR09870
  250 CONTINUE                                                          POR09880
      KAIR = (IM*JM) - KMET - KELEC                                     POR09890
C                                                                       POR09900
C   NOW ACCUMULATE TALLY FOR WHOLE LATTICE                              POR09910
      NMET = NMET + KMET                                                POR09920
      NELEC = NELEC + KELEC                                             POR09930
      NAIR = NAIR + KAIR                                                POR09940
      NSC = NSC + KSC                                                   POR09950
C   END PLANE DISTRIBUTION CALC                                         POR09960
C  ******************************                                       POR09970
C ---- IF ISHOW IS SET THEN DISPLAY LATTICE----                         POR09980
C                                                                       POR09990
```

```
      IF (ISHOW.EQ.0) GOTO 600                                          POR10000
C                                                                       POR10010
C                                                                       POR10020
      WRITE(6,5) K                                                      POR10030
    5 FORMAT ('0 LEVEL ',I2)                                            POR10040
      HOLES= KELEC + KAIR                                               POR10050
      PPOR= HOLES / (IM*JM)                                             POR10060
      DO 600 I=1,IM                                                     POR10070
      DO 8 J=1,JM                                                       POR10080
      IAT=A(I,J,K)                                                      POR10090
      LAB1(J)=LBL(IAT+1)                                                POR10100
    8 CONTINUE                                                          POR10110
      GOTO (10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29)POR10120
     C,I                                                                POR10130
   10 WRITE (6,40)(LAB1(J),J=1,JM)                                      POR10140
   40 FORMAT (T40,20A2)                                                 POR10150
      GOTO 600                                                          POR10160
   11 WRITE(6,41) (LAB1(J),J=1,JM)                                      POR10170
   41 FORMAT (' E/A/M =',T39,20A2)                                      POR10180
      GOTO 600                                                          POR10190
   12 WRITE(6,42) KELEC,KAIR,KMET,(LAB1(J),J=1,JM)                      POR10200
   42 FORMAT (' ',I3,'/',I3,'/',I3,T38,20A2)                            POR10210
      GOTO 600                                                          POR10220
   13 WRITE(6,43) PPOR,(LAB1(J),J=1,JM)                                 POR10230
   43 FORMAT (' PLANE POR=',F3.2,T37,20A2)                              POR10240
      GOTO 600                                                          POR10250
   14 WRITE(6,44) KSC,(LAB1(J),J=1,JM)                                  POR10260
   44 FORMAT (' # SC INT=',I4,T36,20A2)                                 POR10270
      GOTO 600                                                          POR10280
   15 WRITE(6,45) (LAB1(J),J=1,JM)                                      POR10290
   45 FORMAT (T35,20A2)                                                 POR10300
      GOTO 600                                                          POR10310
   16 WRITE(6,46) (LAB1(J),J=1,JM)                                      POR10320
   46 FORMAT (T34,20A2)                                                 POR10330
      GOTO 600                                                          POR10340
   17 WRITE(6,47) (LAB1(J),J=1,JM)                                      POR10350
   47 FORMAT (T33,20A2)                                                 POR10360
      GOTO 600                                                          POR10370
   18 WRITE(6,48) (LAB1(J),J=1,JM)                                      POR10380
   48 FORMAT (T32,20A2)                                                 POR10390
      GOTO 600                                                          POR10400
   19 WRITE(6,49) (LAB1(J),J=1,JM)                                      POR10410
   49 FORMAT (T31,20A2)                                                 POR10420
      GOTO 600                                                          POR10430
   20 WRITE(6,50) (LAB1(J), J=1,JM)                                     POR10440
   50 FORMAT (T30,20A2)                                                 POR10450
      GOTO 600                                                          POR10460
   21 WRITE(6,51) (LAB1(J), J=1,JM)                                     POR10470
   51 FORMAT (T29,20A2)                                                 POR10480
```

```
      GOTO 600                                                          POR10490
   22 WRITE(6,52) (LAB1(J), J=1,JM)                                     POR10500
   52 FORMAT (T28,20A2)                                                 POR10510
      GOTO 600                                                          POR10520
   23 WRITE(6,53) (LAB1(J), J=1,JM)                                     POR10530
   53 FORMAT (T27,20A2)                                                 POR10540
      GOTO 600                                                          POR10550
   24 WRITE(6,54) (LAB1(J), J=1,JM)                                     POR10560
   54 FORMAT (T26,20A2)                                                 POR10570
      GOTO 600                                                          POR10580
   25 WRITE(6,55) (LAB1(J), J=1,JM)                                     POR10590
   55 FORMAT (T25,20A2)                                                 POR10600
      GOTO 600                                                          POR10610
   26 WRITE (6,56) (LAB1(J),J=1,JM)                                     POR10620
   56 FORMAT (T24,20A2)                                                 POR10630
      GOTO 600                                                          POR10640
   27 WRITE (6,57) (LAB1(J),J=1,JM)                                     POR10650
   57 FORMAT (T23,20A2)                                                 POR10660
      GOTO 600                                                          POR10670
   28 WRITE (6,58) (LAB1(J),J=1,JM)                                     POR10680
   58 FORMAT (T22,20A2)                                                 POR10690
      GOTO 600                                                          POR10700
   29 WRITE (6,59) (LAB1(J),J=1,JM)                                     POR10710
   59 FORMAT (T21,20A2)                                                 POR10720
  600 CONTINUE                                                          POR10730
C  SUMMARY                                                            POR10740
C                                                                       POR10750
      IS=IM*JM*KM                                                       POR10760
C COMPUTE TOTAL # INTFC SITES POSSIBLE                                  POR10770
      INTOT = (KM-2)*(IM*(JM-1) + JM*(IM-1)) + IM*JM*(KM-1)             POR10780
      PAREA = IM*JM                                                     POR10790
      ROUGH = NSC/PAREA                                                 POR10800
      WRITE (6,700) NELEC,NAIR,NMET,NSC, ROUGH                          POR10810
      WRITE (7,700) NELEC,NAIR,NMET,NSC ,ROUGH                          POR10820
  700 FORMAT ('0TOTALS: (E/A/M)=',I4,'/',I4,'/',I4,10X,'# SC INTFC=',   POR10830
     1I6,' ROUGH. = ',F5.2)                                             POR10840
      WRITE(6,710) IS,INTOT                                             POR10850
      WRITE(7,710) IS,INTOT                                             POR10860
  710 FORMAT(' TOTAL LATTICE PTS - ',I6,5X,'# INTERFACE SITES- ',I6)    POR10870
      RETURN                                                            POR10880
      END                                                               POR10890
*-------------------------------                                        POR10900
C                                                                       POR10910
C                                                                       POR10920
      SUBROUTINE FILABL(A,LBLA)                                         POR10930
      INTEGER*4 A(IM,JM,KM),LBLA(IM,JM,KM)                             POR10940
      COMMON/LATICE/IM,JM,KM,KMSTOP                                     POR10950
*                                                                       POR10960
*     INITIALIZE ARRAY LBLA TO REFLECT LATTICE NODE #'S                 POR10970
*                                                                       POR10980
```

```
*                                                                       POR10990
*  ICNTR COUNTS THE NUMBER OF NONZERO NODES IN THE LATTICE              POR11000
      ICNTR = 0                                                         POR11010
      DO 100 K=1,KM                                                     POR11020
      DO 100 I=1,IM                                                     POR11030
      DO 100 J=1,JM                                                     POR11040
*    -- IF A(I,J,K) = 0 THEN LBLA IS ALSO 0                            POR11050
         LBLA(I,J,K) = A(I,J,K)                                         POR11060
         IF (A(I,J,K).NE.0) THEN                                        POR11070
              ICNTR = ICNTR + 1                                         POR11080
              LBLA(I,J,K) = ICNTR                                       POR11090
         ENDIF                                                          POR11100
C                                                                       POR11110
  100 CONTINUE                                                          POR11120
      RETURN                                                            POR11130
      END                                                               POR11140
*-------------------------------                                        POR11150
*                                                                       POR11160
*                                                                       POR11170
      SUBROUTINE CONDUC(A,G,JA,IA,B,N,NT,MODEL)                         POR11180
C                                                                       POR11190
C   S.R. TO CONSTRUCT G MATRIX     REV 5/28/84                          POR11200
C                                                                       POR11210
      INTEGER*4 A(IM,JM,KM)                                             POR11220
      COMMON/DIMS/IDIM1,IDIM2,IDIM3,IDIM4                               POR11230
      COMMON/LATICE/IM,JM,KM,KMSTOP                                     POR11240
      COMMON/ELEC/Z(6,2),CUR(2),NCUR,NGND,REALI,RMAGI                   POR11250
      COMMON/GTEMP/ NG1,NG2,NG3,NG4,NG5,ICOL(6),GREAL(6),GIMAG(6)       POR11260
      DIMENSION G(IDIM3),JA(IDIM3)                                      POR11270
      DIMENSION IA(IDIM2),B(IDIM2)                                      POR11280
C                                                                       POR11290
C    SEE SUBROUTINE NODES FOR THE SIGNIFICANCE OF MODEL                 POR11300
C                                                                       POR11310
C ---> HERE-IF MODEL > 50 THEN PERIODIC BOUNDARY CONDITIONS             POR11320
C ---> HAVE BEEN REQUESTED                                              POR11330
C                                                                       POR11340
      ICONT = 0                                                         POR11350
      IF (MODEL.GT.50) ICONT = 1                                        POR11360
C                                                                       POR11370
C  G  IS COEFFICIENT MATRIX STORED IN SPARSE-SYMMETRIC MODE             POR11380
C  JA IS COLUMN ADDRESSES OF EACH ELEMENT OF PACKED G MATRIX            POR11390
C  IA IS THE ROW POINTER TO THE BEGINNING OF EACH ROW--                 POR11400
C         IA(N) CONTAINS THE INDEX OF THE ELEMENT IN G THAT             POR11410
C         STARTS THE N'TH ROW OF MATRIX G                               POR11420
C   B IS THE RHS OF THE SIMULTANEOUS EQU.                               POR11430
C                                                                       POR11440
C                                                                       POR11450
      DO 50 I=1,IDIM3                                                   POR11460
      G(I)=0.0                                                          POR11470
```

```
      JA(I)=0                                                          POR11480
   50 CONTINUE                                                         POR11490
      DO 60 I=1,IDIM2                                                  POR11500
      B(I)=0.0                                                         POR11510
      IA(I)=0                                                          POR11520
   60 CONTINUE                                                         POR11530
C                                                                      POR11540
C  NNA COUNTS WHICH ABS NODE NO. WE'RE UP TO (INCL. 0'S)               POR11550
C  NN  COUNTS WHICH NODE # WE'RE UP TO (EXCLUD. 0'S)                   POR11560
C  N  COUNTS WHICH LINE OF MATRIX WE'RE UP TO                          POR11570
C  NT COUNTS THE # OF ELEMENTS IN G                                    POR11580
C                                                                      POR11590
      NNA = 0                                                          POR11600
      NN  = 0                                                          POR11610
      N   = 0                                                          POR11620
      NT  = 0                                                          POR11630
C                                                                      POR11640
C  INITIALIZE ADDRESS OF GND ADJACENT NODES                            POR11650
C                                                                      POR11660
      NG1 = 0                                                          POR11670
      NG2 = 0                                                          POR11680
      NG3 = 0                                                          POR11690
      NG4 = 0                                                          POR11700
      NG5 = 0                                                          POR11710
C                                                                      POR11720
C                                                                      POR11730
      DO 500 K=1,KM                                                    POR11740
      DO 500 I=1,IM                                                    POR11750
      DO 500 J=1,JM                                                    POR11760
      NNA = NNA + 1                                                    POR11770
      IAT = A(I,J,K)                                                   POR11780
C   NNA IS ALSO DEFINED AS:                                            POR11790
C     NNA = (K-1)*JM*IM + (I-1)*JM + J                                 POR11800
      IF ((IAT.EQ.0).OR.(NNA.EQ.NGND)) GOTO 500                        POR11810
C                                                                      POR11820
      NN = NN + 1                                                      POR11830
C                                                                      POR11840
*  FIRST FIND OUT THE CONDUCTANCE OF BRANCHES IN SURROUNDING NODES     POR11850
*                                                                      POR11860
*                                                          5*5*5       POR11870
*  REL                                  1  2  3  4  5      NUMB        POR11880
*  NODE #    5  1              K       .  .  .  .  .      SYS         POR11890
*  DEF'S     |  /               |                         EXAMPLE     POR11900
*            | /                |  DIRECTIONS  6  7  8  9  10  1 PLANE POR11910
*     4 ----CUR ---- 2          |_____ J       .  .  .  .  .         POR11920
*           /|                  /                           J        POR11930
*          / |                 /                 11 12 13 14 15  |----> POR11940
*         /  6               I                   .  .  .  .  .   |    POR11950
*        3                                                       |    POR11960
*                                              16 17 18 19 20   \/    POR11970
```

```
*                                                                       .  .  .  .  .     1     POR11980
*                                                                                               POR11990
*                                                                      21 22 23 24 25           POR12000
*                                                                       .  .  .  .  .           POR12010
*                                                                                               POR12020
*                                                                                               POR12030
      Z1 = 0.0                                                                                  POR12040
      Z2 = 0.0                                                                                  POR12050
      DO 65 JJ=1,6                                                                              POR12060
      GREAL(JJ)=0.0                                                                             POR12070
      GIMAG(JJ)=0.0                                                                             POR12080
   65 CONTINUE                                                                                  POR12090
      GIMAGO =0.0                                                                               POR12100
      GREALO =0.0                                                                               POR12110
*                                                                                               POR12120
      IF (I-1.LT.1) THEN                                                                        POR12130
*           THIS IS A POINT ON THE BOUNDARY.                                                    POR12140
*           HAVE CONTINUOUS BOUNDARY CONDITIONS BEEN REQUESTED?                                 POR12150
*                                                                                               POR12160
            IF (ICONT.EQ.1) THEN                                                                POR12170
                  NEIGH = A(IM,J,K)                                                             POR12180
                  IGND = NNA + IM*(JM-1)                                                        POR12190
                  CALL NOD146(A,1,IAT,NEIGH,IGND,Z1,Z2)                                         POR12200
            END IF                                                                              POR12210
*                                                                                               POR12220
      ELSE                                                                                      POR12230
            NEIGH = A(I-1,J,K)                                                                  POR12240
            IGND = NNA - IM                                                                     POR12250
            CALL NOD146(A,1,IAT,NEIGH,IGND,Z1,Z2)                                               POR12260
      END IF                                                                                    POR12270
*                                                                                               POR12280
C     ASSUMES GND NODE IS #S                                                                    POR12290
C  NODE #2                                                                                      POR12300
   70 IF (J+1.GT.JM)  THEN                                                                      POR12310
            IF (ICONT.EQ.1) THEN                                                                POR12320
                  NEIGH = A(I,1,K)                                                              POR12330
                  IGND = NNA - JM + 1                                                           POR12340
                  CALL NOD235(A,2,IAT,NEIGH,IGND,Z1,Z2)                                         POR12350
            END IF                                                                              POR12360
*                                                                                               POR12370
      ELSE                                                                                      POR12380
            NEIGH = A(I,J+1,K)                                                                  POR12390
            IGND = NNA + 1                                                                      POR12400
            CALL NOD235(A,2,IAT,NEIGH,IGND,Z1,Z2)                                               POR12410
      END IF                                                                                    POR12420
*                                                                                               POR12430
C  NODE #3                                                                                      POR12440
   80 IF (I+1.GT.IM) THEN                                                                       POR12450
            IF (ICONT.EQ.1) THEN                                                                POR12460
```

```
                NEIGH = A(1,J,K)                                        POR12470
                IGND = NNA - IM*(JM-1)                                  POR12480
                CALL NOD235(A,3,IAT,NEIGH,IGND,Z1,Z2)                   POR12490
            END IF                                                      POR12500
*                                                                       POR12510
      ELSE                                                              POR12520
            NEIGH = A(I+1,J,K)                                          POR12530
            IGND = NNA + JM                                             POR12540
            CALL NOD235(A,3,IAT,NEIGH,IGND,Z1,Z2)                       POR12550
      END IF                                                            POR12560
*                                                                       POR12570
C  NODE #4                                                              POR12580
  90  IF (J-1.LT.1) THEN                                                POR12590
            IF (ICONT.EQ.1) THEN                                        POR12600
                NEIGH = A(I,JM,K)                                       POR12610
                IGND = NNA + JM - 1                                     POR12620
                CALL NOD146(A,4,IAT,NEIGH,IGND,Z1,Z2)                   POR12630
            END IF                                                      POR12640
*                                                                       POR12650
      ELSE                                                              POR12660
            NEIGH = A(I,J-1,K)                                          POR12670
            IGND = NNA - 1                                              POR12680
            CALL NOD146(A,4,IAT,NEIGH,IGND,Z1,Z2)                       POR12690
      END IF                                                            POR12700
*                                                                       POR12710
*                                                                       POR12720
*   THERE ARE NO PERIODIC BC'S                                          POR12730
*                                                                       POR12740
C  NODE #5                                                              POR12750
  100 IF (K+1.GT.KM) GOTO 110                                           POR12760
      CALL NOD235(A,5,IAT,A(I,J,K+1),NNA+IM*JM,Z1,Z2)                   POR12770
C  NODE #6                                                              POR12780
  110 IF (K-1.LT.1) GOTO 120                                            POR12790
      CALL NOD146(A,6,IAT,A(I,J,K-1),NNA-IM*JM,Z1,Z2)                   POR12800
C                                                                       POR12810
  120 DO 130 JO= 1,6                                                    POR12820
      GREALO = GREALO - GREAL(JO)                                       POR12830
      GIMAGO = GIMAGO - GIMAG(JO)                                       POR12840
  130 CONTINUE                                                          POR12850
      IF ((Z1.EQ.0.0).AND.(Z2.EQ.0.0)) GOTO 155                         POR12860
      GREALO = GREALO + Z1/(Z1**2+Z2**2)                                POR12870
      GIMAGO = GIMAGO - Z2/(Z1**2+Z2**2)                                POR12880
C                                                                       POR12890
C   ------------------------------------------------------------        POR12900
C                                                                       POR12910
C    NOW BEGIN CONSTRUCTING THE TWO LINES IN THE MATRIX G               POR12920
C         THAT RELATE TO NODE #NN                                       POR12930
  155 N = N + 1                                                         POR12940
*********************************************************               POR12950
C  DETERMINE NON-ZERO COLUMN INDICES:                                   POR12960
```

```
C  ICOL(II) WILL CONTAIN COLUMN INFORMATION ABOUT THE ADJ. NODES          POR12970
C  IF ICOL(II)=0  THEN EITHER: 1-THERE IS NO NODE THERE (AIR)            POR12980
C                           OR: 2-IT BELONGS TO LOWER HALF TRIANGLE      POR12990
C                                  OF THE MATRIX--LEAVE OUT              POR13000
C                           OR: 3-IT IS THE GROUND NODE                  POR13010
C                                                                        POR13020
      DO 160 II=1,6                                                      POR13030
      ICOL(II) = 0                                                       POR13040
  160 CONTINUE                                                           POR13050
      IZERO = 0                                                          POR13060
      ITEMP=0                                                            POR13070
C     HOW MANY ZERO ELEMENTS ARE IN THE REST OF THIS ROW?                POR13080
      IF (J.EQ.JM) GOTO 162                                              POR13090
      DO 175 JJ=J,JM                                                     POR13100
      IF (A(I,JJ,K).EQ.0) ITEMP=ITEMP+1                                  POR13110
  175 CONTINUE                                                           POR13120
C                                                                        POR13130
  162 IF (J+1.GT.JM) GOTO 170                                            POR13140
      IF (A(I,J+1,K).EQ.0) GOTO 170                                      POR13150
      IF (NNA+1.EQ.NGND) NG2 = N                                         POR13160
      IF (NNA+1.EQ.NGND) GOTO 170                                        POR13170
      ICOL(2) = N + 2                                                    POR13180
  170 IF (I+1.GT.IM) GOTO 190                                            POR13190
      IF (NNA+JM.EQ.NGND) NG1 = N                                        POR13200
      IF (NNA+JM.EQ.NGND) GOTO 190                                       POR13210
      IF (A(I+1,J,K).EQ.0) GOTO 190                                      POR13220
C    HOW MANY ELEMENTS BTWN HERE & NODE IN FRONT OF THIS (NON-ZERO)      POR13230
      IZERO=ITEMP                                                        POR13240
      II = I + 1                                                         POR13250
      DO 180 JJ=1,J                                                      POR13260
      IF (A(II,JJ,K).EQ.0) IZERO=IZERO+1                                 POR13270
  180 CONTINUE                                                           POR13280
      ICOL(3) = N + 2*(JM-IZERO)                                         POR13290
  190 IF (K+1.GT.KM) GOTO 220                                            POR13300
      IF ((NNA+IM*JM).EQ.NGND) NG3 = N                                   POR13310
      IF ((NNA+IM*JM).EQ.NGND) GOTO 220                                  POR13320
      IF (A(I,J,K+1).EQ.0) GOTO 220                                      POR13330
C    HOW MANY ZEROS BTWN HERE & NODE ON TOP?                             POR13340
C FIRST COUNT ZEROS ON REST OF CURRENT PLANE                             POR13350
      IZERO=ITEMP                                                        POR13360
C      IF (IZERO.GT.0) WRITE (6,997) N,IZERO                             POR13370
C  997 FORMAT (' LINE # ',I3,' CKPT#1----IZERO=',I3)                     POR13380
      IF (I+1.GT.IM) GOTO 202                                            POR13390
      ISTART=I+1                                                         POR13400
      DO 200 II=ISTART,IM                                                POR13410
      DO 200 JJ=1,JM                                                     POR13420
      IF (A(II,JJ,K).EQ.0) IZERO=IZERO+1                                 POR13430
  200 CONTINUE                                                           POR13440
C      IF (IZERO.GT.0) WRITE (6,998) N,IZERO                             POR13450
C  998 FORMAT (' LINE # ',I3,' CKPT#2----IZERO=',I3)                     POR13460
C NEXT PLANE UP                                                          POR13470
```

```
  202 KK=K+1                                                              POR13480
      IEND=I-1                                                            POR13490
      IF (IEND.LT.1) GOTO 207                                             POR13500
      DO 205 II=1,IEND                                                    POR13510
      DO 205 JJ=1,JM                                                      POR13520
      IF (A(II,JJ,KK).EQ.0) IZERO=IZERO+1                                 POR13530
  205 CONTINUE                                                            POR13540
C       IF (IZERO.GT.0) WRITE (6,999) N,IZERO                             POR13550
C  999 FORMAT (' LINE # ',I3,' CKPT#3----IZERO=',I3)                      POR13560
  207 DO 210 JJ=1,J                                                       POR13570
      IF (A(I,JJ,KK).EQ.0) IZERO=IZERO+1                                  POR13580
  210 CONTINUE                                                            POR13590
C       IF (IZERO.GT.0) WRITE (6,996) N,IZERO                             POR13600
C  996 FORMAT (' LINE # ',I3,' CKPT#4----IZERO=',I3)                      POR13610
C                                                                         POR13620
      ICOL(5) = N + 2*(IM*JM-IZERO)                                       POR13630
C                                                                         POR13640
C*****************                                                        POR13650
C  PERIODIC BOUNDARY CONDITIONS???                                        POR13660
C                                                                         POR13670
  220  IF (ICONT.NE.1) GOTO 250                                           POR13680
*   CONTINUOUS BOUNDARY CONDITIONS -                                      POR13690
*   IN THIS CASE, NODES IN RELATIVE POS #1 AND #4 ARE IN UPR RH TRIANGLE  POR13700
*   OF THE MATRIX.                                                        POR13710
*    E.G. FOR SIZE = 11 NODE #1 HAS AN ADJACENT NODE TO THE LEFT WITH     POR13720
*   MATRIX ADDRESS (1,11).                                                POR13730
*                                                                         POR13740
      IF (J-1.LT.1) THEN                                                  POR13750
*                          WE ARE ON LH EDGE                            POR13760
            IF (A(I,JM,K).NE.0) THEN                                      POR13770
                  IF (NNA+JM-1.EQ.NGND) THEN                              POR13780
                        NG4 = N                                           POR13790
                  ELSE                                                    POR13800
                        IZERO = ITEMP                                     POR13810
*                                                                         POR13820
                        ICOL(4) = N + 2 * (JM-IZERO-1)                    POR13830
                  END IF                                                  POR13840
             END IF                                                       POR13850
      END IF                                                              POR13860
*                                                                         POR13870
      IF  (I-1.LT.1) THEN                                                 POR13880
               IF (A(I-1,J,K).NE.0) THEN                                  POR13890
                        IF (NNA+JM*(IM-1).EQ.NGND) THEN                   POR13900
                              NG5 = N                                     POR13910
                        ELSE                                              POR13920
                                                                          POR13930
                              IZERO = ITEMP                               POR13940
                              DO 225 II = I+1,IM                          POR13950
                              DO 225 JJ = 1,JM                            POR13960
```

```
                                    IF (II.EQ.IM.AND.JJ.GE.J) GOTO 225     POR13970
                                    IF (A(II,JJ,K).EQ.0) IZERO = IZERO + 1POR13980
  225                             CONTINUE                                 POR13990
                                  ICOL(1) = N. + 2 * (JM*(IM-1)-IZERO)       POR14000
                            END IF                                         POR14010
                                                                           POR14020
                     END IF                                                POR14030
           END IF                                                          POR14040
C                                                                          POR14050
* BECAUSE OF PERIODIC B.C. COLUMN INDICES MAY BE OUT OF ORDER              POR14060
* SORT SO THAT THEY ARE IN ASCENDING ORDER                                 POR14070
*                                                                          POR14080
      DO 240 LL = 1,5                                                      POR14090
         DO 240 IN = LL+1,6                                                POR14100
              IF (ICOL(LL).GT.ICOL(IN)) THEN                               POR14110
*             SWAP                                                         POR14120
                   IHCOL = ICOL(LL)                                        POR14130
                   TEMPGR = GREAL(LL)                                      POR14140
                   TEMPGI = GIMAG(LL)                                      POR14150
                   ICOL(LL) = ICOL(IN)                                     POR14160
                   GREAL(LL) = GREAL(IN)                                   POR14170
                   GIMAG(LL) = GIMAG(IN)                                   POR14180
                   ICOL(IN) = IHCOL                                        POR14190
                   GREAL(IN) = TEMPGR                                      POR14200
                   GIMAG(IN) = TEMPGI                                      POR14210
              END IF                                                       POR14220
  240  CONTINUE                                                           POR14230
C NOW START THE 1ST ROW OF REAL-IMAG PAIR                                  POR14240
  250 NT=NT+1                                                              POR14250
C POINTER TO FIRST ELEMENT IN ROW                                          POR14260
      IA(N) = NT                                                           POR14270
C COLUMN INDEX                                                             POR14280
      JA(NT) = N                                                           POR14290
C  NOW MATRIX ITSELF                                                       POR14300
      G(NT) = GREALO                                                       POR14310
      IF (GIMAGO.EQ.0.0) GOTO 380                                          POR14320
C  INCREMENT POSITION IN G MATRIX                                          POR14330
      NT =  NT + 1                                                         POR14340
      JA(NT) = N + 1                                                       POR14350
      G(NT) = -GIMAGO                                                      POR14360
C                                                                          POR14370
  380 DO 390 LL=1,6                                                        POR14380
      IF (ICOL(LL).EQ.0) GOTO 390                                          POR14390
      NT = NT + 1                                                          POR14400
      JA(NT) = ICOL(LL)                                                    POR14410
      G(NT) = GREAL(LL)                                                    POR14420
      IF (GIMAG(LL).EQ.0.0) GOTO 390                                       POR14430
      NT = NT + 1                                                          POR14440
      JA(NT) = ICOL(LL)+1                                                  POR14450
```

```
      G(NT) = -GIMAG(LL)                                                POR14460
  390 CONTINUE                                                          POR14470
C                                                                       POR14480
C  DO NEXT LINE (IMAG LINE)                                             POR14490
C                                                                       POR14500
      N = N +1                                                          POR14510
      NT = NT + 1                                                       POR14520
      IA(N) = NT                                                        POR14530
      JA(NT) = N                                                        POR14540
      G(NT) = -GREALO                                                   POR14550
C                                                                       POR14560
      DO 400 LL=1,6                                                     POR14570
      IF (ICOL(LL).EQ.0) GOTO 400                                       POR14580
      IF (GIMAG(LL).EQ.0.0) GOTO 395                                    POR14590
      NT = NT + 1                                                       POR14600
      JA(NT) = ICOL(LL)                                                 POR14610
      G(NT) = -GIMAG(LL)                                                POR14620
  395 NT = NT +  1                                                      POR14630
      JA(NT) = ICOL(LL) + 1                                             POR14640
      G(NT) = -GREAL(LL)                                                POR14650
  400 CONTINUE                                                          POR14660
  500 CONTINUE                                                          POR14670
C    MARK LAST ELEMENT                                                  POR14680
      IA(N+1) = NT + 1                                                  POR14690
*     WRITE (7,499) NG1,NG2,NG3,NG4,NG5                                 POR14700
  499 FORMAT (' ADJACENT GND NODE MATRIX ADDRESSES: ',316)              POR14710
      WRITE (6,505) NNA,NN,N,NT                                         POR14720
  505 FORMAT (' TOTAL NODES= ',I5,'  NONZERO NODES= ',I5,               POR14730
     1'  # OF EQU-S= ',I8,'  PACKED ELEMENTS= ',I8)                     POR14740
************    BEGIN CONDITIONAL SECTION   ************               POR14750
C   NOW LIST OFF MATRIX                                                 POR14760
*     L=1                                                               POR14770
*     WRITE (6,510) L                                                   POR14780
* 510 FORMAT (' ***************',//,' ROW ',I4)                        POR14790
*     DO 580 I=1,NT                                                     POR14800
*     WRITE (6,560) JA(I),G(I)                                          POR14810
* 560 FORMAT (' ELEMENT #',I4,'  =  ',1PE13.4)                          POR14820
*     IF (I+1.LT.IA(L+1)) GOTO 580                                      POR14830
*     L = L + 1                                                         POR14840
*     WRITE (6,510) L                                                   POR14850
* 580 CONTINUE                                                          POR14860
*******   END CONDITIONAL SECTION *****************                    POR14870
C                                                                       POR14880
C    SET UP RHS = INPUT CURRENTS                                        POR14890
      B(NCUR) = CUR(1)                                                  POR14900
      B(NCUR+1) = -CUR(2)                                               POR14910
      RETURN                                                            POR14920
      END                                                               POR14930
*-------------------                                                    POR14940
C                                                                       POR14950
```

```
      SUBROUTINE NOD146(A,IPOS,IAT,NEIGH,IGND,Z1,Z2)                    POR14960
C                                                                       POR14970
C     SR TO EVALUATE CONDUCTANCE ELEMENTS AT NODES 1,4,6                POR14980
C                                                                       POR14990
      INTEGER*4 A(IM,JM,KM)                                             POR15000
      COMMON/LATICE/IM,JM,KM,KMSTOP                                     POR15010
      COMMON/ELEC/Z(6,2),CUR(2),NCUR,NGND,REALI,RMAGI                   POR15020
      COMMON/GTEMP/ NG1,NG2,NG3,NG4,NG5,ICOL(6),GREAL(6),GIMAG(6)       POR15030
C                                                                       POR15040
      IAGND=A(IM,JM,KM)                                                 POR15050
      G1 = Z(IAT,1)                                                     POR15060
      G2 = Z(IAT,2)                                                     POR15070
C   NOTHING THERE AT POS IPOS                                           POR15080
      IF (NEIGH.EQ.0) GOTO 100                                          POR15090
C   GND INTERFACE?                                                      POR15100
      IF (IGND.EQ.NGND) GOTO 10                                         POR15110
C   SEMIC INTFC?                                                        POR15120
      IF (IAT+NEIGH.EQ.3) GOTO 20                                       POR15130
C NORMAL INTFC                                                          POR15140
      GOTO 30                                                           POR15150
C       GND NODE                                                        POR15160
  10  Z1  = G1 + Z(IAGND,1)                                             POR15170
      Z2  = G2 + Z(IAGND,2)                                             POR15180
      IF (IAT+NEIGH.NE.3) GOTO 100                                      POR15190
      Z1  = G1 + Z(NEIGH+2,1)                                           POR15200
      Z2  = G2 + Z(NEIGH+2,2)                                           POR15210
      GOTO 100                                                          POR15220
C     S.C. INTFC                                                        POR15230
  20  A1 = Z(NEIGH+2,1) + G1                                            POR15240
      A2 = Z(NEIGH+2,2) + G2                                            POR15250
      GOTO 50                                                           POR15260
C     NORMAL  INTFC                                                     POR15270
  30  A1 = Z(NEIGH,1) + G1                                              POR15280
      A2 = Z(NEIGH,2) + G2                                              POR15290
  50  GREAL(IPOS) = -A1/(A1**2+A2**2)                                   POR15300
      GIMAG(IPOS) = A2/(A1**2+A2**2)                                    POR15310
  100 RETURN                                                            POR15320
      END                                                               POR15330
*------------------                                                     POR15340
C                                                                       POR15350
C                                                                       POR15360
      SUBROUTINE NOD235(A,IPOS,IAT,NEIGH,IGND,Z1,Z2)                    POR15370
C                                                                       POR15380
C SR TO EVALUATE G MATRIX AT NODES 2,3,5                                POR15390
C                                                                       POR15400
      INTEGER*4 A(IM,JM,KM)                                             POR15410
      COMMON/LATICE/IM,JM,KM,KMSTOP                                     POR15420
      COMMON/ELEC/Z(6,2),CUR(2),NCUR,NGND,REALI,RMAGI                   POR15430
      COMMON/GTEMP/ NG1,NG2,NG3,NG4,NG5,ICOL(6),GREAL(6),GIMAG(6)       POR15440
C                                                                       POR15450
```

```
      IAGND=A(IM,JM,KM)                                                 POR15460
      A1=0.0                                                            POR15470
      A2=0.0                                                            POR15480
      G1=Z(IAT,1)                                                       POR15490
      G2=Z(IAT,2)                                                       POR15500
C  NOTHING THERE                                                        POR15510
      IF (NEIGH.EQ.0) GOTO 100                                          POR15520
C  GND INTFC?                                                           POR15530
      IF (IGND.EQ.NGND) GOTO 10                                         POR15540
C S.C. INTERFACE?                                                       POR15550
      IF (IAT+NEIGH.EQ.3) GOTO 20                                       POR15560
C  NORMAL                                                               POR15570
      GOTO 30                                                           POR15580
C                                                                       POR15590
C     GND NODE                                                          POR15600
  10  Z1 = G1 + Z(IAGND,1)                                              POR15610
      Z2 = G2 + Z(IAGND,2)                                              POR15620
      IF (IAT+NEIGH.NE.3) GOTO 100                                      POR15630
      Z1 = Z(IAT+2,1) + Z(IAGND,1)                                      POR15640
      Z2 = Z(IAT+2,2) + Z(IAGND,2)                                      POR15650
      GOTO 100                                                          POR15660
C  SC INTFC                                                             POR15670
  20  G1 = Z(IAT+2,1)                                                   POR15680
      G2 = Z(IAT+2,2)                                                   POR15690
  30  A1 = Z(NEIGH,1) + G1                                              POR15700
      A2 = Z(NEIGH,2) + G2                                              POR15710
      GREAL(IPOS) = -A1/(A1**2+A2**2)                                   POR15720
      GIMAG(IPOS) = A2/ (A1**2+A2**2)                                   POR15730
  100 CONTINUE                                                          POR15740
      RETURN                                                            POR15750
      END                                                               POR15760
*-------------------                                                    POR15770
C                                                                       POR15780
      SUBROUTINE CURCAL (A,G,MODEL)                                     POR15790
C                                                                       POR15800
C  SR TO CALCULATE 'CURRENT OUT' VALUE                                  POR15810
C   TO CHECK RESULT                                                     POR15820
C                                                                       POR15830
      INTEGER*4 A(IM,JM,KM)                                             POR15840
      COMMON /DIMS/IDIM1,IDIM2,IDIM3,IDIM4                              POR15850
      COMMON/LATICE/ IM,JM,KM,KMSTOP                                    POR15860
      COMMON/ELEC/Z(6,2),CUR(2),NCUR,NGND,REAL1,RMAG1                   POR15870
      COMMON/GTEMP/ NG1,NG2,NG3,NG4,NG5,ICOL(6),DUMMY(12)               POR15880
      DIMENSION G(IDIM3)                                                POR15890
*    DEFINE NODE #'S:     1-AT (IM-1)                                  POR15900
*           1             2-AT (JM-1)                                  POR15910
*          /              3- AT (KM-1)                                 POR15920
*  2 ---GND--- 4          4- AT JM+1 (FOR CONT BOUND COND.)            POR15930
*       /|                5- AT IM+1 (FOR COND BOUND COND.)            POR15940
*      / |                                                             POR15950
```

```
*      5  3                                                              POR15960
*                                                                       POR15970
*  CONTINUOUS BOUND. COND. ?                                            POR15980
      ICONT = 0                                                         POR15990
      IF (MODEL.GT.50) ICONT = 1                                        POR16000
*                                                                       POR16010
      IAGND = A(IM,JM,KM)                                               POR16020
C     WRITE (7,10) NG1,NG2,NG3,NG4,NG5                                  POR16030
C 10  FORMAT (' MATRIX ADDRESS OF ADJACENT GND NODES= ',5I7)            POR16040
      IA1 = A(IM-1,JM,KM)                                               POR16050
      IA2 = A(IM,JM-1,KM)                                               POR16060
      IA3 = A(IM,JM,KM-1)                                               POR16070
      IF (ICONT.EQ.1) THEN                                              POR16080
            IA4 = A(IM,1,KM)                                            POR16090
            IA5 = A(1,JM,KM)                                            POR16100
      ELSE                                                              POR16110
            IA4 = 0                                                     POR16120
            IA5 = 0                                                     POR16130
      END IF                                                            POR16140
*                                                                       POR16150
      REALI = 0.0                                                       POR16160
      RMAGI = 0.0                                                       POR16170
C                                                                       POR16180
C                                                                       POR16190
C   ASSUME GND NODE = METAL                                             POR16200
      IF (IA1.EQ.0) GOTO 50                                             POR16210
      V1R = G(NG1)                                                      POR16220
      V1I = G(NG1+1)                                                    POR16230
      IF (IA1+IAGND.EQ.3) IA1 = IA1+2                                   POR16240
      ZR = Z(IA1,1) + Z(IAGND,1)                                        POR16250
      ZI = Z(IA1,2)                                                     POR16260
      DIV = ZR*ZR + ZI*ZI                                               POR16270
      REALI = (V1R*ZR + V1I*ZI)/DIV                                     POR16280
      RMAGI = (V1I*ZR - V1R*ZI)/DIV                                     POR16290
   50 IF (IA2.EQ.0) GOTO 100                                            POR16300
      V2R = G(NG2)                                                      POR16310
      V2I = G(NG2+1)                                                    POR16320
      IF (IA2+IAGND.EQ.3) IA2 = IA2 + 1                                 POR16330
      ZR = Z(IA2,1) + Z(IAGND,1)                                        POR16340
      ZI = Z(IA2,2)                                                     POR16350
      DIV = ZR*ZR + ZI*ZI                                               POR16360
      REALI = REALI + (V2R*ZR + V2I*ZI)/DIV                             POR16370
      RMAGI = RMAGI + (V2I*ZR - V2R*ZI)/DIV                             POR16380
  100 IF (IA3.EQ.0) GOTO 150                                            POR16390
      V3R = G(NG3)                                                      POR16400
      V3I = G(NG3+1)                                                    POR16410
      IF (IA3+IAGND.EQ.3) IA3 = IA3 +2                                  POR16420
      ZR = Z(IA3,1) + Z(IAGND,1)                                        POR16430
      ZI = Z(IA3,2)                                                     POR16440
```

```
      DIV = ZR*ZR + ZI*ZI                                                POR16450
      REALI = REALI + (V3R*ZR + V3I*ZI)/DIV                              POR16460
      RMAGI = RMAGI + (V3I*ZR - V3R*ZI)/DIV                              POR16470
  150 IF (IA4.EQ.0) GOTO 200                                             POR16480
      V4R = G(NG4)                                                       POR16490
      V4I = G(NG4+1)                                                     POR16500
      IF (IA4+IAGND.EQ.3) IA4 = IA4+2                                    POR16510
      ZR = Z(IA4,1) + Z(IAGND,1)                                         POR16520
      ZI = Z(IA4,2)                                                      POR16530
      DIV = ZR*ZR + ZI*ZI                                                POR16540
      REALI = REALI + (V4R*ZR + V4I*ZI)/DIV                              POR16550
      RMAGI = RMAGI + (V4I*ZR - V4R*ZI)/DIV                              POR16560
  200 IF (IA5.EQ.0) GOTO 250                                             POR16570
      V5R = G(NG5)                                                       POR16580
      V5I = G(NG5+1)                                                     POR16590
      IF (IA5+IAGND.EQ.3) IA5 = IA5+2                                    POR16600
      ZR = Z(IA5,1) + Z(IAGND,1)                                         POR16610
      ZI = Z(IA5,2)                                                      POR16620
      DIV = ZR*ZR + ZI*ZI                                                POR16630
      REALI = REALI + (V5R*ZR + V5I*ZI)/DIV                              POR16640
      RMAGI = RMAGI + (V5I*ZR - V5R*ZI)/DIV                              POR16650
  250 CONTINUE                                                           POR16660
C     WRITE (7,110) V1R,V1I,V2R,V2I,V3R,V3I,V4R,V4I,V5R,V5I              POR16670
C 110 FORMAT (' V ON NODES =  ',//,                                      POR16680
C    C5(' ',1P,E12.2,' +(J*) ',E12.2,/))                                 POR16690
      RETURN                                                             POR16700
      END                                                                POR16710
*-------------------                                                     POR16720
*********************************************************************** POR16730
*                                                                     * POR16740
*      STORED ON FILE: CURPRF  FORTRAN  12/5/84                       * POR16750
*                                                                     * POR16760
*      GENERATES A CURRENT PROFILE OF THE LATTICE                     * POR16770
*********************************************************************** POR16780
      SUBROUTINE CURPRF (A,LBLA,G)                                       POR16790
                                                                         POR16800
                                                                         POR16810
                                                                         POR16820
      REAL*4 NRMCUR                                                      POR16830
      CHARACTER*1 LBL(0:3),CHOLD1(0:20),CHOLD2(0:20)                     POR16840
      COMPLEX*8 CI                                                       POR16850
      INTEGER*4 A(IM,JM,KM),LBLA(IM,JM,KM)                              POR16860
      DATA LBL /'A','M','E','X'/                                         POR16870
      COMMON /LATICE/IM,JM,KM,KMSTOP                                     POR16880
      COMMON /DIMS/IDIM1,IDIM2,IDIM3,IDIM4                               POR16890
      COMMON /ELEC/Z(6,2),CUR(2),NCUR,NGND,REALI,RMAGI                   POR16900
      DIMENSION G(IDIM3),RTHOLD(20),FHOLD(20)                            POR16910
                                                                         POR16920
                                                                         POR16930
*                                                                        POR16940
```

```
*   THIS SUBROUTINE OUTPUTS THE CURRENT DISTRIBUTION FROM SELECTED      POR16950
*   NODES IN THE LATTICE.                                              POR16960
*                                                                      POR16970
*   INPUT: A (LATTICE CHARACTER) LBLA (LATTICE NUMBERING SCHEME)       POR16980
*   THE ACTUAL NUMBER OF LINKS IN THE NETWORK ARE                      POR16990
*   NLINKS = (JM-1)*IM*KM + (IM-1)*JM*KM + (KM-1)*IM*JM                POR17000
*          = 3*IM**3-3*IM**2 IF   IM=JM=KM                             POR17010
*                                                                      POR17020
*     THE RETURNED CURRENTS ARE IN 3-TUPLES WHERE FOR                  POR17030
*     EACH NODE AS FOLLOWS:                3|                        POR17040
*     NODE 0: RIGHT, FORWARD, UP            |       1                 POR17050
*                                           0--------                POR17060
*                                          /                         POR17070
*                                         / 2                        POR17080
*                                                                    POR17090
  50   FORMAT (' ',I4,',',I4,'  CUR (%)= ',2P,F7.3,3X,F7.3,-2P,F7.3,3X, POR17100
     1       F7.3)                                                    POR17110
      NRMCUR = SQRT (CUR(1)*CUR(1)+CUR(2)*CUR(2))                     POR17120
*********************************************************************POR17130
*  THIS IS A COUNTER THAT WILL COUNT THE NUMBER OF                  *  POR17140
*  SEMIC-ELEC INTERFACES THAT ARE PASING 'APRECIABLE'               *  POR17150
*  CURRENT. THIS IS THE BEST STRUCTURAL MEASURE OF THE  (?)         *  POR17160
*  EFFECTIVE CAPACITANCE.                                           *  POR17170
      INTCNT = 0                                                      POR17180
      NOCNT  = 0                                                      POR17190
*********************************************************************POR17200
*  DEFINE THE THRESHOLD LEVEL - ABOVE THIS % WE WILL CONSIDER       *  POR17210
*  THE INTERFACE TO BE PASSING CURRENT.                             *  POR17220
      THRESH = .2                                                     POR17230
*********************************************************************POR17240
      DO 400 K=1,KM                                                   POR17250
         DO 300 I=1,IM                                                POR17260
             JPNTR = 0                                                POR17270
             DO 100 J=1,JM                                            POR17280
                NODE1 = LBLA(I,J,K)                                   POR17290
                IAT1 = A(I,J,K)                                       POR17300
                JPNTR = JPNTR + 1                                     POR17310
                IF (J.LT.JM) THEN                                     POR17320
                        NODE2 = LBLA(I,J+1,K)                         POR17330
                        IAT2 = A(I,J+1,K)                             POR17340
                        IF (IAT2.NE.0.AND.IAT1.NE.0) THEN             POR17350
                            CALL CURLNK(G,NODE1,NODE2,IAT1,IAT2,CI)   POR17360
                            PCTCUR = 100.0 * ABS(CI) / NRMCUR         POR17370
                            IF (IAT1+IAT2.EQ.3.AND.PCTCUR.GT.THRESH)THENPOR17380
                                INTCNT = INTCNT + 1                   POR17390
                            ELSE IF (IAT1+IAT2.EQ.3) THEN             POR17400
                                NOCNT=NOCNT+1                         POR17410
                            ENDIF                                     POR17420
                        ELSE                                          POR17430
```

```
                              PCTCUR = 0.0                                      POR17440
                          ENDIF                                                 POR17450
*                         WRITE (7,50) NODE1,NODE2,CI                           POR17460
                          CHOLD1(0) =  LBL(IAT1)                                POR17470
                          RTHOLD(JPNTR) = PCTCUR                                POR17480
                          CHOLD2(JPNTR) = LBL(IAT2)                             POR17490
*                         WRITE (7,51) RTHOLD,CHOLD2(JPNTR)                     POR17500
* 51                      FORMAT (' ', F7.3,A1)                                 POR17510
                    ENDIF                                                       POR17520
                    IF (I.LT.IM) THEN                                           POR17530
                          NODE2 = LBLA(I+1,J,K)                                 POR17540
                          IAT2 = A(I+1,J,K)                                     POR17550
                          IF (IAT2.NE.0.AND.IAT1.NE.0) THEN                     POR17560
                              CALL CURLNK(G,NODE1,NODE2,IAT1,IAT2,CI)           POR17570
                              PCTCUR = 100.0 * ABS(CI) / NRMCUR                 POR17580
                              IF (IAT1+IAT2.EQ.3.AND.PCTCUR.GT.THRESH)THENPOR17590
                                 INTCNT = INTCNT + 1                            POR17600
                              ELSE IF (IAT1+IAT2.EQ.3) THEN                     POR17610
                                 NOCNT=NOCNT+1                                  POR17620
                              ENDIF                                             POR17630
                          ELSE                                                  POR17640
                                PCTCUR = 0.0                                    POR17650
                          ENDIF                                                 POR17660
*                         WRITE (7,50) NODE1,NODE2,CI                           POR17670
                          FHOLD(JPNTR) = PCTCUR                                 POR17680
*                         WRITE (7,51) FHOLD                                    POR17690
                    ENDIF                                                       POR17700
                                                                                POR17710
  100               CONTINUE                                                    POR17720
                                                                                POR17730
                    WRITE(6,110) CHOLD1(0),(RTHOLD(IX),CHOLD2(IX),IX=1,JM-1)    POR17740
                    IF (I.LT.IM) WRITE (6,120) (FHOLD(IX), IX=1,JM)             POR17750
  110               FORMAT (' ',4X,A1,1X,20(F7.3,1X,A1,1X))                     POR17760
  120               FORMAT (' ',20(F7.3,3X))                                    POR17770
  300           CONTINUE                                                        POR17780
                WRITE (6,310) K                                                 POR17790
  310           FORMAT ('1 K PLANE = ',I2//)                                    POR17800
*           FINISHED A WHOLE I-J PLANE, NOW PRINT THE VERTICAL LINKS            POR17810
*           BETWEEN THE CURRENT K PLANE AND K+1:                                POR17820
                TOTPCT = 0.0                                                    POR17830
                IF (K.LT.KM) THEN                                               POR17840
                DO 350 I=1,IM                                                   POR17850
                    JPNTR = 0                                                   POR17860
                    DO 325 J=1,JM                                               POR17870
                       NODE1 = LBLA(I,J,K)                                      POR17880
                       IAT1 = A(I,J,K)                                          POR17890
                       JPNTR = JPNTR + 1                                        POR17900
                       NODE2 = LBLA(I,J,K+1)                                    POR17910
                       IAT2 = A(I,J,K+1)                                        POR17920
```

```
                IF (IAT2.NE.0.AND.IAT1.NE.0) THEN                       POR17930
                      CALL CURLNK(G,NODE1,NODE2,IAT1,IAT2,CI)           POR17940
                      PCTCUR = 100.0 * ABS(CI) / NRMCUR                 POR17950
     0                IF (IAT1+IAT2.EQ.3.AND.PCTCUR.GT.THRESH) THEN     POR17960
                         INTCNT = INTCNT + 1                            POR17970
                      ELSE IF (IAT1+IAT2.EQ.3) THEN                     POR17980
                          NOCNT=NOCNT+1                                 POR17990
                      ENDIF                                             POR18000
                 ELSE                                                   POR18010
                      PCTCUR = 0.0                                      POR18020
                 ENDIF                                                  POR18030
                 TOTPCT = TOTPCT + PCTCUR                               POR18040
                       CHOLD1(JPNTR) = LBL(IAT1)                        POR18050
                       FHOLD(JPNTR) = PCTCUR                            POR18060
                       CHOLD2(JPNTR) = LBL(IAT2)                        POR18070
*                      WRITE (6,50) NODE1,NODE2,CI                      POR18080
  325          CONTINUE                                                POR18090
               WRITE (6,326)(CHOLD1(IX),IX=1,JM)                       POR18100
               WRITE (6,327)(FHOLD(IX),IX=1,JM)                        POR18110
               WRITE (6,328)(CHOLD2(IX),IX=1,JM)                       POR18120
  326          FORMAT (' ',4X,20(A1,9X))                               POR18130
  327          FORMAT (' ',20(F7.3,3X))                                POR18140
  328          FORMAT (' ',20(A1,9X))                                  POR18150
  350       CONTINUE                                                   POR18160
            WRITE (6,360) TOTPCT                                       POR18170
  360       FORMAT ('  TOTAL CURRENT (%) = ',F7.3//)                   POR18180
            ENDIF                                                      POR18190
  400 CONTINUE                                                         POR18200
      WRITE (6,410) INTCNT,NOCNT,INTCNT+NOCNT                          POR18210
  410 FORMAT (' ACTIVE INTERFACES = ',I6,' PASSIVE = ',I6,' TOTAL= ',I6)POR18220
        RETURN                                                         POR18230
        END                                                            POR18240
*------------------                                                    POR18250
*                                                                      POR18260
*                                                                      POR18270
      SUBROUTINE CURLNK(G,NODE1,NODE2,IAT1,IAT2,CI)                    POR18280
*        SUBROUTINE TO CALCULATE THE CURRENT IN A GIVEN CIRCUIT BRANCH POR18290
*                                                                      POR18300
                                                                       POR18310
      COMMON /LATICE/IM,JM,KM,KMSTOP                                   POR18320
      COMMON /DIMS/IDIM1,IDIM2,IDIM3,IDIM4                             POR18330
      COMMON /ELEC/Z(6,2),CUR(2),NCUR,NGND,REALI,RMAGI                 POR18340
      DIMENSION G(IDIM3)                                               POR18350
                                                                       POR18360
                                                                       POR18370
      REAL*8 V1R,V2R,V1I,V2I,ZR,ZI                                     POR18380
      COMPLEX*8 CI                                                     POR18390
      COMPLEX*16 CZ,CV1,CV2,CV0                                        POR18400
                                                                       POR18410
```

```
      IATT1 = IAT1                                                      POR18420
      INDX1 = 2*NODE1 - 1                                               POR18430
      INDX2 = 2*NODE2 - 1                                               POR18440
      V1R = G(INDX1)                                                    POR18450
      V1I = G(INDX1+1)                                                  POR18460
      V2R = G(INDX2)                                                    POR18470
      V2I = G(INDX2+1)                                                  POR18480
                                                                        POR18490
      CV1 = DCMPLX(V1R,V1I)                                             POR18500
      CV2 = DCMPLX(V2R,V2I)                                             POR18510
      CV0 = CV2 - CV1                                                   POR18520
      IF (IATT1+IAT2.EQ.3) IATT1 = IATT1+2                              POR18530
      ZR = Z(IATT1,1) + Z(IAT2,1)                                       POR18540
      ZI = Z(IATT1,2) + Z(IAT2,2)                                       POR18550
      CZ = DCMPLX(ZR,ZI)                                                POR18560
*                                                                       POR18570
*        I = V /Z IN COMPLEX ARITHMETIC                                 POR18580
      CI = CV0 / CZ                                                     POR18590
 50   FORMAT (' ',I4,',',I4,'  CUR (%)= ',1P,E13.3,3X,E13.3,E13.3)     POR18600
      RETURN                                                            POR18610
      END                                                               POR18620
*------------------                                                     POR18630
*  SUBROUTINE TO CALCULATE LATTICE CORRELATIONS                         POR18640
*  TO TRY AND QUANTIFY THE PROPERTIES OF                                POR18650
*  THE GEOMETRICAL CHARACTERISTICS OF EACH POROUS ELECTRODE             POR18660
*   -- DENSITY-DENSITY CORRELATION CALCULATION ---                      POR18670
*   -- OF THE SURFACE SHELL                  ---                       POR18680
*         WRITTEN ON 10/30/85 IN VS FORTRAN77                           POR18690
      SUBROUTINE SHLCOR (A,LBLA,LPLANE)                                 POR18700
*                                                                       POR18710
      PARAMETER (MAXR=100)                                              POR18720
      INTEGER    A(IM,JM,KM),LBLA(IM,JM,KM)                             POR18730
      COMMON /LATICE/ IM,JM,KM,KMSTOP                                   POR18740
      DIMENSION COR(MAXR),R(MAXR)                                       POR18750
*                                                                       POR18760
* SET UP PARAMETERS FOR NEARN SEARCH                                    POR18770
            ISTAY = 1                                                   POR18780
            ILOOK = 2                                                   POR18790
            NPART = 0                                                   POR18800
            N = 0                                                       POR18810
            DO 100 K=1,KM                                               POR18820
            DO 100 J=1,JM                                               POR18830
            DO 100 I=1,IM                                               POR18840
                IFND=0                                                  POR18850
                CALL NEARN(A,I,J,K,ISTAY,ILOOK,IFND,KFND)               POR18860
                IF (IFND.EQ.1 .AND. K.LT.KM) THEN                       POR18870
*                   WE WANT TO LEAVE A() THE WAY THAT IT WAS            POR18880
                    A(I,J,K) = 1                                        POR18890
                    LBLA(I,J,K) = 1                                     POR18900
```

```
                    NPART = NPART + 1                                   POR18910
                ELSE                                                    POR18920
                    LBLA(I,J,K) = 0                                     POR18930
                ENDIF                                                   POR18940
  100       CONTINUE                                                    POR18950
*                                                                       POR18960
      ISHOW = 0                                                         POR18970
      CALL DISPLA(LBLA,ISHOW,MODEL)                                     POR18980
      WRITE (6,*) 'SURFACE CORRELATIONS:'                               POR18990
      WRITE (7,*) 'SURFACE CORRELATIONS:'                               POR19000
      WRITE (6,*) NPART                                                 POR19010
      ROOT2 = SQRT(2.0)                                                 POR19020
      WRITE (7,*) NPART                                                 POR19030
      DO 500 IR=1,IM-3                                                  POR19040
*   CALCULATE C(R) FOR EACH R:                                          POR19050
        N = N+1                                                         POR19060
        R(N) = IR                                                       POR19070
        COR(N) = 0.0                                                    POR19080
        DO 200 K=1,KM                                                   POR19090
        DO 200 J=1,JM                                                   POR19100
        DO 200 I=1,IM                                                   POR19110
*     AVERAGE C(R) OVER THE SIX NEAREST NEIGHBOR CUBIC DIRECTIONS       POR19120
          NLBLA = LBLA (I,J,K)                                          POR19130
            IF (I+IR.LT.IM)                                             POR19140
     1       COR(N) = COR(N) + NLBLA * LBLA(I+IR,J,K)                   POR19150
            IF (I-IR.GT.1)                                              POR19160
     1       COR(N) = COR(N) + NLBLA * LBLA(I-IR,J,K)                   POR19170
            IF (J+IR.LT.JM)                                             POR19180
     1       COR(N) = COR(N) + NLBLA * LBLA(I,J+IR,K)                   POR19190
            IF (J-IR.GT.1)                                              POR19200
     1       COR(N) = COR(N) + NLBLA * LBLA(I,J-IR,K)                   POR19210
            IF (K+IR.LT.KM)                                             POR19220
     1       COR(N) = COR(N) + NLBLA * LBLA(I,J,K+IR)                   POR19230
*  K=1 PLANE ALWAYS CONTAINS ELECTROLYTE                                POR19240
            IF (K-IR.GT.1)                                              POR19250
     1       COR(N) = COR(N) + NLBLA * LBLA(I,J,K-IR)                   POR19260
  200   CONTINUE                                                        POR19270
        COR(N) = COR(N) / (NPART*6.)                                    POR19280
        WRITE (6,*) R(N),COR(N)                                         POR19290
        WRITE (7,*) R(N),COR(N)                                         POR19300
* ----- NOW CALCULATE 2ND NEAREST NEIGHBOR PAIR CORRELATIONS           POR19310
*   CALCULATE C(R) FOR EACH R:                                          POR19320
        N = N+1                                                         POR19330
        R(N) = IR*ROOT2                                                 POR19340
        COR(N) = 0.0                                                    POR19350
        DO 300 K=1,KM                                                   POR19360
        DO 300 J=1,JM                                                   POR19370
        DO 300 I=1,IM                                                   POR19380
        NLBLA = LBLA(I,J,K)                                             POR19390
*     AVERAGE C(R) OVER THE 12 2ND NEAREST NEIGHBOR CUBIC DIRECTIONS    POR19400
```

```
            IF (J+IR.LT.JM) THEN                                          POR19410
               IF (I+IR.LT.IM)                                            POR19420
     1          COR(N) = COR(N) + NLBLA * LBLA(I+IR,J+IR,K)               POR19430
               IF (I-IR.GT.0)                                             POR19440
     1          COR(N) = COR(N) + NLBLA * LBLA(I-IR,J+IR,K)               POR19450
               IF (K+IR.LT.KM)                                            POR19460
     1          COR(N) = COR(N) + NLBLA * LBLA(I,J+IR,K+IR)               POR19470
               IF (K-IR.GT.0)                                             POR19480
     1          COR(N) = COR(N) + NLBLA * LBLA(I,J+IR,K-IR)               POR19490
             ENDIF                                                        POR19500
            IF (J-IR.GT.0) THEN                                           POR19510
               IF (I+IR.LT.IM)                                            POR19520
     1          COR(N) = COR(N) + NLBLA * LBLA(I+IR,J-IR,K)               POR19530
               IF (I-IR.GT.0)                                             POR19540
     1          COR(N) = COR(N) + NLBLA * LBLA(I-IR,J-IR,K)               POR19550
               IF (K+IR.LT.KM)                                            POR19560
     1          COR(N) = COR(N) + NLBLA * LBLA(I,J-IR,K+IR)               POR19570
               IF (K-IR.GT.0)                                             POR19580
     1          COR(N) = COR(N) + NLBLA * LBLA(I,J-IR,K-IR)               POR19590
             ENDIF                                                        POR19600
            IF (K-IR.GT.1) THEN                                           POR19610
               IF (I+IR.LT.IM)                                            POR19620
     1          COR(N) = COR(N) + NLBLA * LBLA(I+IR,J,K-IR)               POR19630
               IF (I-IR.GT.0)                                             POR19640
     1          COR(N) = COR(N) + NLBLA * LBLA(I-IR,J,K-IR)               POR19650
             ENDIF                                                        POR19660
            IF (K+IR.LT.KM) THEN                                          POR19670
               IF (I+IR.LT.IM)                                            POR19680
     1          COR(N) = COR(N) + NLBLA * LBLA(I+IR,J,K+IR)               POR19690
               IF (I-IR.GT.0)                                             POR19700
     1          COR(N) = COR(N) + NLBLA * LBLA(I-IR,J,K+IR)               POR19710
             ENDIF                                                        POR19720
  300    CONTINUE                                                        POR19730
         COR(N) = COR(N) / (NPART*12.)                                   POR19740
         WRITE (6,*) R(N),COR(N)                                         POR19750
         WRITE (7,*) R(N),COR(N)                                         POR19760
  500 CONTINUE                                                           POR19770
         RETURN                                                          POR19780
         END                                                             POR19790
*-------------------------------                                         POR19800
*  SUBROUTINE TO CALCULATE LATTICE CORRELATIONS                         POR19810
*  TO TRY AND QUANTIFY THE PROPERTIES OF                                POR19820
*  THE GEOMETRICAL CHARACTERISTICS OF EACH POROUS ELECTRODE             POR19830
*   -- DENSITY-DENSITY CORRELATION CALCULATION ---                      POR19840
*         WRITTEN ON 6/16/85 IN VS FORTRAN77                            POR19850
         SUBROUTINE CORLAT (A,PORVOL,VOLNOR,ILOOK,COR,IMAX)              POR19860
*                                                                        POR19870
         PARAMETER (MAXR=25)                                             POR19880
         INTEGER    A(IM,JM,KM)                                          POR19890
```

```
      COMMON /LATICE/ IM,JM,KM,KMSTOP                                   POR19900
      DIMENSION ICOR(MAXR),COR(MAXR)                                    POR19910
      INTEGER*4 IR                                                      POR19920
*                                                                       POR19930
      IMAX = MAXR                                                       POR19940
      IMIN = MIN (IM,JM,KM)                                             POR19950
      IF (MAXR.GT.IMIN) THEN                                            POR19960
            WRITE (*,*) 'ERROR - SPEC. MAX CORRELATION LENGTH TOO HIGH' POR19970
            WRITE (*,*) 'SETTING MAXR = ',IMIN                          POR19980
            IMAX = IMIN                                                 POR19990
      ENDIF                                                             POR20000
*                                                                       POR20010
*     FIRST VARY R TO GET THE CORRELATION FUNCTION, ICOR(IR)            POR20020
*                                                                       POR20030
      DO 500 IR=1,IMAX                                                  POR20040
      N = 0                                                             POR20050
      ICOR(IR) = 0                                                      POR20060
*                                                                       POR20070
*     FOR EACH R - SUM OVER THE WHOLE LATTICE.                          POR20080
         DO 100 K=1,KM                                                  POR20090
         DO 100 J=1,JM                                                  POR20100
         DO 100 I=1,IM                                                  POR20110
*                                                                       POR20120
          IAT = A(I,J,K)                                                POR20130
          IF (IAT.EQ.ILOOK) THEN                                        POR20140
            N = N + 1                                                   POR20150
*****       UP                                                          POR20160
            IQ = K + IR                                                 POR20170
            IF (IQ.GT.KM)  IQ = IQ - KM                                 POR20180
            IF (IAT.EQ.A(I,J,IQ)) ICOR(IR) = ICOR(IR) + 1               POR20190
*****       DOWN                                                        POR20200
            IQ = K - IR                                                 POR20210
            IF (IQ.LT.1) IQ = IQ + KM                                   POR20220
            IF (IAT.EQ.A(I,J,IQ)) ICOR(IR) = ICOR(IR) + 1               POR20230
*****       RIGHT                                                       POR20240
            IQ = J+IR                                                   POR20250
            IF (IQ.GT.JM) IQ = IQ - JM                                  POR20260
            IF (IAT.EQ.A(I,IQ,K)) ICOR(IR) = ICOR(IR) + 1               POR20270
*****       LEFT                                                        POR20280
            IQ = J-IR                                                   POR20290
            IF (IQ.LT.1) IQ = IQ + JM                                   POR20300
            IF (IAT.EQ.A(I,IQ,K)) ICOR(IR) = ICOR(IR) + 1               POR20310
*****       FRONT                                                       POR20320
            IQ = I + IR                                                 POR20330
            IF (IQ.LT.IM) IQ = IQ - IM                                  POR20340
            IF (IAT.EQ.A(IQ,J,K)) ICOR(IR) = ICOR(IR) + 1               POR20350
*****        BACK                                                       POR20360
            IQ = I - IR                                                 POR20370
            IF (IQ.LT.1) IQ = IQ + IM                                   POR20380
```

```
            IF (IAT.EQ.A(IQ,J,K)) ICOR(IR) = ICOR(IR) + 1                POR20390
          ENDIF                                                          POR20400
  100    CONTINUE                                                        POR20410
      COR(IR) = REAL(ICOR(IR)) / (REAL(N) * 6.)                          POR20420
      WRITE (6,150) IR,COR(IR),ALOG10(FLOAT(IR)),ALOG10(COR(IR))         POR20430
      WRITE (7,150) IR,COR(IR),ALOG10(FLOAT(IR)),ALOG10(COR(IR))         POR20440
  150  FORMAT (' ',I4,1P,3E15.5)                                         POR20450
  500 CONTINUE                                                           POR20460
      RETURN                                                             POR20470
      END                                                                POR20480
*************************************************************************POR20490
*                                                                       *POR20500
*        END OF PROGRAM POR_MET AND SUBROUTINES                         *POR20510
*                                                                       *POR20520
*************************************************************************POR20530
*************************************************************************POR20540
*                                                                       *POR20550
*                                                                       *POR20560
*                                                                       *POR20570
*            DISSOLUTION - PRECIPITATION SIMULATION                     *POR20580
*                      PROGRAM DP2                                      *POR20590
*                 VERSION 2.15 FROM 01/07/86                            *POR20600
*              TRYING TO GENERATE BROWN FRACTALS                        *POR20610
*          THIS IMPLEMENTATION: LET TOP ROW LOSE PARTICLES              *POR20620
*        KEEP TRACK OF MAX Y, # OF ATTACHED PARTICLES, MIN Y            *POR20630
* ALLOWS FOR CLUSTER ATTACHMENT: NO CLUSTERS (ICLUST=0), ALL CLUSTERS   *POR20640
* (ICLUST=1) OR CLUSTERS LARGER THAN A CERTAIN SIZE (ICLUST>1)          *POR20650
* ALLOWS FOR DISSOLUTION OF PARTICLES AS WELL                           *POR20660
* THE PARTICLE MOVEMENT CAN BE REGULATED BY PROB3 - PROBABILITY         *POR20670
* TO TRY TO MOVE A PARTICLE                                             *POR20680
* COR1 - CORRELATION MATRIX FOR # OF POINTS ON THE INTERFACE            *POR20690
* COR2 - CORRELATION MATRIX FOR # OF INTERFACES                         *POR20700
*                                                                       *POR20710
*                                                                       *POR20720
*************************************************************************POR20730
      PARAMETER (MXSPAC =050,MYSPAC =050,MXPRT = 1*MXSPAC*MYSPAC)       POR20740
      INTEGER*2 A(MXSPAC,MYSPAC),B(MXPRT,2),D(MXPRT,2)                  POR20750
*  AS IS A TEMPORARY MATRIX TO CONTAIN ONLY SURFACE PARTICLES            POR20760
      INTEGER*2 AS(MXSPAC,MYSPAC)                                       POR20770
      INTEGER*2 MODE,IPALET,ICOLOR(3),ICLR,MAXR                         POR20780
      INTEGER*2 DEVICE,I2ZERO,I2ONE,I2THRE,I2FOUR,IDIS,KCOUNT,JCOUNT    POR20790
      INTEGER*2 I2TWO                                                   POR20800
       PARAMETER (I2ZERO=0,I2ONE=1,I2THRE=3,I2FOUR=4,I2TWO=2)           POR20810
      INTEGER*4 NPART,MXGEN,KGEN,BGROW(MXPRT),JENCNT(50),MAXHT(50)      POR20820
      INTEGER*4 NPSTUK(50),NPFRE(50),NSURPT(50),MXFREE(50)              POR20830
      INTEGER*2 MINIY(50),ICNTR(-1:1)                                   POR20840
      LOGICAL*1 EMPTY,NOTSAV,IQUIT                                      POR20850
      REAL*8 DSEEDO,DSEED,GETTIM,GOTMIN,ZMANO,ZMAN                      POR20860
C DCONC IS THE CONCENTRATION PROFILE FOR DISSOLVED PARTICLES            POR20870
C SCONC IS THE CONCENTRATION PROFILE FOR STUCK PARTICLES                 POR20880
```

```
      REAL*4 RUF(50),DCONC(10000),SCONC(10000)                          POR20890
      REAL*4 COR1(100000),COR2(100000)                                  POR20900
       CHARACTER*8 IDNAM                                                POR20910
       CHARACTER*4 CDUM,MSG1,MSG2,MSG3,MSG4                             POR20920
       CHARACTER*1 GRAP(-1:5)                                           POR20930
       DATA GRAP /'+',' ','*','?','#','-','!'/                         POR20940
     1,MSG1/'CONC'/,MSG2/'PART'/,                                       POR20950
     1 MSG3/'COR1'/,MSG4/'COR2'/                                        POR20960
      COMMON /SIZPRM/  NXSPAC,NYSPAC,MXNPRT,NPART,NNUC,NLAY             POR20970
      COMMON /PARAM/ NPARTO,PROB,PROB2,PROB3,CONC,IZ,MX1Y,ICLUST,KGEN,  POR20980
     1 MN1Y,MXFRE,IAMBLK                                                POR20990
C*********************************************************************** POR21000
      WRITE (*,*) 'IF U CHANGE ANYTHING, PLZ CHGE REV AND INSERT A COM'  POR21010
      REV = 2.15                                                        POR21020
      WRITE (*,10) REV                                                  POR21030
 10   FORMAT (' PROGRAM DP - REVISION ',F5.2)                           POR21040
C*********************************************************************** POR21050
      DUMMY=0.0                                                         POR21060
      EMPTY =  .TRUE.                                                   POR21070
      NOTSAV = .FALSE.                                                  POR21080
      IQUIT = .FALSE.                                                   POR21090
      MODE = 1                                                          POR21100
      IPALET = 0                                                        POR21110
      ICOLOR(1) = 1                                                     POR21120
      ICOLOR(2) = 0                                                     POR21130
      ICOLOR(3) = 2                                                     POR21140
      NXSPAC= MXSPAC                                                    POR21150
      NYSPAC= MYSPAC                                                    POR21160
      MXNPRT= MXPRT                                                     POR21170
      MXFRE = 0                                                         POR21180
      MAXR=NXSPAC/2                                                     POR21190
      KCOUNT= I2ZERO                                                    POR21200
      JCOUNT= I2ZERO                                                    POR21210
      NSURF=0                                                           POR21220
      NSURF1=0                                                          POR21230
       IORD = -1                                                        POR21240
C    BULK IS THE LOCAL 'Y' CONCENTRATION THAT ALMOST INSURES            POR21250
C    THAT THIS LINE IS CONNECTED TO THE BULK - USED WITH PARAM "IAMBLK" POR21260
      BULK = .80                                                        POR21270
      IDBULK  = BULK * NXSPAC                                           POR21280
      DCONC(1) = 1.                                                     POR21290
      SCONC(1) = 1.                                                     POR21300
       DO 100 J=1,NYSPAC                                                POR21310
       DO 100 I=1,NXSPAC                                                POR21320
 100   A(I,J) = I2ZERO                                                  POR21330
      CALL ICOUNT (A)                                                   POR21340
      CALL ICPROF (A,DCONC,SCONC)                                       POR21350
  20  IF (IQUIT) GOTO 3000                                              POR21360
      WRITE (*,*) 'ENTER CHOICE:'                                       POR21370
```

```
      WRITE (*,*) '1-  INITIALIZE A NEW PROBLEM'                          POR21380
      WRITE (*,*) '2-  RESUME AN OLD PROBLEM'                             POR21390
      WRITE (*,*) '3-  ---- QUIT ------'                                  POR21400
      WRITE (*,*) '4-  GO!'                                               POR21410
      WRITE (*,*) '5-  STORE THE CURRENT SIMULATION'                      POR21420
      WRITE (*,*) '6-  LIST THE CURRENTLY DEFINED PARAMETERS'             POR21430
      WRITE (*,*) '7-  GRAPHICAL DISPLAY OF THE CURRENT SIMULATION'       POR21440
      CALL RDTERM(IOPT,DUM,CDUM,0,0,*20)                                  POR21450
      IF (EMPTY.AND.(IOPT.GT.3)) THEN                                     POR21460
            WRITE (*,*) ' NO SIMULATION CURRENTLY DEFINED'                POR21470
            GOTO 20                                                       POR21480
      ENDIF                                                               POR21490
      GOTO (1000,2000,3000,4000,5000,6000,7000), IOPT                     POR21500
      GOTO 20                                                             POR21510
*================= INITIALIZE A NEW PROBLEM ================            POR21520
 1000 WRITE (*,*) ' ENTER ID NAME FOR THIS RUN (DATAID-8 CHAR ',          POR21530
     1  'ENCLOSED IN SINGLE QUOTES)'                                     POR21540
      READ (*,*) IDNAM                                                    POR21550
      WRITE (*,*) 'ENTER DESIRED CLUSTER ATTACHMENT:',                    POR21560
     1' (0 - NO CLUSTERS, 1 - ALL CLUSTERS, ETC.)'                        POR21570
      READ (*,*) ICLUST                                                   POR21580
      WRITE (*,*) 'ENTER DESIRED PARTICLE CONCENTRATION (<.5):'           POR21590
      READ (*,*) CONC                                                     POR21600
      WRITE (*,*) 'ENTER DESIRED NUCLEI CONCENTRATION :'                  POR21610
      READ (*,*) CONUC                                                    POR21620
      WRITE (*,*) 'ENTER DESIRED # OF LAYERS :'                           POR21630
      READ (*,*) NLAY                                                     POR21640
      IF (NLAY.GT.2*NYSPAC/3) THEN                                        POR21650
            WRITE (*,*) ' TOO MANY LAYERS OF BULK SPECIFIED'              POR21660
            STOP                                                          POR21670
      ENDIF                                                               POR21680
      WRITE (*,*) 'ENTER DESIRED STICKING PROBABILITY .'                  POR21690
      READ (*,*) PROB                                                     POR21700
      WRITE (*,*) 'ENTER DESIRED UN-STICKING PROBABILITY :'               POR21710
      READ (*,*) PROB2                                                    POR21720
      WRITE (*,*) 'ENTER DESIRED MOVING PROBABILITY :'                    POR21730
      READ (*,*) PROB3                                                    POR21740
       DSEED0 = 2345.0D0                                                  POR21750
       DSEED = DSEED0                                                     POR21760
       NNUC=CONUC*NXSPAC                                                  POR21770
       IZ=NNUC+(NLAY-1)*NXSPAC                                            POR21780
       NFREE= CONC*NXSPAC*(NYSPAC-NLAY-CONUC)                             POR21790
       NPART= NFREE+IZ                                                    POR21800
*  NTOP IS THE NUMBER OF PARTICLES ON THE TOP (NON-NUCLEATING) PLANE      POR21810
*  THIS MAY BE KEPT CONSTANT IN SUCCEEDING GENERATIONS BY ADDING LOST     POR21820
C PARTICLES, COUNTED BY 'CNTTOP'                                          POR21830
C      NTOP = CONC*NXSPAC+.5                                              POR21840
       NPARTO = NPART                                                     POR21850
       DO 1050 J=1,NYSPAC                                                 POR21860
```

```
      DO 1050 I=1,NXSPAC                                                POR21870
 1050  A(I,J) = I2ZERO                                                  POR21880
      CALL GENER (A,B,BGROW,DSEED)                                      POR21890
*  MXIY KEEPS TRACK OF THE MAXIMUM LEVEL THAT DENDRITE HAS GROWN TO     POR21900
      MXIY = NLAY                                                       POR21910
      MNIY = NLAY                                                       POR21920
*  IAMBLK KEEPS TRACK OF WHERE THE BULK APROXIMATELY BEGINS             POR21930
      IAMBLK = NLAY                                                     POR21940
      KGENO = 1                                                         POR21950
*   NSTOR IS A COUNTER FOR THE STORED DATA - AVGHT AND MAXHT            POR21960
      NSTOR = 0                                                         POR21970
      EMPTY = .FALSE.                                                   POR21980
      ZMAN = 0.0                                                        POR21990
*  RETURN TO MENU                                                       POR22000
      GOTO 20                                                           POR22010
*==================== READ IN DATA FROM DISK ======================   POR22020
 2000 READ (7,2005) IDNAM,DUMMY                                         POR22030
      READ (7,2010) REV,EZMAN                                           POR22040
      READ (7,2020) NXSPAC,NYSPAC,MXNPRT                                POR22050
      IF (NXSPAC.NE.MXSPAC.OR.NYSPAC.NE.MYSPAC) THEN                    POR22060
       WRITE (*,*) ' ERROR: STORED SIMULATION SPACE DIMENSION INCORRECT'POR22070
       WRITE (*,*) ' EXPECTING DIMENSIONS:',MXSPAC,' X ',MYSPAC         POR22080
       WRITE (*,*) ' BUT STORED SIMULATION IS:',NXSPAC,' X ',NYSPAC     POR22090
       MXNPRT = MXPRT                                                   POR22100
       NYSPAC = MYSPAC                                                  POR22110
       NXSPAC = MXSPAC                                                  POR22120
       GOTO 20                                                          POR22130
      ENDIF                                                             POR22140
      READ (7,2030) CONC,CONUC,NPARTO,NPART,FRAC                        POR22150
      READ (7,2040) DSEEDO,DSEED,PROB3                                  POR22160
      READ (7,2050) ICLUST,PROB,PROB2,DUMMY                             POR22170
      READ (7,2060) KGEN,MXGEN,IZ,MXIY,MNIY,NSTOR,NLAY                 POR22180
      READ (7,2070)                                                     POR22190
     1  (JENCNT(II),NPSTUK(II),NPFRE(II),NSURPT(II)                    POR22200
     1,MAXHT(II),RUF(II),MINIY(II),MXFREE(II),II=1,NSTOR)               POR22210
      READ (7,2075) MSG1                                                POR22220
      READ (7,2078)  DCONC(1)                                           POR22230
      IPTO= DCONC(1)                                                    POR22240
      READ (7,2079) (DCONC(II),II=2,IPTO)                               POR22250
      READ (7,2078)  SCONC(1)                                           POR22260
      IPT = SCONC(1)                                                    POR22270
      READ (7,2079) (SCONC(II),II=2,IPT)                                POR22280
      READ (7,2075) MSG2                                                POR22290
      READ (7,2080) ((B(II,JJ),JJ=1,2),BGROW(II),II=1,NPART)            POR22300
C IORD  IS THE STARTING POINT FOR FUTURE FILLING OF THE CORREL. MATRICESPOR22310
      IORD =NSTOR-1                                                     POR22320
      ICOR = IORD*MAXR                                                  POR22330
      READ (7,2075) MSG3                                                POR22340
      READ (7,2100) (COR1(I),I=1,ICOR)                                  POR22350
```

```
      READ (7,2075) MSG4                                                    POR22360
      READ (7,2100) (COR2(I),I=1,ICOR)                                      POR22370
                                                                            POR22380
 2100 FORMAT (1X,9F8.5)                                                     POR22390
 2005 FORMAT (1X,A8,F5.2)                                                   POR22400
 2010 FORMAT (1X,F6.2,F10.2)                                                POR22410
 2020 FORMAT (1X,2I5,I8)                                                    POR22420
 2030 FORMAT (1X,2F5.2,2I8,F5.2)                                            POR22430
 2040 FORMAT (1X,2D15.0,F5.2)                                               POR22440
 2050 FORMAT (1X,I8,3F5.2)                                                  POR22450
 2060 FORMAT (1X,2I10,I8,2I5,I7,I5)                                         POR22460
 2070 FORMAT (1X,5I7,F7.3,2I7)                                              POR22470
 2075 FORMAT (1X,A4)                                                        POR22480
 2078 FORMAT (1X,F8.0)                                                      POR22490
 2079 FORMAT (1X,8F9.4)                                                     POR22500
 2080 FORMAT (1X,2I5,I9,2I5,I9,2I5,I9)                                      POR22510
      DO 2200 J=1,NYSPAC                                                    POR22520
      DO 2200 I=1,NXSPAC                                                    POR22530
 2200   A(I,J) = I2ZERO                                                     POR22540
                                                                            POR22550
*   NOW DECODE THE INFORMATION FROM THE STORED B MATRIX.                    POR22560
      DO 2300 I = 1,NPART                                                   POR22570
        IX = B(I,1)                                                         POR22580
        IY = B(I,2)                                                         POR22590
        A(IX,IY) = -I2ONE                                                   POR22600
*    PARTICLE STUCK??                                                       POR22610
        IF (BGROW(I).GT.0) A(IX,IY) = I2ONE                                 POR22620
 2300 CONTINUE                                                              POR22630
*  FIRST APROXIMATION; WILL GET SET EXACTLY NEXT GENERATION                 POR22640
      IAMBLK = MNIY                                                         POR22650
      NNUC = CONUC*NXSPAC                                                   POR22660
C     NTOP = CONC * NXSPAC +.5                                              POR22670
      KGENO = KGEN + 1                                                      POR22680
      EMPTY = .FALSE.                                                       POR22690
      GOTO 20                                                               POR22700
*==================== QUIT ============================================    POR22710
 3000 IF (NOTSAV) THEN                                                      POR22720
        WRITE (*,*) 'NEW SIMULATION RESULTS NOT SAVED. SAVE THEM?(Y/N)'     POR22730
        IF (IGTYES(5,6).EQ.1) GOTO 5000                                     POR22740
      ENDIF                                                                 POR22750
      STOP                                                                  POR22760
*********   START SIMULATION  ***************************************     POR22770
 4000 NOTSAV = .TRUE.                                                       POR22780
C IF ONE WANTS TO KEEP THE TOP ROW AT CONSTANT CONC THEN:                   POR22790
C     CALL ICNTTO (A,B,BGROW,NTOP,DSEED)                                    POR22800
C IF ONE WANTS TO KEEP THE BOTTOM ROW AT CONSTANT CONC THEN:                POR22810
C     CALL ICNTBO (A,B,BGROW,NTOP,DSEED)                                    POR22820
C  KCOUNT IS A NEIGHBOUR COUNTER  JCOUNT - SECOND NEIGHB                    POR22830
      CALL IUPDAT (A,B,BGROW,DSEED,D,KCOUNT,JCOUNT)                         POR22840
```

```
      CALL IDISP  (A,AS,B,BGROW,DEVICE, REV,OLDGEN,ZMANO,CONUC          POR22850
     1,COR1,COR2,IORD,NSURF1,IDNAM,ZMAN,DCONC,SCONC,IPTO,MAXR)          POR22860
C                                                                       POR22870
C                                                                       POR22880
      WRITE (*,*) 'CURRENT VALUES: MXGEN=',MXGEN,' STOP FRACTION=',FRAC POR22890
      WRITE (*,*) 'ENTER MAXIMUM # OF GENERATIONS     :'               POR22900
      READ (*,*)  MXGEN                                                 POR22910
      WRITE (*,*) 'ENTER MAXIMUM ALLOWED LOST  FRACTION:'               POR22920
      READ (*,*) FRAC                                                   POR22930
*                                                                       POR22940
*                                                                       POR22950
*  INITIALIZE INTERRUPT FLAG TO ZERO:                                   POR22960
      INIT = ITYBIT (-1)                                                POR22970
C IORD IS A COUNTER FOR SEQUENTIAL WRITING OF CORREL MATRICES           POR22980
C  INITIALIZE DISSOLUTION  FLAG                                         POR22990
      IDIS=I2ONE                                                        POR23000
*  INITIALIZE TIMER VARIABLES                                           POR23010
      OGEN = KGEN                                                       POR23020
*   INITIAL TIME FOR TOTAL SIMULATION CALCULATION.                      POR23030
*    ZMANO WILL ALWAYS BE SUBTRACTED FROM CURRENT TIME.                 POR23040
      ZMANO  = GETTIM(GOTMIN)                                           POR23050
      WRITE (*,*) 'ENTER DEVICE #:                                      POR23060
     1(0=NONE,1=PC,2=VM/TEK,3=VM/PRT,4=VM/SCREEN)'                      POR23070
      READ (*,*) DEVICE                                                 POR23080
*   IBOTOM IS THE BOTTOM BELOW WHICH WE DONT ALLOW THIS TO PROGRESS     POR23090
      IBOTOM = NLAY / 10                                                POR23100
C DISPLAY IN LOGARITHMIC SCALE, E.G. IFREQ=3 ==> GENERATIONS DISPLAYED  POR23110
C WILL BE 3, 6, 9, 30, 60, 90, 300, 600, ETC.                           POR23120
      IFREQ = 2                                                         POR23130
* SET UP TO AUTOMATICALLY QUIT AT THE END OF SIMULATION:                POR23140
      IQUIT = .TRUE.                                                    POR23150
*  INITIALIZE DECADE COUNTER                :                          POR23160
      IDECAD = 10 ** INT(ALOG10(FLOAT(KGEN0)))                          POR23170
*                                                                       POR23180
        DO 4050 I = NPART+1,MXNPRT                                      POR23190
            BGROW(I)=0                                                  POR23200
 4050   CONTINUE                                                        POR23210
*                                                                       POR23220
      DO 4500 KGEN=KGEN0,MXGEN                                          POR23230
C IF ONE WANTS TO KEEP THE TOP ROW AT CONSTANT CONC THEN:               POR23240
C     CALL CNTTOP (A,B,BGROW,NTOP,DSEED)                                POR23250
C IF ONE WANTS TO KEEP THE BOTTOM ROW AT CONSTANT CONC THEN:            POR23260
C     CALL CNTBOT (A,B,BGROW,NTOP,DSEED)                                POR23270
         IF (NPART.LE.0) GOTO 20                                        POR23280
        CALL UPDATE                                                     POR23290
*     NOW UPDATE BGROW, IZ TO BE PROPERLY RECORDED                      POR23300
         DO 4200 I = 1,NPART                                            POR23310
            IX = B(I,1)                                                 POR23320
            IY = B(I,2)                                                 POR23330
```

```
            IF (A(IX,IY).GT.I2ZERO) THEN                                    POR23340
               IF (BGROW(I).EQ.0) THEN                                      POR23350
                  IZ=IZ+1                                                   POR23360
                  BGROW(I) = KGEN                                           POR23370
               ENDIF                                                        POR23380
            ELSE IF (A(IX,IY).LT.I2ZERO) THEN                               POR23390
*         PARTICLE IS FREE - MAKE SURE BGROW IS 0 AND MXFRE IS CORRECT      POR23400
               IF (BGROW(I) .GT. 0) THEN                                    POR23410
                  BGROW(I) = 0                                              POR23420
                  IZ=IZ-1                                                   POR23430
               ENDIF                                                        POR23440
               IF (IY.GT.MXFRE) MXFRE = IY                                  POR23450
            ENDIF                                                           POR23460
 4200     CONTINUE                                                          POR23470
*-----------------------------------                                        POR23480
*    NOW FIND IAMBLK VALUE                                                  POR23490
      IAMBLK = MXIY                                                         POR23500
 4210  ICNTR(1) = 0                                                        POR23510
         ICNTR(-1) = I2ZERO                                                 POR23520
         ICNTR(0)  = I2ZERO                                                 POR23530
*   COUNT THE NUMBER OF STUCK PARTICLES PER LINE - IN ICNTR(1)              POR23540
      DO 4250 II=1,NXSPAC                                                   POR23550
 4250    ICNTR(A(II,IAMBLK)) = ICNTR(A(II,IAMBLK)) + 1                      POR23560
      IAMBLK = IAMBLK - 1                                                   POR23570
      IF (ICNTR(1).LT.IDBULK) GOTO 4210                                     POR23580
*    IAMBLK IDENTIFIED - BELOW THAT WILL BE CONSIDERED TO BE BULK           POR23590
* -------------------------------------------------------------            POR23600
C IF IAMBLK IS TOO LOW, EXIT                                                POR23610
      IF (IAMBLK.LE.IBOTOM) GOTO 4700                                       POR23620
         IF (KGEN.GT.IDECAD*10) IDECAD = IDECAD * 10                        POR23630
         IF (MOD(KGEN,IFREQ*IDECAD).EQ.0 ) THEN                             POR23640
*  RESET COUNTERS EVERY NOW AND THEN SO THAT THEY DONT OVERFLOW              POR23650
            IORD = IORD+1                                                   POR23660
*     DISPLA ALSO CALCULATES NSURF, MNIY, MXIY, NSURF1, COR1,COR2           POR23670
            CALL DISPLA (NSURF)                                             POR23680
            CALL CPROF(IPTO)                                                POR23690
            NSTOR = NSTOR + 1                                               POR23700
            JENCNT(NSTOR) = KGEN                                            POR23710
            NPSTUK(NSTOR) = IZ                                              POR23720
            MINIY(NSTOR) = MNIY                                             POR23730
            NPFRE (NSTOR) = NPART - IZ                                      POR23740
            MAXHT(NSTOR) = MXIY                                             POR23750
            RUF(NSTOR) = FLOAT(NSURF) / FLOAT(NXSPAC)                       POR23760
            NSURPT(NSTOR)=NSURF1                                            POR23770
            MXFREE(NSTOR)=MXFRE                                             POR23780
         ENDIF                                                              POR23790
*      LOOP EXIT CONDITION:                                                 POR23800
*                                                                           POR23810
      IF (NPART.LT.NPART0*(1-FRAC).OR.MXIY+1.GE.NYSPAC) GOTO 4700           POR23820
*     CHECK INTERRUPT FLAG - HAS THE USER REQUESTED A HALT?                 POR23830
*     (THIS MAY BE DONE BY THE CP COMMAND 'ST 464 1' AT A CP ATTN INTRPPOR23840
```

```
          IF (ITYBIT(0).GT.0) THEN                                          POR23850
               IQUIT = .FALSE.                                              POR23860
               WRITE (*,*) ' USER REQUESTED HALT IN PROGRAM EXECUTION'      POR23870
*              NOW SET UP LOOP PARAMETER FOR POSSIBLE LATER CONTINUE        POR23880
               KGENO = KGEN + 1                                             POR23890
               GOTO 4700                                                    POR23900
          ENDIF                                                             POR23910
 4500 CONTINUE                                                              POR23920
 4700 IORD=IORD+1                                                           POR23930
      CALL DISPLA (NSURF)                                                   POR23940
      CALL CPROF(IPTO)                                                      POR23950
          IF (JENCNT(NSTOR).NE.KGEN) THEN                                   POR23960
            NSTOR = NSTOR + 1                                               POR23970
            JENCNT(NSTOR) = KGEN                                            POR23980
           NPSTUK(NSTOR) = IZ                                               POR23990
           MINIY(NSTOR) = MNIY                                              POR24000
           NPFRE (NSTOR) = NPART - IZ                                       POR24010
            MAXHT(NSTOR) = MXIY                                             POR24020
            RUF(NSTOR) = FLOAT(NSURF) / FLOAT(NXSPAC)                       POR24030
           NSURPT(NSTOR)=NSURF1                                             POR24040
           MXFREE(NSTOR)=MXFRE                                              POR24050
          ENDIF                                                             POR24060
*  NOW GO AND STORE AND QUIT/MENU                                           POR24070
*============== STORE THE CURRENT SIMULATION PARAMETERS =========          POR24080
 5000 WRITE (8,5005) IDNAM,DUMMY                                            POR24090
      EZMAN = ZMAN / 60.DO                                                  POR24100
      WRITE (8,5010) REV,EZMAN                                              POR24110
      WRITE (8,5020) MXSPAC,MYSPAC,MXNPRT                                   POR24120
      WRITE (8,5030) CONC,CONUC,NPARTO,NPART,FRAC                           POR24130
      WRITE (8,5040) DSEEDO,DSEED,PROB3                                     POR24140
      WRITE (8,5050) ICLUST,PROB,PROB2,DUMMY                                POR24150
      WRITE (8,5060) KGEN,MXGEN,IZ,MXIY,MNIY,NSTOR,NLAY                     POR24160
      WRITE (8,5070)                                                        POR24170
     1  (JENCNT(II),NPSTUK(II),NPFRE(II),NSURPT(II)                        POR24180
     1,MAXHT(II),RUF(II),MINIY(II),MXFREE(II),II=1,NSTOR)                   POR24190
      WRITE (8,5075) MSG1                                                   POR24200
      WRITE(8,5078)  DCONC(1)                                               POR24210
      IPT = DCONC(1)                                                        POR24220
      WRITE(8,5079) (DCONC(II),II=2,IPT)                                    POR24230
      WRITE(8,5078)  SCONC(1)                                               POR24240
      IPT = SCONC(1)                                                        POR24250
      WRITE(8,5079) (SCONC(II),II=2,IPT)                                    POR24260
      WRITE(8,5075) MSG2                                                    POR24270
      WRITE (8,5080) ((B(II,JJ),JJ=1,2),BGROW(II),II=1,NPART)               POR24280
      ICOR =(IORD+1) * MAXR                                                 POR24290
      WRITE(8,5075) MSG3                                                    POR24300
      WRITE(8,5100) (COR1(I),I=1,ICOR)                                      POR24310
      WRITE(8,5075) MSG4                                                    POR24320
      WRITE(8,5100) (COR2(I),I=1,ICOR)                                      POR24330
```

```
 5100 FORMAT (' ',9F8.5)                                                  POR24340
 5005 FORMAT (' ',A8,F5.2)                                                POR24350
 5010 FORMAT (' ',F6.2,F10.2)                                             POR24360
 5020 FORMAT (' ',215,I8)                                                 POR24370
 5030 FORMAT (' ',2F5.2,2I8,F5.2)                                         POR24380
 5040 FORMAT (' ',F13.0,'D0',F13.0,'D0',F5.2)                             POR24390
 5050 FORMAT (' ',I8,3F5.2)                                               POR24400
 5060 FORMAT (' ',2I10,I8,2I5,I7,I5)                                      POR24410
 5070 FORMAT (' ',5I7,F7.3,2I7)                                           POR24420
 5075 FORMAT (' ',A4)                                                     POR24430
 5078 FORMAT (1X,F8.0)                                                    POR24440
 5079 FORMAT (1X,8F9.4)                                                   POR24450
 5080 FORMAT (' ',2I5,I9,2I5,I9,2I5,I9,2I5,I9)                           POR24460
                                                                          POR24470
      NOTSAV = .FALSE.                                                    POR24480
      GOTO 20                                                             POR24490
*============= LIST THE CURRENTLY DEFINED PARAMETERS ============       POR24500
 6000 WRITE (*,*) 'FEATURE NOT IMPLEMENTED YET'                           POR24510
      GOTO 20                                                             POR24520
*============= GRAPHICAL DISPLAY OF SIMULATION ===================       POR24530
 7000 WRITE (*,*) 'ENTER DEVICE #:(1=PC,2=VM/TEK,3=VM/PRT,4=VM/SCREEN)'  POR24540
      READ (*,*) DEVICE                                                   POR24550
*  INITIALIZE TIMER VARIABLES                                            POR24560
      OGEN = KGEN                                                         POR24570
      OTIM = GETTIM (GOTMIN)                                              POR24580
      CALL DISPLA (NSURF)                                                 POR24590
      GOTO 20                                                             POR24600
*=================================================================      POR24610
       END                                                                POR24620
                                                                          POR24630
                                                                          POR24640
                                                                          POR24650
                                                                          POR24660
      SUBROUTINE GENER  (A,B,BGROW,DSEED)                                 POR24670
      INTEGER*2 I2ZERO,I2ONE,IDIS                                         POR24680
       PARAMETER (I2ZERO=0,I2ONE=1)                                       POR24690
C     POPULATE A USING NPART PARTICLES AND INITIALIZE                     POR24700
C     B TO PROPER VALUES                                                  POR24710
      COMMON /SIZPRM/  NXSPAC,NYSPAC,MXNPRT,NPART,NNUC,NLAY               POR24720
      COMMON /PARAM/ NPARTO,PROB,PROB2,PROB3,CONC,IZ,MXIY,ICLUST,KGEN,    POR24730
     1 MNIY,MXFRE,IAMBLK                                                  POR24740
      INTEGER*4 NXSPAC,NYSPAC,MXNPRT,NPART,BGROW(MXNPRT)                  POR24750
      INTEGER*2 A(NXSPAC,NYSPAC),B(MXNPRT,2)                              POR24760
      REAL*8 DSEED                                                        POR24770
C SET UP THE FIXED PARTICLES                                              POR24780
C DEEP DOWN                                                               POR24790
      NSTUK=0                                                             POR24800
      DO 50 JJ=1,NLAY-1                                                   POR24810
      DO 50 II=1,NXSPAC                                                   POR24820
```

```
      NSTUK=NSTUK+1                                                    POR24830
       B(NSTUK,1)=II                                                   POR24840
       B(NSTUK,2)=JJ                                                   POR24850
       BGROW(NSTUK)=1                                                  POR24860
       A(II,JJ)=I2ONE                                                  POR24870
 50    CONTINUE                                                        POR24880
       IF (NNUC.NE.NXSPAC) THEN                                        POR24890
            DO 500 I=NSTUK+1,IZ                                        POR24900
 400        IX=(NXSPAC*GGUBFS(DSEED)+1.)                               POR24910
            IY=1                                                       POR24920
            IF (A(IX,IY).EQ.I2ZERO) THEN                               POR24930
                 A(IX,IY)=I2ONE                                        POR24940
                 B(I,1)=IX                                             POR24950
                 B(I,2)=IY                                             POR24960
                 BGROW(I) = 1                                          POR24970
               ELSE                                                    POR24980
                 GOTO 400                                              POR24990
               ENDIF                                                   POR25000
 500        CONTINUE                                                   POR25010
            GOTO 600                                                   POR25020
       ELSE                                                            POR25030
            DO 550 I=1,NXSPAC                                          POR25040
              NSTUK=NSTUK+1                                            POR25050
              B(NSTUK,1)=I                                             POR25060
              B(NSTUK,2)=NLAY                                          POR25070
              BGROW(NSTUK)=1                                           POR25080
              A(I,NLAY)=I2ONE                                          POR25090
 550        CONTINUE                                                   POR25100
       ENDIF                                                           POR25110
*   NOW TO SET UP THE FREE PARTICLES:                                  POR25120
      IF (NPART.EQ.IZ) RETURN                                          POR25130
 600   DO 1000 I=IZ+1,NPART                                            POR25140
 800  IX=(NXSPAC*GGUBFS(DSEED)+1.)                                     POR25150
      IY=(NYSPAC*GGUBFS(DSEED)+1.)                                     POR25160
      IF (A(IX,IY).EQ.I2ZERO) THEN                                     POR25170
           A(IX,IY)=-I2ONE                                             POR25180
           B(I,1)=IX                                                   POR25190
           B(I,2)=IY                                                   POR25200
           BGROW(I) = 0                                                POR25210
       ELSE                                                            POR25220
           GOTO 800                                                    POR25230
       ENDIF                                                           POR25240
 1000  CONTINUE                                                        POR25250
      RETURN                                                           POR25260
      END                                                              POR25270
                                                                       POR25280
                                                                       POR25290
      SUBROUTINE ISHUFL (DSEED,B,BGROW)                                POR25300
                                                                       POR25310
```

```
      INTEGER*4 NPART                                                    POR25320
      COMMON /SIZPRM/  NXSPAC,NYSPAC,MXNPRT,NPART,NNUC,NLAY             POR25330
      COMMON /PARAM/ NPARTO,PROB,PROB2,PROB3,CONC,IZ,MXIY,ICLUST,KGEN,   POR25340
     1 MNIY,MXFRE,IAMBLK                                                 POR25350
      REAL*8 DSEED                                                       POR25360
      INTEGER*2 B(MXNPRT,2),ITEMP                                        POR25370
      INTEGER*4 BGROW(MXNPRT)                                            POR25380
      RETURN                                                             POR25390
*                                                                        POR25400
      ENTRY SHUFLE                                                       POR25410
      M=NPART                                                            POR25420
      KM1 = NPART-1                                                      POR25430
C               NOW RANDOMLY PERMUTE                                     POR25440
      DO 10 I=1,KM1                                                      POR25450
         J = 1+GGUBFS(DSEED)*M                                           POR25460
         ITEMP = B(M,1)                                                  POR25470
         B(M,1) = B(J,1)                                                 POR25480
         B(J,1) = ITEMP                                                  POR25490
         ITEMP = B(M,2)                                                  POR25500
         B(M,2) = B(J,2)                                                 POR25510
         B(J,2) = ITEMP                                                  POR25520
         ITEMP = BGROW(M)                                                POR25530
         BGROW(M) = BGROW(J)                                             POR25540
         BGROW(J) = ITEMP                                                POR25550
         M = M-1                                                         POR25560
   10 CONTINUE                                                           POR25570
      RETURN                                                             POR25580
      END                                                                POR25590
                                                                         POR25600
                                                                         POR25610
      SUBROUTINE IUPDAT (A,B,BGROW,DSEED,D,KCOUNT,JCOUNT)                POR25620
      INTEGER*2 I2ZERO,I2ONE,IDIS,KCOUNT,I2FOUR,I2TWO,I2THRE,JCOUNT      POR25630
       PARAMETER (I2ZERO=0,I2ONE=1,I2FOUR=4,I2TWO=2,I2THRE=3)            POR25640
C UPDATES  POSITION OF THE PARTICLES; IZ IS A STUCK PARTICLE COUNTER      POR25650
C  MXIY RECORDS THE MAXIMUM HEIGHT OF DENDRITE, MNIY - THE 'FLOOR LEVEL' POR25660
      COMMON /SIZPRM/  NXSPAC,NYSPAC,MXNPRT,NPART,NNUC,NLAY              POR25670
      COMMON /PARAM/ NPARTO,PROB,PROB2,PROB3,CONC,IZ,MXIY,ICLUST,KGEN,   POR25680
     1 MNIY,MXFRE,IAMBLK                                                 POR25690
      INTEGER*4 NXSPAC,NYSPAC,MXNPRT,NPART,BGROW(MXNPRT)                 POR25700
      INTEGER*2 A(NXSPAC,NYSPAC),B(MXNPRT,2),D(MXNPRT,2)                 POR25710
      INTEGER*2  ILIST(4),JLIST(4)                                       POR25720
      REAL*8 DSEED                                                       POR25730
      CALL ISHUFL (DSEED,B,BGROW)                                        POR25740
      CALL IMOVE (A,B,BGROW,I,IX,IY,DSEED,ITRY)                          POR25750
      CALL INCLUS (A,D)                                                  POR25760
      CALL INEIGH (A)                                                    POR25770
      RETURN                                                             POR25780
*                                                                        POR25790
      ENTRY UPDATE                                                       POR25800
```

```
      CALL SHUFLE                                                       POR25810
* THE VALUE OF NPART CHANGES AS PARTICLES LEAVE AND ENTER THE BOX        POR25820
      DO 1000 I=1,MXNPRT                                                POR25830
*   EVEN THOUGH NPART MAY HAVE DECREASED WE STILL WANT TO GET ALL THE   POR25840
*          PARTICLES CHECKED IN THIS GENERATION                         POR25850
  5   IF (I.GT.NPART) RETURN                                            POR25860
      IX=B(I,1)                                                         POR25870
      IY=B(I,2)                                                         POR25880
C IS THE PARTICLE STUCK??                                               POR25890
*************** IF SO, BEGIN ATTEMPT TO DISLODGE STUCK PARTICLE        POR25900
C UNLESS IT IS ON THE BOTTOM PLANE                                      POR25910
      IF (IY.EQ.1) GOTO 1000                                            POR25920
      IF (A(IX,IY).GT.I2ZERO) THEN                                      POR25930
*          IF SURROUNDED IT STAYS STUCK - AND GOTO NEXT PARTICLE        POR25940
         CALL COUNT (IX,IY,KCOUNT,JCOUNT,1)                             POR25950
         IF (KCOUNT.EQ.I2FOUR) GOTO 1000                                POR25960
*          IF THE PARTICLE IS ISOLATED, IT WILL DISSOLVE                POR25970
         IF (KCOUNT.EQ.I2ZERO) THEN                                     POR25980
           A(IX,IY)=-I2ONE                                              POR25990
*   TRY TO MOVE IT                                                      POR26000
           GOTO 15                                                      POR26010
         ENDIF                                                          POR26020
* PARTICLE IS STUCK AND ON THE SURFACE                                  POR26030
*   DOES IT DISSOLVE?                                                   POR26040
         R=GGUBFS(DSEED)                                                POR26050
*          PROB2 IS THE DISSOLUTION PROBABILITY                         POR26060
*          IF IT DOESNT DISSOLVE, THEN FORGET THIS PARTICLE.            POR26070
        IF (R.GE.PROB2) GOTO 1000                                       POR26080
*          IF R IS LT PROB2  THEN THE PARTICLE DISSOLVES                POR26090
        A(IX,IY)=-I2ONE                                                 POR26100
* NEXT QUESTION - DID DISSOLVED PARTICLE DISCONNECT A CLUSTER?          POR26110
C IDIS IS A DISSOLUTION (1)/PRECIPITATION (-1) FLAG                     POR26120
      IDIS = I2ONE                                                      POR26130
C IF IT IS A TIP OF DENDRITE, JUST TRY TO MOVE IT                       POR26140
      IF (KCOUNT.EQ.I2ONE) GOTO 15                                      POR26150
**********************                                                  POR26160
      CALL NEIGH(IX,IY,ILIST,JLIST)                                     POR26170
*          ILIST AND JLIST CONTAIN NEIGHBOR'S COORDINATES               POR26180
*==========BEGIN AN ATTEMPT TO SEE WHAT ELSE DISSOLVES                 POR26190
*==========WITH CURRENT PARTICLE                                       POR26200
      IF (KCOUNT.EQ.I2TWO) THEN                                         POR26210
        DO 10 IGO=1,4                                                   POR26220
*          WE KNOW THERE ARE TWO NEIGHBORS - FIND THEM!                 POR26230
           IF (ILIST(IGO) .EQ. I2ZERO) GOTO 10                          POR26240
*          FOUND ONE;  HOW MANY NEIGHBORS DOES THIS NEIGHBOR HAVE?      POR26250
           IP = ILIST(IGO)                                              POR26260
           IQ = JLIST(IGO)                                              POR26270
           CALL COUNT (IP,IQ,KCOUNT,JCOUNT,1)                           POR26280
           IF (KCOUNT.EQ.I2ZERO) THEN                                   POR26290
```

```
                A(IP,IQ) = -I2ONE                                       POR26300
*               DONT LOOK FOR OTHER NEIGHBORS - THEY GOTTA BE STUCK     POR26310
                GOTO 15                                                 POR26320
             ELSE                                                       POR26330
                CALL CLUSTR (IP,IQ,IDIS)                                POR26340
             ENDIF                                                      POR26350
  10      CONTINUE                                                      POR26360
        ELSE                                                            POR26370
*        THERE ARE THREE NEIGHBORS FOR THE DISSOLVED PARTICLE           POR26380
         DO 12 IGO=1,4                                                  POR26390
*           WE KNOW THERE ARE THREE NEIGHBORS - FIND THEM!              POR26400
            IF (ILIST(IGO) .EQ. I2ZERO) GOTO 12                         POR26410
*           HOW MANY NEIGHBORS DOES THIS NEIGHBOR HAVE?                 POR26420
            IP = ILIST(IGO)                                             POR26430
            IQ = JLIST(IGO)                                             POR26440
            CALL COUNT (IP,IQ,KCOUNT,JCOUNT,1)                          POR26450
            IF (KCOUNT.EQ.I2ZERO) THEN                                  POR26460
                A(IP,IQ) = -I2ONE                                       POR26470
            ELSE                                                        POR26480
                IF (KCOUNT.LT.I2THRE) CALL CLUSTR (IP,IQ,IDIS)          POR26490
            ENDIF                                                       POR26500
  12      CONTINUE                                                      POR26510
*=======  END OF ATTEMPT TO SEE HOW MUCH DISSOLVES WITH CURR PARTICLE    POR26520
        ENDIF                                                           POR26530
*%%%%%%%% END OF ATTEMPT TO DISLODGE STUCK PARTICLE                     POR26540
        ENDIF                                                           POR26550
*+++++++++IS THE PARTICLE FREE?  THEN TRY TO MOVE IT & THEN PRECIPITATE  POR26560
 15     IF (A(IX,IY).LT.I2ONE) THEN                                     POR26570
          IF (GGUBFS(DSEED).LE.PROB3) CALL MOVE                         POR26580
C IF A PARTICLE IS LOST THEN ANOTHER ONE TAKES ITS PLACE AND IS CHECKED POR26590
          IF (ITRY.LT.0) GOTO 5                                         POR26600
*  AFTER EACH ADJACENT PARTICLE IS CHECKED, FIND OUT IF IT'S STUCK YET  POR26610
*    PERIODIC BOUNDARY CONDITION ON THE X-AXIS                         POR26620
*    RIGHT HAND SIDE                                                   POR26630
              IQ=IX+1                                                   POR26640
          IF (IX.EQ.NXSPAC) IQ=1                                        POR26650
            IF (A(IQ,IY).GT.I2ZERO) GOTO 950                            POR26660
*         LEFT HAND SIDE                                                POR26670
              IQ=IX-1                                                   POR26680
          IF (IQ.EQ.0) IQ=NXSPAC                                        POR26690
            IF (A(IQ,IY).GT.I2ZERO) GOTO 950                            POR26700
*                                                                       POR26710
*     TOWARDS TOP - NON-NUCLEATING SIDE                                 POR26720
          IF (IY.LT.NYSPAC) THEN                                        POR26730
            IF (A(IX,IY+1).GT.I2ZERO) GOTO 950                          POR26740
          ENDIF                                                         POR26750
*     BOTTOM - TOWARDS NUCLEATING PLANE                                 POR26760
          IF (IY.GT.1)      THEN                                        POR26770
            IF (A(IX,IY-1).GT.I2ZERO) GOTO 950                          POR26780
C IF IT NEIGHBORS THE NUCLEATING PLANE, TRY TO STICK IT                 POR26790
```

```
        ELSE                                                              POR26800
           GOTO 950                                                       POR26810
        ENDIF                                                             POR26820
C  NO STUCK NEIGHBORS, SO GO TO THE NEXT PARTICLE                        POR26830
        GOTO 1000                                                         POR26840
*********  ATTEMPT TO STICK                                               POR26850
 950  R=GGUBFS(DSEED)                                                    POR26860
      IF (R.LT.PROB)  THEN                                              POR26870
*        PARTICLE HAS STUCK                                             POR26880
         A(IX,IY)=I2ONE                                                 POR26890
C ADJUSTING IDIS FOR PRECIPITATION                                      POR26900
         IF (ICLUST.GT.0) THEN                                          POR26910
             IDIS = -I2ONE                                              POR26920
             CALL CLUSTR (IX,IY,IDIS)                                   POR26930
         ENDIF                                                          POR26940
        ENDIF                                                           POR26950
C A 'SHAPE CHECK' WILL BE INTRODUCED HERE                               POR26960
C     IF (A(IX,IY).GT.I2ZERO) CALL SHCHEK (                )            POR26970
C                                                                       POR26980
*+++++++++  END OF FREE PARTICLE MOVE AND PRECIPITATE                    POR26990
      ENDIF                                                             POR27000
 1000 CONTINUE                                                          POR27010
      RETURN                                                            POR27020
      END                                                               POR27030
                                                                        POR27040
                                                                        POR27050
                                                                        POR27060
        SUBROUTINE IMOVE(A,B,BGROW,I,IX,IY,DSEED,ITRY)                  POR27070
      INTEGER*2 I2ZERO,I2ONE,IDIS                                       POR27080
       PARAMETER (I2ZERO=0,I2ONE=1)                                     POR27090
      COMMON /SIZPRM/  NXSPAC,NYSPAC,MXNPRT,NPART,NNUC,NLAY             POR27100
      COMMON /PARAM/ NPARTO,PROB,PROB2,PROB3,CONC,IZ,MXIY,ICLUST,KGEN,  POR27110
     1 MNIY,MXFRE,IAMBLK                                                POR27120
        INTEGER*4 NXSPAC,NYSPAC,MXNPRT,NPART,BGROW(MXNPRT)              POR27130
        INTEGER*2 A(NXSPAC,NYSPAC),B(MXNPRT,2)                          POR27140
        REAL*8 DSEED                                                    POR27150
        RETURN                                                          POR27160
*                                                                       POR27170
      ENTRY MOVE                                                        POR27180
       ITRY = 0                                                         POR27190
       IX=B(I,1)                                                        POR27200
       IY=B(I,2)                                                        POR27210
C    CHOOSE A DIRECTION                                                 POR27220
       IDIR=(4*GGUBFS(DSEED)+1.)                                        POR27230
       GOTO (1000,4000,2000,3000) IDIR                                  POR27240
*   PERIODIC BOUNDARY CONDITIONS ON X AXIS                              POR27250
 1000  IQ=IX+1                                                          POR27260
       IF (IQ.GT.NXSPAC) IQ=1                                           POR27270
         IF (A(IQ,IY).EQ.I2ZERO) THEN                                   POR27280
```

```
            A(IQ,IY)=-I2ONE                                                  POR27290
C EMPTY THE PREVIOUS POSITION                                                POR27300
            A(IX,IY)=I2ZERO                                                  POR27310
C CHANGE COORD OF THE I PARTICLE                                             POR27320
            B(I,1)=IQ                                                        POR27330
            IX=IQ                                                            POR27340
          ENDIF                                                              POR27350
        RETURN                                                               POR27360
 2000   IQ=IX-1                                                             POR27370
         IF (IQ.LT.1) IQ=NXSPAC                                              POR27380
          IF (A(IQ,IY).EQ.I2ZERO) THEN                                       POR27390
            A(IQ,IY)=-I2ONE                                                  POR27400
C EMPTY THE PREVIOUS POSITION                                                POR27410
            A(IX,IY)=I2ZERO                                                  POR27420
C CHANGE COORD OF THE I PARTICLE                                             POR27430
            B(I,1)=IQ                                                        POR27440
            IX=IQ                                                            POR27450
          ENDIF                                                              POR27460
        RETURN                                                               POR27470
C Y-AXIS MOVEMENT                                                            POR27480
 3000   IF (IY+1.LE.NYSPAC) THEN                                            POR27490
          IF (A(IX,IY+1).EQ.I2ZERO) THEN                                     POR27500
            A(IX,IY+1)=-I2ONE                                                POR27510
C EMRTY THE PREVIOUS POSITION                                                POR27520
            A(IX,IY)=I2ZERO                                                  POR27530
C CHANGE COORD OF THE I PARTICLE                                             POR27540
            B(I,2)=IY+1                                                      POR27550
            IY=IY+1                                                          POR27560
          ENDIF                                                              POR27570
        ELSE                                                                 POR27580
*         TOP EDGE -- LET IT MOVE OUT AND DISAPEAR!!                         POR27590
*    BUT ONLY IF THERE IS NO IMAGINARY PARTICLE ADJOINING IT                 POR27600
C         R=GGUBFS(DSEED)                                                    POR27610
C         IF (R.GT.0) THEN                                                   POR27620
*           OK, LET IT GET LOST                                              POR27630
            A(IX,IY) = I2ZERO                                                POR27640
*   NOTE THAT THIS PARTICLE IS NOT IN THE ITH POSITION ANYMORE               POR27650
*   SO FILL THAT WITH THE NPART'TH PARTICLE AND DECREMENT NPART              POR27660
            B(I,1) = B(NPART,1)                                              POR27670
            B(I,2) = B(NPART,2)                                              POR27680
            IX = B(NPART,1)                                                  POR27690
            IY = B(NPART,2)                                                  POR27700
            BGROW(I)=BGROW(NPART)                                            POR27710
            NPART = NPART - 1                                                POR27720
*   INDICATE THAT MOVING THE ITH PARTICLE WAS NOT SUCCESSFUL (IT'S LOST)     POR27730
            ITRY = - 1                                                       POR27740
C         ENDIF                                                              POR27750
        ENDIF                                                                POR27760
        RETURN                                                               POR27770
```

```
 4000  IF (IY-1.GE.1)      THEN                                          POR27780
         IF (A(IX,IY-1).EQ.I2ZERO) THEN                                POR27790
            A(IX,IY-1)=-I2ONE                                          POR27800
C EMPTY THE PREVIOUS POSITION                                          POR27810
            A(IX,IY)=I2ZERO                                            POR27820
C CHANGE COORD OF THE I PARTICLE                                       POR27830
            B(I,2)=IY-1                                                POR27840
            IY=IY-1                                                    POR27850
          ENDIF                                                        POR27860
        ENDIF                                                          POR27870
        RETURN                                                         POR27880
         END                                                           POR27890
                                                                       POR27900
       SUBROUTINE INCLUS (A,D)                                         POR27910
       INTEGER*2 I2ZERO,I2FIVE,I2ONE,IDIS                              POR27920
       PARAMETER (I2ZERO=0,I2ONE=1,I2FIVE=5)                           POR27930
       COMMON /S1ZPRM/  NXSPAC,NYSPAC,MXNPRT,NPART,NNUC,NLAY           POR27940
       COMMON /PARAM/ NPARTO,PROB,PROB2,PROB3,CONC,IZ,MXIY,ICLUST,KGEN, POR27950
      1 MNIY,MXFRE,IAMBLK                                              POR27960
         INTEGER*4 NXSPAC,NYSPAC,MXNPRT,NPART ,ICNTR(-1:1)             POR27970
C D(NPART,2) DEPOSITORY FOR STICKING CANDIDATES                        POR27980
       INTEGER*2 A(NXSPAC,NYSPAC),D(NPART,2)                           POR27990
       RETURN                                                          POR28000
*                                                                      POR28010
       ENTRY CLUSTR (IX,IY,IDIS)                                       POR28020
*    INITIALIZE A SEARCH WINDOW IN LOCAL REGION OF SPACE WHERE CLUSTER ISPOR28030
*    THIS WINDOW IS EXPANDED AS THE CLUSTER SEARCH GROWS               POR28040
       IXMIN = IX                                                      POR28050
       IXMAX = IX                                                      POR28060
       IYMIN = IY                                                      POR28070
       IYMAX = IY                                                      POR28080
C                                                                      POR28090
        JCLUST=1                                                       POR28100
        D(1,1)=IX                                                      POR28110
        D(1,2)=IY                                                      POR28120
        A(IX,IY) = I2FIVE                                              POR28130
                                                                       POR28140
****************  BEGIN SCAN TO FIND  5'S IN OUR SEARCH WINDOW ******** POR28150
 50     IFOUND=0                                                       POR28160
                                                                       POR28170
        DO 100 I=IXMIN,IXMAX                                           POR28180
        DO 100 J=IYMIN,IYMAX                                           POR28190
C CHECK WHETHER A NEWLY CONSIDERED  PARTICLE IS THERE                  POR28200
          IF (A(I,J).NE.I2FIVE) GOTO 100                               POR28210
C IF YES, AND A PARTICLE NEIGHBORS IT, STICK THE NEW                   POR28220
         IFOUND=1                                                      POR28230
           IQ=I+1                                                      POR28240
       IF (I.EQ.NXSPAC) IQ=1                                           POR28250
        IF (A(IQ,J).EQ.IDIS) THEN                                      POR28260
```

```
            A(IQ,J)=I2FIVE                                              POR28270
             JCLUST=JCLUST+1                                            POR28280
             D(JCLUST,1)=IQ                                             POR28290
             D(JCLUST,2)=J                                              POR28300
             IXMAX = MAX(IQ,IXMAX)                                      POR28310
             IF (IQ.EQ.1) THEN                                          POR28320
               IXMAX=NXSPAC                                             POR28330
               IXMIN=1                                                  POR28340
             ENDIF                                                      POR28350
          ENDIF                                                         POR28360
*     LEFT  HAND SIDE                                                   POR28370
          IQ=I-1                                                        POR28380
        IF (IQ.EQ.0) IQ=NXSPAC                                          POR28390
         IF (A(IQ,J).EQ.IDIS) THEN                                      POR28400
             A(IQ,J)=I2FIVE                                             POR28410
              JCLUST=JCLUST+1                                           POR28420
              D(JCLUST,1)=IQ                                            POR28430
              D(JCLUST,2)=J                                             POR28440
              IXMIN = MIN(IQ,IXMIN)                                     POR28450
              IF (IQ.EQ.NXSPAC) THEN                                    POR28460
                IXMAX=NXSPAC                                            POR28470
                IXMIN=1                                                 POR28480
              ENDIF                                                     POR28490
          ENDIF                                                         POR28500
C     Y AXIS                                                            POR28510
          IF (J.EQ.NYSPAC) GOTO 75                                      POR28520
             IQ=J+1                                                     POR28530
          IF (A(I,IQ).EQ.IDIS) THEN                                     POR28540
              A(I,IQ)=I2FIVE                                            POR28550
              JCLUST=JCLUST+1                                           POR28560
              D(JCLUST,1)=I                                             POR28570
              D(JCLUST,2)=IQ                                            POR28580
              IYMAX = MAX(IQ,IYMAX)                                     POR28590
          ENDIF                                                         POR28600
   75     IF (J.EQ.1) GOTO 90                                           POR28610
             IQ=J-1                                                     POR28620
          IF (A(I,IQ).EQ.IDIS) THEN                                     POR28630
              A(I,IQ)=I2FIVE                                            POR28640
              JCLUST=JCLUST+1                                           POR28650
              D(JCLUST,1)=I                                             POR28660
              D(JCLUST,2)=IQ                                            POR28670
              IYMIN = MIN(IQ,IYMIN)                                     POR28680
          ENDIF                                                         POR28690
C NOW EMPTY THE CHECKED SPOT, WITHOUT EMPTYING A                        POR28700
 90     A(I,J)=-IDIS                                                    POR28710
 100     CONTINUE                                                       POR28720
*                                                                       POR28730
*****************  END WINDOW SCAN FOR 5'S ***************************** POR28740
                                                                        POR28750
C IF IN THIS SCAN A CONNECTION HAS BEEN FOUND, DON'T CHECK FURTHER      POR28760
```

```
      IF (IYMIN.LT.IAMBLK.AND.IDIS.EQ.I2ONE) GOTO 240                    POR28770
C NOW CHECK WHETHER NEW PARTICLES STUCK TO THE CLUSTER                    POR28780
      IF (IFOUND.NE.0) GOTO 50                                           POR28790
 240  IF (IDIS.GT.I2ZERO)   THEN                                         POR28800
*   WE ARE TRYING TO DISSOLVE - IS CLUSTER CONNECTED TO BULK?            POR28810
      IF (IYMIN.LT.IAMBLK) THEN                                          POR28820
*   YUP - PART OF BULK - LEAVE THEM STUCK                                POR28830
            DO 500 II=1,JCLUST                                           POR28840
            A(D(II,1),D(II,2))=I2ONE                                     POR28850
 500        CONTINUE                                                     POR28860
      ELSE                                                               POR28870
C IF NOT CONNECTED TO THE BULK, DISSOLVE IT                              POR28880
            DO 550 II=1,JCLUST                                           POR28890
            A(D(II,1),D(II,2))=-I2ONE                                    POR28900
 550        CONTINUE                                                     POR28910
      ENDIF                                                              POR28920
     ELSE                                                                POR28930
*    WE ARE TRYING TO STICK CLUSTER - IS IT BIG ENOUGH?                  POR28940
        IF (JCLUST.LT.ICLUST) THEN                                       POR28950
C UN-STICK THE ALREADY STUCK PARTICLES INCLUDING THE INITIAL ONE         POR28960
C STORED IN D(1,1)                                                       POR28970
          DO 250 I=1,JCLUST                                              POR28980
            A(D(I,1),D(I,2))=-I2ONE                                      POR28990
 250      CONTINUE                                                       POR29000
       ELSE                                                              POR29010
          DO 255 I=1,JCLUST                                              POR29020
            A(D(I,1),D(I,2))=I2ONE                                       POR29030
 255      CONTINUE                                                       POR29040
       ENDIF                                                             POR29050
      ENDIF                                                              POR29060
      RETURN                                                             POR29070
      END                                                                POR29080
                                                                         POR29090
                                                                         POR29100
                                                                         POR29110
      SUBROUTINE IDISP  (A,AS,B,BGROW,DEVICE, REV,OLDGEN,ZMANO,CONUC     POR29120
     1,COR1,COR2,IORD,NSURF1,IDNAM,ZMAN,DCONC,SCONC,IPTO,MAXR)           POR29130
      INTEGER*2 I2ZERO,I2ONE,I2THRE,I2FOUR,IDIS,I2EIGT                   POR29140
       PARAMETER (I2ZERO=0,I2ONE=1,I2THRE=3,I2FOUR=4,I2EIGT=8)           POR29150
      COMMON /SIZPRM/  NXSPAC,NYSPAC,MXNPRT,NPART,NNUC,NLAY              POR29160
      COMMON /PARAM/ NPARTO,PROB,PROB2,PROB3,CONC,IZ,MXIY,ICLUST,KGEN,   POR29170
     1 MNIY,MXFRE,IAMBLK                                                 POR29180
      REAL*4 DCONC(10000),SCONC(10000),COR1(100000),COR2(100000)         POR29190
      REAL*8 ZMAN,GETTIM,GOTMIN,ZMANO,ZMANOW,DELTAT                      POR29200
      INTEGER*4 NXSPAC,NYSPAC,BGROW(MXNPRT)                              POR29210
      INTEGER*2 A(NXSPAC,NYSPAC),B(MXNPRT,2),ICNTR(-1:1),KCOUNT,JCOUNT   POR29220
      INTEGER*2 AS(NXSPAC,NYSPAC)                                        POR29230
      INTEGER*2 MODE,IPALET,ICOLOR(3),ICLR,II,JJ,IZERO,IFIVH             POR29240
      INTEGER*4 NPART,MXGEN,KGEN                                         POR29250
```

```
      INTEGER*2 ITWO,DEVICE,IONE,MAXR                                   POR29260
       CHARACTER*1 GRAP(-1:5)                                           POR29270
       CHARACTER*8 IDNAM                                                POR29280
       CHARACTER*130 CPLOT,CBLANK                                       POR29290
       DATA CBLANK /'                                                   POR29300
     1                                                                  POR29310
     1             '/                                                   POR29320
       DATA GRAP /'+',' ','*','?','#','-','!'/                         POR29330
      DATA MODE,IPALET,ICOLOR /1,0,1,0,2/                               POR29340
      CALL ICORE (AS,MAXR,NSURF1,COR1,COR2,IORD)                        POR29350
      RETURN                                                            POR29360
                                                                        POR29370
         ENTRY DISPLA (NSURF)                                           POR29380
          DO 10 I=1,NXSPAC                                              POR29390
          DO 10 J=1,NYSPAC                                              POR29400
  10      AS(I,J) = I2ZERO                                              POR29410
*     FIND AVERAGE EXECUTION TIME FOR SINGLE GENERATION                 POR29420
*  HOW LONG SIMULATION RUN FROM BEGINNING TO NOW?                       POR29430
         ZMANOW = GETTIM(GOTMIN)                                        POR29440
         IF (ZMANOW.LT.ZMANO) THEN                                      POR29450
             WRITE (*,*) 'ACCOUNTING BOUNDARY CROSSED!'                 POR29460
             ZMANO = ZMANOW                                             POR29470
         ENDIF                                                          POR29480
         DELTAT = ZMANOW - ZMANO                                        POR29490
      ZMAN = ZMAN + DELTAT                                              POR29500
      ZMANO = ZMANOW                                                    POR29510
* ZMAN IN MINUTES                                                      POR29520
      EZMAN = ZMAN / 60.DO                                              POR29530
      DELTAG = KGEN - OLDGEN                                            POR29540
      SPEED = 0.0                                                       POR29550
                                                                        POR29560
      IF (DELTAG.NE.0.) SPEED = DELTAT / DELTAG                         POR29570
      OLDGEN = KGEN                                                     POR29580
                                                                        POR29590
*   UPDATE THE MAXIMUM HEIGHT COUNTER                                   POR29600
      DO 400 J=NYSPAC,1,-1                                              POR29610
         DO 400 I=1,NXSPAC                                              POR29620
            IF (A(I,J) .EQ. I2ONE) GOTO 410                             POR29630
 400     CONTINUE                                                       POR29640
 410     MXIY = J                                                       POR29650
*    UPDATE MNIY COUNTER                                                POR29660
      DO 500 J = 1,NYSPAC                                               POR29670
         ICNT = 0                                                       POR29680
         DO 490 I = 1,NXSPAC                                            POR29690
             ICNT = ICNT + A(I,J)                                       POR29700
  490    CONTINUE                                                       POR29710
         IF (ICNT.LT.NXSPAC) GOTO 510                                   POR29720
  500 CONTINUE                                                          POR29730
  510 MNIY = J                                                          POR29740
C NSURF IS THE # OF INTERFACES EXPOSED BY THE BULK                      POR29750
```

```
          NSURF = 0                                                              POR29760
C NSURF1 IS THE # OF SURFACE POINTS                                              POR29770
          NSURF1 = 0                                                             POR29780
          DO  300 I = 1,NPART                                                    POR29790
             IX = B(I,1)                                                         POR29800
             IY = B(I,2)                                                         POR29810
             IF (A(IX,IY).GT.I2ZERO .AND. IY.GE.MNIY-2) THEN                     POR29820
                CALL COUNT (IX,IY,KCOUNT,JCOUNT,2)                               POR29830
C THE TRUE 'ROUGHNESS' IS THE # OF INTERFACES STUCK/REST OF THE WORLD            POR29840
C (FREE PART OR EMPTY SPACE)                                                     POR29850
                NSURF=NSURF+(4-KCOUNT)                                           POR29860
       KJCNT = KCOUNT + JCOUNT                                                   POR29870
       IF (KJCNT.LT.I2EIGT.AND.KJCNT.GT.I2ZERO) THEN                             POR29880
             NSURF1=NSURF1+1                                                     POR29890
C FILLING THE "SURFACE MAP" AS(I,J)                                              POR29900
C 1 IS ADDED TO 4-KCOUNT TO AVOID 0 FOR 4-COORDINATED SURFACE PARTICLES          POR29910
             AS(IX,IY) = I2FOUR-KCOUNT+I2ONE                                     POR29920
          ENDIF                                                                  POR29930
       ENDIF                                                                     POR29940
  300    CONTINUE                                                               POR29950
C CALCULATE THE CORRELATION FUNCTION                                             POR29960
       CALL CORE                                                                 POR29970
*  NOW DISPLAY -                                                                 POR29980
      GOTO (1000,2000,3000,4000) DEVICE                                          POR29990
*   DEVICE 0 - JUST RETURN                                                       POR30000
      RETURN                                                                     POR30010
*--------- PC VERSION:                                                          POR30020
                                                                                 POR30030
 1000  CONTINUE                                                                  POR30040
*      CALL CLS                                                                  POR30050
       WRITE (*,*) '                                    ',KGEN,NPARTO,NPART      POR30060
                                                                                 POR30070
*      DO 1115 KK=1,NPART                                                        POR30080
*         ICLR = ICOLOR(A(B(KK,1),B(KK,2))+2)                                    POR30090
*         CALL PSET(B(KK,1),B(KK,2),ICLR)                                        POR30100
*1115    CONTINUE                                                               POR30110
        RETURN                                                                   POR30120
                                                                                 POR30130
*--------- MAINFRAME VERSION - TEK SCREEN                                       POR30140
 2000 IONE = 1                                                                   POR30150
      ICLR = 1                                                                   POR30160
      ITWO = 2                                                                   POR30170
      IZERO = 0                                                                  POR30180
      IFIVH = 500                                                                POR30190
*     CALL INIT (IZERO,TWO)                                                      POR30200
*     II = 1                                                                     POR30210
*     JJ = 2 * (NYSPAC + 5)                                                      POR30220
*     CALL PLOT (II,JJ,IZERO)                                                    POR30230
*     II = 500                                                                   POR30240
*     CALL PLOT (II,JJ,ITWO)                                                     POR30250
```

```
*     JJ = 3 *NYSPAC                                                    POR30260
*     DO 2115 J=1,NYSPAC                                                POR30270
*     DO 2115 I=1,NXSPAC                                                POR30280
*        II =7*I                                                        POR30290
*        JJ=5*J                                                         POR30300
*        IF (A(I,J).EQ.1) CALL PLOT(II,JJ,IONE)                         POR30310
*2115    CONTINUE                                                       POR30320
*     CALL TSET(IFIVH,IONE,ITWO)                                        POR30330
      IO = 6                                                            POR30340
      ASSIGN 4010 TO IFORM                                              POR30350
      GOTO 9000                                                         POR30360
* ---- MAINFRAME -- PRINTER (NO GRAPHICS SCREEN)                        POR30370
 3000 IO = 3                                                            POR30380
      ASSIGN 4015 TO IFORM                                              POR30390
      GOTO 9000                                                         POR30400
* ---- MAINFRAME --  SCREEN ONLY                                        POR30410
 4000 IO = 6                                                            POR30420
      ASSIGN 4010 TO IFORM                                              POR30430
9000       AVGHT = FLOAT(IZ) / FLOAT(NXSPAC)                            POR30440
      PORO=100.*(1. - AVGHT/MXIY)                                       POR30450
      RUFN=FLOAT(NSURF)/FLOAT(NXSPAC)                                   POR30460
      WRITE (IO,4105) IDNAM,EZMAN,SPEED,REV,KGEN                        POR30470
      WRITE (IO,4120) NXSPAC,NYSPAC,NLAY,NPARTO,ICLUST,PROB,PROB2,PROB3,POR30480
     1CONC,CONUC                                                        POR30490
      WRITE (IO,4140) AVGHT,MXIY,MNIY,PORO,RUFN                         POR30500
      WRITE (IO,4150) NPART,IZ,NPART-IZ                                 POR30510
      IF (DEVICE.LT.2) RETURN                                           POR30520
      IF (DEVICE.EQ.4.AND.NXSPAC.GT.78) GOTO 4004                       POR30530
      IF (DEVICE.EQ.3.AND.NXSPAC.GT.130) GOTO 4004                      POR30540
C FOR QUIK DISPLAY ON SCREEN, SHOW ONLY THE 'ROUGH' PART                POR30550
*   ALWAYS SHOW ONLY UP TO THE MAXIMUM PARTICLE HEIGHT                  POR30560
      MNJ=MNIY-1                                                        POR30570
      MXJ = MAX (MXFRE,MXIY)                                            POR30580
      IF (IO.EQ.3) MNJ=1                                                POR30590
       WRITE (IO,IFORM) ((GRAP(A(I,J)),I=1,NXSPAC),J=MNJ,MXJ)           POR30600
       WRITE (IO,IFORM) ((GRAP(AS(I,J)),I=1,NXSPAC),J=MNJ,MXJ)          POR30610
 4004 WRITE (IO,4006)                                                   POR30620
 4006 FORMAT ('1')                                                      POR30630
       WRITE (IO,*) '0.           ---- CONCENTRATION------             POR30640
     1             .5                                                 POR30650
     1---------1.'                                                      POR30660
       WRITE (IO,*) '-----------------------------------------------------POR30670
     1------------|-----------------------------------------------------POR30680
     1-----------'                                                      POR30690
      IPT  = DCONC(1)                                                   POR30700
      IEND = 130                                                        POR30710
      ASSIGN 4008 TO IFORM                                              POR30720
      IF (DEVICE.EQ.4) THEN                                             POR30730
          IEND = 80                                                     POR30740
```

```
         ASSIGN 4007 TO IFORM                                                  POR30750
      ENDIF                                                                    POR30760
      DO 4009 I = IPTO+1,IPT                                                   POR30770
        IPOS = DCONC(I)*(IEND-1)+1                                             POR30780
C IF IPOS IS TOO BIG, IPOS+2 IS LARGER THAN IEND AND NEEDS ADJUSTMENT          POR30790
        IF (IPOS.EQ.IEND) IPOS=IEND-2                                          POR30800
            CPLOT = CBLANK(1:IPOS) // '*' // CBLANK(IPOS+2:IEND)               POR30810
            IPOS = SCONC(I)*(IEND-1) + 1                                       POR30820
        IF (IPOS.EQ.IEND) IPOS=IEND-2                                          POR30830
            CPLOT = CPLOT(1:IPOS) // '+' // CPLOT(IPOS+2:IEND)                 POR30840
        WRITE (IO,IFORM) CPLOT                                                 POR30850
 4007   FORMAT (1X,A80)                                                        POR30860
 4008   FORMAT (1X,A130)                                                       POR30870
 4009 CONTINUE                                                                 POR30880
 4010  FORMAT (' ',50A1)                                                       POR30890
 4015  FORMAT (' ',050A1)                                                      POR30900
 4105 FORMAT ('1',A8,' TIME: ',F8.2,' MIN.',' SPEED=',F8.3,'SEC/GEN',          POR30910
     1' VERSION#',F5.3,' GEN#',I7)                                             POR30920
 4120 FORMAT (' ','LAT. SZE',I5,' X',I5,' #LYRS',I5,' NPRTO',I7,               POR30930
     1' CLST SZE',I4,' PRB (PP,DIS,MOVE)',3F5.2                                POR30940
     1,' CONC',F5.3,' CONUC',F5.3)                                             POR30950
 4140 FORMAT (' AVG.HT=',F7.3,' MAXHT=',I5,' MNHT=',I5,' PORO=',F7.3,          POR30960
     1' ROUGH=',F9.3)                                                          POR30970
 4150 FORMAT (' TOTAL # ',I8,' # STUCK=',I8,' # FREE=',I8)                     POR30980
      RETURN                                                                   POR30990
      END                                                                      POR31000
                                                                               POR31010
       SUBROUTINE ICOUNT (A)                                                   POR31020
      INTEGER*2 I2ZERO,I2ONE,IDIS,KCOUNT,ICNTR(-1:1),JCOUNT,I2TWO              POR31030
       PARAMETER (I2ZERO=0,I2ONE=1,I2TWO=2)                                    POR31040
      COMMON /SIZPRM/  NXSPAC,NYSPAC,MXNPRT,NPART,NNUC,NLAY                    POR31050
      COMMON /PARAM/ NPARTO,PROB,PROB2,PROB3,CONC,IZ,MXIY,ICLUST,KGEN,         POR31060
     1 MNIY,MXFRE,IAMBLK                                                       POR31070
        INTEGER*4 NXSPAC,NYSPAC,MXNPRT,NPART                                   POR31080
        INTEGER*2 A(NXSPAC,NYSPAC)                                             POR31090
      RETURN                                                                   POR31100
C                                                                              POR31110
      ENTRY COUNT (IX,IY,KCOUNT,JCOUNT,JUMP)                                   POR31120
       KCOUNT=I2ZERO                                                           POR31130
       JCOUNT=I2ZERO                                                           POR31140
       ICNTR(1)=I2ZERO                                                         POR31150
           ICNTR(-1) = I2ZERO                                                  POR31160
           ICNTR(0)  = I2ZERO                                                  POR31170
C LOOK TO THE RIGHT                                                            POR31180
       IQ=IX+1                                                                 POR31190
       IF (IX.EQ.NXSPAC) IQ=1                                                  POR31200
       ICNTR(A(IQ,IY)) = ICNTR(A(IQ,IY)) + I2ONE                               POR31210
C LOOK TO THE LEFT                                                             POR31220
       IQ=IX-1                                                                 POR31230
```

```
      IF (IQ.EQ.0) IQ=NXSPAC                                          POR31240
      ICNTR(A(IQ,IY)) = ICNTR(A(IQ,IY)) + I2ONE                       POR31250
C LOOK UP                                                             POR31260
      IQ=IY+1                                                         POR31270
C THE SPACE IS CONSIDERED VOID OF 'SOLUTE' OUTSIDE THE BOX            POR31280
      IF (IY.EQ.NYSPAC) GOTO 100                                      POR31290
      ICNTR(A(IX,IQ)) = ICNTR(A(IX,IQ)) + I2ONE                       POR31300
C LOOK DOWN                                                           POR31310
100   IQ=IY-1                                                         POR31320
C THE BOX IS INFINITELY FULL UNDER THE BOTTOM                         POR31330
      IF (IQ.EQ.0) THEN                                               POR31340
          ICNTR(1) = ICNTR(1) + I2ONE                                 POR31350
          KCOUNT = ICNTR(1)                                           POR31360
          GOTO 150                                                    POR31370
      ENDIF                                                           POR31380
      ICNTR(A(IX,IQ)) = ICNTR(A(IX,IQ)) + I2ONE                       POR31390
      KCOUNT = ICNTR(1)                                               POR31400
 150  IF (JUMP.EQ.1) RETURN                                           POR31410
C LOOKING ON THE DIAGONAL                                             POR31420
C ICNTR SHOULD BE RESET                                               POR31430
      ICNTR(1)=I2ZERO                                                 POR31440
          ICNTR(-1) = I2ZERO                                          POR31450
          ICNTR(0)  = I2ZERO                                          POR31460
C NORTH-EAST                                                          POR31470
      IQX=IX+1                                                        POR31480
      IQY=IY+1                                                        POR31490
      IF (IY.EQ.NYSPAC) GOTO 200                                      POR31500
      IF (IX.EQ.NXSPAC) IQX=1                                         POR31510
      ICNTR (A(IQX,IQY))=ICNTR(A(IQX,IQY)) + I2ONE                    POR31520
C NORTH-WEST                                                          POR31530
      IQX=IX-1                                                        POR31540
      IQY=IY+1                                                        POR31550
      IF (IQX.EQ.0) IQX=NXSPAC                                        POR31560
      ICNTR (A(IQX,IQY))=ICNTR(A(IQX,IQY)) + I2ONE                    POR31570
C SOUTH-EAST                                                          POR31580
 200  IQX=IX+1                                                        POR31590
      IQY=IY-1                                                        POR31600
      IF (IQY.EQ.0) GOTO 300                                          POR31610
      IF (IX.EQ.NXSPAC) IQX=1                                         POR31620
      ICNTR (A(IQX,IQY))=ICNTR(A(IQX,IQY)) + I2ONE                    POR31630
C SOUTH-WEST                                                          POR31640
      IQX=IX-1                                                        POR31650
      IQY=IY-1                                                        POR31660
      IF (IX.EQ.1) IQX=NXSPAC                                         POR31670
      ICNTR (A(IQX,IQY))=ICNTR(A(IQX,IQY)) + I2ONE                    POR31680
      GOTO 350                                                        POR31690
300   ICNTR(1)=ICNTR(1)+I2TWO                                         POR31700
350   JCOUNT=ICNTR(1)                                                 POR31710
400   RETURN                                                          POR31720
```

```
      END                                                                    POR31730
                                                                             POR31740
                                                                             POR31750
      SUBROUTINE INEIGH (A)                                                  POR31760
      INTEGER*2 I2ZERO,I2ONE,IDIS,ILIST(4),JLIST(4)                          POR31770
      PARAMETER (I2ZERO=0,I2ONE=1)                                           POR31780
      COMMON /SIZPRM/  NXSPAC,NYSPAC,MXNPRT,NPART,NNUC,NLAY                  POR31790
      COMMON /PARAM/ NPARTO,PROB,PROB2,PROB3,CONC,IZ,MXIY,ICLUST,KGEN,       POR31800
     1 MNIY,MXFRE,IAMBLK                                                     POR31810
        INTEGER*4 NXSPAC,NYSPAC,MXNPRT,NPART                                 POR31820
        INTEGER*2 A(NXSPAC,NYSPAC)                                           POR31830
      RETURN                                                                 POR31840
C                                                                            POR31850
      ENTRY NEIGH (IX,IY,ILIST,JLIST)                                        POR31860
*                                                                            POR31870
*      RETURNS THE COORDINATES OF THE FOUR NEIGHBORS IF THEY EXIST           POR31880
*      1- TOP   2- RIGHT  3-BOTTOM  4-LEFT                                   POR31890
C LOOK TO THE RIGHT                                                          POR31900
      IQ=IX+1                                                                POR31910
      IF (IX.EQ.NXSPAC) IQ=1                                                 POR31920
      IF (A(IQ,IY).EQ.I2ONE) THEN                                            POR31930
          ILIST(2) = IQ                                                      POR31940
          JLIST(2) = IY                                                      POR31950
      ELSE                                                                   POR31960
          ILIST(2) = I2ZERO                                                  POR31970
      ENDIF                                                                  POR31980
C LOOK TO THE LEFT                                                           POR31990
      IQ=IX-1                                                                POR32000
      IF (IQ.EQ.0) IQ=NXSPAC                                                 POR32010
      IF (A(IQ,IY).EQ.I2ONE) THEN                                            POR32020
          ILIST(4) = IQ                                                      POR32030
          JLIST(4) = IY                                                      POR32040
      ELSE                                                                   POR32050
          ILIST(4) = I2ZERO                                                  POR32060
      ENDIF                                                                  POR32070
C LOOK UP                                                                    POR32080
      IQ=IY+1                                                                POR32090
C THE SPACE IS CONSIDERED VOID OF 'SOLUTE' OUTSIDE THE BOX                   POR32100
      ILIST(1) = I2ZERO                                                      POR32110
      IF (IY.LT.NYSPAC) THEN                                                 POR32120
           IF (A(IX,IQ).EQ.I2ONE) THEN                                       POR32130
              ILIST(1) = IX                                                  POR32140
              JLIST(1) = IQ                                                  POR32150
           ENDIF                                                             POR32160
      ENDIF                                                                  POR32170
C LOOK DOWN                                                                  POR32180
      ILIST(3) = I2ZERO                                                      POR32190
      IQ=IY-1                                                                POR32200
C THE BOX IS INFINITELY FULL UNDER THE BOTTOM                                POR32210
*    BUT NEIGHBOR COORDINATES ARE NOT VALID BELOW IX = 1                     POR32220
*    SO LEAVE ILIST = 0 FOR THE Y=0 ROW                                      POR32230
```

```
      IF (IQ.EQ.0) THEN                                                 POR32240
          RETURN                                                        POR32250
      ENDIF                                                             POR32260
      IF (A(IX,IQ).EQ.I2ONE) THEN                                       POR32270
          ILIST(3) = IX                                                 POR32280
          JLIST(3) = IQ                                                 POR32290
      ENDIF                                                             POR32300
      RETURN                                                            POR32310
      END                                                               POR32320
                                                                        POR32330
                                                                        POR32340
      SUBROUTINE ICPROF(A,DCONC,SCONC)                                  POR32350
     INTEGER*2 I2ZERO,I2ONE,ICNTR(-1:1)                                 POR32360
      PARAMETER (I2ZERO=0,I2ONE=1)                                      POR32370
     COMMON /SIZPRM/  NXSPAC,NYSPAC,MXNPRT,NPART,NNUC,NLAY              POR32380
     COMMON /PARAM/ NPARTO,PROB,PROB2,PROB3,CONC,IZ,MXIY,ICLUST,KGEN,   POR32390
    1 MNIY,MXFRE,IAMBLK                                                 POR32400
       INTEGER*4 NXSPAC,NYSPAC,MXNPRT,NPART                             POR32410
       INTEGER*2 A(NXSPAC,NYSPAC)                                       POR32420
       REAL*4 DCONC(10000),SCONC(10000)                                 POR32430
       RETURN                                                           POR32440
                                                                        POR32450
      ENTRY CPROF (IPTO)                                                POR32460
*   ON EXIT, IPTO WILL PT TO THE FIRST ENTRY OF LATEST SET              POR32470
*************************************************************************POR32480
*      CALCULATE A CONCENTRATION PROFILE FROM MNIY TO THE TOP           POR32490
*      LOCAL CONCENTRATION IS A FUNCTION OF Y:                          POR32500
*                          (NUMBER OF FREE ON A ROW)                    POR32510
*    FOR EACH Y, DCONC(Y)= ------------------------------              POR32520
*                          (NUMBER OF FREE+NUMBER BLANK SPACES)         POR32530
*************************************************************************POR32540
*   DCONC DATA STRUCTURE:  DCONC(1) = TOTAL # OF VALID PTS IN DCONC     POR32550
*                          DCONC(2) = TOTAL #OF PTS IN DATASET 1        POR32560
*                          DCONC(3) ... DCONC(M)=FIRST SET'S CONC PROF  POR32570
*                          DCONC(M+1) = TOTAL # OF PTS IN DATSET #2     POR32580
*                          DCONC(M+2)...DCONC(N) = 2ND SET'S CONC PROF  POR32590
*************************************************************************POR32600
      IPT = DCONC(1) + 1                                                POR32610
      IPTO = IPT                                                        POR32620
      DO 100 J = MNIY,MXFRE                                             POR32630
         ICNTR(-1) = I2ZERO                                             POR32640
         ICNTR(0)  = I2ZERO                                             POR32650
         ICNTR(1)  = I2ZERO                                             POR32660
         DO 90 I =1,NXSPAC                                              POR32670
             ICNTR(A(I,J)) = ICNTR(A(I,J)) + I2ONE                      POR32680
  90     CONTINUE                                                       POR32690
         NAIR = ICNTR(0)                                                POR32700
         NFRE = ICNTR(-1)                                               POR32710
         NSTIK = ICNTR(1)                                               POR32720
```

```
            IPT = IPT + 1                                                    POR32730
            SCONC(IPT) = 1-FLOAT(NSTIK) / FLOAT(NXSPAC)                      POR32740
            IF (NFRE+NAIR .EQ. 0) THEN                                       POR32750
                  DCONC(IPT) = 0.001                                         POR32760
            ELSE                                                             POR32770
                  DCONC(IPT)= FLOAT(NFRE) / FLOAT(NFRE + NAIR)               POR32780
            ENDIF                                                            POR32790
  100   CONTINUE                                                           POR32800
        DCONC(1) = IPT                                                     POR32810
        DCONC(IPTO) = IPT - IPTO                                           POR32820
        SCONC(1) = IPT                                                     POR32830
        SCONC(IPTO) = IPT - IPTO                                           POR32840
        RETURN                                                             POR32850
        END                                                                POR32860
                                                                           POR32870
                                                                           POR32880
      SUBROUTINE ICORE (AS,MAXR,NSURF1,COR1,COR2,IORD)                     POR32890
      COMMON /SIZPRM/  NXSPAC,NYSPAC,MXNPRT,NPART,NNUC,NLAY                POR32900
      COMMON /PARAM/ NPARTO,PROB,PROB2,PROB3,CONC,IZ,MXIY,ICLUST,KGEN,     POR32910
     1 MNIY,MXFRE,IAMBLK                                                   POR32920
      REAL*4 COR1(100000),COR2(100000)                                     POR32930
      INTEGER*2 AS(NXSPAC,NYSPAC),MAXR,I2ZERO                              POR32940
      PARAMETER (I2ZERO=0)                                                 POR32950
      INTEGER*4 NPART,MXGEN,KGEN,ICALC1,ICALC2                             POR32960
      RETURN                                                               POR32970
                                                                           POR32980
      ENTRY CORE                                                           POR32990
                                                                           POR33000
      IARD=IORD*MAXR                                                       POR33010
C CHOOSE A PARTICLE                                                        POR33020
      DO 200 IR=1,MAXR                                                     POR33030
       ICRR=IARD+IR                                                        POR33040
        ICALC1= 0                                                          POR33050
        ICALC2= 0                                                          POR33060
        DO 100 I=1,NXSPAC                                                  POR33070
        DO 100 J=1,NYSPAC                                                  POR33080
              IF (AS(I,J).EQ.I2ZERO) GOTO 100                              POR33090
C X -NEIGHBORS                                                             POR33100
C RIGHT                                                                    POR33110
              IQ = I + IR                                                  POR33120
              IF (IQ.GT.NXSPAC) IQ=IQ-NXSPAC+1                             POR33130
              IF (AS(IQ,J).NE.I2ZERO) THEN                                 POR33140
                  ICALC1=ICALC1+1                                          POR33150
                  ICALC2=ICALC2+AS(IQ,J)-1                                 POR33160
              ENDIF                                                        POR33170
C LEFT                                                                     POR33180
              IQ = I - IR                                                  POR33190
              IF (IQ.LE.0) IQ=NXSPAC+IQ-1                                  POR33200
              IF (AS(IQ,J).NE.I2ZERO) THEN                                 POR33210
```

```
                    ICALC1=ICALC1+1                                    POR33220
                    ICALC2=ICALC2+AS(IQ,J)-1                           POR33230
                ENDIF                                                  POR33240
C Y -NEIGHBORS                                                         POR33250
C UP                                                                   POR33260
                IQ = J + IR                                            POR33270
                IF (IQ.GT.NYSPAC) IQ=IQ-NYSPAC+1                       POR33280
                IF (AS(I,IQ).NE.I2ZERO) THEN                           POR33290
                    ICALC1=ICALC1+1                                    POR33300
                    ICALC2=ICALC2+AS(I,IQ)-1                           POR33310
                ENDIF                                                  POR33320
C DOWN                                                                 POR33330
                IQ = J - IR                                            POR33340
                IF (IQ.LE.0) IQ=NYSPAC+IQ-1                            POR33350
                IF (AS(I,IQ).NE.I2ZERO) THEN                           POR33360
                    ICALC1=ICALC1+1                                    POR33370
                    ICALC2=ICALC2+AS(I,IQ)-1                           POR33380
                ENDIF                                                  POR33390
 100        CONTINUE                                                   POR33400
C AVERAGE COR1(R) OVER 4 NEIGHBOR DIRECTIONS                           POR33410
            COR1(ICRR)=ICALC1/ (4.0 * NSURF1)                          POR33420
C AVERAGE COR2(R) OVER 8 NEIGHBOR DIRECTIONS                           POR33430
            COR2(ICRR)=ICALC2/ (8.0 * NSURF1)                          POR33440
 200       CONTINUE                                                    POR33450
          RETURN                                                       POR33460
          END                                                          POR33470
```

REFERENCES

1. R. Zallen, "Introduction to Percolation: A Model for All Seasons", in Annals of the Israeli Physical Society, Vol. 5, Adam Hilger and the Israeli Physical Society, Bristol, England,

2. S. Kirkpatrick, Solid State Commun. 12, 1279.

3. S. Kirkpatrick, Rev. Mod. Phys., 45, 574 (1973).

4. B.J. Last and D.J. Thouless, Phys. Rev. Lett. 27, 1719 (1971).

5. S. Kirkpatrick, Phys. Rev. Lett. 27, 1722 (1971).

6. I. Webman, J. Jortner, M.H. Cohen, Phys Rev B11 2885 (1975).

7. J.P. Straley, Annals of the Israel Physical Soc., 5 355, Adam Hilger and the Israel Physical Soc., Bristol England, 1983.

8. M.H. Cohen, J. Jortner, I. Webman, AIP Conference Proceedings, 40 63 (1978).

9. D.J. Bergman, J. Imry, Phys Rev. Let. 39,1222 (1977).

10. J.P. Straley, Phys. Rev. B15, 5733 (1977)

11. R. Bass, R., M.J. Steven, J. Phys. C, 8,L281 (1975).

12. J.P. Straley, J. Phys. C9, 783 (1976).

13. S. Kirkpatrik, Ill Condensed Matter, eds. R. Balian, R. Maynard, G. Toulouse (North Holland, 1979).

14. D. Stauffer, Scaling Theory of Percolation Clusters, Phys. Rep. 54 1-74.

15. B. Mandelbrot, The Fractal Geometry of Nature, W Freeman & Co., N.Y. (1983).

16. S. Kirkpatrik, Ill Condensed Matter, p. 372.

17. B. Mandelbrot, Fractals- Form, Chance and Dimension, W. Freeman & sons, 1977, p. 196.

18. D. Stauffer, D., Phys Rep, 54 p. 50.

19. T.A. Witten and L.M. Sander, Phys. Rev. Lett. 47,1400 (1981).

20. I. Webman, in Proceedings of the Symposium on the Chemistry and Physics of Porous Media, The Electrochemical Society, (1985).

21. S.K. Sinha, T. Freltoft, and K. Kjems in Kinetics of Aggregation and Gelation, F. Family and D.P. Landau eds., North Holland (1984).

22. D.W. Schafer, and J.E. Martin, Phys. Rev. Lett., 25, 2371.

23. D.W. Schaefer, A.J. Hurd, in "Proceedings of the Symposium on the Chemistry and Physics of Composite Media", The Electrochemical Society, (1985) p. 55.

24. H.D. Bale, and P.W. Schmidt, Phys. Rev. Lett. 53, 596 (1984).

25. A. Palevski, and G. Deutscher, J. Phys. A 17,L895 (1984).

26. R. Landauer, in "Electrical Transport and Optical Properties of Inhomogeneous Media", ed. J.C. Garland and D.B. Tanner, AIP Conf. Proc. #40 (1978), p. 2.

27. D.E. Aspnes, Am. J. Phys. 50, 704 (1982).

28. D. Stroud, in "Proceedings of the Symposium on The Chemistry and Physics of Composite Media", P.N. Sen and M. Tomkiewicz, eds., The Electrochemical Society, 1985, p. 27.

29. R. Landauer, AIP Conf. Proc. 40 (1978), p.21.

30. J.O.M. Bokris, and A.K.N. Reddy, Modern Electrochemistry, Plenum Publishing (1970) p.725.

31. C. Kittel, Solid State Physics, 5th edition, Wiley (1978).

32. M. Tomkiewicz, J. Elec. Soc. 126,2221 (1978).

33. J. O'M. Bockris and D. Drazie, Electrochemical Science, Barnes and Noble Books (1972) Figure 8.7.

34. E. Yeager and A. Salkind eds., Techniques in Electrochemistry Volume 3, Wiley Interscience (1978), p. 305.

35. M. Hepel, and M. Tomkiewicz, J. Elec. Soc. 131, 1288 (1984).

36. R. De Levie, in Advances in Electrochemistry and Electrochemical Engineering, Vol. 6, Wiley Interscience New York, p. 329.

37. S.A.G.R. Karunathilaka, and N.A. Hampson, J. Appl. Electrochem., 10, 357 and 603 (1980).

38. D.O. Raleigh, J. Electrochem. Soc. 121,639 (1974).

39. B.V. Tilak, R.S. Perkins, H.A. Kozlowska, B.E. Conway, Electrochim. Acta 26,1447 (1972).

40. B.G. Pound, and D.D. Macdonald, Electrochim. Acta. 27, 1489 (1982).

41. P.H. Bottelberghs and G.H.J. Broers, J. Electroanal. Chem. 67, 155 (1976).

42. R.D. Armstrong and R.A. Burnham, J. Electroanal. Chem. 72,257 (1976).

43. G.J. Brug, A.L.G. Van Den Eeden, M. Sluyters-Rehbach, J.H. Sluyters, J. Electroanal. Chem. 176,275 (1984).

44. W. Scheider, J. Phys. Chem. 79, 127 (1974).

45. J. Lyden, M.H. Cohen, M. Tomkiewicz, Phys. Rev. Lett. 47,961

46. M.H. Cohen and M. Tomkiewicz, Phys. Rev. B 26,7097 (1982).

47. S.H. Liu, Bull. Am. Phys. Soc., 30,564 (1985).

48. Mandelbrot, p. 82.

49. Nyikos and Pajkossy, Preprint-1985, accepted for publication in Electrochim. Acta.

50. A. LeMehaute and G. Crepy, Solid State Ionics, 9 and 10, 17 (1983).

51. Carnahan, Luther, and Wilkes, Applied Numerical Methods, p. 300, John Wiley and Sons, New York (1969).

52. R. De Levie, Electrochim. Acta 9, 1231 (1964).

53. Bard, A.J. and Faulkner, L.R., Electrochemical Methods, Wiley (1980), p. 316.

54. N.A. Hampson and A.J.S. McNeil in Electrochemistry volume 9, The Royal Society of Chemistry, Burlington House, London (1984), p. 31.

55. P. Meakin, Phys Rev A27, 604 (1983)

56. R.F. Voss, J. Stat. Phys., 36,861 (1984).

57. R.F. Voss and M. Tomkiewicz, J. Electrochem. Soc., 132,371 (1985).

58. S.R. Forest and T.A. Witten Jr., J. Phys. A12,L109 (1979).

59. R.F. Voss, R.B. Laibowitz, E.I. Allessandrini, Phys. Rev. Lett. 49, 1441 (1982).

60. J. Feder, T. Jossang, E Rosenqvist, Phys. Rev. Lett. 53,1403 (1984).

61. D.A. Weitz, M. Oliveria, Phys. Rev. Lett. 52,1433 (1984).

62. M. Matsushita, M. Sano, Y. Hayakawa, H. Honjo, Y. Sawada, Phys. Rev. Lett. 53,286 (1984).

63. S. Szpak and C.J. Gabriel, J. ELectrochem. Soc. 126,1914 (1979).

64. D.D. Macdonald, Transient Techniques in Electrochemistry, Plenum Press (1977), p. 85.

65. P. Meakin and Z.R. Wasserman, Chemical Physics 91,391 (1984).

66. S. Wilke, Phys. Lett. A 96,467 (1983).

67. D.W. Herrman, D. Stauffer, Z. Physik B 44 339 (1981).

68. B. Derrida, D. Stauffer, H.J. Herrmann, J. Vannimenus, J. Physique Lett. 44 L701 (1983).

69. H.J. Herrmann, B. Derrida, J. Vannimenus, Phys Rev B 30 4080 (1984).

70. A.L.R. Bug, G.S. Grest, M.H. Cohen, I. Webman, "A.C. Response Near Percolation Threshold:Transfer Matrix Calculation in 2-D", preprint (1986).

71. M. Kramer, M. Tomkiewicz, J. Elec. Soc. 131 1283 (1984).

72. M. Kramer, M. Tomkiewicz, Proceedings of the Symposium on the Chemistry and Physics of Composite Media, The Electrochemical Society, 1985.