296
4/9/68
fey
AEC RESEARCH AND DEVELOPMENT REPORT

K-1740

MASTER

A COMPUTER PROGRAM
TO SOLVE EDDY-CURRENT PROBLEMS

AUTHORS:

J. Nelson Tunstall
C. V. Dodd, III

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

**UNION
CARBIDE**

UNION CARBIDE CORPORATION
NUCLEAR DIVISION

Operating the

OAK RIDGE GASEOUS DIFFUSION PLANT • OAK RIDGE NATIONAL LABORATORY
OAK RIDGE Y-12 PLANT • PADUCAH GASEOUS DIFFUSION PLANT

*For the Atomic Energy Commission
Under U.S. Government Contract W7405 eng 26*

DISCLAIMER

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

DISCLAIMER

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

Printed in the United States of America. Available from Clearinghouse for Federal
Scientific and Technical Information, National Bureau of Standards,
U.S. Department of Commerce, Springfield, Virginia 22151
Price: Printed Copy \$3.00; Microfiche \$0.65

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

- A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
- B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Date of Issue: April 8, 1968

Report Number: K-1740

Subject Category: Mathematics
and
Computers

A COMPUTER PROGRAM TO SOLVE

EDDY-CURRENT PROBLEMS

J. Nelson Tunstall
Computing Technology Center

C. V. Dodd, III
Metals and Ceramics Division
Oak Ridge National Laboratory

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

UNION CARBIDE CORPORATION
NUCLEAR DIVISION
Oak Ridge Gaseous Diffusion Plant
Oak Ridge, Tennessee

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

TABLE OF CONTENTS

	Page
Abstract	5
1. Introduction	7
2. Numerical Technique	8
3. Adaptation for a Computer	10
4. The Computer Code	15
4.1 Geometric Configuration	15
4.2 Logical Flow	16
4.3 Execution Options	18
4.4 Plotting Options	20
4.5 Definition of Program Variables	22
4.6 Description of Card Input	25
4.7 Miscellaneous Comments	29
References	31
Appendix A. Calculation of Physical Phenomena	32
Appendix B. Example Problem	34
Appendix C. Code Listing	45

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

A COMPUTER PROGRAM TO SOLVE
EDDY-CURRENT PROBLEMS

Abstract

The partial differential equation

$$\Delta A - \frac{A}{r^2} = -\mu I + i\omega\mu\sigma A - \mu \left[\left(\frac{\partial(1/\mu)}{\partial r} \right) \left(\frac{1}{r} \frac{\partial r A}{\partial r} \right) + \left(\frac{\partial(1/\mu)}{\partial z} \right) \frac{\partial A}{\partial z} \right],$$

where Δ is the differential operator

$$\Delta = \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{\partial^2}{\partial z^2}$$

and $A = A(z, r)$, with fixed boundary conditions is considered. A numerical technique of solution and a description of a computer code which uses this technique are presented.

This equation arises from considering Maxwell's equations in a conductive, isotropic, inhomogeneous medium at low frequencies and assuming symmetry about the axis of the coil and a sinusoidal driving current.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

1. INTRODUCTION

Starting with Maxwell's equations in a conductive, isotropic, inhomogeneous medium at low frequencies and assuming symmetry about the axis of the coil and a sinusoidal driving current, Dodd and Deeds [1], [2] derived the equation for the vector potential A,

$$(1) \quad \Delta A - \frac{A}{r^2} = -\mu I + i\omega\mu\sigma A - \mu \left[\left(\frac{\partial(1/\mu)}{\partial r} \right) \left(\frac{1}{r} \frac{\partial r A}{\partial r} \right) + \left(\frac{\partial(1/\mu)}{\partial z} \right) \frac{\partial A}{\partial z} \right],$$

where Δ is the differential operator

$$\Delta = \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{\partial^2}{\partial z^2},$$

and where μ , I , ω , and σ are respectively the permeability, applied current density, angular frequency, and conductivity. Once the vector potential is known, the induced voltage in the driving coil, the coil impedance, the induced eddy currents, and the eddy-current forces can be determined.

In this report, we give a numerical technique which solves (1) in a bounded domain and describe a computer code which utilizes this technique.

2. NUMERICAL TECHNIQUE

Due to the symmetry assumed in the derivation of Equation (1), we need only consider the eddy-current problem in the quarter-plane E defined by

$$E = \{(z, r) \mid z \geq 0 \text{ and } r \geq 0\}.$$

Let E_h , $h > 0$, be the discrete subset of the point set E defined by

$$E_h = \{(z_n, r_m) \mid z_n = (n-1)h, r_m = (m-1)h, n = 1, 2, \dots, N; \\ m = 1, 2, \dots, M\},$$

where N and M are integers such that $(N-1)h \geq 4R$ and $(M-1)h \geq 4R$, where R is the mean coil radius. Approximating the derivatives in Equation (1) at the point (z_n, r_m) , $1 < n < N$, $1 < m < M$, with finite differences [1], [2], we can express the vector potential A at the point (z_n, r_m) by

$$(2) \quad A_{n,m} = \left\{ \left(1 + \frac{1}{m-1}\right) \left(\frac{\mu_{n,m}}{\mu_{n,m+1}}\right) A_{n,m+1} + A_{n,m-1} + \frac{\mu_{n,m}}{\mu_{n+1,m}} A_{n+1,m} \right. \\ \left. + A_{n-1,m} + h^2 \mu_{n,m} I_{n,m} \right\} \div \\ \left\{ 2 + \frac{1}{m-1} + \frac{1}{(m-1)^2} + \frac{\mu_{n,m}}{\mu_{n,m+1}} + \frac{\mu_{n,m}}{\mu_{n+1,m}} + ih^2 \omega \mu_{n,m} \sigma_{n,m} \right\}$$

where

$$A_{n,m} = A(z_n, r_m), \quad n = 2, \dots, N-1; \quad m = 2, \dots, M-1.$$

We defined $A_{n,m}$ to be zero on the boundaries of E_h . This boundary condition is exact along the coil axis, and from our restriction on N and M , $A \approx 0$ on the other three boundaries [1]. With these boundary conditions and Equation (2) we can solve for the vector potential for every point in E_h . We use a point successive overrelaxation (SOR) iteration technique where at k^{th} iterate

$$(3) \quad A_{n,m}^{(k)} = A_{n,m}^{(k-1)} + W(A_{n,m}^{\text{calc}} - A_{n,m}^{(k-1)}), \quad k \geq 1, \quad 0 < W < 2,$$

where $A_{n,m}^{\text{calc}}$ denotes the calculated value obtained from (2). After many iterations, the vector potential will converge to a value determined by Maxwell's equations [1].

3. ADAPTATION FOR A COMPUTER

In order to use Equation (2) in a computer code, the values of $\mu_{n,m}$, $\sigma_{n,m}$, and $I_{n,m}$ must be stored for each point in E_h . Therefore, it behooves us to alter the form of Equation (2) at the expense of imposing certain restrictions on the geometric configuration of the problem. However, from the derivation of Equation (1), we have two properties [1], [2] which can simplify Equation (2) depending on the location of the point (z_n, r_m) in E_h . They are

$$(4) \quad \begin{aligned} \sigma_{n,m} &= 0 \quad \text{if } (z_n, r_m) \text{ is not in a conductor,} \\ I_{n,m} &= 0 \quad \text{if } (z_n, r_m) \text{ is not in the coil.} \end{aligned}$$

In altering Equation (2), we will take advantage of these properties.

Consider the discrete point set E_h , where for each point (z_n, r_m) in E_h , the values of $\mu_{n,m}$, $\sigma_{n,m}$, and $I_{n,m}$ are known. We define a material S to be a set of points (z, r) in E_h such that for all points (z, r) in S the respective values of $\mu_{z,r}$, $\sigma_{z,r}$, and $I_{z,r}$ are equal. That is, if the points (z_{n_1}, r_{m_1}) and (z_{n_2}, r_{m_2}) are in the same material S , then $\mu_{n_1, m_1} = \mu_{n_2, m_2}$, $\sigma_{n_1, m_1} = \sigma_{n_2, m_2}$, and $I_{n_1, m_1} = I_{n_2, m_2}$. Thus E_h can be written as the union of ℓ disjoint materials,

$$E_h = S_1 \cup S_2 \cup \dots \cup S_\ell.$$

Associated with each material S_k , $k = 1, 2, \dots, \ell$, there are values u'_k , v_k , and w_k such that for all points (z, r) in S_k , we have that $\mu_{z,r} = u'_k$, $\sigma_{z,r} = v_k$, and $I_{z,r} = w_k$. We say that each point in the

material S_k has a material index of k . Thus we define a point set function $\beta(z_n, r_m)$ on E_h by

$$\beta_{n,m} = \beta(z_n, r_m) = k,$$

where k is the material index of the point (z_n, r_m) .

We define the relative permeability $\bar{\mu}$ by

$$\bar{\mu} = \mu/\mu_0,$$

where μ_0 is the permeability of free space, and we "normalize" the $\mu_{n,m}$ in this manner for $n = 1, 2, \dots, N$ and $m = 1, 2, \dots, M$. Thus we obtain the relative permeability values u_k , $k = 1, \dots, \ell$.

We make a classification of the materials $S_1, S_2, \dots, S_\ell$ based on the properties given by (4), and we impose the restriction that a material can be one and only one of these types. A material S_k , $1 \leq k \leq \ell$, is called:

coil material if and only if $w_k \neq 0$ and $v_k = 0$;

conductor material if and only if $v_k \neq 0$ and $w_k = 0$;

ferrite material if and only if $v_k = w_k = 0$ and $u_k > 1$;

air material if and only if $v_k = w_k = 0$ and $u_k = 1$.

We now define two functions $\alpha(k)$ and $\gamma(k)$ which are dependent on the material indices k , $k = 1, 2, \dots, \ell$. The function $\alpha(k)$ is given by

$$\alpha(k) = \begin{cases} 1, & \text{if } k \text{ is the index of a coil material} \\ 0, & \text{otherwise.} \end{cases}$$

Suppose there are c conductor materials. Let $k_1, k_2, \dots, k_c$ be the corresponding material indices. Define

$$(5) \quad \hat{F} = h^2 \omega u_{k_1}^r v_{k_1},$$

$$F_i = (h^2 \omega u_{k_i}^r v_{k_i}) / \hat{F}, \quad i = 1, 2, \dots, c.$$

We define the function $\gamma(k)$ by

$$\gamma(k) = \begin{cases} 0, & \text{if } k \text{ is not the index of a conductor material} \\ F_i, & \text{if } k = k_i \text{ for some } i, i = 1, 2, \dots, c. \end{cases}$$

Using the functions $\beta(z_n, r_m)$, $\alpha(k)$, and $\gamma(k)$, we can express the vector potential A at the point (z_n, r_m) by

$$(6) \quad A_{n,m} = \frac{(1 + \frac{1}{m-1}) \frac{u_k}{u_s} A_{n,m+1} + A_{n,m-1} + \frac{u_k}{u_t} A_{n+1,m} + A_{n-1,m} + \alpha(k) u_k}{2 + \frac{1}{m-1} + \frac{1}{(m-1)^2} + \frac{u_k}{u_s} + \frac{u_k}{u_t} + i \hat{F} \gamma(k)},$$

where $k = \beta_{n,m}$, $s = \beta_{n,m+1}$, and $t = \beta_{n+1,m}$. This is the form of Equation (2) which the computer code utilizes. The replacing of the term $h^2 \mu_{n,m} I_{n,m}$ in Equation (2) with $\alpha(k) u_k$ automatically sets the current density I equal to $1/(h^2 \mu_0)$. Since the vector potential is a linear function of the current, it may be scaled to correspond to a different value of the current. The current i in a test coil is related to the current density in the computer program by

$$\hat{i} = \frac{N h^2 I}{n} = \frac{N}{n \mu_0},$$

where n is the number of turns in the test coil and N is the number of points of E_h in the coil material.

In order to provide initial estimates and normalizing factors, the code calculates an "air case" - the case where no conductor materials

are present. For the air case, Equation (6) reduces to

$$(7) \quad \hat{A}_{n,m} = \frac{(1 + \frac{1}{m-1}) \frac{u_k}{u_s} \hat{A}_{n,m+1} + \hat{A}_{n,m-1} + \frac{u_k}{u_t} \hat{A}_{n+1,m} + \hat{A}_{n-1,m} + \alpha(k) u_k}{2 + \frac{1}{m-1} + \frac{1}{(m-1)^2} + \frac{u_k}{u_s} + \frac{u_k}{u_t}}$$

where k , s , and t are as before except now they represent indices for only air, coil, and ferrite materials. Note that $\hat{A}_{n,m}$ is a real-valued quantity.

The following summarizes the general iterative scheme for the eddy-current problem on a $N \times M$ mesh with ℓ materials of which c are conductor materials:

Step 1. Construct α_k , γ_k , and u_k for $k = 1, 2, \dots, \ell - c$, and $\hat{\beta}_{n,m}$ for $n = 1, 2, \dots, N$; $m = 1, 2, \dots, M$.

Step 2. Set $\hat{A}_{n,m} = 0$ for $n = 1, 2, \dots, N$; $m = 1, 2, \dots, M$.

Step 3. Fix $m = 2$.

Step 4. Calculate $\hat{A}_{n,m}$ for $n = 2, \dots, N-1$ using (3) with (7).

Step 5. Do Step 4 for $m = 3, \dots, M-1$.

Step 6. Test for convergence.

Step 7. Do Steps 3-6 until the convergence criterion is met.

Step 8. Construct α_k , γ_k , and u_k for $k = c, c+1, \dots, \ell$ and $\beta_{n,m}$ for $n = 1, 2, \dots, N$; $m = 1, 2, \dots, M$.

Step 9. Set $\text{Re}(A_{n,m}) = \hat{A}_{n,m}$ and $\text{Im}(A_{n,m}) = 0$ for $n = 1, 2, \dots, N$; $m = 1, 2, \dots, M$.

Step 10. Set $m = 2$.

Step 11. Calculate $A_{n,m}$ for $n = 2, \dots, N-1$ using (3) with (6).

Step 12. Do Step 11 for $m = 3, \dots, M-1$.

Step 13. Test for convergence.

Step 14. Do Steps 10-13 until the convergence criterion is met.

Note that Steps 1-7 constitute the calculation of the air case and that Steps 8-14 are for a fixed F.

4. THE COMPUTER CODE

In this section we describe a computer code EDDY, written in FORTRAN IV for the IBM 7090, which solves (1) in a bounded domain of E using the numerical technique in Section 2 and the general iterative scheme in Section 3. We define the program variables and describe the input and logical flow of the code in this section. A sample eddy-current problem is given in Appendix B, and the FORTRAN listing of the code is given in Appendix C. We assume in this section that a grid E_h of $N \times M$ points has been chosen and that the "geometric configuration," that is, the functions $\beta(z_n, r_m)$, u_k , $\alpha(k)$, and $\gamma(k)$, is known for the ℓ materials of E_h . A method of determining E_h and the geometric configuration is described in Appendix B.

4.1 Geometric Configuration

We now give a general description of the input necessary to define for the code the geometric configuration. Let k be any material index, $k=1,2,\dots,\ell$. Then the material S_k ,

$$S_k = \{(z_n, r_m) \mid (z_n, r_m) \in E_h \text{ and } \beta_{n,m} = k\},$$

can be expressed as

$$S_k = L_1^{(k)} \cup L_2^{(k)} \cup \dots \cup L_{p_k}^{(k)},$$

where

$$(8) \quad L_i^{(k)} = \{(z_n, r_m) \mid \underline{n}_i^{(k)} \leq n \leq \bar{n}_i^{(k)} \text{ and } \underline{m}_i^{(k)} \leq m \leq \bar{m}_i^{(k)}\}.$$

That is, we can express S_k as the union of rectangles $L_i^{(k)}$, $i=1,2,\dots,p_k$, of points from E_h with vertices $(\underline{n}_i^{(k)}, \underline{m}_i^{(k)})$, $(\bar{n}_i^{(k)}, \underline{m}_i^{(k)})$, $(\bar{n}_i^{(k)}, \bar{m}_i^{(k)})$,

and $(\underline{n}_i^{(k)}, \underline{m}_i^{(k)})$, $i=1,2,\dots,p_k$. Of course, if the material S_k is itself a rectangle of points, then $p_k=1$. Hence, the code can construct the function $\beta_{n,m}$ by having as input for each k , $k=1,2,\dots,\ell$, the values of $\underline{n}$, $\underline{\bar{n}}$, $\underline{m}$, and $\underline{\bar{m}}$ for each $L_i^{(k)}$ in (8). We note in the construction of the air case geometric configuration in Step 1 of the iterative scheme in Section 3 and the construction of the geometric configuration in Step 8 that the only difference besides the number of materials involved is in $\hat{\beta}_{n,m}$ and $\beta_{n,m}$. However, by the definition of the air case, we have that $\hat{\beta}_{n,m} = \beta_{n,m}$ for all points (z_n, r_m) not in a conductor material, and $\hat{\beta}_{n,m}$ equals the material index for the air material if (z_n, r_m) is in a conductor material.

4.2 Logical Flow

We now describe the general logical flow of the code EDDY using the iterative scheme given in Section 3 as a reference point. The air case geometric configuration is constructed, and the air case is calculated as given by Steps 1-7. The convergence criterion used at the k^{th} iteration is

$$(9) \quad \frac{|\hat{S}^{(k)} - \hat{S}^{(k-1)}|}{|\hat{S}^{(k)}|} < \hat{\epsilon},$$

where

$$(10) \quad \hat{S}^{(k)} = 2\pi \sum_{\text{coil}} (m-1) \hat{A}_{n,m}^{(k)}.$$

When this convergence criterion is met or when the number of iterations k exceeds k_{max} , a limit which is input to the code, the air case geometric configuration and the values of $\hat{A}_{n,m}$, $n=1,\dots,N$; $m=1,\dots,M$, are stored on magnetic tape (logical unit number 9).

After completion of the air case, the code reads the input that extends the functions u_k , $\alpha(k)$, $\gamma(k)$ and "adds" the conductor materials S_k , $k=c, c+1, \dots, \ell$, to obtain the conductor geometric configuration. This is also stored on magnetic tape (logical unit number 11). The values of the air case vector potential are used as the initial estimate as given by Step 9.

With this fixed geometric configuration, the code will calculate (as given by Steps 10-14) n "conductor cases," each identified by the input value of $\hat{F}_j$, $j=1, 2, \dots, n$, where $\hat{F}_j$ is given by the first expression in (5). The code does require that $\hat{F}_{j+1} > \hat{F}_j$. After each conductor case has been calculated, the $\hat{F}$ for that case and $A_{n,m}$, $n=1, \dots, N$; $m=1, \dots, M$, are stored on the logical unit number 11. The values of $A_{n,m}$ for the j^{th} conductor case (i.e., case where $\hat{F} = \hat{F}_j$) are used as initial estimates for the $(j+1)$ st conductor case. As in the air case, the calculation of a conductor case is completed when the convergence criteria at the k^{th} iteration,

$$(11) \quad \frac{|\operatorname{Re}(S^{(k)}) - \operatorname{Re}(S^{(k-1)})|}{|\operatorname{Re}(S^{(k)})|} < \epsilon_1$$

and

$$(12) \quad \frac{|\operatorname{Im}(S^{(k)}) - \operatorname{Im}(S^{(k-1)})|}{|\operatorname{Im}(S^{(k)})|} < \epsilon_2,$$

where

$$(13) \quad S^{(k)} = 2\pi \sum_{\text{coil}} (m-1) A_{n,m}^{(k)},$$

is met or when the number of iterations exceeds a limit which is input to the code.

4.3 Execution Options

The code EDDY allows the user four options of execution, three of which are merely variations on the general logical flow outlined in Section 4.2. If the input variable NOPT=1, the code only calculates the air case and writes the air case tape. If NOPT=2, the code calculates the air case, writes the air case tape, calculates the n conductor cases, and writes these cases on the conductor tape. If NOPT=3, the code reads the values from the associated air case tape and using these values as initial estimates, calculates the n conductor cases as in the NOPT=2 case. NOPT=4 signals the code that the user wishes to "update" a conductor tape previously created. The user can update a conductor tape by means of two update operations, REDOIT and EXTEND, which are defined as follows.

REDOIT Operation: The code reads n' , the number of cases to be updated with this operation, and a value $\hat{F}_s$. The code searches the conductor tape until the case where $\hat{F}_j = \hat{F}_s$ is found, and the values of $A_{n,m}$, $n=1,2,\dots,N$; $m=1,2,\dots,M$ are read. Using these values of A as initial estimates, the code calculates the n' cases (the user furnishes the values $\hat{F}_j$, $j=1,\dots,n'$) in the manner outlined in Section 4.2.

EXTEND Operation: The code reads n' , the number of cases to be updated with this operation. For each case j , $j=1, 2,\dots,n'$, the code reads a value F_j , searches the conductor tape until the

case where $\hat{F} = \hat{F}_j$ is found, reads the values $A_{n,m}$, $n=1,2,\dots,N$; $m=1,2,\dots,M$, for that case, and continues calculation on that case for the additional number of iterations the user desires.

The REDOIT operation was designed to handle situations that arise when, on the previous creation of the conductor tape,

- (a) the computer run was abnormally terminated due to floating point overflows, allotted time exceeded, etc., and the conductor tape was not completed as planned, or
- (b) the cases after a certain point all diverged due to a "too large" overrelaxation constant or poor initial estimates.

The EXTEND operation was designed to handle the situation when

- (a) cases on the conductor tape did not converge and the user feels that convergence can be achieved by continuing the calculation for more iterations, or
- (b) the user wishes to achieve "better convergence" on certain cases on the conductor tape by reducing the sizes of ϵ_1 and ϵ_2 given by (11) and (12).

Both of these operations signaled by NUPT=4 can be carried out during a single run of the code. However, they must be ordered in such a way that the cases are updated in the same order that they appear on the tape ($\hat{F}_{i+1} > \hat{F}_i$). Also during a single run of the code the following ambiguous orders of update operations are not allowed:

- (1) "REDOIT" same case twice,
- (2) "REDOIT" a case which has just been updated by "EXTEND",

- (3) using "REDOIT" to extend a case, i.e., have $\hat{F}_s$ equal the $\hat{F}$ of the first case to be updated,
- (4) "EXTEND" the same case twice, and
- (5) "EXTEND" a case which has just been updated by "REDOIT".

An example is given in Appendix B to illustrate the NOPT=4 operation.

4.4 Plotting Options

The code EDDY has the capability of creating a tape for the Benson-Lehner Electro-plotter from which the user can obtain contour plots of the four quantities, $N(|A_{n,m}|)$, $\arg(A_{n,m})$, $(f_z)_{n,m}$, and $(f_r)_{n,m}$, where $(f_z)_{n,m}$ and $(f_r)_{n,m}$ denote normalized net electromagnetic forces (see Appendix A), and where $N(|A_{n,m}|)$,

$$N(|A_{n,m}|) = \frac{([\operatorname{Re}(A_{n,m})]^2 + [\operatorname{Im}(A_{n,m})]^2)^{1/2}}{\max \hat{A}_{n,m}},$$

is the normalized magnitude of $A_{n,m}$, and θ ,

$$\theta = \arg(A_{n,m}) = \tan^{-1} \frac{\operatorname{Im}(A_{n,m})}{\operatorname{Re}(A_{n,m})}, \quad 0 \leq \theta < 2\pi,$$

is the phase of $A_{n,m}$.

The user has the option of obtaining contour plots for any of these four quantities by means of the input array ITB(I), $I=1, \dots, 4$, with the definition that:

ITB(1) $\neq$ 0 $\rightarrow$ code creates contour plot of $N(|A_{n,m}|)$,

ITB(2) $\neq$ 0 $\rightarrow$ code creates contour plot of $\arg(A_{n,m})$,

ITB(3) $\neq$ 0 $\rightarrow$ code creates contour plot of $(f_z)_{n,m}$,

ITB(4) $\neq 0 \rightarrow$ creates contour plot of $(f_r)_{n,m}$.

If

$$\sum_{I=1}^4 |\text{ITB}(I)| = 0,$$

then of course no plot tape is created.

For the normalized magnitude plot, the user not only specifies the contour levels desired but also a rectangular subregion of E_h defined by the values $\underline{M}_1$, $\overline{M}_1$, $\underline{N}_1$, and $\overline{N}_1$, where

$$(14) \quad \begin{aligned} 1 &\leq \underline{M}_1 < \overline{M}_1 \leq M, \\ 1 &\leq \underline{N}_1 < \overline{N}_1 \leq N, \end{aligned}$$

along with spacing increments ΔM_1 and ΔN_1 . The contours are plotted based on the normalized magnitude at the points (z_N, r_M) , where

$$M = \underline{M}_1, \underline{M}_1 + \Delta M_1, \underline{M}_1 + 2(\Delta M_1), \dots, \overline{M}_1;$$

$$N = \underline{N}_1, \underline{N}_1 + \Delta N_1, \underline{N}_1 + 2(\Delta N_1), \dots, \overline{N}_1.$$

There must be at least two r-spacings and two z-spacings, that is

$$(15) \quad \frac{\overline{M}_1 - \underline{M}_1}{\Delta M_1} + 1 \geq 3; \quad \frac{\overline{N}_1 - \underline{N}_1}{\Delta N_1} + 1 \geq 3.$$

Similarly for the phase plot, the user specifies the contour levels desired, a subregion of interest defined by the values $\underline{M}_2$, $\overline{M}_2$, $\underline{N}_2$, and $\overline{N}_2$, and the spacing increments ΔM_2 and ΔN_2 such that the inequalities in (14) and (15) are satisfied.

The plots are created on E_h itself if $\underline{M}_i = \underline{N}_i = 1$, $\overline{M}_i = M$, and $\overline{N}_i = N$, $i=1$ or 2 . The user need only to specify the contour levels desired for the $(f_z)_{n,m}$ and $(f_r)_{n,m}$ plots since the region of interest is determined by the code to be the conductor materials.

4.5 Definition of Program Variables

We now give the definition of the FORTRAN variables in EDDY. Whenever possible, these program variables are defined in terms of the variables used previously in the report. The notation $CA(N,M)$ indicates that CA is an array and that the value of CA at the (N,M) position corresponds to the value at the point (z_N, r_M) in the discrete set E_h .

MM:	The number of columns, M , of E_h (see Section 2).
NN:	The number of rows, N , of E_h .
CA(N,M):	The complex-valued vector potential, $A_{n,m}$.
RA(N,M):	The real-valued vector potential $\hat{A}_{n,m}$, calculated in the air case.
MATNO(N,M):	The material index function, $\beta_{n,m}$, defined in Section 3.
PER(K):	The material index dependent function u_k defined in Section 3.
VCN(K):	The material index dependent function $\gamma(k)$ defined in Section 3.
IZERO(K):	The material index dependent function $\alpha(k)$ defined in Section 3.
FMU:	The value $\hat{F}$ defined in (5).

TOTMAT: The total number of materials, ℓ .

NDCTOR: The number of conductor materials, c .

NREGA: The number of materials in the air case, $(\ell - c)$.

RELAX: The overrelaxation constant W .

MAXCT: The maximum number, $k_{\max}$, of iterations to be calculated.

KOUT: The code executes the first KOUT iterations before testing for convergence.

EPS: The limit $\hat{\epsilon}$ in (9) for the air case or the limit ϵ_1 in (11) for conductor cases.

EPSI: The limit ϵ_2 in (12).

SUMA: The sum over the coil, $\hat{S}$, defined by (10).

SUMC: The sum over the coil, S , defined by (13).

AIRMAX: The value of $\max \hat{A}_{n,m}$.

ITB(I): The plot option array ($I=1, \dots, 4$) defined in Section 4.4.

ML(I)
MU(I)
MDELTA(I) } : The constants $\underline{M}_i$, $\overline{M}_i$, and ΔM_i , $i=1,2$, defined in Section 4.4.

NL(I)
NU(I)
NDELTA(I) } : The constants $\underline{N}_i$, $\overline{N}_i$, and ΔN_i , $i=1,2$, defined in Section 4.4.

NCONT1: The number of contours to be used in the normalized magnitude plots.

NCONT2: The number of contours to be used in the phase plots.

NCONT3: The number of contours to be used in the $(f_z)_{n,m}$ plots.
 NCONT4: The number of contours to be used in the $(f_r)_{n,m}$ plots.
 CONT1(I): The contour values for the normalized magnitude plots ($I=1, \dots, NCONT1$).
 CONT2(I): The contour values for the phase plots ($I=1, \dots, NCONT2$).
 CONT3(I): The contour values for the $(f_z)_{n,m}$ plots ($I=1, \dots, NCONT3$).
 CONT4(I): The contour values for the $(f_r)_{n,m}$ plots ($I=1, \dots, NCONT4$).
 NOPT: The execution option value defined in Section 4.3.
 OPTION: The update indicator (used when $NOPT=4$) in which the operations "REDOIT", "EXTEND", or "FINISH" are specified (see Sections 4.3 and 4.6).
 NCASES: The number of conductor cases, n , to be run if $NOPT=2, 3$ or the number of cases, n' , to be updated by a particular operation if $NOPT=4$.
 ZN: The normalized impedance (see Appendix A).
 IMAT: The variable name used in reading the material index in constructing the geometric configuration (see Sections 4.1 and 4.6).

NLG	}	:	The variable names used, corresponding to a set
NUG			
MLG			
MUG			
FMUST:			The value $\hat{F}_s$ in Section 4.3 used when NOPT=4 and OPTION $\equiv$ REDOIT.

4.6 Description of Card Input

We now describe the card input for EDDY. Each card is described by listing the order that the variables from Section 4.5 are to be specified and giving the FORTRAN format.

Since the value of NOPT (on the first card below) determines the specific logical flow (see Section 4.3) that the code takes, this input description is outlined based on the different values that this variable can take. However, the cards that provide the necessary plotting values (referred to as "plot cards" hereafter) are independent of the NOPT value; and we describe these cards separately at the end of this section, giving initially only the location in the input package where these cards should appear if plotting is requested.

Variables Read	Format
<u>Card A:</u> NOPT	(I5)
If NOPT=1, the (air case) cards B, C, D _j , E _i , and F are needed.	
<u>Card B:</u> MAXCT, KOUT, EPS, RELAX:	(2I5,2F10.5)
<u>Card C:</u> NREGA, NN, MM:	(3I5)

Variables Read

Format

Cards D_j : IMAT, PER(IMAT), VCN(IMAT),

IZERO(IMAT): (I2,3X,2F10.5,4X,I1)

for $j = 1, 2, \dots, NREGA$

(Cards D_j define the functions u_k , $\gamma(k)$, and $\alpha(k)$ for $k = 1, 2, \dots, NREGA$).

Cards E_i : IMAT, NLG, NUG, MLG, MUG: (I2,3X,4I5)

for $i = 1, 2, \dots, P$, where $P = \sum_{k=1}^{NREGA} p_k$.

(Each card E_i describes a rectangle L , defined in Section 4.1, giving the vertex "coordinates" and material index of the points in L).

Card F: "-1" (I2)

(Card F signals the code that the air case geometry has been defined).

If NOPT=3, the (conductor cases) cards G, H, I_j , J_i , K, L, and M_i are needed.

Card G: EPS, EPSI, ITB(1), ITB(2), ITB(3)

ITB(4): (2F10.0,5X,4I1)

If $\sum_{I=1}^4 |ITB(I)| \neq 0$, the user inserts the Plot Cards here.

Card H: NCDTOR: (I5)

Cards I_j : IMAT, PER(IMAT), VCN(IMAT),

IZERO(IMAT): (I2,3X,2F10.5,4X,I1)

for $j = 1, 2, \dots, NCDTOR$.

(Cards I_j define the functions u_k , $\gamma(k)$, and $\alpha(k)$ for $k = [NREGA+1], \dots, \ell$).

Variables Read	Format
<u>Cards</u> J_i : IMAT, NLG, NUG, MLG, MUG:	(I2,3X,4I5)

for $i = 1, 2, \dots, Q$, where $Q = \sum_{k=NREGA+1}^{\ell} P_k$.

(Each card J_i describes a rectangle L for a conductor material - see cards E_i description.)

<u>Card</u> K: "-2" :	(I2)
-----------------------	------

(Card K signals the code that the complete geometric configuration has been defined.)

<u>Card</u> L: NCASES:	(I5)
------------------------	------

<u>Cards</u> M_i : MAXCT, KOUT, FMU, RELAX:	(2I5,2F10.5)
---	--------------

for $i = 1, 2, \dots, NCASES$.

If NOPT=2, then both the air case cards (for NOPT=1) and the conductor cases cards (for NOPT=3) are needed. That is, cards B, C, D_j , E_i , F, G, H, I_j , J_i , K, L, M_i are needed.

If NOPT=4, then the cards N, O, Q_i are needed.

<u>Card</u> N: EPS, EPSI, ITB(1), ITB(2), ITB(3),

ITB(4):	(2F10.0,5X,4I1)
---------	-----------------

If $\sum_{I=1}^4 |ITB(I)| \neq 0$, the user inserts the Plot Cards here.

<u>Card</u> O: OPTION, NCASES:	(A6,I4)
--------------------------------	---------

If OPTION = REDOIT, then both of the following set of cards P and Q_i are needed.

<u>Card</u> P: FMUST:	(F10.5)
-----------------------	---------

<u>Cards</u> Q_i : MAXCT, KOUT, FMU, RELAX	(2I5,2F10.5)
--	--------------

for $i = 1, 2, \dots, NCASES$.

Variables Read

Format

If $\text{OPTION} \equiv \text{EXTEND}$, then only cards Q_i are needed. (Note that Cards Q_i and M_i are of the same form.)

The user may have as many Cards 0 (update operation cards) as necessary - each followed by the sets P and Q_i depending on the nature of the update.

Card R: "FINISH":

(A6)

(Card R signals the code that the update requests have been processed.)

Plot Cards

If at least one $\text{ITB}(I)$, $I=1, \dots, 4$ is nonzero, then the user must provide some portion of the plot cards described now.

If $\text{ITB}(1) \neq 0$, then cards S and T are needed.

Variables Read

Format

Card S: $\text{ML}(1), \text{MU}(1), \text{MDELTA}(1), \text{NL}(1),$

$\text{NU}(1), \text{NDELTA}(1), \text{NCONT1}:$

(7I5)

Cards T: $\text{CONT1}(I):$

(8F10.5)

for $I = 1, 2, \dots, \text{NCONT1}$

(The code continues to read Cards T until all NCONT1 contour values have been read.) If $\text{ITB}(2) \neq 0$, then cards U and V are needed.

Card U: $\text{ML}(2), \text{MU}(2), \text{MDELTA}(2), \text{NL}(2),$

$\text{NU}(2), \text{NDELTA}(2), \text{NCONT2}:$

(7I5)

Cards V: $\text{CONT2}(I):$

(8F10.5)

for $I = 1, 2, \dots, \text{NCONT2}$

If $\text{ITB}(3) \neq 0$, then cards W and X are needed:

Card W: $\text{NCONT3}:$

(I5)

Variables Read	Format
<u>Cards</u> X: CONT3(I):	(8F10.5)
for I = 1,2,...,NCONT3	
If ITB(4) $\neq$ 0, then cards Y and Z are needed.	
<u>Card</u> Y: NCONT4:	(I5)
<u>Cards</u> Z: CONT4(I):	(8F10.5)
for I = 1,2,...,NCONT4	

4.7 Miscellaneous Comments

In this section we comment on some of the limitations of the code as presented in Appendix C, point out some areas where the performance of the code can be improved by the user, and outline the magnetic tape input and output requirements.

The code was written in FORTRAN IV, version 13, to be run on the IBM 7090 under the IBSYS operating system with the IBJOB processor. It was necessary to utilize the overlay feature and to require that ALTIO be specified. If the user does not wish to obtain plots, he can eliminate the second dependent link (making the program changes necessary to satisfy the loader requirements), and the need for the overlay structure is no longer necessary.

Since many of the calculations in the code are performed in complex-valued arithmetic, the execution time of EDDY is relatively long. For instance, in the example presented in Appendix B, convergence for the air case was achieved in 116 iterations, taking 8.4 minutes; and convergence in a conductor case was achieved in 287 iterations, taking 51 minutes. The creation of a plot tape with the magnitude

(10 contours), phase (8 contours), and $(f_z)_{n,m}$ (12 contours) plots for that case took an additional 18 minutes. However, assuming the user has n cases (with the fixed geometric configuration) he wishes to compute, the update operations with NOPT=4 allow him some freedom in dividing the total execution time into several separate runs. The user can run only a few of the n cases at a time making use of the update operation REDOIT each time (after the first time). Alternatively, the user can run all n cases at once but only running each case for a "few" number of iterations, and then in later runs he can use the EXTEND operation on each case. If the user desires plots, he may find that running all cases without a plot request and then at a later time running each case (for which he desires plots) for one iteration with the EXTEND operation and with a plot request will be a more favorable procedure.

We now summarize the tape input and output requirements. The air vector potential tape (logical unit 9) is an output tape if NOPT=1 or 2, and an input tape if NOPT=3, and is not required if NOPT=4. The conductor vector potential tape (master tape) is not required if NOPT=1 and is an output tape if NOPT=2 or 3. For the updating operations (NOPT=4), the "old" conductor master is the input tape on logical unit 12, and the "new" conductor master is created on logical unit 13. The plot tape is created on logical unit 15 if plots are requested.

REFERENCES

1. Dodd, C. V., Solutions to Electromagnetic Induction Problems, Oak Ridge National Laboratory, Oak Ridge, Tennessee, June, 1967, (ORNL-TM-1842).
2. Dodd, C. V., and Deeds, W. E., "Electromagnetic Forces in Conductors," Journal of Applied Physics, Vol. 38, No. 13, December, 1967, pp. 5045-5051.
3. Dodd, C. V., and Deeds, W. E., "Computer Design of Eddy-Current Tests," to be published in Proceedings of Fifth International Conference for Nondestructive Testing, Montreal, Canada, May, 1967.

APPENDIX A

Calculation of Physical Phenomena

Once the vector potential has been determined at each point (z_n, r_m) in E_h , any physically observable electromagnetic phenomenon can be calculated [1]. Some of these phenomena in their finite difference representation [1], [2], [3] are:

- (1) the coil impedance Z , given by

$$Z = i\omega(n/N)^2 \mu_o h \sum_N 2\pi(m-1) A_{n,m},$$

where n is the number of turns in a test coil and N is the number of points of E_h in the coil;

- (2) the net electromagnetic force in the z direction, given by

$$F_z = \pi h^2 \omega \sum_{\text{metal}} \sigma_{n,m} (m-1) [\text{Re}(A_{n,m}) \text{Im}(A_{n-1,m}) - \text{Im}(A_{n,m}) \text{Re}(A_{n-1,m})];$$

and

- (3) the electromagnetic force in the r direction given by

$$F_{r_{n,m}} = \pi(m-1) h^2 \omega \sigma_{n,m} |A_{n,m}| \cdot \left| \left(\frac{m+1}{m} \right) A_{n,m+1} - A_{n,m} \right| \cdot [\sin(2\omega t + \phi + \phi') + \sin(\phi - \phi')],$$

where ϕ and ϕ' are the phase shift of $|A_{n,m}|$ and

$|A_{n+1,m} - A_{n,m}|$, respectively.

The code EDDY makes the following calculations necessary to determine these quantities. The sum over the coil S ,

$$S = 2\pi \sum_{\text{coil}} (m-1) A_{n,m},$$

is calculated and then normalized by dividing by $\hat{S}$,

$$\hat{S} = 2\pi \sum_{\text{coil}} (m-1) \hat{A}_{n,m},$$

to obtain the normalized coil impedance Z_n ,

$$Z_n = i S/\hat{S}.$$

The expressions

$$(f_z)_{n,m} = \text{Re}(A_{n,m}) \text{Im}(A_{n-1,m}) - \text{Im}(A_{n,m}) \text{Re}(A_{n-1,m})$$

and

$$(f_r)_{n,m} = \left(\frac{m}{m-1}\right) [\text{Re}(A_{n,m+1}) \text{Im}(A_{n,m}) - \text{Im}(A_{n,m+1}) \text{Re}(A_{n,m})]$$

are calculated at all points in conductor materials and are normalized by dividing by $\max[(f_z)_{n,m}]$ and $\max[(f_r)_{n,m}]$, respectively. Contour plots of these force quantities can be obtained.

APPENDIX B

Example Problem

We consider the eddy-current problem for a coil with a ferrite cup above a conducting plane where:

- (a) the coil has an outside radius of 2.16 cm., an inside radius of 1.62 cm., frequency of .5 kc., and a lift-off of .1 cm.;
- (b) the conducting plane is copper of conductivity 1.732 $\mu\text{ohm-cm.}$ with a .18 cm. aluminum cladding of conductivity 4.08 $\mu\text{ohm-cm.}$;
- (c) the ferrite cup has an outside radius of 2.61 cm., an inside radius of 1.17 cm., and a relative permeability of 1000.

In this section we illustrate a procedure to "set-up" the above problem for the code EDDY and give the corresponding card format for the input to the code.

We choose the set E_h to have dimensions $NN = 70$ and $MM = 70$. Then a choice of $h = .09$ cm. will give rise to the problem geometry as illustrated by Figure 1. From Figure 1 and the properties given above, we see that there are five materials, two of which are conductor materials. We now can determine the geometric configuration. Defining

$$S_1 = \{(n,m) \mid 1 \leq n \leq 70, 1 \leq m \leq 70\} \sim \{(n,m) \mid 33 \leq n \leq 49, 14 \leq m \leq 30\},$$

$$S_2 = \{(n,m) \mid 33 \leq n \leq 39, 19 \leq m \leq 25\},$$

and

$$S_3 = \{(n,m) \mid 33 \leq n \leq 39, 14 \leq m \leq 18\} \cup \{(n,m) \mid 33 \leq n \leq 39, 26 \leq m \leq 30\} \cup \{(n,m) \mid 40 \leq n \leq 49, 14 \leq m \leq 30\},$$

Figure 1. Geometry of Example Problem

we have the materials corresponding to the air case. (The reason why we describe S_1 , the air material, as above will be shown later.) Since S_1 is the air material (for the air case), S_2 is the coil material, and S_4 is a ferrite material, we have that:

$$\text{PER}(1) = 1., \text{VCN}(1) = 0., \text{and } \text{IZERO}(1) = 0.;$$

$$\text{PER}(2) = 1., \text{VCN}(2) = 0., \text{and } \text{IZERO}(2) = 1.;$$

$$\text{PER}(3) = 1000., \text{VCN}(3) = 0., \text{and } \text{IZERO}(3) = 0.$$

and $\text{NREGA} = 3$.

The conductor materials can be defined as follows:

$$S_4 = \{(n,m) \mid 1 \leq n \leq 29, 1 \leq m \leq 70\}$$

and

$$S_5 = \{(n,m) \mid 30 \leq n \leq 31, 1 \leq m \leq 70\},$$

where $\text{NCDTOR} = 2$.

Now for the points in S_4 we have that

$$\begin{aligned} h^2_{\omega\mu\sigma} &= (.0009)^2 (3.14159 \times 10^3) (12.56636 \times 10^{-7}) (5.77367 \times 10^7)^* \\ &= .1846. \end{aligned}$$

and similarly for points in S_5 we have that $h^2_{\omega\mu\sigma} = .0784$. Thus

choosing $\text{FMU} = .1846$, we have that:

$$\text{PER}(4) = 1., \text{VCN}(4) = 1., \text{and } \text{IZERO}(4) = 0.;$$

$$\text{PER}(5) = 1., \text{VCN}(5) = .4245, \text{and } \text{IZERO}(5) = 0.$$

Thus, we now have the necessary information for the geometric configuration and have determined that one conductor case of interest will be

* All calculations are done in MKS units (see [1]).

with $FMU = .1846$. We also consider the cases where (on this geometric configuration) $FMU = .2769$, $.3693$, and $.7385$. These additional cases of interest were obtained by considering $\omega = .75$, $1.$, and $2.$, respectively. While all four conductor cases could be computed in a single run of EDDY, we describe the input using alternate procedures outlined in Section 4.6 in order to illustrate the input when $NOPT = 4$.

The procedure for running these cases is as follows:

Run 1: With $NOPT = 2$, run the air case and the conductor case with $FMU = .1846$.

Run 2: With $NOPT = 4$, EXTEND the $FMU = .1846$ case for 1 iteration with a plot request.

Run 3: With $NOPT = 4$, use REDOIT operation on the conductor tape created in Run 1 to run the cases with $FMU = .2769$, $.3693$, and $.7385$.

The formats for the input cards for Run 1, Run 2, and Run 3 are found in Table 1, Table 2, and Table 3, respectively.

The seventh card of Table 1 (corresponding to card type E in Section 4.6) instructs EDDY to initially set up the entire MATNO array as an air material, but the succeeding cards cause EDDY to "over-write" those portions of the array that are coil and ferrite materials. After the air case is completed, the code again "over-writes" portions of the air material with conductor materials as instructed in the 17th and 18th cards of Table 1. This procedure is possible due to the structure of the subroutine MESH (see Appendix C). The procedure to set up the air case geometric configuration illustrated in Table 1 is not a unique one, but the author finds it the easiest.

Table 1
DATA FOR RUN 1

1	5	6	10	11	15	16	20	21	25	26	30	31	35	36	40	41	45	46	50	51	55	56	60	61	65	66	70	71	73	80
3																														
5.0.0		1.0.0		0.0.0	0.1		1.9																							
3		7.0		7.0																										
1		1.0			0.0						0																			
2		1.0			0.0						1																			
3		1.0.0.			0.0						0																			
1		1		7.0		1		7.0																						
2		3.3		3.9		1.9		2.5																						
3		3.3		3.9		1.4		1.8																						
3		3.3		3.9		2.6		3.0																						
3		4.0		4.9		1.4		3.0																						
-1																														
.0.0.0.0.1				.0.0.0.0.1						0.0.0.0																				
2																														
4		1.0			1.0					0																				
5		1.0			0.4.2.4.5					0																				
4		1		2.9		1		7.0																						
5		3.0		3.1		1		7.0																						
-2																														
1																														
5.0.0		1.0.0		0.1.8.4.6		1.2																								

Table 2
DATA FOR RUN 2

[illegible]

Table 3
DATA FOR RUN 3

[illegible]

The results of these cases are as follows:

- (1) The air case converged in 116 iterations, and

$$\hat{S} = .34859 \times 10^6.$$

- (2) The conductor case FMU = .1846 converged in 287 iterations,
and

$$S = (.23016 \times 10^6, - .38849 \times 10^5)$$

$$Z_n = (.11145, .66026).$$

- (3) The conductor case FMU = .2769 converged in 414 iterations,
and

$$S = (.22018 \times 10^6, - .35923 \times 10^5)$$

$$Z_n = (.10305, .63163).$$

- (4) The conductor case FMU = .3693 converged in 369 iterations,
and

$$S = (.21372 \times 10^6, - .34165 \times 10^5)$$

$$Z_n = (.09801, .61311).$$

- (5) The conductor case FMU = .7385 converged in 418 iterations,
and

$$S = (.19920 \times 10^6, - .29848 \times 10^5)$$

$$Z_n = (.08563, .57145).$$

The results of Run 2 are illustrated in Figures 2, 3, and 4.

Figure 2
MAG CONTOURS FMU = 0.185

Figure 3
PHA CONTOURS $FMU = 0.185$

Figure 4
FZ CONTOURS FMU = 0.185

APPENDIX C

CODE LISTING

```

C ***** EDDY *****
C *** MAIN CONTROL DECK ***
COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1      SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2      ML(2),MU(2),MDELTA(2),NL(2),NU(2),NDELTA(2),
3      NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWITCH
COMMON/LIMITS/NNDIM,MMDIM
COMPLEX CA
INTEGER TOTMAT
REAL RA(70,70)
EQUIVALENCE (RA,CA)
DATA CK1/6HEXTEND/,CK2/6HREDOIT/,CK3/6HFINISH/
C      ALTIO MUST BE SPECIFIED.
C      LOG. NO. . . PURPOSE
C          9 (A6)  AIR VECTOR POTENTIAL TAPE--OUTPUT IF NOPT#1,2
C                                  INPUT IF NOPT#3
C          11 (A8) CONDUCTOR VECTOR POTENTIAL TAPE--OUTPUT IF NOPT#2,3
C                                  SCRATCH IF NOPT#4
C          12 (B5)  OLD CONDUCTOR MASTER--USED IFF NOPT#4 AND IS INPUT
C          13 (B6)  NEW CONDUCTOR MASTER--USED IFF NOPT#4 AND IS OUTPUT
C          15 (B8)  PLOT TAPE
C ***INITIALIZE***
      NNDIM#70
      MMDIM#70
      ISWITCH#0
      FMUSAV#0
      DO 1 M#1,70
      DO 1 N#1,70
      1 CA(N,M)#(0.0,0.0)
      READ(5,8000)NOPT
      8000 FORMAT(I5)
      GO TO(10,10,100,200),NOPT
C ***AIR CASE***
      10 READ(5,8010)MAXCT,KOUT,EPS,RELAX
      8010 FORMAT(2I5,2F10.5)
      WRITE(6,8020)NOPT,MAXCT,KOUT,EPS,RELAX
      8020 FORMAT(20H1AIR CASE WITH NOPT#,11/11HOTO BE RUN ,I5,2X,
1          26HMAX. ITERATIONS WITH KOUT#,I5,2X,4HEPS#,E15.7,2X,
2          20HAND WITH RELAXATION#,F10.5)
      CALL GEOM(1)
      CALL AIRVP
      GO TO(50,30),NOPT
C ***CONDUCTOR CASES***
      30 CALL CNVERT
      40 READ(5,8030)EPS,EPSI,(ITB(1),I#1,4)
      8030 FORMAT(2F10.0,5X,4I1)
      WRITE(6,8040)EPS,EPSI,(ITB(1),I#1,4)
      8040 FORMAT(19H1CONDUCTOR CASES.../10H0WITH EPS#,E17.8,2X,5HEPSI#,
1          E17.8,2X,11HAND ITB(1)#,4I2)
      IF((ITB(1)+ITB(2)+ITB(3)+ITB(4)).NE.0) CALL INPLT

```

```

      CALL GEOM(2)
      READ(5,8000) NCASES
      CALL CONDTI(NCASES,FMUSAV,1)
      CALL ENDTPE
50  RETURN
100 CALL AIRRD
    GO TO 40
200 READ(5,8030) EPS, EPSI, (ITB(1), I#1, 4)
    WRITE(6,8040) EPS, EPSI, (ITB(1), I#1, 4)
    IF((ITB(1)+ITB(2)+ITB(3)+ITB(4)).NE.0) CALL INPLT
210 READ(5,8050) OPTION, NCASES
8050 FORMAT(A6, I4)
C *** TEST FOR EXTEND REQUEST ***
    IF(OPTION.EQ.CK1) GO TO 220
C *** TEST FOR REDOIT REQUEST ***
    IF(OPTION.EQ.CK2) GO TO 300
C *** TEST FOR FINISH ***
    IF(OPTION.EQ.CK3) GO TO 400
C ERROR CONDITION -- FAULTY OPTION CARD
    WRITE(6,8055) OPTION
8055 FORMAT(28H1 FAULTY OPTION CARD..OPTION#, A6/60H0 PROGRAM WILL UPDATE
    IOLD CONDUCTOR TAPE AS MUCH AS POSSIBLE.)
    GO TO 400
220 CALL CONDTI(NCASES, FMUSAV, 2)
    GO TO 210
300 READ(5,8060) FMUST
8060 FORMAT(F10.5)
    IF(FMUST.GE.FMUSAV) GO TO 315
    WRITE(6,8070) FMUSAV, FMUST, NCASES
8070 FORMAT(31H0 FMU START IS LESS THAN FMUSAV#, F6.3, 2X, I2H WHERE FMUST#,
    F6.3, 2X, 25H NO CALCULATIONS FOR THESE, I4, 2X, 6H CASES.)
    DO 310 K#1, NCASES
    READ(5,8080)
8080 FORMAT(80H
1
310 WRITE(6,8080)
    GO TO 210
315 IF(FMUST.EQ.FMUSAV) GO TO 320
    FMUSAV#FMUST
    FMU#FMUST
    CALL CONDRD
320 CALL CONDTI(NCASES, FMUSAV, 1)
    GO TO 210
400 CALL ENDTPE
    CALL UPDATE
    WRITE(6,8090)
8090 FORMAT(17H0 UPDATE COMPLETED)
    RETURN
    END

```

```

SUBROUTINE INPLT
COMMON//CA(70,70), MATNO(70,70), MAXCT, KOUT, EPS, EPSI, RELAX, MM, NN,
1      SUMA, FMU, PER(9), VCN(9), IZERO(9), ITB(4),
2      ML(2), MU(2), MDELTA(2), NL(2), NU(2), NDELTA(2),
3      NCONT1, CONT1(20), NCONT2, CONT2(20), NCONT3, CONT3(20),

```

```

      4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWTC
      COMPLEX CA
8000 FORMAT(7I5)
8010 FORMAT(8F10.5)
      IF(ITB(1).EQ.0)GO TO 10
      READ(5,8000)ML(1),MU(1),MDELTA(1),NL(1),NU(1),NDELTA(1),NCONT1
      READ(5,8010)(CONT1(I),I#1,NCONT1)
10  IF(ITB(2).EQ.0)GO TO 20
      READ(5,8000)ML(2),MU(2),MDELTA(2),NL(2),NU(2),NDELTA(2),NCONT2
      READ(5,8010)(CONT2(I),I#1,NCONT2)
20  IF(ITB(3).EQ.0)GO TO 30
      READ(5,8000)NCONT3
      READ(5,8010)(CONT3(I),I#1,NCONT3)
30  IF(ITB(4).EQ.0)GO TO 40
      READ(5,8000)NCONT4
      READ(5,8010)(CONT4(I),I#1,NCONT4)
40  RETURN
      END

```

```

      SUBROUTINE CONDT1(NCASES,FMUSAV,INDEX)
      COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1      SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2      ML(2),MU(2),MDELTA(2),NL(2),NU(2),NDELTA(2),
3      NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWTC
      COMPLEX CA
      INTEGER TOTMAT
      DO 20 K#1,NCASES
      K#K
      READ(5,8000)MAXCT,KOUT,FMU,RELAX
8000 FORMAT(2I5,2F10.5)
      WRITE(6,8010)FMU,RELAX,MAXCT,KOUT
8010 FORMAT(22H1 CONDUCTOR CASE...FMU#,F10.5,2X,11HRELAXATION#,F10.5/
1      7HMAXCT#,I5,2X,5HKOUT#,I5)
      IF(FMU-FMUSAV)30,40,10
10  GO TO(16,15),INDEX
15  CALL CONDRD
16  CALL CONDVP
      IF((ITB(1)+ITB(2)+ITB(3)+ITB(4)).NE.0) CALL PLOTS
      FMUSAV#FMU
20  CONTINUE
      RETURN
30  WRITE(6,8020)FMU,FMUSAV
8020 FORMAT(5H1FMU#,F6.3,21H IS LESS THAN FMUSAV#,F6.3)
      WRITE(6,8025)
8025 FORMAT(71H NO CALCULATIONS FOR THIS AND THE REMAINING CASES IN THI
      IS OPTION BATCH.)
      GO TO 50
40  WRITE(6,8030)FMU,FMUSAV
8030 FORMAT(5H1FMU#,F6.3,20H IS EQUAL TO FMUSAV#,F6.3)
      WRITE(6,8025)
      GO TO(41,45),INDEX
41  WRITE(6,8040)
8040 FORMAT(14H0YOU CANNOT.../31H CALCULATE THE SAME CASE TWICE/
1      29H OR REDO A CASE JUST EXTENDED/

```

```

      2      54H OR USE REDO TO EXTEND A CASE IE HAVE FMUST#FIRST FMU.)
      GO /TO 50
45 WRITE(6,8050)
8050 FORMAT(14H0YOU CANNOT.../29H EXTEND A CASE JUST EXTENDED/
      1      45H OR EXTEND A CASE JUST OPERATED ON BY REDCIT.)
50 NLEFT#NCASES-K
   IF(NLEFT.EQ.0)GO TO 60
   DO 55 K#1,NLEFT
   READ(5,8060)
8060 FORMAT(
      1
55 WRITE(6,8060)
60 RETURN
   END

```

```

$ORIGIN      ALPHA
$INCLUDE     FIOB,FRDB.,FWRB.,FBIN.,FRWT.,FCAS

```

```

      SUBROUTINE GEOM(INDEX)
      COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
      1      SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
      2      ML(2),MU(2),MDelta(2),NL(2),NU(2),NDelta(2),
      3      NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
      4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWCH
      COMMON/LIMITS/NNDIM,MMDIM
      COMPLEX CA
      INTEGER TOTMAT
C ****INDEX#1 IMPLIES CREATE AIR GEOMETRY MESH WHICH IS WRITTEN ON
C      LOG. TAPE 9 (A6).
C ****INDEX#2 IMPLIES CREATE CONDUCTOR GEOMETRY MESH WHICH IS WRITTEN ON
C      LOG. TAPE 11 (A8).
      DATA IFLAG1,IFLAG2/-1,-2/
      GO TO(1,100),INDEX
      1 READ(5,8000)NREGA,NN,MM
8000 FORMAT(3I5)
      IF(NN.GT.NNDIM) GO TO 5
      IF(MM.LE.MMDIM) GO TO 10
      5 WRITE(6,8010)NN,MM
8010 FORMAT(11HIEITHER NN#,13,7H OR MM#,13,51H EXCEEDS ITS ALLOWABLE LI
      MIT. OPERATION TERMINATED.)
      CALL EXIT
      10 CONTINUE
      CALL INPUT(NREGA)
      CALL MESH(IFLAG1)
      CALL OUTPUT(1)
      RETURN
      100 READ(5,8000)NCDTOR
      CALL INPUT(NCDTOR)
      CALL MESH(IFLAG2)
      TOTMAT#NREGA+NCDTOR
      CALL OUTPUT(2)
      RETURN
      END

```



```

SUBROUTINE INPUT(K)
COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1      SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2      ML(2),MU(2),MDELTA(2),NL(2),NU(2),NDELTA(2),
3      NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWCH
COMPLEX CA
C *****THIS ROUTINE READS PROPERTIES FOR K MATERIALS.
DO 10 L#1,K
10 READ(5,8000)IMAT,PER(IMAT),VCN(IMAT),IZERO(IMAT)
8000 FORMAT(12,3X,2F10.5,4X,11)
IF((VCN(IMAT)*FLOAT(IZERO(IMAT))).EQ.0.) GO TO 20
WRITE(6,8010)IMAT,PER(IMAT),VCN(IMAT),IZERO(IMAT)
8010 FORMAT(21H)THE MATERIAL INDEXED,I2,11H WHERE PER#,E15.7,5H VCN#,
1E15.7,11H AND IZERO#,I2/40H IS NOT ALLOWABLE. OPERATION TERMINATED
2.)
20 CONTINUE
RETURN
END

```

```

SUBROUTINE MESH(IFLG)
COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1      SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2      ML(2),MU(2),MDELTA(2),NL(2),NU(2),NDELTA(2),
3      NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWCH
COMPLEX CA
C *****THIS ROUTINE READS THE RECTANGULAR BOUNDS OF THE REGIONS THAT
C APPROXIMATES THE MATERIALS AND SETS UP THE MESH.
10 READ(5,8000)IMAT,NLG,NUG,MLG,MUG
8000 FORMAT(12,3X,4I5)
IF(IMAT.EQ.1)GO TO 30
DO 20 N#NLG,NUG
DO 20 M#MLG,MUG
20 MATNO(N,M)#IMAT
GO TO 10
30 RETURN
END

```

```

SUBROUTINE OUTPUT(JNT)
COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1      SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2      ML(2),MU(2),MDELTA(2),NL(2),NU(2),NDELTA(2),
3      NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWCH
COMPLEX CA
INTEGER TOTMAT
C *****THIS ROUTINE OUTPUTS THE MESH AND WRITES THE GEOMETRY SPECIFICATIONS
C ON EITHER THE AIR OR CONDUCTOR TAPE (DEPENDING ON JNT).
GO TO(10,20),JNT
10 WRITE(6,8000)
8000 FORMAT(32H)A MAP OF THE AIR CASE GEOMETRY /(
C **WRITE AIR CASE TAPE GEOMETRY

```

```

WRITE(9)NREGA,NN,MM,(PER(1),VCN(1),IZERO(1),I#1,NREGA),
1      ((MATNO(N,M),M#1,MM),N#1,NN)
  LASTR#NREGA
  GO TO 30
20 WRITE(6,8010)
8010 FORMAT(38H1A MAP OF THE CONDUCTOR CASE GEOMETRY //)
C  **WRITE CONDUCTOR CASES TAPE GEOMETRY
  WRITE(11)TOTMAT,NN,MM,(PER(1),VCN(1),IZERO(1),I#1,TOTMAT),
1      ((MATNO(N,M),M#1,MM),N#1,NN)
  LASTR#TOTMAT
30 DO 40 N#1,NN
  N1#(NN+1)-N
40 WRITE(6,8020)(MATNO(N1,M),M#1,MM)
8020 FORMAT(1H ,7011)
  WRITE(6,8030)
8030 FORMAT(33H0WITH THE FOLLOWING PROPERTIES.../8H0MAP NO.,2X,
1      21HRELATIVE PERMEABILITY,2X,12H FMU WEIGHT ,2X,
2      14HCURRENT SWITCH)
  WRITE(6,8040)(I,PER(1),VCN(1),IZERO(1),I#1,LASTR)
8040 FORMAT(1H ,3X,11,5X,F12.6,11X,F12.6,9X,11)
  RETURN
  END

```

```

SUBROUTINE AIRVP
COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1      SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2      ML(2),MU(2),MDelta(2),NL(2),NU(2),NDelta(2),
3      NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWICH
  COMPLEX CA
  REAL RA(70,70)
  EQUIVALENCE (RA,CA)
  DATA TPI/6.2831853/
  MMM#MM-1
  NNN#NN-1
  KOUNTA#0
  OLDSUM#0.
  WRITE(6,8000)
8000 FORMAT(49H0SUM OVER THE COIL PER ITERATION FOR AIR CASE....)
10 SUMA#0.
  DO 20 M#2,MMM
  R#M-1
  RINV#1./R
  GAM#1.+RINV
  PSI#1.+GAM+RINV*RINV
  DO 20 N#2,NNN
  K#MATNO(N,M)
  K1#MATNO(N,M+1)
  K2#MATNO(N+1,M)
  VPER#PER(K)
  CURI#IZERO(K)
  ALPHA#VPER/PER(K1)
  BETA#VPER/PER(K2)
  RA(N,M)#RA(N,M)+RELAX*((GAM*ALPHA*RA(N,M+1) + RA(N,M-1) +
1      BETA*RA(N+1,M) + RA(N-1,M) + VPER*CURI)/(PSI+ALPHA+BETA)

```

```

      2      - RA(N,M))
      IF(CURI.EQ.0.)GO TO 20
      SUMA#SUMA+TPI*R*RA(N,M)
20  CONTINUE
      KOUNTA#KOUNTA+1
      IF(KOUNTA.LE.KOUT)GO TO 30
      IF(ABS((SUMA-OLDSUM)/SUMA).LE.EPS)GO TO 40
30  IF(KOUNTA.GE.MAXCT)GO TO 50
      OLDSUM#SUMA
      WRITE(6,8010)KOUNTA,SUMA
8010 FORMAT(1H ,I4,4X,E17.9)
      GO TO 10
40  WRITE(6,8020)KOUNTA,SUMA
8020 FORMAT(23H0CONVERGENCE FOR AIR IN,I6,2X,10HITERATIONS,2X,
1      14HSUM OVER COIL#,E17.9)
      GO TO 60
50  WRITE(6,8010)KOUNTA,SUMA
      WRITE(6,8030)
8030 FORMAT(35H0ITERATION COUNTER EXCEEDED FOR AIR)
C ****CALCULATE AIRMAX****
60  AIRMAX#0.
      DO 70 N#1,NN
      DO 70 M#1,MM
70  IF(RA(N,M).GT.AIRMAX) AIRMAX#RA(N,M)
      WRITE(6,8040)AIRMAX
8040 FORMAT(37H0THE MAXIMUM VECTOR POTENTIAL IN AIR#,E17.9)
      WRITE(9)SUMA,AIRMAX,((RA(N,M),M#1,MM),N#1,NN)
      END FILE 9
      REWIND 9
      RETURN
      END

```

```

SUBROUTINE CONDVP
COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1      SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2      ML(2),MU(2),MDelta(2),NL(2),NU(2),NDelta(2),
3      NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWCH
COMPLEX CA
INTEGER TOTMAT
REAL RA(70,70)
EQUIVALENCE (RA,CA)
COMPLEX OLDSUM,SUMC,ZI,E,ZN
DATA TPI/6.2831853/
MMM#MM-1
NNN#NN-1
KOUNTC#0
OLDSUM#(0.0,0.0)
ZI#(0.0,1.0)
WRITE(6,8000)
8000 FORMAT(55H0SUM OVER THE COIL PER ITERATION FOR THE CONDUCTOR CASE)
10 SUMC#(0.0,0.0)
DO 20 M#2,MMM
R#M-1
RINV#1./R

```

```

GAM#1.+RINV
PSI#1.+GAM+RINV*RINV
DO 20 N#2,NNN
K#MATNO(N,M)
K1#MATNO(N,M+1)
K2#MATNO(N+1,M)
VPER#PER(K)
FMUBAR#VCN(K)*FMU
CURI#IZERO(K)
ALPHA#VPER/PER(K1)
BETA#VPER/PER(K2)
CA(N,M)#CA(N,M)+RELAX*((GAM*ALPHA*CA(N,M+1) + CA(N,M-1) +
1      BETA*CA(N+1,M) + CA(N-1,M) + VPER*CURI)/
2      (PSI+ALPHA+BETA + ZI*FMUBAR) - CA(N,M))
20 IF(CURI.NE.0.) SUMC#SUMC+TPI*R*CA(N,M)
KOUNTC#KOUNTC+1
IF(KOUNTC.LE.KOUT)GO TO 30
IF(ABS((REAL(SUMC)-REAL(OLDSUM))/REAL(SUMC)).GT.EPS)GO TO 30
IF(ABS((AIMAG(SUMC)-AIMAG(OLDSUM))/AIMAG(SUMC)).LE.EPS)GO TO 40
30 IF(KOUNTC.GE.MAXCT)GO TO 50
OLDSUM#SUMC
WRITE(6,8010)KOUNTC,SUMC
8010 FORMAT(1H ,14,4X,E17.9,2X,E17.9)
GO TO 10
C ***CASE CONVERGES***
40 WRITE(6,8020)KOUNTC,SUMC
8020 FORMAT(39H0CONVERGENCE FOR THIS CONDUCTOR CASE IN,16,2X,
1      10HITERATIONS/15H0SUM OVER COIL#,E17.9,2X,E17.9)
GO TO 60
50 WRITE(6,8010)KOUNTC,SUMC
WRITE(6,8030)
8030 FORMAT(41H0ITERATION COUNTER EXCEEDED FOR THIS CASE)
60 E#ZI*SUMC
ZN#ZI*SUMC/SUMA
WRITE(6,8040)E,ZN,AIRMAX
8040 FORMAT(5H0E#Z#,2(E16.8,2X),4H ZN#,2(E16.8,2X)/8H AIRMAX#,E16.8)
WRITE(11)FMU,SUMA,AIRMAX
WRITE(11)((CA(N,M),M#1,MM),N#1,NN)
RETURN
END

```

```

SUBROUTINE AIRRD
COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1      SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2      ML(2),MU(2),MDelta(2),NL(2),NU(2),NDelta(2),
3      NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWCH
COMPLEX CA
INTEGER TOTMAT
REAL RA(70,70)
EQUIVALENCE (RA,CA)
REWIND 9
READ(9)NREGA,NN,MM,(PER(1),VCN(1),IZERO(1),I#1,NREGA),((MATNO(N,M)
1,M#1,MM),N#1,NN)
READ(9)SUMA,AIRMAX,((RA(N,M),M#1,MM),N#1,NN)

```

```

REWIND 9
CALL CNVERT
RETURN
END

```

```

SUBROUTINE CNVERT
COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1      SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2      ML(2),MU(2),MDELTA(2),NL(2),NU(2),NDELTA(2),
3      NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWCH
COMPLEX CA
REAL RA(70,70)
EQUIVALENCE (RA,CA)
DO 10 LI#1,70
M#71-L1
DO 10 L2#1,70
N#71-L2
10 CA(N,M)#RA(N,M)
RETURN
END

```

```

SUBROUTINE CONDRD
COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1      SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2      ML(2),MU(2),MDELTA(2),NL(2),NU(2),NDELTA(2),
3      NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWCH
COMPLEX CA
INTEGER TOTMAT
DATA EOTPE/999.999/
IF(ISWCH.NE.0)GO TO 10
ISWCH#123456
REWIND 12
READ(12)TOTMAT,NN,MM,(PER(1),VCN(1),IZERO(1),I#1,TOTMAT),
1      ((MATNO(N,M),M#1,MM),N#1,NN)
WRITE(6,8010)
8010 FORMAT(38H1A MAP OF THE CONDUCTOR CASE GEOMETRY //)
LASTR#TOTMAT
DO 1 N#1,NN
NI#(NN+1)-N
1 WRITE(6,8020)(MATNO(NI,M),M#1,MM)
8020 FORMAT(1H ,70I1)
WRITE(6,8030)
8030 FORMAT(33H0WITH THE FOLLOWING PROPERTIES.../8H0MAP NO.,2X,
1      21HRELATIVE PERMEABILITY,2X,12H FMU WEIGHT ,2X,
2      14HCURRENT SWITCH)
WRITE(6,8040)(I,PER(I),VCN(I),IZERO(I),I#1,LAstr)
8040 FORMAT(1H ,3X,11,5X,F12.6,11X,F12.6,9X,11)
10 READ(12)FMUIDT,SUMA,AIRMAX
IF(FMU.EQ.FMUIDT)GO TO 20
IF(FMUIDT.EQ.EOTPE)GO TO 30
READ(12)DUMMY

```

```

      GO TO 10
20  READ(12)((CA(N,M),M#1,MM),N#1,NN)
      RETURN
30  WRITE(6,8000)FMU
8000 FORMAT(5HIFMU#,F10.5,59H WAS NOT FOUND ON THE INPUTED CONDUCTOR TA
      IPE. CALLING EXIT.)
      REWIND 12
      CALL EXIT
      RETURN
      END

```

```

SUBROUTINE UPDATE
COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPS1,RELAX,MM,NN,
1    SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2    ML(2),MU(2),MDELTA(2),NL(2),NU(2),NDELTA(2),
3    NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4    NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWCH
COMPLEX CA
INTEGER TOTMAT
COMMON/UPDTE/NFILES
DATA EOTPE/999.999/
C THIS ROUTINE TAKES THE OLD MASTER--UNIT 12--AND MERGES THE LATEST
C CALCULATIONS ON THE SCRATCH--UNIT 11--TO FORM A NEW MASTER
C ON UNIT 13.
      REWIND 11
      REWIND 13
      REWIND 12
      READ(12)DUMMY
      WRITE(13)TOTMAT,NN,MM,(PER(1),VCN(1),IZERO(1),I#1,TOTMAT),
      ((MATNO(N,M),M#1,MM),N#1,NN)
      WRITE(6,8000)
8000 FORMAT(52HIDictionary of the CONDUCTOR CASES ON THE NEW MASTER/
110HDFMU CASE ,5X,35HTAPE PULLED FROM(12#OLD,11#SCRATCH))
      READ(12)FMU12,SUMA12,AIRM12
      READ(11)FMU11,SUMA11,AIRM11
      NFILES#0
      10 IF(FMU11.EQ.EOTPE)GO TO 50
      11 IF(FMU12.EQ.EOTPE)GO TO 60
      IF(FMU12-FMU11)20,30,40
C ***** FMU12 IS LESS THAN FMU11 *****
      20 CALL OPTAPE(12,FMU12,SUMA12,AIRM12)
      GO TO 11
C ***** FMU12 IS EQUAL TO FMU11 *****
      30 CALL OPTAPE(11,FMU11,SUMA11,AIRM11)
      READ(12)DUMMY
      READ(12)FMU12,SUMA12,AIRM12
      GO TO 10
C ***** FMU11 IS LESS THAN FMU12 *****
      40 CALL OPTAPE(11,FMU11,SUMA11,AIRM11)
      GO TO 10
C ***** EOF ENCOUNTERED ON UNIT 11 -- COPY REST OF UNIT 12 *****
      50 IF(FMU12.EQ.EOTPE)GO TO 70
      CALL OPTAPE(12,FMU12,SUMA12,AIRM12)
      GO TO 50
C ***** EOF ENCOUNTERED ON UNIT 12 -- COPY REST OF UNIT 11 *****

```

```

60 IF(FMU11.EQ.EOTPE)GO TO 70
   CALL OPTAPE(11,FMU11,SUMA11,AIRM11)
   GO TO 60
70 REWIND 11
   REWIND 12
   XFLES#NFILES
   WRITE(13)EOTPE,XFLES,EOTPE
   END FILE 13
   REWIND 13
   WRITE(6,8010)NFILES
8010 FORMAT(29H0THE NO. OF CONDUCTOR FILES# ,14)
   WRITE(6,8020)
8020 FORMAT(11H1END OF RUN)
   RETURN
   END

```

```

SUBROUTINE OPTAPE(NTAPE,X1,X2,X3)
COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1      SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2      ML(2),MU(2),MDELTA(2),NL(2),NU(2),NDELTA(2),
3      NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWCH
COMPLEX CA
INTEGER TOTMAT
COMMON/UPDTE/NFILES
WRITE(6,8000)X1,NTAPE
8000 FORMAT(1H ,F9.5,5X,12)
IF(NTAPE.EQ.12) GO TO 10
READ(11)((CA(N,M),M#1,MM),N#1,NN)
WRITE(13)X1,X2,X3
READ(11)X1,X2,X3
GO TO 20
10 READ(12)((CA(N,M),M#1,MM),N#1,NN)
   WRITE(13)X1,X2,X3
   READ(12)X1,X2,X3
20 WRITE(13)((CA(N,M),M#1,MM),N#1,NN)
   NFILES#NFILES+1
   RETURN
   END

```

```

SUBROUTINE ENDTPE
COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1      SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2      ML(2),MU(2),MDELTA(2),NL(2),NU(2),NDELTA(2),
3      NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWCH
COMPLEX CA
DATA EOTPE/999.999/
WRITE(11)EOTPE,EOTPE,EOTPE
END FILE 11
REWIND 11
RETURN
END

```

\$ORIGIN

ALPHA

```

SUBROUTINE PLOTS
COMMON/ARRAY/F(70,70)
COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1      SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2      ML(2),MU(2),MDELTA(2),NL(2),NU(2),NDELTA(2),
3      NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWCH
COMPLEX CA
WRITE(6,8000)
8000 FORMAT(1H1)
10 IF(ITB(1).EQ.0)GO TO 20
CALL PLTMAG
20 IF(ITB(2).EQ.0)GO TO 30
CALL PLTPHS
30 IF(ITB(3).EQ.0)GO TO 40
CALL PLTFC(1)
40 CONTINUE
RETURN
END

```

```

SUBROUTINE PLTMAG
COMMON/ARRAY/F(70,70)
COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1      SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2      ML(2),MU(2),MDELTA(2),NL(2),NU(2),NDELTA(2),
3      NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWCH
COMPLEX CA
DIMENSION HOL(5)
DATA HOL(1)/18H MAG CONTOURS FMU#/,HOL(5)/07777777777777/
EXTERNAL EVAL
N1#NL(1)
N2#NU(1)
YNL#N1
ND#NDELTA(1)
YD#ND
M1#ML(1)
M2#MU(1)
XML#M1
MD#MDELTA(1)
XD#MD
10 WRITE(0,8000)FMU
8000 FORMAT(F6.3)
READ(0,8010)HOL(4)
8010 FORMAT(A6)
NR#0
DO 20 N#N1,N2,ND
NR#NR+1
NC#0
DO 20 M#M1,M2,MD
NC#NC+1
20 F(NC,NR)#SQRT(REAL(CA(N,M))**2 + AIMAG(CA(N,M))**2)/AIRMAX

```



```

      CALL CONPLT(XML,YNL,XD,YD,NC,NR,10.,10.,NCONT1,CONT1,2,EVAL,F,
1       HOL(1))
      WRITE(6,8020)FMU,M1,M2,MD,N1,N2,ND,NC,NR
8020  FORMAT(34H0MAGNITUDE PLOT COMPLETED FOR FMU#,F6.3,5X,33H WITH THE
      FOLLOWING PARAMETERS.../4H ML#,I3,2X,4H MU#,I3,2X,8H MDELTA#,I3,
2       2X,4H NL#,I3,2X,4H NU#,I3,2X,8H NDELTA#,I3/
3       13H NO. OF COLS#,I3,2X,13H NO. OF ROWS#,I3)
      RETURN
      END

```

```

SUBROUTINE PLTPHS
COMMON/ARRAY/F(70,70)
COMMON/CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1       SUMA,FMU,PER(9),VCN(9),IZERO(9),ITB(4),
2       ML(2),MU(2),MDELTA(2),NL(2),NU(2),NDELTA(2),
3       NCONT1,CONT1(20),NCONT2,CONT2(20),NCONT3,CONT3(20),
4       NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWCH
COMPLEX CA
DIMENSION HOL(5)
DATA HOL(1)/18H PHA CONTOURS FMU#/,HOL(5)/07777777777777/
DATA PI/3.14159265/
EXTERNAL EVAL
N1#NL(2)
N2#NU(2)
YNL#N1
ND#NDELTA(2)
YD#ND
M1#ML(2)
M2#MU(2)
XML#M1
MD#MDELTA(2)
XD#MD
10  WRITE(0,8000)FMU
8000  FORMAT(F6.3)
      READ(0,8010)HOL(4)
8010  FORMAT(A6)
      NR#0
      DO 20 N#N1,N2,ND
      NR#NR+1
      NC#0
      DO 20 M#M1,M2,MD
      NC#NC+1
      PHS#ATAN2(AIMAG(CA(N,M)),REAL(CA(N,M)))
      IF(PHS.LT.0.) PHS#2.*PI + PHS
20  F(NC,NR)#PHS
      CALL CONPLT(XML,YNL,XD,YD,NC,NR,10.,10.,NCONT2,CONT2,2,EVAL,F,
1       HOL(1))
      WRITE(6,8020)FMU,M1,M2,MD,N1,N2,ND,NC,NR
8020  FORMAT(30HOPHASE PLOT COMPLETED FOR FMU#,F6.3,5X,33H WITH THE FOLL
      OWING PARAMETERS.../4H ML#,I3,2X,4H MU#,I3,2X,8H MDELTA#,I3,2X,
2       4H NL#,I3,2X,4H NU#,I3,2X,8H NDELTA#,I3/
3       13H NO. OF COLS#,I3,2X,13H NO. OF ROWS#,I3)
      RETURN
      END

```

```

SUBROUTINE PLTFC(INDEX)
COMMON/ARRAY/F(70,70)
COMMON//CA(70,70),MATNO(70,70),MAXCT,KOUT,EPS,EPSI,RELAX,MM,NN,
1      SUMA,FMU,PER(9),VCN(9),IZERC(9),ITS(4),
2      ML(2),MU(2),MDELTA(2),NL(2),NU(2),NDELTA(2),
3      NCONT1,CONT1(20),NCONT2,CCNT2(20),NCONT3,CONT3(20),
4      NCONT4,CONT4(20),NREGA,TOTMAT,AIRMAX,ISWCH
COMPLEX CA
DIMENSION HOL1(5),HOL2(5)
DATA HOL1(1)/18H FZ CONTOURS FMU#/,HOL1(5)/077777777777777/
DATA HOL2(1)/18H FR CONTOURS FMU#/,HOL2(5)/077777777777777/
EXTERNAL EVAL
MUC#0
NUC#0
MLC#70
NLC#70
MMM#MM-1
DO 10 M#1,MMM
DO 10 N#2,NN
IREG#MATNO(N,M)
IF(VCN(IREG).EQ.0.)GO TO 10
IF(N.LT.NLC) NLC#N
IF(N.GT.NUC) NUC#N
IF(M.LT.MLC) MLC#M
IF(M.GT.MUC) MUC#M
10 CONTINUE
NC#(MUC-MLC)+1
NR#(NUC-NLC)+1
XML#MLC
YNL#NLC
20 GO TO(30,70),INDEX
30 WRITE(0,8000)FMU
8000 FORMAT(F6.3)
READ(0,8010)HOL1(4)
8010 FORMAT(A6)
FZSUM#0.
FMAX#0.
DO 50 M#MLC,MUC
R#M-1
DO 50 N#NLC,NUC
IREG#MATNO(N,M)
IF(VCN(IREG).EQ.0.)GO TO 40
F(M,N)#REAL(CA(N,M))*AIMAG(CA(N-1,M))
- AIMAG(CA(N,M))*REAL(CA(N-1,M))
FZSUM#FZSUM + R*F(M,N)
ABSFZ#ABS(F(M,N))
IF(ABSFZ.GT.FMAX) FMAX#ABSFZ
GO TO 50
40 F(M,N)#0.
50 CONTINUE
DO 60 N#1,NR
NV#(N-1)+NLC
DO 60 M#1,NC
MV#(M-1)+MLC
60 F(M,N)#F(MV,NV)/FMAX
CALL CONPLT(XML,YNL,1.,1.,NC,NR,10.,10.,NCONT3,CONT3,2,EVAL,F,
1      HOL1(1))

```

```

      WRITE(6,8020)FMU,FZSUM,MLC,MUC,NLC,NUC,NC,NR
8020 FORMAT(39H0FORCE IN Z CONTOURS COMPLETED FOR FMU#,F6.3,3X,
1      12HWITH FZ SUM#,E17.9/52H WHERE THE REGION PLOTTED IS DEFIN
2ED BY FOLLOWING.../4H ML#,I3,2X,4H MU#,I3,2X,4H NL#,I3,2X,4H NU#,
3      13,2X,I3H NO. OF COLS#,I3,2X,I3H NO. OF ROWS#,I3)
      RETURN
70 WRITE(0,8000)FMU
   READ(0,8010)HOL2(4)
   FRSUM#0.
   FMAX#0.
   DO 90 M#MLC,MUC
   R#M-1
   DO 90 N#NLC,NUC
   IREG#MATNO(N,M)
   IF(VCN(IREG).EQ.0.)GO TO 80
   F(M,N)#((R+1.)/R)*(REAL(CA(N,M+1))*AIMAG(CA(N,M))
   - AIMAG(CA(N,M+1))*REAL(CA(N,M)))
   FRSUM#FRSUM + F(M,N)
   ABSFR#ABS(F(M,N))
   IF(ABSFR.GT.FMAX) FMAX#ABSFR
   GO TO 90
80 F(M,N)#0.
90 CONTINUE
   DO 100 N#1,NR
   NV#(N-1)+NLC
   DO 100 M#1,NC
   MV#(M-1)+MLC
100 F(M,N)#F(MV,NV)/FMAX
   CALL CONPLT(XML,YNL,1.,1.,NC,NR,10.,10.,NCONT4,CONT4,2,EVAL,F,
1      HOL2(1))
   WRITE(6,8030)FMU,FRSUM,MLC,MUC,NLC,NUC,NC,NR
8030 FORMAT(39H0FORCE IN R CONTOURS COMPLETED FOR FMU#,F6.3,3X,
1      12HWITH FR SUM#,E17.9/52H WHERE THE REGION PLOTTED IS DEFIN
2ED BY FOLLOWING.../4H ML#,I3,2X,4H MU#,I3,2X,4H NL#,I3,2X,4H NU#,
3      13,2X,I3H NO. OF COLS#,I3,2X,I3H NO. OF ROWS#,I3)
      RETURN
      END

```

```

SUBROUTINE CONPLT(XX0,YY0,DDX,DDY,NNX,NNY,XXA,YYA,NC,CON,NSYMB,
1 EVAL,F,HOL)
CONPLT --- G W STEWART, III, CDPF, ORGDP, 9/10/65 ORD 0007
DIMENSION F(70,70)
DIMENSION CON(1),HOL(1),X(50),Y(50),XP(50),YP(50),
1 IT(50),XS(100),YS(100)
DELTA # .1
XD # XX0
YD # YY0
NX # NNX
NXI # NX-1
FNXI # NXI
NY # NNY
NYI # NY-1
FNYI # NYI
XA # AMINI(XXA,15.)
YA # AMINI(YYA,9.5)

```

```

DX # DDX
DY # DDY
NPX # 1. + AMAXI(4.,AMINI(20.,XA/(DELTA*FNXI)))
NPY # 1. + AMAXI(4.,AMINI(20.,YA/(DELTA*FNXI)))
DPX # 1./FLOAT(NPX-1)
DPY # 1./FLOAT(NPY-1)
DPXI # DPX + 1.E-6
DPYI # DPY + 1.E-6
DX # DDY
DY # DDY
X(1) # XD
X(2) # XD + DX*FNXI
Y(1) # YD
Y(2) # YD + DY*FNXI
CALL SCALE(X,Y,2,XA,YA,2)
CALL LEGEND(0.,-0.45,0.14,0.,0.025,0.,HOL)
C
C   SET UP THE GRID
C
DO 10 I#1,NX
XX # XD + DX*FLOAT(I-1)
DO 10 J#1,NY
10 CALL EVAL(XX,YD+DY*FLOAT(J-1),F(I,J))
C
C   START THE CONTOUR PLOTTING
C
DO 1000 IC#1,NC
KS # 0
ISYMB # -1
IF(NSYMB.LE.0) GO TO 20
IF(MOD(IC-1,NSYMB).EQ.0) ISYMB # (IC-1)/NSYMB
20 DO 750 IYY#1,NYI
IY # IYY + 1
YL # -1.
IF(IY.LT.NY) GO TO 50
IY # NYI
YL # 0.
50 YU # 1. + YL
YLI # YL - 1.E-6
YUI # YU + 1.E-6
YC # YD + DY*FLOAT(IY-1)
DO 500 IXX#1,NXI
IX # IXX + 1
XL # -1.
IF(IX.LT.NX) GO TO 75
IX # NXI
XL # 0.
75 XU # 1. + XL
XLI # XL - 1.E-6
XUI # XU + 1.E-6
XC # XD + DX*FLOAT(IX-1)
C
C   COMPUTE THE APPROXIMATION FOR THIS PANEL
C
F1 # F(IX-1,IY+1)
F2 # F(IX,IY+1)
F3 # F(IX+1,IY+1)

```

```

F4 # F(IX-1,IY)
F5 # F(IX,IY)
F6 # F(IX+1,IY)
F7 # F(IX-1,IY-1)
F8 # F(IX,IY-1)
F9 # F(IX+1,IY-1)
C1 # F5
C2 # (F6-F4)/2.
C3 # (F4-2.*F5+F6)/2.
B1 # (F2-F8)/2.
B2 # (F3-F1-F9+F7)/4.
B3 # (F1-2.*F2+F3-F7+2.*F8-F9)/4.
A1 # (F2-2.*F5+F8)/2.
A2 # (F3-F1+2.*F4-2.*F6+F9-F7)/4.
A3 # (F1-2.*F2+F3-2.*F4+4.*F5-2.*F6+F7-2.*F8+F9)/4.
DO 100 I#1,8
100 IT(I) # -1
KB # 0
KI # 8

C
C COMPUTE CONTOUR POINTS ALONG THE X DIRECTION
C
DO 200 I#1,NPX
KKK # 1
XX # XL + DPX*FLOAT(I-1)
A # XX*(A3*XX+A2)+A1
B # XX*(B3*XX+B2)+B1
C # XX*(C3*XX+C2)+C1-CON(IC)
IF(A.NE.0.) GO TO 110
IF(B.EQ.0.) GO TO 200
KKK # 2
YY # -C/B
GO TO 120
110 D # B**2 - 4.*A*C
IF(D.LT.0.) GO TO 200
D # SIGN(SQRT(D),-B)
YY # (D-B)/(2.*A)
120 IF(YY.LT.YLI .OR. YY.GT.YUI) GO TO 150
IF(I.EQ.1 .OR. I.EQ.NPX) GO TO 140
KI # KI+1
X(KI) # XX
Y(KI) # YY
IT(KI) # 1
GO TO 150
140 KB # KB+1
X(KB) # XX
Y(KB) # YY
IT(KB) # 1
150 GO TO(160,200),KKK
160 KKK # 2
YY # C/(YY*A)
GO TO 120
200 CONTINUE

C
C COMPUTE CONTOUR POINTS ALONG THE Y DIRECTION
C
DO 300 I#1,NPY

```

```

      KKK # 1
      YY # YL + DPY*FLOAT(I-1)
      A # YY*(A3*YY+B3)+C3
      B # YY*(A2*YY+B2)+C2
      C # YY*(A1*YY+B1)+C1-CON(IC)
      IF(A.NE.0.) GO TO 210
      IF(B.EQ.0.) GO TO 300
      KKK # 2
      XX # -C/B
      GO TO 220
210  D # B**2 - 4.*A*C
      IF(D.LT.0.) GO TO 300
      D # SIGN(SQRT(D),-B)
      XX # (D-B)/(2.*A)
220  IF(XX.LT.XLI .OR. XX.GT.XUI) GO TO 250
      IF(I.EQ.1 .OR. I.EQ.NPY) GO TO 240
      KI # KI+1
      X(KI) # XX
      Y(KI) # YY
      IT(KI) # I
      GO TO 250
240  KB # KB+1
      X(KB) # XX
      Y(KB) # YY
      IT(KB) # I
250  GO TO(260,300),KKK
260  KKK # 2
      XX # C/(XX*A)
      GO TO 220
300  CONTINUE
C
C      CONNECT AND PLOT THE CONTOUR POINTS
C
310  IF(KB.EQ.0 .AND. KI.EQ.8) GO TO 500
320  KP # 0
      DO 325 IS#1,KI
      IF(IT(IS).GT.0) GO TO 330
325  CONTINUE
      GO TO 500
330  IT(IS) # 0
      II # IS
335  KP # KP+1
      XP(KP) # X(II)
      YP(KP) # Y(II)
      R # 1.
      II # 0
      DO 350 I#IS,KI
      IF(IT(I))350,340,345
340  IF(KP.LT.4) GO TO 350
345  DXX # ABS(X(I)-XP(KP))
      IF(DXX.GT.DPXI) GO TO 350
      DYY # ABS(Y(I)-YP(KP))
      IF(DYY.GT.DPYI) GO TO 350
      RR # DXX**2 + DYY**2
      IF(RR.GE.R) GO TO 350
      R # RR
      II # I

```

```

350 CONTINUE
   IF(II.FO.0) GO TO 360
   IT(II) # -1
   GO TO 335
360 IT(IS) # -1
   IF(KP.LT.2) GO TO 320
   CALL SCPLLN(XC+DX*XP(I),YC+DY*YP(I),I)
   DO 370 JANE#2,KP
370 CALL SCPLLN(XC+DX*XP(JANE),YC+DY*YP(JANE),2)
   IF(ISYMB.LT.0) GO TO 320
   IF(KS.LT.100) GO TO 390
   KS # 0
   DO 380 I#1,100
380 CALL SCPLPT(XS(I),YS(I),ISYMB)
390 I # KP/2
   KS # KS + 1
   XS(KS) # XC + DX*XP(I)
   YS(KS) # YC + DY*YP(I)
   GO TO 320
500 CONTINUE
750 CONTINUE
   IF(KS.EQ.0) GO TO 1000
   DO 800 I#1,KS
800 CALL SCPLPT(XS(I),YS(I),ISYMB)
1000 CONTINUE
   CALL SCPLLN(X0,Y0+DY*FNYI,-1)
   END FILE 15
   RETURN
   END

```

```

$IBFTC EVALD   DECK
SUBROUTINE EVAL(X,Y,Z)
RETURN
END

```