ORNL/TM-6858

OAK RIDGE NATIONAL LABORATORY

UNION CARBIDE

Computer-Aided Design of Multifrequency Eddy-Current Tests for Layered Conductors with Multiple Property Variations

W. E. Deeds
C. V. Dodd
G. W. Scott

APPLIED TECHNOLOGY

A[illegible] of this document or [illegible] th[illegible] interests [illegible] m[illegible] of U.S. [illegible] Reactor Research and Technology, Department of Energy.

OK
DRH
6/22/05

OPERATED BY
UNION CARBIDE CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

ORNL/TM-6858
Distribution
Category UC-79b,
-h, and -k

Contract No W-7405-eng-26

METALS AND CERAMICS DIVISION

COMPUTER-AIDED DESIGN OF MULTIFREQUENCY EDDY-CURRENT TESTS FOR LAYERED CONDUCTORS WITH MULTIPLE PROPERTY VARIATIONS

W. E. Deeds, C. V. Dodd, and G. W. Scott

Date Published: October 1979

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

CONTENTS

COMPUTER-AIDED DESIGN OF MULTIFREQUENCY EDDY-CURRENT TESTS FOR LAYERED CONDUCTORS WITH MULTIPLE PROPERTY VARIATIONS*

W. E. Deeds, C. V. Dodd, and G. W. Scott

ABSTRACT

Our program is part of a larger project designed to develop multifrequency eddy-current inspection techniques for multi-layered conductors with parallel planar boundaries. To reduce the need to specially program each new problem, we developed a family of programs that handle a large class of related problems with only minor editorial and interactive changes. We developed programs for two types of cylindrical coil probes: the reflection probe, which contains the driver and pickup coils and is used from one side of the specimen, and the through-transmission probe set, which places the driver and pickup coils on opposite sides of the conductor stack.

The programs perform these basic functions: (1) simulation of an ideal instrument's response to specific conductor and defect configurations, (2) control of an eddy-current instrument interfaced to a minicomputer to acquire and record actual instrument responses to test specimens, (3) construction of complex function expansions to relate instrument response to conductor and defect properties by using measured or computed responses and properties, and (4) simulation of a microcomputer on board the instrument by the interfaced minicomputer to test the analytical programming for the microcomputer. The programs are written mostly in Fortran IV for the ModComp IV FR4 Compiler, which is standard (ANSI X3.9-1966) Fortran IV with a few additional features. A few segments are written in the ModComp assembly language for control of the computer-instrument data acquisition interface.

Our report contains the basic equations for the computations, the main and subroutine programs, instructions for editorial changes and program execution, analyses of the main programs, file requirements, and other miscellaneous aids for the user.

INTRODUCTION

The eddy currents induced in conductors by changing currents in nearby coils are affected by many things, including the proximity and design of the

*Work performed under DOE/RRT AG 10 20 45 0, Task 04 (OH061), "Nondestructive Testing."

coils; the size, shape, and electrical properties of the conductors; the presence of the conductors; and the rate of change of the fields produced by the currents. This makes it possible to study various properties of conductors, but it also requires considerable care to distinguish the effects of interest from those caused by changes in the other properties.

To determine a given property unambiguously when several other properties may be simultaneously varying, one needs at least as many pieces of information as there are parameters that may change to eliminate the unwanted variables. The usual ways of obtaining the necessary information from eddy-current measurements include the use of either driving currents that contain multiple sinusoidal harmonics, pulsed amplitudes, or swept-frequency currents that can be Fourier analyzed into various harmonic components or sampled at various times. The objective of these methods is to obtain more bits of information.

Since the theoretical analysis of eddy-current problems has been developed more extensively for sinusoidal driving currents, we shall consider only the use of multiple sinusoidal driving currents; nevertheless, our results and programs (with some modifications) will apply equally well to the Fourier components of pulsed or swept-frequency currents. The data-fitting programs are also applicable to time-domain readings, that is, readings sampled at various times during current pulses or frequency sweeps.

Our programs handle applications involving coaxial cylindrical coils with their common axis perpendicular to a system of conductors bounded by parallel planes.

Four stages in the complete process of determining the properties of unknown samples are: design optimization, acquisition of data from standard samples, fitting the known properties of the standards with empirical formulas involving the eddy-current readings, and using the empirical formulas to calculate the properties of unknown samples from measured readings.

The first step, optimizing the coil and circuit designs for detecting certain properties, is performed with either the Multilayer Reflection Coil Design Program (MULRFD) or Multilayer Through-Transmission Design Program (MULTRU). These prompt the user to type in design parameters.

They then make calculations and permit the user to determine the effectiveness of the design for detecting various properties. The MULTRU program is used for multilayer through-transmission problems and is presently written to include no defects. In both programs the user can try various combinations of test frequencies and various empirical formulas of polynomial expansions to fit the various properties. He or she can quickly obtain a fairly accurate idea of how well a given property can be measured in a given experiment.

The second step of the process, recording data from standard samples, is accomplished by the Reflection Coil Reading Program (RFLRDG), which is applicable to multilayer problems using reflection-type coils. The program prompts the user to type in the data for the various properties, takes the corresponding readings from a multifrequency eddy-current instrument, and then writes all the data in a random access data (RAD) file. Defect properties can be included.

The third and fourth steps of the process, finding the best approximation formulas and then using them to determine the properties of unknown samples, are handled by the Reflection Coil Fitting Program (RFLFIT). It takes the data recorded in the RAD file by the RFLRDG program, prompts the user to give the type of property to be fitted and the type of polynomial to be used, performs a least squares fitting of the coefficients in the polynomial to approximate the property, stores the resulting coefficients in another RAD file, and uses the readings from unknown samples with the coefficients to calculate the desired properties. It is presently set up to handle six properties: resistivity, permeability, thickness, lift-off, defect size, and defect location in one direction in any layer of a multilayer set of conductors.

A brief summary of the theoretical basis is in the next section, followed by users' guides to the various programs. Detailed analyses and explanations of the programs followed by printouts of sample runs are given in Appendices A through D. The subroutines used in the various programs are briefly described and listed in Appendix E. Appendix F contains brief programs that obtain and manipulate the data for various existing coils.

THEORY

Determination of Multiple Test Properties

Properties of conductors

If a coil is excited by a sinusoidal driving current, one can generally measure two quantities at a time, either: the magnitude and phase of the voltage induced in the coil by eddy currents in nearby conductors, the in-phase and out-of-phase components of the induced voltage, the real and imaginary parts of the coil impedance, or other combinations. However, these two quantities are affected by the coil-to-conductor spacing; the frequency of the sinusoidal current; the shape, size, resistivity, and permeability of any conductors in the vicinity; and the sizes, shapes, orientations, and locations of any defects in the conductors. The conductor resistivities and permeabilities may also depend upon the past history of the sample, present field strengths, and the composition of the conductors.

Obviously, more than two measurements are necessary to eliminate undesirable variables and to unambiguously determine those of interest. Among the ways to accomplish this are: (1) take readings at several frequencies simultaneously (the multifrequency technique), (2) use nonsinusoidal amplitude-modulated (*pulsed*) driving currents, or (3) use frequency-modulated (*swept-frequency*) eddy currents. Combinations of these techniques can also be used. The nonsinusoidal currents can have their outputs Fourier analyzed into harmonic components so that the data can be handled as in the first method. However, the pulsed or swept-frequency currents can also be measured as functions of time, and various pattern recognition techniques can be used to correlate the patterns and properties. By the Fourier theorem one can theoretically transform back and forth between the time and frequency domains, but there are practical differences and advantages for each method.

The theory of multiple test properties will be applied to the frequently occurring problems of: (1) fitting an expansion of functions of a set of computed readings to simulate the calculated response of an instrument to the conductor properties from which the readings were computed; (2) fitting a similar expansion by using instrument readings from standards to the properties of those standards, as measured by an independent method; and (3) using a fitted function, determined by method (1) or (2) above, to compute an unknown property from a set of actual instrument readings (measurements). These problems occur in the design, calibration, and operation phases, respectively, of many multiconductor inspections. In each case the task is to assure independence of desired variables through adequate numbers of readings, property sets, and expansion terms.

At present, let us assume that we have N pieces of information or "readings" taken by either multifrequency, pulsed, or swept-frequency methods. Let us also assume that all N readings R_n $(n = 1,2,\ldots,N)$ are linearly independent. We initially assume that a given property, PROP_m, is a linear function of the readings so that

$$\mathrm{PROP}_m = C_{m0} + \sum_{n=1}^{N} C_{mn}R_n \ ,$$

or (1)

$$P_m = \mathrm{PROP}_m - C_{m0} = \sum_{n=1}^{N} C_{mn}R_n \ .$$

The constant term in the expansion, C_{m0}, will be the "offset"; the deviation from the offset, P_m, is a measure of the property value.

If K sets of readings, R_n', $R_n''\ldots$, $R_n^{(K)}$, are made on specimens with the values P_m', P_m'',..., $P_m^{(K)}$, respectively, of the given property, then we can write a set of linear equations:

$$\begin{aligned}
P_m' &= C_{m1}R_1' + C_{m2}R_2' + \ldots + C_{mN}R_N' \\
P_m'' &= C_{m1}R_1'' + C_{m2}R_2'' + \ldots + C_{mN}R_N'' \\
&\ \ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad\qquad \vdots \\
P_m^{(K)} &= C_{m1}R_1^{(K)} + C_{m2}R_2^{(K)} + \ldots + C_{mN}R_N^{(K)} \ ,
\end{aligned} \tag{2}$$

which can be solved uniquely for the coefficients C_{mn} if $K = N$ and the equations are linearly independent.

If some specimens have the same values of the given property, then some of their other properties must differ if the readings are to be linearly independent. For example, the readings for a given sample with and without a defect should produce slightly different readings. Defects with different sizes or locations should also produce different (i.e., linearly independent) readings, even though the properties such as resistivity, permeability, and thickness are unchanged. If $K < N$ a unique set of coefficients cannot be found. Usually, an infinite set of combinations of coefficients can be found to satisfy the underdetermined system. If $K > N$, the system is overdetermined, and no set of coefficients will satisfy all of the equations (except in the unusual case when at least $K - N$ of the equations are *not* linearly independent). In the non-degenerate overdetermined case ($K > N$) we must be satisfied with the compromise set of coefficients that will give the most reasonable approximation to the true properties P_m when the various sets of readings $R_n^{(k)}$ are put into Eq. (2). The usual way is to define the differences,

$$P_m^{(k)} - \sum_{n=1}^{N} C_{mn} R_n^{(k)} = \varepsilon_k, \quad (k = 1, \ldots, K) \ , \tag{3}$$

and then minimize the sum of the squares of the differences,

$$S = \sum_{k=1}^{K} \varepsilon_k^2 \ , \tag{4}$$

by adjusting the set of N coefficients, C_{mn}. These are called the linear least squares coefficients for the property P_m since they occur only to the first power in the equations; furthermore, they give the "best" fit (in the linear least squares sense) to the set of properties $P_m^{(K)}$ in terms of the K sets of readings $R_n^{(k)}$.

Any number of properties P_m can be fitted in this way if enough sets of readings are taken to determine the coefficients. However, this does *not* mean that *any* number of properties can be uniquely determined from a finite number, N, of independent readings. If N is less than the number of properties that may vary, several combinations of properties may produce the *same* set of readings. Therefore, we could not distinguish which combination was actually occurring. If only N properties can be determined with N independent readings, and a larger number of properties may influence the readings, then one must either control some of the properties so they do not vary in the given experiment or increase the number, N, of independent readings. Thus, if M is the number of properties that can be uniquely determined with a single set of readings, N is the number of independent readings taken in that set, and L is the number of sets of independent or "calibration" measurements taken to determine those coefficients, then we have the following relationships:

$$M \leqslant N \leqslant L \leqslant K \quad . \tag{5}$$

Calibration measurements are taken on samples with known (i.e., independently measured) values of a property to determine the coefficients used for measuring samples with unknown values of the property.

The important limitation to remember is this: although the number of properties that can be *fitted* is limited in principle only by the total number of calibration readings (K), the number of properties that can be *uniquely determined* (M) with a single test is limited by the number of readings made in that test (N).

The inequality between N and L may seem surprising, but we need not assume that a linear expansion [such as Eq. (2)] is the best or only type that can be used. In fact it is not ideal, except for *very small* variations in the property.

A more general expansion might contain higher powers and cross products of the readings. For simplicity assume that only the magnitude, M_1, and phase, Ph_1, at a single frequency are measured. We can easily generalize to more readings at more frequencies. For the kth pair of single-frequency readings, $M_1^{(k)}$ and $Ph_1^{(k)}$, we could determine the following polynomial approximation for property $P^{(k)}$:

$$P^{(k)} = \mathrm{PROP}^{(k)} - C_0 = \sum_{l=1}^{m} \left[C_l \left(M_1^{(k)}\right)^l + D_l \left(Ph_1^{(k)}\right)^l + \sum_{j=1}^{l-1} E_{jl} \left(M_1^{(k)}\right)^j \left(Ph_1^{(k)}\right)^{l-j} \right], \; k = 1, \ldots, K \; . \tag{6}$$

If the total number of coefficients, L, is less than or equal to the number of independent sets of measurements, K, the coefficients can be determined, and L can obviously be much greater than $N = 2$ readings. In general, the greater L the more accurately a given set of values of the property $P^{(k)}$ can be fitted. However, it is risky to use very high powers of the readings in the expansion because small errors in the readings may produce unreasonably large fluctuations in the computed value of P. A polynomial of high degree may fit a given set of property values exactly but give absurd values at intermediate points. Thus, the highest degree of any term in the polynomial expansion should not exceed the number of different property values to be fitted and, in fact, should generally be less than that number.

However, there is another method that can produce excellent fits with very little risk of errors caused by faulty readings. Suppose, for example, that a particular property varied exactly as the logarithm of the magnitude at the first frequency, call it $\log M_1$:

$$P \equiv \log M_1 \; . \tag{7}$$

Then a really good fit would require a very long sequence of powers of M_1, and the coefficients of all the other readings would be zero. However, if a power series in $\log M_1$ were used instead of one in M_1, a perfect fit would be obtained with only one term rather than a poorer fit with many powers of M_1. This rather extreme example shows the possibility of greatly improving accuracy and simplicity by using appropriate functions of the readings as the variables in a power series expansion. If only first powers of the functions (such as logarithm, exponential, reciprocal, and others) are required for the fit, the polynomial is said to be linearized; but this is not always possible for large ranges of the variables, at least not when only simple functions of the readings are used.

To be more definite assume that $F_i = F_i(R_i)$ is some function of the ith reading (R_i), a calculated function or one obtained from a look-up table. Assume also that a desired property can be expanded in a power series of various chosen functions of the readings:

$$\begin{aligned} \text{PROP} - C_0 = {} & C_{11}F_1 + C_{12}F_1^2 + \ldots + C_{1k}F_1^k \\ & + C_{21}F_2 + C_{22}F_2^2 + \ldots + C_{2l}F_2^l \\ & + D_{11}F_1F_2 + D_{12}F_1F_2^2 + D_{21}F_1^2F_2 + D_{22}F_1^2F_2^2 + \ldots \\ & + C_{31}F_3 + C_{32}F_3^2 + \ldots \,. \end{aligned} \tag{8}$$

Since the various powers of the functions F_1, F_2, ... are linearly independent, as many of the coefficients C_{ij} and D_{ij} can be determined as there are values of PROP $- C_0$ for which independent readings R_i have been made (and which determine the values of the functions F_i). If more sets of properties have been measured than the number of coefficients to be determined, then an optimizing process such as least squares must be used.

Once the best-fitting set of coefficients has been found for a given property, a number of other properties can be fitted in a similar manner. The total number that can be fitted is, in principle, limited only by the number of different sets of readings. In fact, if different functions of a given reading, such as log M_1, exp M_1, $1/M_1$, and others are allowed, even more properties might be determined. *However,* they could not be *uniquely* determined. Several combinations of various properties might produce the same set of readings, and one could not tell which combination had occurred. In fact, the expansions using those readings would lead one to believe that all such combinations had occurred. So the safe procedure is to measure at least as many magnitudes and phases (at various frequencies) on each sample as there are parameters that might vary in the test.

As a simple example, in a single-frequency measurement a change in the thickness of the sample can produce the same effect as a combination of lift-off and resistivity changes. With only two pieces of data available, one could not tell which case had actually occurred or if a combination of all three changes had occurred. In the general case, one cannot uniquely determine from a single set of readings more properties than there are pieces of information available (twice the number of frequencies used). Therefore, one must limit the variables in a given problem to not more than twice the number of frequencies (in a multifrequency experiment) or not more than the number of data channels in a pulsed-current or swept-frequency experiment.

The readings that determine the expansion coefficients taken on samples whose properties are known from other types of measurements are "calibration readings." Once coefficients fit the known properties of the calibration standards, they can be used to calculate the properties of unknown samples from measurements taken on them. One can also obtain sets of coefficients for the various properties entirely by use of the computer programs, which calculate expected readings for various combinations of the properties by simulating instrument response. However, these coefficients will usually be less accurate than those obtained empirically from the standards. This is because the actual coil and circuit parameters and the instrumental calibration will generally be slightly different from the values assumed in the computer simulation.

The principal limitations on the determination of multiple properties can be summarized as follows:

(1) The number of different properties that can be uniquely determined for a given sample is less than or equal to the number of readings taken on that sample (twice the number of frequencies used in a multifrequency measurement).

(2) The highest useful degree of any term in a polynomial expansion for a given property is less than the number of *different* values of the property to which the polynomial is fitted. Thus, if only two different values of a property are used to determine the polynomial, then making more than a linear interpolation is not justifiable.

(3) The total number of coefficients that can be determined for a given property expansion is less than or equal to the total number of values of that property fitted by the polynomial. This includes property values that are constant while other properties are varying. That is, if there are N_1 *different* values of the given property each of which is combined with N_2 different sets of values of the other properties, then a total of N_1N_2 equations can be written to determine the values of up to N_1N_2 expansion coefficients.

Properties of defects

For programming purposes defects are characterized by their (r,z) location, referenced to the coil and conductor positions, and their size parameter, referenced to a sphere having the same volume as the defect. Thus defects, like conductors, have three associated properties. If there is more than one defect in the conductor stack, positioned so that all defects of concern interact with the probe coils simultaneously, then the complete defect property set includes all the properties of all the defects.

To determine the effects of defects two sets of computations are performed — the first on the conductor stack without defects and the second with defects. The no-defect property sets have zero for the defect volume and nominal values for the defect location. In practical cases some defect parameters can be fixed, usually the r-coordinate and the size.

Instrument Design Theory

Eddy-current measurements are based on the electromagnetic induction effects between coils and systems of conductors. Induction coil impedance changes in predictable ways, which depend on the properties of conductors in close proximity. The methods developed by ORNL for solving eddy-curent induction problems have been extensively documented elsewhere;[1-6] hence, we shall include only the outline of the method, appropriate results, and references to earlier developments. We first consider the complete instrument circuit and calculate its response in terms of induction (probe) coil impedances. Then we examine the calculation of those impedances for reflection probes and transmission coils and the effect of conductor properties on the results.

Figures 1 and 2 show the coil and conductor arrangements for reflection and transmission testing methods. These have both similarities and differences of methods for obtaining the analytical and computational solutions to these problems.

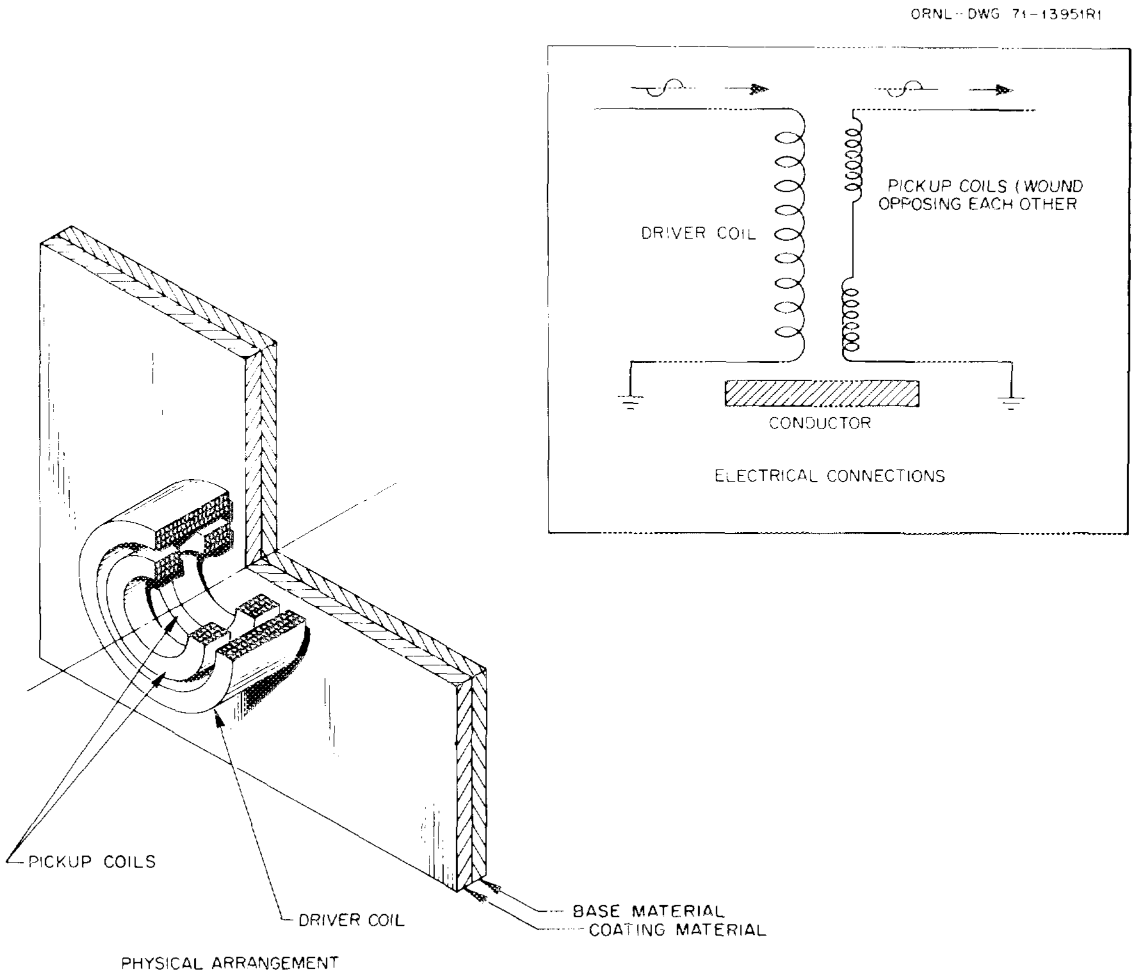


Fig. 1. Coil and Conductor Arrangements for Reflection Test Method.

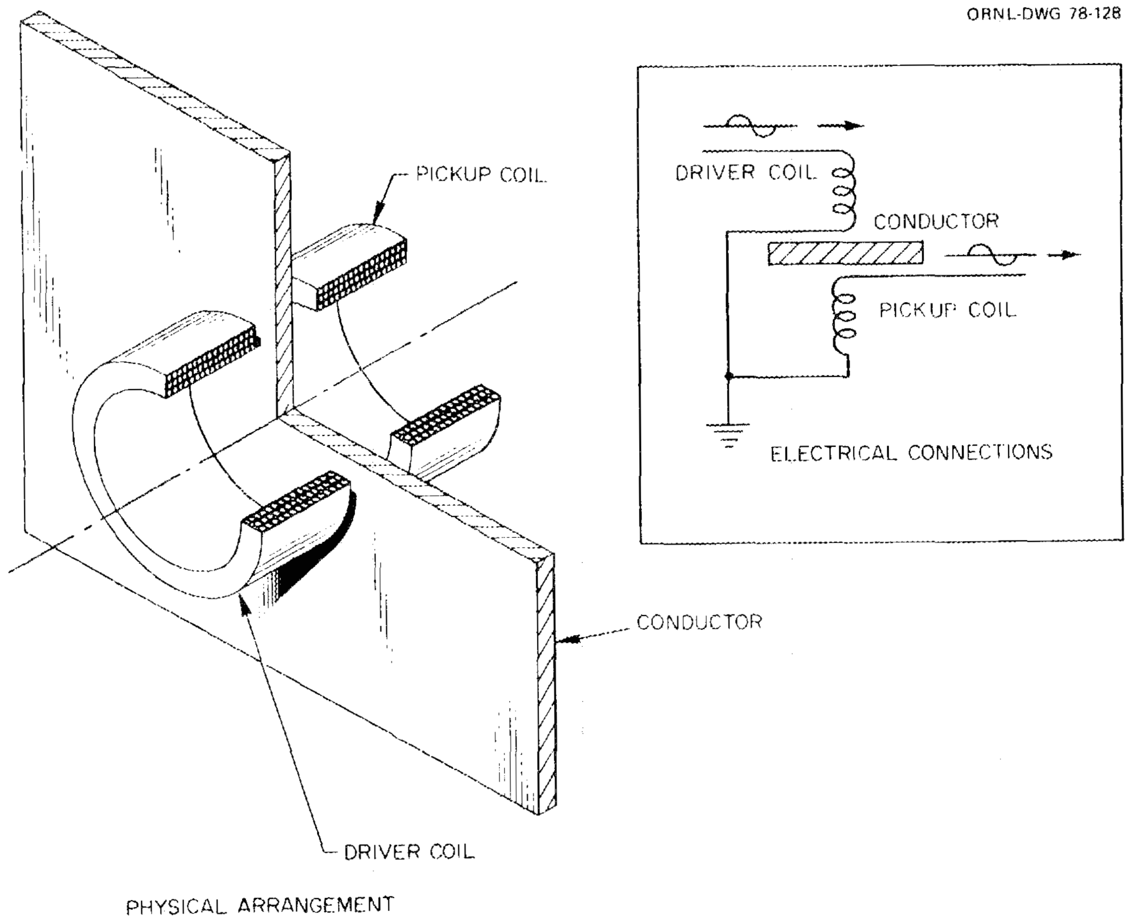
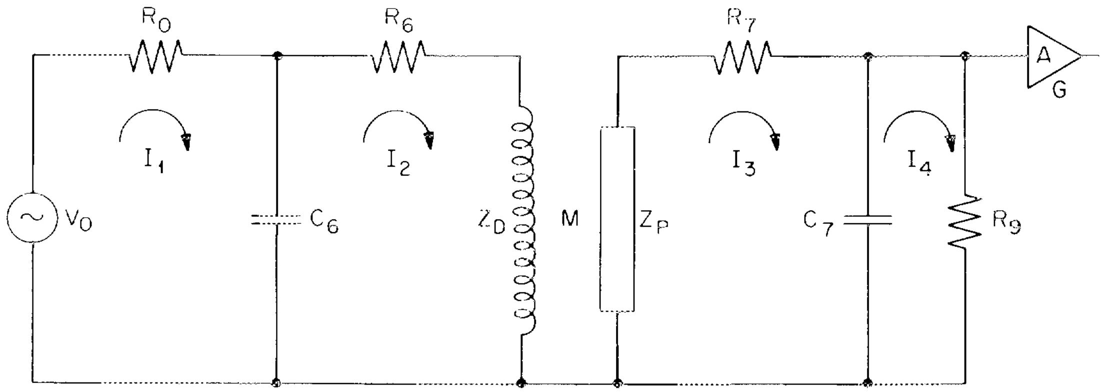


Fig. 2. Coil and Conductor Arrangements for Transmission Test Method.

Instrument circuits

Figure 3 shows the instrument circuit common to both reflection and transmission methods. The physical differences in the pickup coil arrangements (see Figs. 1 and 2) produce changes in M and Z_p and in the electrical response of the circuit. The opposed pickup coil windings in the reflection probe form a bridge circuit, which gives zero output voltage when the probe is removed from conductors. (During probe assembly the pickup coils are positioned symmetrically with respect to the driver coil to produce a null output.) The transmission pickup is not a bridge; it produces zero output only when the coils are separated by a large distance or when enough conductive material is interposed between the probes to reduce penetrating fields below a detectable amplitude.

ORNL-DWG 72-671R



V_0 DRIVING VOLTAGE
R_0 SERIES RESISTANCE IN THE DRIVING CIRCUIT
C_6 SHUNT CAPACITANCE OF THE DRIVING CIRCUIT
R_6 D.C. RESISTANCE OF THE DRIVER COIL
Z_D IMPEDANCE OF THE DRIVER COIL
M MUTUAL IMPEDANCE BETWEEN THE DRIVER AND PICK-UP COILS
Z_P IMPEDANCE OF THE PICK-UP COIL
R_7 D.C. RESISTANCE OF THE PICK-UP COIL
C_7 SHUNT CAPACITANCE OF THE PICK-UP CIRCUIT
R_9 AMPLIFIER INPUT IMPEDANCE
I LOOP CURRENT
G AMPLIFIER GAIN

Fig. 3. Instrument Circuit Common to Both Reflection and Transmission Test Methods.

To determine the circuit output, we first write four loop equations,[1] summing voltage drops around each loop:

$$I_1\left(R_0 - \frac{j}{\omega C_6}\right) - I_2\left(-\frac{j}{\omega C_6}\right) = V_0 \quad (9)$$

$$-I_1\left(-\frac{j}{\omega C_6}\right) + I_2\left(-\frac{j}{\omega C_6} + R_6 + Z_D\right) - I_3 j\omega M = 0 \quad (10)$$

$$-I_2 j\omega M + I_3\left(Z_P + R_7 - \frac{j}{\omega C_7}\right) - I_4\left(-\frac{j}{\omega C_7}\right) = 0 \quad (11)$$

$$-I_3\left(-\frac{j}{\omega C_7}\right) + I_4\left(-\frac{j}{\omega C_7} + R_9\right) = 0, \quad (12)$$

where j is the square root of -1 and ω is the angular operating frequency.

We use determinants to solve for the current in the final loop, I_4, produced by an applied voltage, V_0:

$$I_4 = \frac{\begin{vmatrix} R_0 - \dfrac{j}{\omega C_6} & \dfrac{j}{\omega C_6} & 0 & V_0 \\ \dfrac{j}{\omega C_6} & \dfrac{-j}{\omega C_6} + R_6 + Z_D & -j\omega M & 0 \\ 0 & -j\omega M & Z_P + R_7 - \dfrac{j}{\omega C_7} & 0 \\ 0 & 0 & \dfrac{j}{\omega C_7} & 0 \end{vmatrix}}{\begin{vmatrix} R_0 - \dfrac{j}{\omega C_6} & \dfrac{j}{\omega C_6} & 0 & 0 \\ \dfrac{j}{\omega C_6} & \dfrac{-j}{\omega C_6} + R_6 + Z_D & -j\omega M & 0 \\ 0 & -j\omega M & Z_P + R_7 - \dfrac{j}{\omega C_7} & \dfrac{j}{\omega C_7} \\ 0 & 0 & \dfrac{j}{\omega C_7} & \dfrac{-j}{\omega C_7} + R_9 \end{vmatrix}} . \tag{13}$$

We multiply I_4 by the resistance R_9 to determine the input voltage to the amplifier and then by the amplifier gain, G, to determine the voltage output:

$$V_{\text{out}} = -j\omega M V_0 R_9 G \div \Big\{(\omega C_6 R_0 - j)(\omega C_7 R_9 - j)\omega^2 M^2$$
$$+ \big[(\omega C_6 R_0 - j)(Z_D + R_6) - jR_0\big]\big[(\omega C_7 R_9 - j)(Z_P + R_7) - jR_9\big]\Big\} . \tag{14}$$

The remainder of the basic instrument includes circuits that measure both the amplitude of V_{out} and its phase relative to that of V_0. The resistances and capacitances are easily measured passive components of the circuit. The driver coil impedances, Z_D, the pickup coil impedance, Z_P, and the mutual inductance, M, are much more difficult to determine. Their computation requires most of the comupter-aided design effort and uses a large fraction of the computer time in the MULFRD and MULTRU programs.

Probe coil impedances

The coil impedances, Z_D and Z_P, and mutual inductance, M, are determined from exact solutions to the electromagnetic induction problem of planar conductors between or alongside cylindrical coils by the following steps:

1. Write the diffusion approximation to the Helmholtz equation for the electromagnetic vector potential, $\vec{A}$. Assume coaxial coils, parallel planar conductor boundaries normal to the coil axis, and linear isotropic homogeneous media. [Write the equation in cylindrical coordinates (see Figs. 4 and 5).] Assume the driving current to be sinusoidal and constrained to a single δ-function current loop.

2. Determine the general solution to the differential equation.

3. Apply boundary conditions for the desired conductor structure and determine the Green's function for the particular problem.

4. Build up the vector potential for a coil of finite cross-sectional area by integrating the Green's function over the coil region.[3]

5. Determine the voltage induced in a second coil by finding the integral

$$V = j\omega \int \vec{A} \cdot d\vec{s} \tag{15}$$

over the second coil[2] ($d\vec{s}$ is the differential element of length along a turn of the coil).

6. The mutual inductance between coils is given by[2]

$$M = \frac{V}{j\omega I} . \tag{16}$$

ORNL-DWG 72-10908

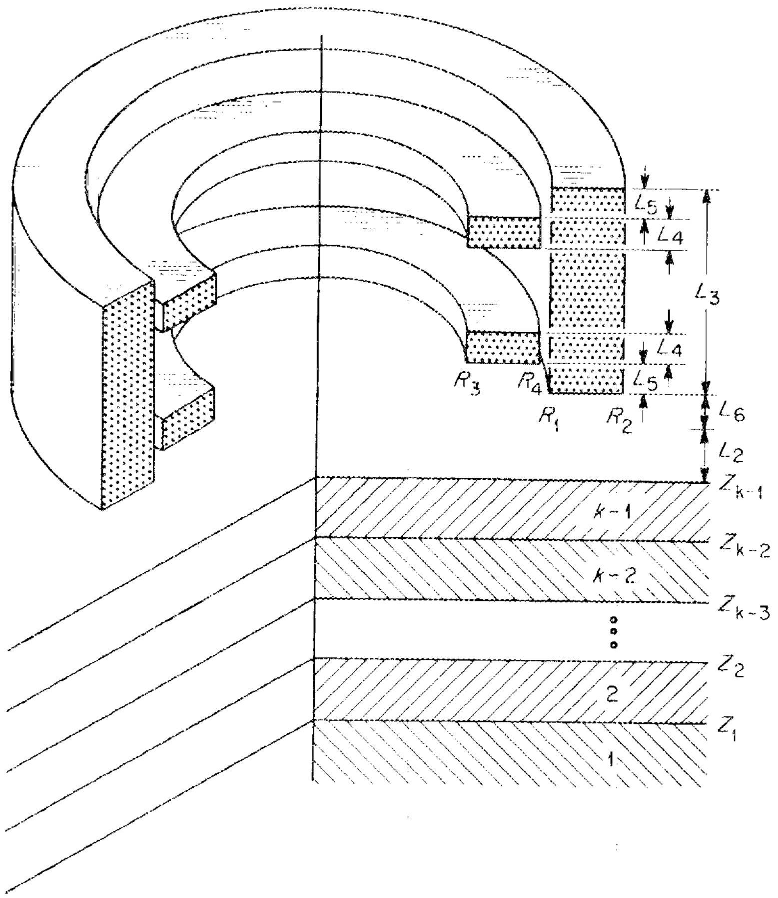


Fig. 4. Reflection Probe Above Multiple Conductors.

7. The separate coil impedances are computed by making each coil, in turn, the driver coil and by integrating the magnetic vector potential over the same coil.[2] Then

$$Z_{coil} = \frac{V}{I} . \tag{17}$$

The through-transmission case is completely solved by the sequential application of steps 1 through 7. Solution of the reflection coil problem requires some repetition of steps 4 through 7 to include the effects of the split pickup coil.[1] These calculations yield the results that follow.

There is no difference in the self-impedance of a driver coil between the reflection and transmission systems, so it is given by:

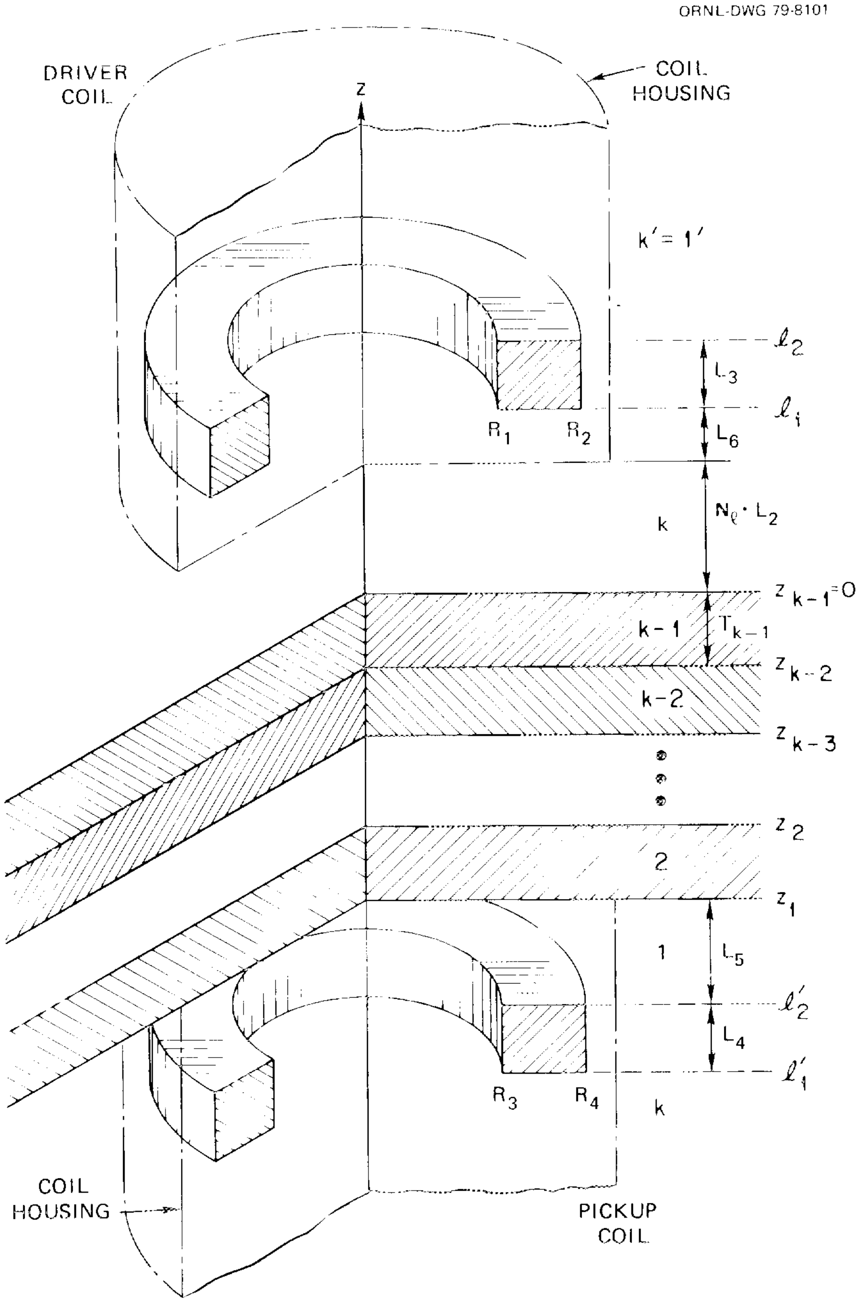


Fig. 5. Coil Arrangement, Conductor Structure, and Coordinate System for Through-Transmission Eddy-Current Instrument Simulations.

$$Z_D = \frac{j\omega\mu_0 \pi N_D^2 R_5}{L_3^2 (R_2 - R_1)^2} \int_0^\infty \frac{J^2(R_2,R_1)}{\alpha^6} \left\{ 2\left[\alpha L_3 + \exp(-\alpha L_3) - 1\right] \right.$$

$$\left. + \gamma_D \left[1 - \exp(-\alpha L_3)\right]^2 \exp\left[-2\alpha(L_6 + N_l \cdot L_2)\right] \right\} d\alpha \ . \qquad (18)$$

Mutual inductances for the two cases are given by (superscript T represents transmission case and R the reflection case in the following equations):

$$M^{(T)} = \frac{\mu_0 \pi N_D N_P R_5}{L_3(R_2 - R_1) \ L_4(R_4 - R_3)} \int_0^\infty \frac{J(R_4,R_3) \ J(R_2,R_1)}{\alpha^6} \left[1 - \exp(-\alpha L_3)\right]$$

$$\cdot \left[1 - \exp(-\alpha L_4)\right] \gamma_M \exp\left[-\alpha(L_5 + L_6 + N_l \cdot L_2)\right] d\alpha \qquad (19)$$

and

$$M^{(R)} = \frac{\mu_0 \pi N_D N_P R_5}{L_3(R_2 - R_1) \ L_4(R_4 - R_3)} \int_0^\infty \frac{J(R_4,R_3) \ J(R_2,R_1)}{\alpha^6} \left[1 - \exp(-\alpha L_3)\right]$$

$$\cdot \left[1 - \exp(-\alpha L_4)\right] \gamma_D \left\{1 - \exp\left[-\alpha(L_3 - L_4 - 2L_5)\right]\right\}$$

$$\cdot \exp\left[-\alpha(L_5 + 2L_6 + 2N_l \cdot L_2)\right] d\alpha \ . \qquad (20)$$

Self-impedances for the pickup coils are given by:

$$Z_P^{(T)} = \frac{j\omega\mu_0 \pi N_P^2 R_5}{L_4^2 (R_4 - R_3)^2} \int_0^\infty \frac{J^2(R_4,R_3)}{\alpha^6} \left\{ 2\left[\alpha L_4 + \exp(-\alpha L_4) - 1\right] \right.$$

$$\left. + \gamma_P \left[1 - \exp(-\alpha L_4)\right]^2 \exp(-2\alpha L_5) \right\} d\alpha \qquad (21)$$

and

$$Z_P^{(R)} = \frac{j\omega\mu_0 \pi N_P^2 R_5}{L_4^2 (R_4 - R_3)^2} \int_0^\infty \frac{J^2(R_4,R_3)}{\alpha^6} \Bigg[4\Big[\alpha L_4 + \exp(-\alpha L_4) - 1\Big]$$

$$+ \Big[1 - \exp(-\alpha L_4)\Big]^2 \Bigg(\gamma_D \Big\{1 - \exp\Big[-\alpha(L_3 - L_4 - 2L_5)\Big]\Big\}^2$$

$$\cdot \exp\Big[-2\alpha(L_5 + L_6 + N_l \cdot L_2)\Big] - 2\exp\Big[-\alpha(L_3 - 2L_4 - 2L_5)\Big]\Bigg)\Bigg] d\alpha \ . \qquad (22)$$

The function

$$J(R_b,R_a) = \int_{\alpha R_a}^{\alpha R_b} x\, J_1(x)\, dx \ , \qquad (23)$$

where J_1 is the first-order Bessel function of the first kind, and the remaining symbols are defined in Table 1 and in Figs. 4 and 5. In these equations all length dimensions (L's and R's) are normalized to the mean radius of the driver coil,

$$R_5 = \bar{r} = \tfrac{1}{2}(r_1 + r_2) \ , \qquad (24)$$

that is $L_3 = (l_2 - l_1)/\bar{r}$ and so forth.

The integration variable, α, is derived from the separation constant for the differential equation (step 1 above). The integrals include the required range of values of α for the complete solution.

The γ-factors (γ_D, γ_M, γ_P) incorporate the effects of conductors placed between the transmission coils or in front of the reflection probe. Their properties include:

1. γ_D and γ_P relate to fields that are effectively reflected from the conductor stack; hence, they differ for the through-transmission coils because they are computed for opposite sides of the stack. (Only γ_D appears in the reflection impedances because both coils are on the same side of the stack.)

Table 1. Symbols[a] for Eddy-Current Impedance and Inductance Equations

Parameter	Driver Coil	Pickup Coil
Inner Radius	R_1	R_3
Outer Radius	R_2	R_4
Mean Radius	R_5	
Coil Length	L_3	L_4
Number of Wire Turns on Coil	N_D	N_P
Fixed Spacing of Coil Insulation (T) or Offset Between Pickup and Driver (R)	L_6	L_5 L_5
Probe Lift-Off from Specimen Surface	$N_l L_2$ ($N_l = 0, 1, \ldots$)	
Boundary Value Coefficient for Self-Impedance	γ_D	γ_P (T)
Boundary Value Coefficient for Mutual Inductance	γ_M (T)	
Magnetic Permeability of Air	μ_0	

[a](T) — transmission case. (R) — reflection case.

2. When the coils are moved away from any conductors, γ_D and γ_P approach zero; the remaining parts of the impedance integrals are the values for the coils in air.

3. γ_M relates to fields that are transmitted through the conductor stack. Mathematically, it may be computed by starting from either side of the stack, but the result will be the same. Thus, it is a symmetric function of the stack variables. When the conductor stack is removed from between two coils, γ_M approaches $e^{-\alpha T}$, where T is the thickness of the removed stack.

All three γ-factors are complex when conductors are present and thus add imaginary terms to otherwise real integrals. The initially pure reactive inductances acquire a resistive component. The imaginary components of the γ-factors depend on the thickness, electrical resistivity (or conductivity), and magnetic permeability of each layer and the operating frequency of the driver coil.[1-3] This frequency dependence allows the determination of the conductor properties from a series of measurements at different frequencies.

For each unknown parameter one must measure at least one magnitude or phase of the output voltage, V_{out}, at some frequency. (Each frequency provides two independent measurements, one magnitude and one phase.)

The probe lift-off appears in all of the inductances and impedances $\left(\text{except } Z_P^{(T)}\right)$ in a way that allows it to affect both of the complex components. It may require treatment as an unknown unless it is carefully controlled, for example, by maintaining the probe(s) in intimate contact on the specimen.

Calculation of gamma factors

As noted above, the γ-factors contain the effects of conductor properties on the individual and mutual probe coil impedances. Efficient techniques to compute the γ-factors contribute significantly to the brevity of our programs. Their origin and structure has been discussed in detail in earlier work.[1,2] We summarize the results below for ready reference.

The general solution to the vector Helmholtz equation in cylindrical coordinates for the case of a single source consisting of a δ-function current ring (step 1 above) has the form

$$\int_0^\infty \left[B_n(\alpha)\ \exp(-\alpha_n z) + C_n(\alpha)\ \exp(\alpha_n z)\right] J_1(\alpha r)\ d\alpha \text{ in the } n\text{th layer,} \qquad (25)$$

where $\alpha_n^2 = \alpha^2 + j\omega\mu_n\sigma_n - \omega^2\mu_n\varepsilon_n$, and μ_n, σ_n, and ε_n are the permeability, conductivity, and permittivity of the nth region, respectively.

Boundary conditions between layers require matching values of $A(r,z)$ and $(1/\mu_n)(\partial A/\partial z)$ and yield boundary equation sets that typically look like

$$B_n \exp(-\alpha_n z_n) + C_n \exp(\alpha_n z_n) = B_{n+1} \exp(-\alpha_{n+1} z_n) + C_{n+1} \exp(\alpha_{n+1} z_n) , \quad (26)$$

and

$$\beta_n \left[B_n \exp(-\alpha_n z_n) - C_n \exp(\alpha_n z_n)\right]$$
$$= \beta_{n+1} \left[B_{n+1} \exp(-\alpha_{n+1} z_n) - C_{n+1} \exp(\alpha_{n+1} z_n)\right] , \quad (27)$$

where $\beta_n = \alpha_n/\mu_n$.

The source coil location, z', must be treated as a boundary, and the solution above the coil, in the k' region, must remain finite; so $C_{k'} = 0$ and the boundary equations become:

$$B_k \exp(-\alpha z') + C_k \exp(\alpha z') = B_{k'} \exp(-\alpha z') \quad (28)$$

and

$$\beta\left[B_k \exp(-\alpha z') - C_k \exp(\alpha z')\right] = \beta B_{k'} \exp(-\alpha z') - \alpha r' J_1(\alpha r') , \quad (29)$$

where the last term in Eq. (29) results from the source at the boundary.

At the lowest boundary, z_1, the allowed solution in the lowest region must remain finite for $z \to -\infty$, so $B_1 = 0$ and the boundary equations are

$$C_1 \exp(\alpha_1 z_1) = B_2 \exp(-\alpha_2 z_1) + C_2 \exp(\alpha_2 z_1) , \quad (30)$$

and

$$\beta_1 C_1 \exp(\alpha_1 z_1) = \beta_2 \left[-B_2 \exp(-\alpha_2 z_1) + C_2 \exp(\alpha_2 z_1)\right] . \quad (31)$$

A system of k layers (excluding k' from the count) has $2k$ boundary equations and $2k$ unknown coefficients. A matrix transformation technique[3] dramatically simplifies the calculation and allows selection of the required solutions.

Equations (26) and (27) can be written as

$$\begin{bmatrix} B_{n+1} \\ C_{n+1} \end{bmatrix} = \begin{bmatrix} T_{11}(n+1,n) & T_{12}(n+1,n) \\ T_{21}(n+1,n) & T_{22}(n+1,n) \end{bmatrix} \begin{bmatrix} B_n \\ C_n \end{bmatrix} , \tag{32}$$

where

$$T_{ij}(n+1,n) = (1/2\beta_{n+1})\left[\beta_{n+1} + (-1)^{i+j}\beta_n\right]\left\{\exp\left[(-1)^j \alpha_n z_n - (-1)^i \alpha_{n+1} z_n\right]\right\} . \tag{33}$$

The T-matrices can be combined to yield

$$\begin{bmatrix} B_k \\ C_k \end{bmatrix} = \begin{bmatrix} V_{11}(k,1) & V_{12}(k,1) \\ V_{21}(k,1) & V_{22}(k,1) \end{bmatrix} \begin{bmatrix} 0 \\ C_1 \end{bmatrix} , \tag{34}$$

where the V-matrix is the product of the T-matrices:

$$[V(k,1)] = [T(k,k-1)][T(k-1,k-2)] \ldots [T(2,1)] . \tag{35}$$

Note that only the second column elements of [V] are needed:

$$B_k = V_{12}(k,1)C_1 ; \tag{36}$$

$$C_k = V_{22}(k,1)C_1 . \tag{37}$$

Combining Eqs. (36) and (37) with (28) and (29) allows solution for $B_{k'}$ and C_1:

$$B_{k'} = \frac{V_{12}(k,1)\exp(-\alpha z') + V_{22}(k,1)\exp(\alpha z')}{2V_{22}(k,1)}\left(\frac{\alpha}{\beta}\right) r' \; J_1(\alpha r') ; \tag{38}$$

$$C_1 = \frac{\exp(-\alpha z_1)\alpha r' \; J_1(\alpha r')}{2V_{22}(k,1)\beta} . \tag{39}$$

From $B_{k'}$ and C_1 any B_n or C_n may be computed and the solution in any region found. When the Green's function solution is integrated over coil regions of finite size, the impedance equations listed in the second section above are obtained.

Three Green's functions are of interest for the reflection and transmission problems. For the region above the coil

$$G^{(k')}(r,z;\ r',z') = \int_0^\infty B_{k'}(\alpha)\ \exp(-\alpha z) J_1(\alpha r)\ d\alpha \ , \tag{40}$$

where $B_{k'}$ is given by Eq. (38). For the region between the coil and the conductor stack,

$$G^{(k)}(r,z;\ r',z') = \int_0^\infty \left[B_k(\alpha)\ \exp(-\alpha z) + C_k(\alpha)\ \exp(\alpha z)\right] J_1(\alpha r)\ d\alpha \ , \tag{41}$$

where

$$B_k = \frac{V_{12}(k,1)\ \exp(-\alpha z_1)\alpha r' J_1(\alpha r')}{2V_{22}(k,1)\beta} \ , \tag{42}$$

and

$$C_k = \frac{\exp(-\alpha z_1)\alpha r' J_1(\alpha r')}{2\beta} \ . \tag{43}$$

For the region below the conductor stack,

$$G^{(1)}(r,z;\ r',z') = \int_0^\infty C_1(\alpha)\ \exp(\alpha z) J_1(\alpha r)\ d\alpha \ , \tag{44}$$

where C_1 is given by Eq. (39).

For a driver coil of finite size the field within the range of coil height at a point z is found by integrating $G^{(k')}$ over that segment of the coil below z and adding the integral of $G^{(k)}$ over the segment of the coil above z.

From the equations for the driver side of the conductor stack we get

$$\gamma_D = \frac{V_{12}(k,1)}{V_{22}(k,1)} \; . \tag{45}$$

From the equation for the pickup side, we get

$$\gamma_M = \frac{1}{V_{22}(k,1)} \; . \tag{46}$$

When the driver coil is set up on the opposite side of the same conductor stack, we can invert the coordinate system, as shown in Fig. 6, to compute γ_P. We get

$$\gamma_P = \frac{V_{12}(k',1')}{V_{22}(k',1')} \; , \tag{47}$$

and also

$$\gamma_{M'} = \frac{1}{V_{22}(k',1')} \; . \tag{48}$$

From reciprocity we know that

$$\gamma_{M'} = \gamma_M, \text{ so } V_{22}(k',1') = V_{22}(k,1) \; . \tag{49}$$

Details of matrix element (V_{ij}) computation are included in the program discussions.

<u>Defects</u>

The effects of defects on instrument response can be incorporated directly into the expression for V_{out} [Eq. (14)] by adding impedance

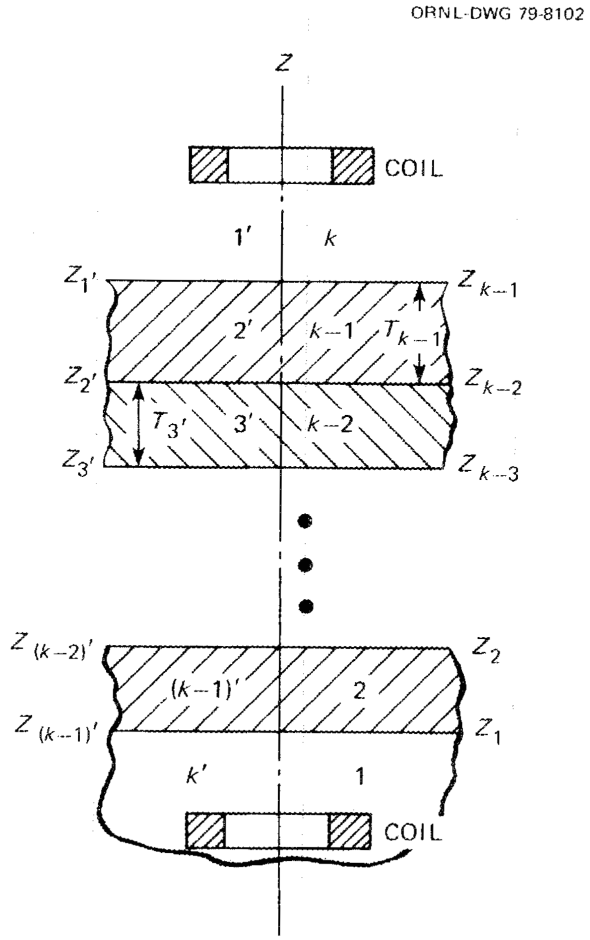


Fig. 6. Inverted Coordinate System Used for Calculating Pickup Coil Impedance for Through-Transmission Case.

changes to the existing coil impedances. For example, the driver coil impedance becomes

$$Z_D(\text{defect}) = Z_D + Z_{D'} \ , \tag{50}$$

where Z_D is given by Eq. (18) and $Z_{D'}$ is calculated below.

Small defects (relative to coil and layer thickness dimensions) can be represented as the sum of a current defect (discontinuity in resistivity) and a magnetic defect[4] (discontinuity in permeability). Our programs were initially designed for use with nonmagnetic materials, so the magnetic terms were not included. (Guidance for development and inclusion of these terms is available in refs. 5 and 6.)

A large class of defects is adequately approximated by a sphere, so we treat the case of a spherical defect located in the nth conducting layer ($1 \leqslant n \leqslant k-1$) detected by a reflection probe. (Only our reflection coil program contains the defect simulation.) The impedance changes for individual coils produced by a defect at (r,z) are given by

$$Z_c' = \frac{3}{2}\sigma\omega^2\left[\frac{A_c^{(n)}(r,z)}{I_c}\right]^2(\alpha_{22}V_f) \ , \tag{51}$$

where $A_c^{(n)}$ is the vector potential in the nth layer at (r,z), I_c is the coil current, V_f is defect ("flaw") volume, and α_{22} is a defect shape and orientation factor, which equals unity for a sphere. The mutual inductance change between driver and pickup coils is given by

$$M' = -\frac{3}{2}\,j\omega\sigma\left[\frac{A_D^{(n)}(r,z)}{I_D}\right]\left[\frac{A_P^{(n)}(r,z)}{I_P}\right]\alpha_{22}V_f \ , \tag{52}$$

where D and P refer to the driver and pickup, respectively. The vector potentials are given by

$$A_D^{(n)}(r,z) = \frac{\mu_0 I_D N_D}{2(R_2-R_1)L_3}\int_0^\infty \frac{J(R_2,R_1)J_1(\alpha r)}{\alpha}\exp\left[-\alpha(L_6+N_l\cdot L_2)\right]$$

$$\cdot\left[1-\exp(-\alpha L_3)\right]\left[\frac{V_{12}(n,1)\ \exp(-\alpha_n z)+V_{22}(n,1)\ \exp(\alpha_n z)}{V_{22}(k,1)}\right]d\alpha \ , \tag{53}$$

and

$$A_P^{(n)}(r,z) = \frac{\mu_0 I_P N_P}{2(R_4-R_3)L_4}\int_0^\infty \frac{J(R_4,R_3)J_1(\alpha r)}{\alpha^3}\exp\left[-\alpha(L_5+L_6+N_l\cdot L_2)\right]$$

$$\cdot\left\{1-\exp\left[-\alpha(L_3-L_4-2L_5)\right]\right\}\left[1-\exp(-\alpha L_4)\right]$$

$$\cdot\left[\frac{V_{12}(n,1)\ \exp(-\alpha_n z)+V_{22}(n,1)\ \exp(\alpha_n z)}{V_{22}(k,1)}\right]d\alpha \ . \tag{54}$$

Computation of the matrix elements $V_{ij}(n,1)$ is similar to that for $V_{ij}(k,1)$.

PROGRAM FEATURES, STRUCTURE, AND FUNCTIONS

Common Program Features

Most of our programming is written in Fortran IV and is compatible with the FR4 Compiler installed in our ModComp Max IV System Processor.[7] This compiler complies with standard Fortran[8] and includes the modifications and extensions listed in Appendix F. Ours are "main" programs and each includes a subroutine cluster.

The exceptions to Fortran include one subroutine, READAT, and the main program, RFLRDG, which both include short sections in ModComp IV Assembly Language. These sections exercise hardware control unique to the ModComp system.

All of the programming has been run and checked on a ModComp IV minicomputer system, which includes teletype and CRT interactive operator terminals, fixed and removable disc storage, line printers for hard-copy output, and a modular data acquisition system (MODACS).

We have attempted to use the same variables from one program to another. Generally, variables are changed when one that is single-valued in one program is converted to an array in another.

Modes of data insertion into the programs include:

1. editing before use to initialize properties, array sizes, and control parameters;
2. interactive operator communications during program execution through teletype or CRT terminals; and
3. direct reading of instrument response through the MODACS (described below).

The operating system (i.e., software package) installed in the ModComp allows several useful manipulations, which likely have equivalent counterparts in other systems:

1. Programs and subroutines may be stored on disc files in source or object language; source and object storage are in separate files.

2. Source files are directoried and may use the program name or an alias.

3. Random Access Data (RAD) files are accessible by the programs.

We typically store main programs in source language and subroutines used by more than one program in object language. (Subroutines unique to one program are often listed in the file as part of that program.) Since the programs must be edited to enter new initializing data, the edited version for a specific problem is refiled under an alias, leaving the "master" version intact. RAD files can also be changed to keep the data recorded for one problem separate from that for others.

Program Functions

The four main programs each perform one or more of the following functions:

1. instrument modeling or simulation for optimization of the instrument design,
2. acquisition and recording of instrument calibration and standardization data,
3. least squares fitting of instrument response to conductor properties, and
4. simulation of the analyses performed by on-board microprocessors in finished instruments.

Modeling includes the calculation of the circuit impedance for specific coil-conductor configurations and the output of the instrument circuit. Data is acquired through manual entry or from the instrument through the MODACS. Least squares fits are performed between calculated instrument responses and assumed (ideal) conductor properties or between actual instrument readings and actual measured properties. Simulated microprocessor action includes the storage of property-fitting equations and the calculation of conductor or defect properties from instrument readings with these stored equations. Table 2 summarizes the main program functions and their subroutines.

The initial design stage of a test development typically requires modeling of both a specific conductor structure and a coil configuration

Table 2. Program Functions and Subroutine Structures

Function	Subroutines Used By Programs[a]			
	MULRFD	MULTRU	RFLRDG	RFLFIT
1. Instrument Modeling			b	b

a. Impedance of Coil-Conductor Configuration	RFCOLM
	BESSEL
	GAMAML
	  BETAM
	  VINITM
	  VMATRM
	  CPXQOT
	BESEL1
		TRCOLM
		BESSEL
		GAMAMT
		  BETAMT
		  VMATRT
		
		  CPXQOT		

b. Circuit Output Voltage	RFCIRK	TRUCIR		
c. Circuit Component Drift	RFCKDT	b		
2. Data Acquisition and Recording	Direct Recording	b	READAT	READAT
3. Least Squares Fitting	POLTYP RDGEXP ALSQS CON8	POLTYP RDGEXP ALSQS CON8	b ALSQS	POLTYP RDGEXP CONVR8
4. Microprocessor Simulation	b	b	b	READAT RDGEXP
5. Miscellaneous (input/output, etc.)	COLFIL CHRFPM PRFCKT PRFCOL PRFVLT	TCOFIL CHRFPM PRFCKT PRFCOL PTRVLT		COLFIL PRFCKT PRFVLT

[a]Indentation indicates subroutine level (e.g., first-level routines are called from the main program).

[b]Not performed.

determined from the problem statement. The model incorporates anticipated variations in conductor and defect properties so that the variation in response of an ideal instrument with property variations can be studied. The instrument response is optimized by changing the probe coil characteristics (dimensions and electrical properties), circuit component values, and operating frequencies. All this modeling is done by computer.

From the computer-aided optimization studies, design parameters are selected and the actual instrument is built. Part of the optimization process is repeated, again assisted by the computer, but with several important differences. First, the instrument must be internally calibrated to assure that its outputs correctly indicate its actual response. Secondly, real conductors are applied to the real probe coil(s) to produce the instrument responses. The properties of these conductors are determined by independent measurements; thus they become calibration standards. Both the conductor properties and the instrument responses to them are entered into the computer programs, which then determine functional relations between the responses and the properties.

Multifrequency instruments are built with on-board microprocessors to analyze the input data and determine the properties of interest while discriminating against variations in the remaining properties of the conductor structure and any uncontrolled variations in the instrumentation system. The final step in adapting an instrument to a specific test is the simulation of microprocessor action by the minicomputer interfaced to the instrument.

Programs MULRFD and MULTRU perform the initial modeling for design optimization of reflection and through-transmission instrument types, respectively. The MULRFD is more advanced with its capability to store initial optimization data directly on RAD files. This storage allows other programs to access the file and simulate microprocessor performance of an ideal instrument.

Program RFLRDG interfaces the instrument to the minicomputer for recording instrument responses to the standard conductors and allows manual entry of the standard conductor properties. It also allows entry and recording of some internal calibration data for storage and later reference.

The RFLFIT reads stored data from the files filled by RFLRDG and performs the least squares fit between the properties of the standards and the instrument readings. It stores the fit coefficients in other RAD files for its own use and in forms suitable for entry into the instrument microprocessors. It also has MODACS control for reading the instrument response to unknown specimens from which it can compute their properties by using the stored functions.

Property Arrays and Critical Array Dimensions

We define a "conductor property set" for a specified conductor stack as a set that includes one value of each property for each conductor in the stack. If there are no defects each conductor has electrical resistivity, relative magnetic permeability, and thickness. If a conductor contains a defect the size and two location parameters of the defect are treated as additional properties (see p. 35 for author definitions of defect size and location parameters.) Defects that are located simultaneously within the sensitive region around the coil have a "defect property set" analogous to the conductor property set.

In design computations the effect of variation in a particular conductor property is determined by inserting a series of different values and observing the change in simulated instrument response. Two or more properties may be varied simultaneously, but when this is done a response must be computed for each possible unique combination of properties. All these possible responses enter the performed computations to maximize the discriminatory capability of the system.

The number of unique property sets is simply the product of the number of values for each property (sometimes called "steps" or "step-values") taken over all the properties in the specified stack. For example, if a 4-conductor stack has 2 varying properties, such as thickness and resistivity in a single layer, and 4 steps are inserted for each of these, then 16 unique property sets can be constructed and must be considered in the computations. If a defect were added in any layer, the numbers of steps of defect size and/or locations would be multiplied by 16 to determine the number of unique property sets.

In design studies the response of conductors with defects is compared to the response without defects, so calculations are done for both. Thus, the total number of calculations and hence the sizes of certain arrays depend on the number of unique property sets without defects plus the number of unique sets possible when defect properties are included. Following the example above the total number of calculations would be 16 plus 16 times the defect property step factors.

Lift-off is not a conductor property, but since its variation also affects instrument response and must often be discriminated out, provision is made for including it in the optimization process. It has the effect of a single property since it occurs at one point in the system. When variations in lift-off are included, the number of variation steps, NLT, is specified and is multiplied by the number of unique property sets to determine the sizes of critical arrays, such as READNG.

To achieve flexibility in the programs conductor properties are distinguished from defect properties and stored in separate arrays. Two array types are used for conductor properties. The first type, which is initialized, is arranged as though the conductor stack were free of defects. The second type is constructed during execution and contains extra entries (or "copies") of specific property sets that must be combined with defect properties for a particular calculation.

The READNG array is sized as follows:

1. Determine the number of conductor properties, 3*(NRT-1), [or 3*(NRT-2) if lowest region is air].

2. Determine the number of unique conductor property sets. NPT equals the product of the number of property value steps for each property taken over all the conductor properties (some factors may equal 1).

3. Determine the number of defect properties, 3*(number of defects).

4. Determine the number of unique defect property sets by multiplying the number of value steps for each property over all the properties (some factors may equal 1). As MULRFD is now written this product is NDFLOC*NDFSIZ.

5. Determine the number of unique combined property sets (conductors plus defects — the result of step 2 times the result of step 4). In MULFRD the number of combined sets is NODF = NPT*NDPS.

6. Determine the total number of property sets used in calculations, the number of conductor sets, and the number of combined sets. In MULRFD, NPTT = NODF + NPT.

7. When lift-off is varied it must be treated as a property for the conductor and combined cases, so NPTT*NLT computations are performed. READNG requires one more row to accommodate the values of properties computed by the function expansion, so its row dimension is NPTT*NLT + 1.

8. The column dimension, IRDPRM + 1, of READNG accommodates IRDPRM expansion terms plus the property values for which the fit is calculated.

Defects and Defect Parameters

Defects are described within the programs by specifying the following information:

1. conducting layer in which contained (identified by region number, NRDF),
2. position within the layer (the z-axis distance from the driver coil side of the layer),
3. location relative to the coil axis (radial distance), and
4. size (explained more fully below).

When treating the problem of a single defect within the entire conductor stack, the radial location parameter is typically fixed at that radius where its influence is maximum, that is, where the coil is most sensitive to it. Layer position, DFLOC (z-distance), is measured to the center of the sphere representing the defect. For many applications the defect is assumed to be at the midplane of its layer. The normalized defect size, DFSIZE, is calculated from DFDIAM by calculating the volume V of the actual defect (shown in Fig. 7), finding the radius of the sphere with the same volume $[r = (3V/4\pi)^{1/3}]$, and then calculating the normalized volume of that sphere:

$$\text{DFSIZE} = (4\pi/3)(r/\text{RBAR})^3 \quad . \tag{55}$$

As long as the defect dimensions are small compared to those of the conductor, the calculated output voltage change caused by a defect is directly proportional to the size, shape, and orientation factor of the defect

ORNL-DWG 79-8103

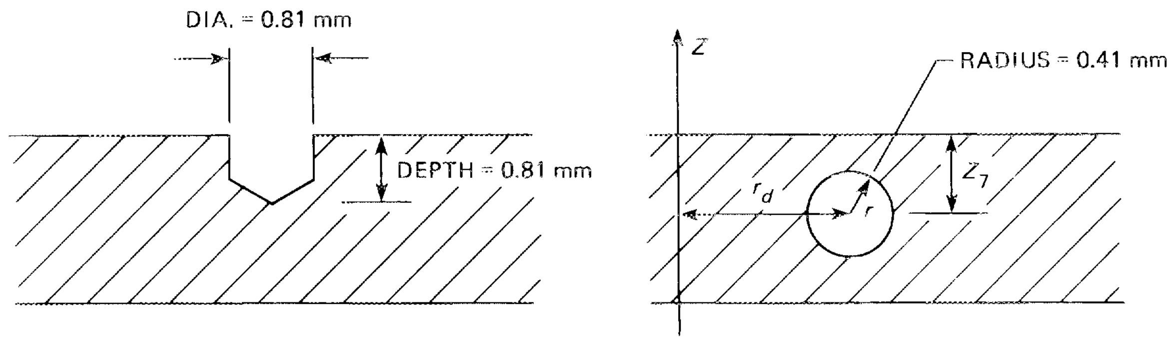


Fig. 7. Cross Section of Defects Assumed in Sample Runs. (a) Defect on top or bottom. (b) Defect in middle.

called α_{22} [Eqs. (51) and (52)]. Therefore, calculating a number of different defect volumes at a given location in a given conductor is unnecessary. Rather, it is sufficient to calculate the output with and without a defect of a given volume; then other volumes can be interpolated or extrapolated from the results with the known volume (and zero volume). From experience this hypothesis proves quite good, even for fairly large defects (compared to the conductor thickness).

For this reason the MULRFD program assumes only a single defect size but allows several possible locations. If desired the program can be edited to cover a range of defect volumes at each of the defect locations to verify the linearity of fitting the defect volume property.

Least Squares Fitting of the Properties

As described in the section on "Theory" under "Properties of Conductors" (p. 4), a polynomial expansion [see Eq. (8)] is assumed for the property in terms of various assumed functions f_i of the observed or calculated magnitudes and phases. This is done for a sufficient number of properties (with their corresponding magnitudes and phases) to overdetermine the system of equations [see Eq. (6)] for the coefficients in the expansion. Since the coefficients are only to the first power, we use a linear least squares fitting to minimize the sum of squares of the differences [see Eq. (4)]. The procedure is as follows.

A matrix array, PROP(M,N), is formed from the values of the properties to be fitted (N is the index of the various properties, and M is the index of the various values of that property). If we select a particular property to be fitted with index N_i, then we can henceforth consider only the column vector Y(M) = PROP(M,N_i), where M=1, ..., MSET. Term MSET is the number of sets of independent measurements that have been made of the various properties. The column vector Y(M) is to be fitted with a set of coefficients, C(I) = COEF(I,N_i), that produce the best (least squares) fit by using the various measured magnitudes and phases (or readings) raised to various powers.

The programs offer the user a wide choice of complex functions to fit conductor or defect properties to instrument readings. (The fitting process is described earlier, pp 4–13.) For each frequency the programs construct sequences of the form

$$\sum_{i=1}^{a} A_i [f(M)]^i + \sum_{i=1}^{b} B_i [g(P)]^i + \sum_{i=1}^{c} C_i [f(M)]^i [g(P)]^{c+1-i} , \qquad (56)$$

where M and P represent output voltage (V_{out}) magnitude and phase, respectively, at that frequency. The choices for f and g are: linear, $f = x$; logarithm, $f = \ln(x)$; exponential (positive), $f = e^x$; and inverse, $f = 1/x$. However, $g(P)$ cannot be $\ln(P)$ if $P < 0$ without causing computer error response and program interruption.

In the program selection sequences f and g are selected independently as "FCN TYP," a and b as "POL DEG," and c as "# CROSS TERMS" (see Tables A3, B2, D2) by frequency. The selections for each frequency and an arbitrary leading constant are combined to form the complete function fitted to the selected property. This information is stored in an array called NPOL. The NPOL array has six rows and NFT columns. In each column the row numbers describe the terms from that frequency (NF) that appear in the complete property function. Table 3 lists the row parameter definitions. Then the RDGEXP subroutine is called to construct the matrix array of the expanded readings. The array is called READNG and would normally

Table 3. Function-Descriptive Row Parameters in the NPOL Array

Row I	Parameter Definitions, NPOL(I,NF)
1	Function type for output voltage magnitude conversion[a]
2	Maximum degree of polynomial in the magnitude function
3	Zero[b]
4	Function type for output voltage phase conversion[c]
5	Maximum degree of polynomial in the phase function
6	Number of cross terms between the magnitude and phase function

[a]1=linear; 2=logarithm; 3=exponential; 4=inverse.

[b]If value ≠ 0 is inserted, an error message results.

[c]Logarithm (2) is not allowed, since phase may be negative.

have MSET rows and IRDPR columns, where IRDPR = number of terms in the expansion for each property value:

$$\text{IRDPR} = \sum_{\text{NF}=1}^{\text{NFT}} [\text{NPOL(2,NF)} + \text{NPOL(5,NF)} + \text{NPOL(6,NF)}] \ .$$

Actually, because of the least squares fitting process, an extra row and column are added to the READNG array to allow space for computational storage. So the dimensions of the array are READNG(MSET+1, IRDPR+1). The calculated values of the property are obtained by multiplying the coefficient vector C(I) by the READNG matrix:

$$\text{CALC(M)} = \text{READNG(M,I)*C(I)} \ .$$

In the programs we compute

$$\text{SUM} = \sum_{\text{I}=1}^{\text{IRDPR}} \text{COEF(I)*READNG(M,I)} \ .$$

To get the minimum squared deviations of the calculated property values from the actual values, we minimize the quantity

$$||(\mathrm{READNG})*(\mathrm{C}) - (\mathrm{Y})||^2 . \tag{57}$$

The procedure uses the Householder reduction of the READNG matrix to triangular form (given by Golub[9]) and is accomplished by the ALSQS subroutine. Upon return from the subroutine the last (IRDPR+1) column of the READNG matrix contains the values of the property calculated by using the least squares coefficients. The maximum allowable row size of READNG is (IRDPRM+1) terms, and IRDPRM is fixed at the start of the program. The READNG matrix is reduced to (IRDPR+1) columns when the RDGEXP subroutine is called. In addition, two quantities, B and R2, are returned to the main program. Term B is a column vector containing the linear least squares coefficients, and R2 is the residual sum of squares of the differences. If the assumed polynomial expansion were exactly the correct functional form for the property, R2 would be the variance, and its square root would be the standard deviation, σ, of the fit. The square root of R2 is called the standard deviation in the printout, but the user should understand that it is only a good approximation to σ when its value is small, meaning that the polynomial is a good fit.

Tests of Fit Quality

Two properties of the fit are calculated: the rms difference between the actual property values and the values computed from the fit and the effect of small errors in the instrument readings on the values computed from the fit function. For the mth property set

$$\mathrm{DIFF}_m = \mathrm{PROP}_m - \sum_{j=1}^{J} C_j f_j(R_{mj}) \ , \tag{58}$$

where

J = IRDPR, the total number of terms in the expansion;

C_j = least squares coefficient for the jth expression;

f_j = a selected function, such as f or g above, raised to the appropriate power, and

R_{mj} = the instrument reading (actual or calculated) corresponding to the jth term in the expansion.

The sum is returned as the array element READNG(M,IRDPR1) from the ALSQS subroutine. Then

$$\mathrm{SSDIFF} = \sum_{m=1}^{M} (\mathrm{DIFF}_m)^2 \ , \tag{59}$$

where M = MSET, the total number of property sets used in the program including conductors, conductors plus defects (if defects are present), and lift-off values.

Finally,

$$\mathrm{SDIFF} = (\mathrm{SSDIFF}/M)^{\frac{1}{2}} \ . \tag{60}$$

To see how sensitive the calculated values are to slight measurement errors, a change of 0.01% in a magnitude reading or 0.01° in a phase reading is made, the values of the properties are recalculated by using one erroneous reading, and the residual sum of squares is obtained.

The term SUM, the property calculated by using the erroneous magnitude or phase and the absolute value of the difference between it and the value calculated by using the "correct" values (which are in the IRDPR+1 column of the READNG matrix returned from ALSQS), is added to the DRIFT. When the absolute values of the differences are added this way, DRIFT accumulates the largest possible error in the calculated property that can be caused by 0.01% errors in each magnitude and 0.0° errors in each phase, assuming that the magnitude and phase errors are all independent and small enough to produce nearly linear influences on the property. The sum of the squares of the drifts is accumulated as

$$\text{SSDRIF} = \sum_{m=1}^{M} (\text{DRIFT}_m)^2 \, , \qquad (61)$$

and finally,

$$\text{SDRIF} = (\text{SSDRIF}/M)^{\frac{1}{2}} \, . \qquad (62)$$

Internal and External Instrument Calibration

The use of programs RFLRDG and RFLFIT is better understood if preceded by a review of the calibration process that they assist. The process includes two basic steps: internal calibration of the phase shift measuring circuits and external standardization of the instrument response against specific test variables.

Internal calibration

Internal calibration assures that phase shift output of the instrument is linear and accurately represents the actual phase difference between the pickup coil output and the input driver coil voltages. It is accomplished by substituting (switching) a known and fixed network (Fig. 8) in place of the instrument probe(s). (The calibration impedance replaces the probe impedances shown in Fig. 3).

Typically, three phase values are used to verify linearity. (Three values for R may be switched in.) Each phase value is checked at two magnitude settings, which represent the upper and lower magnitude values expected in the test, to assure that magnitude changes do not affect the phase calibration.

Phase calibrators are unique to a given problem. They are built to provide specific phase shifts at specific frequencies. A multifrequency instrument will contain one such network for each frequency used.

ORNL-DWG 79-8104

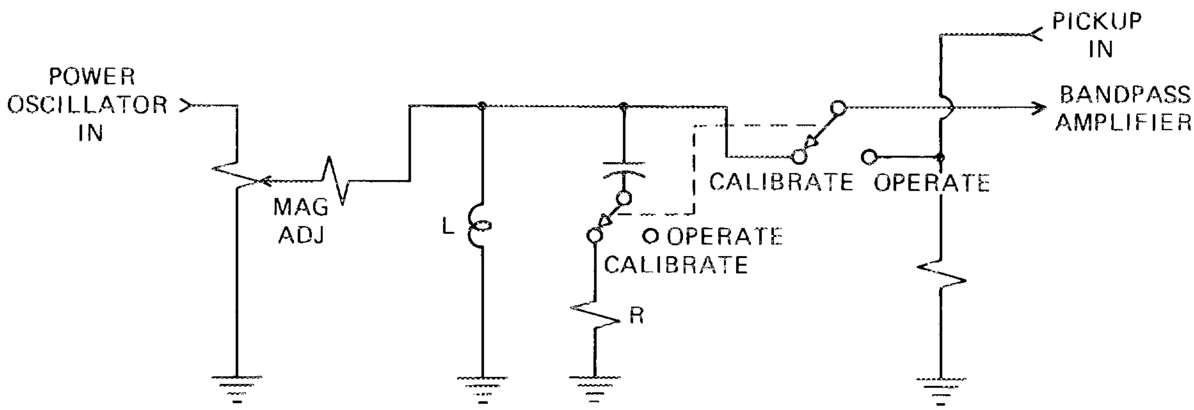


Fig. 8. Diagram of Internal Calibration Method.

External standardization

External standardization sets up the on-board microprocessor in the instrument to correctly process instrument readings and determine other properties, such as conductor thickness, resistivity, and others. It is accomplished in several steps:

1. Record the instrument responses to a number of standard specimens whose properties have been measured by another independent method.

2. Record the independently measured specimen properties. For steps 1 and 2 note that a test plate having areas with and without defects represents two (or more) specimens.

3. Perform a least squares fit of the instrument readings to the specimen properties.

4. Store the appropriate fitting equations for the desired properties in the on-board microprocessor.

A possible intermediate step between steps 3 and 4 is the simulation of the on-board microprocessor by an external minicomputer to verify the accuracy of the fitting process.

An alternate method, often useful in design studies, is fitting an idealized property set to a set of instrument readings computed from that set with use of idealized model (MULRFD and MULTRU are examples). The previous method is more accurate for field use because it uses actual properties of real materials and includes effects (such as minor coil property variations) that may not have been accurately determined and introduced into the idealized computation.

Calibration steps

The following is a general outline of the complete calibration, using RFLRDG and RFLFIT:

1. Apply the probe to each of the standards in the set. Determine the minimum and maximum magnitudes and the range of phase shifts.

2. For phase shifts exceeding 180°, switch in the inverting stages in the bandpass amplifiers to bring the phase shifts back within range of the indicating circuits.

3. Set amplifier gains to keep their outputs no more than 10 V over the range of specimen magnitude readings. (Amplifiers should not be allowed to overdrive or saturate.)

4. Check phase outputs that are using the phase calibrators.

5. Record the independently measured properties of the standard specimens.

6. Record instrument readings from each specimen.

7. Perform least squares fits between readings and properties.

8. Transfer fit equations to on-board microprocessor. (This step requires one additional program, not reported here.)

Applications to through-transmission

The RFLRDG and RFLFIT programs were designed and verified for non-magnetic materials and reflection coils. They and the procedures would be useful for through-transmission with the following modifications:

1. Add storage for separate pickup coil identification data.

2. Check phase shifts for conductors against appropriately sized air gaps to determine phase switch positions.

3. Center or otherwise adjust phase shifts no more than 180° within the indicating range of the instrument by using the balance control on the discriminator modules.

4. Determine the magnitude range by inserting and removing conductors (specimens). The minimum magnitude should correspond to the air gap necessary to accommodate the thickest specimen. The maximum magnitude will correspond (roughly) to the thickest or highest conductivity specimen.

Data acquisition system

The minicomputer analog interface for which this programming was used is based on the Digital I/O (DIO) module of the ModComp MODACS (modular data acquisition system). A locally built 14-bit analog-digital converter (ADC) is controlled by the MODACS DIO, and the programming, in turn, controls the DIO.

The ADC has 12 analog input channels. One channel is assigned to the analog voltage corresponding to each measured magnitude and phase (i.e., 2 channels per test frequency), 1 channel per frequency reads the power oscillator output (V_0) for that frequency, and channel 12 is reserved for control of the reading process. A foot pedal switch grounds channel 12 to initiate the reading cycle (using subroutine READAT). The programming tests channel 12 and limits the execution of READAT to one reading cycle for each depression of the foot pedal.

One reading cycle reads each channel 512 times and sums the results as "counts" for the ADC. READAT returns these counts to the controlling program in an array, from which they are converted to voltages. (The factor 819.2 in both programs is the ADC counts per volt for the particular instrument in use; the 14th bit of the ADC is used for the sign.)

ACKNOWLEDGMENTS

Preparation of a complex report of this size for publication requires the cooperative efforts of several people. W. H. Butler and B. S. Borie reviewed the report. The very comprehensive editing was done by B. G. Ashdown. S. G. Frykman prepared the reproducible copy.

REFERENCES

1. C. V. Dodd, C. C. Cheng, W. A. Simpson, D. A. Deeds, and J. H. Smith, *The Analysis of Reflection Type Coils for Eddy-Current Testing*, ORNL/TM-4107 (April 1973).
2. C. V. Dodd and W. E. Deeds, "Analytical Solutions to Eddy-Current Probe Coil Problems," *J. Appl. Phys.* 39(6): 2829–38 (1968).

3. C. C. Cheng, C. V. Dodd, and W. E. Deeds, "General Analysis of Probe Coils Near Stratified Conductors," *Int. J. Nondestr. Test.* 3(2): 109–30 (1971).

4. M. L. Burrows, *A Theory of Eddy Current Flaw Detection,* University Microfilms, Ann Arbor, Mich., 1964.

5. C. V. Dodd and W. E. Deeds, *Calculations of Magnetic Fields from Time-Varying Currents in the Presence of Conductors,* ORNL/TM-4958 (July 1975).

6. C. V. Dodd, W. E. Deeds, J. W. Luquire, and W. G. Spoeri, *Some Eddy-Current Problems and Their Integral Solutions,* ORNL-4384 (April 1969).

7. Modular Computer Systems, Inc., Ft. Lauderdale, Fla., *MAX IV System Processors FORTRAN IV, Reference Manual,* Publication 210-610500-010B00, 1976.

8. *ANSI X3.9,* American National Standards Institute, New York, 1966.

9. C. Golub, "Number Methods for Solving Linear Least Squares Problems," *Numer. Math.* 7: 202–16 (1965).

APPENDIX A

PROGRAM MULRFD

The functions of this program are described earlier. Table A1 lists the editorial changes required before running the program. Note that, in addition to the table requirements, a data file must be separately established for use by the program. If file storage of coil data is not available, subroutine COLFIL (Appendix E, p. 160) can be modified to enter coil data normally. Table A2 lists additional definitions of program parameters to assist the user in setting up a calculation. Table A3 lists step-by-step instructions for program execution. The final section contains a detailed analysis of program construction and operation.

Table A1. Summary of Editorial Changes Preparatory to Running MULRFD

Line(s)	Actions
59–73	Adjust array dimensions; follow key provided in lines 42–56 (see pp. 34–35 for discussion of READNG).
89	Adjust control parameters (NRT, NPT, NLT, NFT, NDFLOC, NDFSIZ); select initial identifier (PROBE).
90	Select print option (NPRINT), output device (LOU), input device (LI). (NPRINT = 0, no initial property or circuit tables.)
91–93	Enter defect parameter values (free format) NRDF(J), J = 1,...NPT; DFDP(J), J = 1,...NDFLOC. DFRAD(J), J = 1,...NPTT; DFDIAM = single value.
94–105	Enter property values (free format) RES(NR,NP) PERM(NR,NP), THICK(NR,NP); NR = 1, NRT, NP = 1, NPT; NR cycles first. [Order of values is (1,1), (2,1),...(NRT,1), (1,2),...(NRT,NPT).]
110	Enter frequency values (free format) FREQ(NF), NF = 1, NFT.
111	Enter gain values (free format) in same order as frequencies. GAIN(NF), NF = 1, NFT.
117	Enter normalized lift-off increment (free format) L2 = $l_2/\bar{r}$
145–149	Enter initial set of circuit parameters (free format). Units as shown in comments lines (139–143).

Table A2. Expanded Definitions of Program Control Parameters and Defect Properties

Control Parameter	Definition
DFDP	Initializing data; mean depth of center of a surface defect (in.) measured from the surface[a] of the layer in which the defect is embedded. (This number is not used for computation.)
DFLOC	Vertical distance of defect[b] below top[a] of layer (in.).
DFRAD	Radial distance of defect (center of sphere) from driver-coil axis (in.); fixed at 0.0832 R_5, unless the problem requires consideration of multiple defects with offsets in radial distance.
DFDIAM	Diameter (in.) of sphere with volume equal to that of actual defect, $\left[\frac{6(\text{defect volume})}{\pi}\right]^{1/3}$.
DFVOL	Normalized defect volume, $\frac{\pi}{6}\left(\frac{\text{DFDIAM}}{R_5}\right)^3$.
IRDPRM	Maximum number of coefficients allowed in the least squares fitting expansions; typically set at 15
LI	Logical input unit; transmits operator commands to computer.
LOU	Logical output unit; supplies cue statements and printed output to operator.
NCOIL	Sequential position index of coil data in File 28. (Local compiler requirement for direct access file statements.)
NDFLOC	Initialized data; number of step values for defect location, DFLOC.
NDFSIZ	Initialized data; number of step values for defect size, entered as DFDIAM; normally set at 1. Can be set greater than 1 to check linearity, but further editing would be required.
NDPS	NDFLOC*NDFSIZ (line 113). Product equals the number of unique defect property sets in a specimen as MULFRD is now written.

Table A2. (Continued)

Control Parameter	Definition
NFT	Number of discrete frequencies for which computations are performed.
NLT	Number of values of driver-coil lift-off for which computations are performed.
NODF	The number of unique property sets obtained by combining defect properties and conductor properties and accounting for all step variations ("combined" property sets).
NPRINT	Print and transfer index; initial value selects printing of tabular output. Values selected during execution determine selected program options.
NPT	The number of unique conductor property sets; includes the number of conductor properties and the steps of variation in each property.
NPTT	The number of combined property sets, including defects as distinct property sets. (The total number of calculations required by conductor and defect properties.)
NRT	Number of regions in the problem, including the air region containing the coil.
NUNIT	Unit control. Initial value selects English or metric output length dimensions. (1 = English, 2 = metric.)

[a]"Top" is the driver-coil side of the layer.

[b]Coordinate locations of normalized defect refer to center of sphere representing the defect.

Table A3. Interactive Program Queries and Operator Responses Occurring During Execution of MULRFD

Line	QUERY or Response
136	From subroutine COLFIL: TYPE COIL NAME (UP TO 6 LETTERS): Enter six-character alphanumeric; may contain leading or trailing blanks (spaces). COIL (Name) NOT FOUND. TYPE FOLLOWING COIL PARAMETERS: RBAR, R1, R2, L3, R3, R4, L4, L5, L6, RDCDR, RDCPU, TNDR, TNPU: Enter free-format number for each variable.
454	TYPE NUMBER OF NEXT ACTION TO BE TAKEN: 1 = PRT PROPS; 2 = FIT PROP; 3 = PRT ENTIRE FIT; 4 = PRT COEF; 5 = CHG FCTN/POL TYP 6 = CHG COIL/CKT VAL; 7 = CHG W PRT; 8 = CAL COIL/CKT DRIFT; 9 = STOP Enter integer 1–9.

The following sections list ensuing interactions resulting from each of the option selections at line 454:

Line	QUERY or Response
273	1. Print properties — No operator input required. (See p. 68 for sample output.)
297	2. Fit Property — Will produce following list of options:
319	SELECT NUMBER OF PROPERTY TO BE FITTED: 1 = RHO 2 = MU 3 = THICKNESS 4 = LIFT-OFF 5 = DEFECT DEPTH 6 = DEFECT SIZE
320	SELECT REGION TO BE FITTED. LOWEST IS NUMBER 1. Enter integer 1–6 (property), delimiting space, integer 1-NRT (region). (The remainder of interactions occurring with this option duplicate those occurring with option 5 and are listed there. Proceed to option 5 from here.)
389	3. Print Entire Fit — No operator input required. (See p. 68 for sample output.)
435	4. Print Coefficients — No operator input required. (See p. 68 for sample output.)
	5. Change (Fitting) Function (or) Polynomial Type — Produces:

Table A3. (Continued)

Line	QUERY or Response
336	TYPE 1 IF THERE IS OFFSET, 0 IF NO OFFSET:
	Enter integer 1 or 0. [OFFSET (1) adds a constant term to the function expansion.]
341	SELECT THE NUMBER OF THE FUNCTION TYPE, POLYNOMIAL, DEGREE, AND NUMBER OF CROSS TERMS FOR EACH MAGNITUDE AND PHASE
344	FUNCTION TYPE: 1 = LINEAR 2 = LOG 3 = EXP 4 = INV
346	(Blank line)
347	FCN TYPE POL DEG #CROSS TERMS
	MAG AT (number) HZ PHA AT (number) HZ ... } Response entered on these lines
	Enter on each MAG line; integer 1—4, delimiting blank (space), integer 1—15, space, integer 0.
	Enter on each PHA line: integer 1, 3, or 4; space; integer 1—15; space; integer 1—15. (Sum of integers 1—15 cannot exceed 15.) (See p. 68 for sample outputs.)
363	ERROR: NUMBER OF TERMS IN POLARY EXCEEDS DIMENSION
	(No response — program returns to line 454 above.) This error message occurs if the sum of POL DEG and #CROSS TERM entries exceeds 15.
	Action included in option 5 continues until line 454 is encountered again. (See p. 68 for samples of sequentially occurring outputs.)
	6. <u>Change Coil (or) Circuit Values</u> — Produces the following:
466	From subroutine CHRFPM:
(S-13)	COIL INN.RAD. OUT.RAD LENGTH SHT CAP DC RES TURNS EA.
(S-16)	DRV (Number) (Number) (Number) (Number) (Number) (Number)
(S-18)	(Blank line)
	Enter free format number or "0" for each parameter; separate each pair by blank (space) or comma delimiter; "0" will leave the corresponding parameter unchanged.

Table A3. (Continued)

Line	QUERY or Response
(S-20)	PIC (Number) (Number) (Number) (Number) (Number) (Number)
	Use same entry format as for "DVR" line.
(S-24) (S-26)	RBAR VOUT DVR AMP RES PIC AMP RES (Number) (Number) (Number) (Number) (Blank line)
	Enter numbers for each parameter as described above.
	7. <u>Change (Coil or Circuit Values) with Printout.</u>
	Printout or display is identical to that shown for option 6 above. Additional hard-copy output is printed (lines 273 and 278).
After execution of CHRFPM control returns to line 170 (statement 15) if coil dimensions are changed or to line 297 (statement 160) if circuit parameters are changed.	
Return of control to line 170 causes repetition of the action selection at line 454 (described above).	
Return of control to line 297 causes repetition of the property fitting sequence (option 2 above).	
460	8. <u>Calculate Coil or Circuit Drift</u> -- No operator input required. (See p. 68 for sample output.)
469	9. <u>Stop</u> -- No operator input required; program executes Fortran Stop Job instruction.

LISTING AND ANALYSIS OF MULRFD

Program Identification

```
 1   C
 2          PROGRAM MULRFD
```

Description of Program Functions and Key Control Variables

See Table A2 for additional information.

```
 3   C
 4   C      21 SEPT 1978 VERSION OF THE MULTI-LAYER PROGRAMS IN ORNL-TM-4107,
 5   C      PP. 8-23. ALL COMPLEX OPERATIONS HAVE BEEN ELIMINATED, AND
 6   C      THE PROGRAM NOW CALCULATES THE MAGNITUDES AND PHASES FOR ALL
 7   C      NPT SETS OF PROPERTIES.VARIATIONS IN THICKNESS,RESISTIVITY
 8   C      AND DEFECTS ARE ALLOWED.THE FREQUENCY CAN BE VARIED OVER NFT
 9   C      AND THE LIFT-OFF VARIED OVER NLT DIFFERENT VALUES.WHEN A DEFECT
10   C      IS INCLUDED THE MAGNITUDE AND PHASE WILL BE CALCULATED WITH AND
11   C      WITHOUT THE DEFECT.NO ADDITIONAL SET OF PROPERTY VALUES WITHOUT
12   C      THE DEFECT IS NEEDED.THE ACTUAL NUMERICAL VALUE OF THE SUBSCRIPT
13   C      MUST BE SUBSTITUTED IN FOR THE SUBSCRIPT NAME IN THE DIMENSION
14   C      STATEMENTS IN LINES 59-73.
15   C
16   C      DFDP=DISTANCE OF DEFECT FROM SURFACE OF LAYER (IN INCHES)
17   C      DFLOC=DISTANCE OF DEFECT BELOW TOP OF THE REGION (IN INCHES)
18   C      DFRAD=DISTANCE OF DEFECT FROM AXIS (IN INCHES)
19   C      DFDIAM=DEFECT DIAMETER(OF EQUIVALENT SPHERE, IN INCHES)
20   C      DFVOL=NORMALIZED VOLUME OF DEFECT
21   C      IRDPRM=MAXIMUM NUMBER OF COEFFICIENTS IN EXPANSION
22   C      LI=LOGICAL INPUT UNIT (1=TTY)
23   C      LOU=LOGICAL OUTPUT UNIT
24   C      NCOIL=POSITION INDEX OF COIL DATA IN FILE 28
25   C      NDFLOC=NUMBER OF DEFECT LOCATIONS IN A GIVEN SAMPLE
26   C      NDFSIZ=NUMBER OF DEFECT SIZES AT A GIVEN LOCATION
27   C      NDPS=NDFLOC*NDFSIZ=NUMBER OF DEFECTS PER SAMPLE
28   C      NFT=NUMBER OF FREQUENCIES
29   C      NLT=NUMBER OF LIFTOFF VALUES
30   C      NODF=TOTAL NUMBER OF DEFECTS
31   C      NPRINT=PRINT AND TRANSFER INDEX=0 FOR NO PRINT,1 FOR INITIAL PRINT
32   C      NPT=NUMBER OF SETS OF PROPERTIES
33   C      NPTT=NPT+NODF=NUMBER OF PROPERTIES+NUMBER OF DEFECTS
34   C      NRDF=NUMBER OF THE REGION WHERE DEFECT IS FOUND(=NRT IF NO DEFECT)
35   C      NUNIT=INDEX OF PRINTOUT UNITS(1=INCHES;2=MILLIMETERS)
36   C
```

Optional High-Precision Variables

Remove the "C" by editing if additional precision is required.

```
37   C      REAL*8 READNG,COEF,SUM
38          REAL*6 NPROBE
```

Conversion of Normally Integer Variables to Real

```
39          REAL L2,L3,L4,L5,L6
```

Required for correspondence with conventional coil dimension usage previously established.

Dimensions for Control Parameters Not Requiring Adjustment

```
40          DIMENSION HDR(2), HPU(2), ITIM(3), IDAY(3),PROPTY(6),CKTPAR(8)
41          DIMENSION CONVRT(2),UNITS(6,2),UN(6)
```

Listing and Identification of Dimensions that Normally Require Adjustments for New Problems

Literal variables in these comment statements locate the positions of the control variables in the array dimensions, lines 61--75.

```
42   C
43   C                DIMENSIONS THAT ARE CHANGED:
44   C       DIMENSION READNG(NPTT*NLT+1,IRDPRM+1),POLARY(IRDPRM+1,5)
45   C       DIMENSION PROP(NPTT*NLT),RL1(NLT),RL2(NLT),WUSR(NRT,NPTT,NFT)
46   C       DIMENSION TMDFT(NFT),PHDFT(NFT),NPOL(6,NFT),COEF(IRDPRM)
47   C       DIMENSION BETAR(NRT),BETAI(NRT),COSF(NRT),SINF(NRT),XPON(NRT)
48   C       DIMENSION RES(NRT,NPT),PERM(NRT,NPT),THICK(NRT,NPT)
49   C       DIMENSION RHO(NRT,NPTT),U(NRT,NPTT),TH(NRT,NPTT)
50   C       DIMENSION DFDP(NDFLOC),DFV(NPTT),DFSIZ(NPTT)
51   C       DIMENSION NRDF(NPTT),DFRAD(NPTT),DFLOC(NPTT),FREQ(NFT),GAIN(NFT)
52   C       DIMENSION GAMAR(NPTT),GAMAI(NPTT),GAMADR(NPTT),GAMADI(NPTT)
53   C       DIMENSION TMAG(NLT,NPTT,NFT),PHASE(NLT,NPTT,NFT)
54   C       DIMENSION TMUTRE(NLT,NPTT,NFT),TMUTIM(NLT,NPTT,NFT)
55   C       DIMENSION DRIVRE(NLT,NPTT,NFT),DRIVIM(NLT,NPTT,NFT)
56   C       DIMENSION PICKRE(NLT,NPTT,NFT),PICKIM(NLT,NPTT,NFT)
57   C       DIMENSION DRVDR(NLT,NPTT,NFT),PICKDR(NLT,NPTT,NFT)
58   C       DIMENSION DRVDI(NLT,NPTT,NFT),PICKDI(NLT,NPTT,NFT)
```

Array Dimensions

```
59   C   THE APPROPRIATE NUMBERS SHOULD BE INSERTED IN THE FOLLOWING DIMENSION
60   C   STATEMENTS:
61          DIMENSION READNG(109,16),POLARY(16,5)
62          DIMENSION PROP(108),RL1(3),RL2(3),WUSR(3,36,3)
63          DIMENSION TMDFT(3),PHDFT(3),NPOL(6,3),COFF(15)
64          DIMENSION BETAR(3),BETAI(3),COSF(3),SINF(3),XPON(3)
65          DIMENSION RES(3,9),PERM(3,9),THICK(3,9)
66          DIMENSION RHO(3,36),U(3,36),TH(3,36)
67          DIMENSION DFDP(1),DFV(36),DFSIZ(36)
68          DIMENSION NRDF(36),DFRAD(36),DFLOC(36),FREQ(3),GAIN(3)
69          DIMENSION GAMAR(36),GAMAI(36),GAMADR(36),GAMADI(36)
```

```
      DIMENSION GAMAR(36),GAMAI(36),GAMADR(36),GAMADI(36)
      DIMENSION TMAG(3,36,3),PHASE(3,36,3)
      DIMENSION TMUTRE(3,36,3),TMUTIM(3,36,3)
      DIMENSION DRIVRE(3,36,3),DRIVIM(3,36,3)
      DIMENSION PICKRE(3,36,3),PICKIM(3,36,3)
      DIMENSION DRVDR(3,36,3),PICKDR(3,36,3)
      DIMENSION DRVDI(3,36,3),PICKDI(3,36,3)
```

Common Block Storage Allocation

B1 — variables transferred to subroutines calculating integrals.

B2,B6 — variables transferred to subroutines for coil or output computation.

```
      COMMON /B1/X,XX,XXXX
      COMMON /B2/R1,R2,R3,R4,L3,L4,RBAR,VO,ZLDR,ZLPU,
     *TNDR,TNPU,RDCDR,RDCPU,RO,R9,CAPDR,CAPPU
      COMMON /B6/ L2,L5,L6
```

RAD File Definitions

File 28 — permanent storage of coil data.

File 21 — temporary storage for least squares fit function description (array NPOL) and term coefficients (array COEF).

```
      DEFINE FILE 28(80,32,U,NCOL)
      DEFINE FILE 21(30,40,U,ICOEF)
```

Data Initialization

See Table A3 for required editorial changes. TWOPI and RAD are angle conversion factors; CONVRT, metric and English length conversions; and PROPTY, UNITS, and CKTPAR contain repeated headings for column printouts. (Note: The single value for DFDIAM is used because the defect size response is linear with volume over a reasonable range.)

```
      DATA TWOPI/6.28318531/,RAD/57.2957795/,CONVRT/1.,25.4/
      DATA PROPTY/4HREST,4HPERM,4HTHIK,4HL.O.,4HDLOC,4HDFSI/
      DATA UNITS/4HMOCM,4HREL.,4HINCH,4HINCH,4HINCH,4HCUIN,
     *           4HMOCM,4HREL.,4H MM ,4H MM ,4H MM ,4HCUMM/
      DATA NUNIT/2/
      DATA CKTPAR/4HDR.T,4HPU.T,4HDR.R,4HPU.R,4HSE.R,4HSH.R,
     *4HDR.C,4HPU.C/
      DATA NRT/3/,NPT/9/,NLT/3/,NFT/3/,NDFLOC/3/,NDFSIZ/1/,NPROBE/0.DO/
      DATA NPRINT/0/,LOU/11/,LI/7/,IR/1/,IRDPRM/15/,ICOEF/1/
      DATA NRDF/36*2/
      DATA DFDP/.016/
      DATA DFRAD/36*.0832/,DFDIAM/.032/
```

Property Value Initialization

Property arrays are listed column by column; each of NPT columns has NRT rows (in this example NRT = 3, NPT = 9). Values are arranged within each array such that if the (NR,NP) elements of each array (RES, PERM, and THICK) are combined as sets, NRT*NPT different sets will result.

```
C
C        THE MATERIAL PROPERTIES NOW FOLLOW,STARTING WITH THE FIRST
C        PROPERTY SET FOR THE LOWERMOST REGION. (RHO=RESISTIVITY IN
C        MICROHM-CM;U=RELATIVE PERMEABILITY;TH=THICKNESS IN INCHES)
C
      DATA RES/1.E10,3.95,2*1.E10,3.95,2*1.E10,3.95,
     *       2*1.E10,5.03,2*1.E10,5.03,2*1.E10,5.03,
     *       2*1.E10,6.07,2*1.E10,6.07,2*1.E10,6.07,1.E10/
      DATA PERM/27*1./
      DATA THICK/1.E10,.055,2*1.E10,.071,2*1.E10,.077,
     *         2*1.E10,.052,2*1.E10,.066,2*1.E10,.076,
     *         2*1.E10,.054,2*1.E10,.067,2*1.E10,.072,1.E10/
```

Initialization of Frequencies and Gains

```
C
C     THE DIFFERENT FREQUENCIES AND GAINS ARE NOW GIVEN.
C     L2=NORMALIZED LIFTOFF INCREMENT.
C
      DATA FREQ/2.E3,2.E4,1.E5/
      DATA GAIN/330.,20.,8./
```

Check Calculation of Control Parameter NPTT

This calculation must be performed *before* editorial sizing of READNG array.

```
C
      NDFS=NDFLOC*NDFSIZ
      NDFS1=NDFS+1
      NODF=NPT*NDFS
      NPTT=NPT+NODF
```

Initialize Normalized Lift-Off Increment

L_2 = actual lift-off increment divided by mean driver-coil radius. By using the increment this way actual increments will change with driver-coil size selection.

```
      L2=.01333
```

Time, Date, and CPU Run Time

Subroutines are part of machine operating system. This step starts the CPU timer.

```
C
      CALL TIME (ITIM)
      CALL DATE (IDAY)
      CALL TIMER(STARTT)
      IT = IDAY(1) - 1900
      IDAY(1) = IDAY(2)
      IDAY(2) = IDAY(3)
      IDAY(3) = IT
      WRITE (LOU,5)
    5 FORMAT ('1',20X)
      WRITE (LOU,10) IDAY,ITIM
   10 FORMAT ('0PROGRAM MULRFD DATE   ',2(I2,1H/),I2,
     *  ' TIME   ',2(I2,1H:),I2,/)
```

Coil Data Selection or Input

Branch to subroutine COLFIL allows selective reading of coil data file or manual insertion of coil data (see Appendix E, p. 160).

(Note: Coil data are now in File 28).

```
C
C        THE INPUT DATA FOR THE PARAMETERS OF THE COILS, MATERIAL,
C        AND CIRCUIT ARE READ FROM FILE 28. SEE FIG.2, P.4
C        AND FIG.4, P.7, ORNL-TM-4107, FOR DEFINITIONS.
C
      CALL COLFIL(LI,LOU,NPROBE)
```

Circuit Parameter Initialization

Editorial change required. Units as shown in comment lines.

```
C
C     THE CIRCUIT PARAMETERS ARE NOW GIVEN.
C     R0=DRIVER SERIES RESISTANCE(OHMS)
C     R9=PICKUP SHUNT RESISTANCE(OHMS)
C     CAPDR=DRIVER SHUNT CAPACITANCE(FARADS)
C     CAPPU=PICKUP SHUNT CAPACITANCE(FARADS)
C     V0=DRIVER OUTPUT VOLTAGE(RMS VOLTS)
C
      R0=200.
      R9=1.E6
      CAPDR=8.47E-11
      CAPPU=8.45E-11
      V0=3.535
```

Unit Conversion

Conversion factors of 1, to retain inch dimensions, or 25.4, to convert to millimeter dimensions are selected.

```
C
C       SET UP PROPERTIES FOR THE INTEGRATION
C
        CNVT=CONVRT(NUNIT)
```

Defect Normalization

Normalized diameter: $d_n = d/\overline{r}$ ($\overline{r}$ = R5),

Volume factor = $d_n^3/8$,

Normalized volume = $\pi d_n^3/6$.

```
        DINORM=DFDIAM/RBAR
        VOLFAC=DINORM*DINORM*DINORM/8.
        DFVOL=TWOPI*VOLFAC/1.5
```

Rearrangement of Property Arrays

Initial arrays RES, THICK, and PERM (for resistivity, thickness, and permeability) are arranged for a defect-free conductor stack. The example is a three-layer system with a nine-property conductor stack, so the property arrays are each (NR,NP) up to (3,9).

When defects are added the conductor properties are transferred to larger arrays (RHO, TH, U) with extra entries of those property sets that correspond to the defect cases. The defect properties are stored in separate arrays, DFLOC and DFV, numbered (indexed) the same way as the conductor properties. Thus, a defect calculation picks up the conductor properties from RHO, etc. and the defect properties for DFLOC and DFV.

As written the program generates a single value of DFV for the defect and three values of DFLOC, corresponding to the driver side, middle, and far side of the conducting layer. The program is limited to a value of

3 for NDPS by the condition statements in lines 167–169, which could be changed.

```
157          DO 15 NS=1,NPT
158          DO 15 ND=1,NDPS1
159          NP=(NS-1)*NDPS1+ND
160          DO 15 NR=1,NRT
161          RHO(NR,NP)=RES(NR,NS)
162          TH(NR,NP)=THICK(NR,NS)
163          U(NR,NP)=PERM(NR,NS)
164          IF(NR.NE.NRDF(NP)) GO TO 15
165          DFLOC(NP)=THICK(NR,NS)/2.
166          DFV(NP)=DFVOL
167          IF(ND.EQ.1) DFV(NP)=0.
168          IF(ND.EQ.2) DFLOC(NP)=DFDP(1)
169          IF(ND.EQ.4) DFLOC(NP)=THICK(NR,NS)-DFDP(1)
170       15 CONTINUE
```

Manipulations to Speed Computations Inside Long Loops

RRBAR = $1/\overline{r}$ (R5 = $\overline{r}$)

$T_{n'} = T_n \mu_n / \overline{r}$; then $\mu_{n'} = (1/\sqrt{2})\ \mu_n$

$Z_{7'} = \sqrt{2}\ Z_7/\mu_n\overline{r}$; $r_{d'} = r_d/\overline{r}$

```
171  C
172  C        COMPUTE RECIPROCALS FOR LATER USE  TO AVOID DIVISIONS
173  C        INSIDE LOOPS.
174  C
175      20 RRBAR = 1./RBAR
176  C
177  C      THE PERMEABILITY IS CHANGED TO SQRT(.5)/U(N) AND THE THICKNESS IS
178  C      CHANGED TO TH(N)*U(N)/RBAR.
179  C
180         DO  30 NP=1,NPTT
181         IF(NPRINT.NE.0) WRITE(LOU,25)
182      25, FORMAT(1X)
183         DO  30 NR=1,NRT
184         TH(NR,NP)=U(NR,NP)*TH(NR,NP)*RRB.R
185         U(NR,NP)=.707106781/U(NR,NP)
186         IF(NR.NE.NRDF(NP)) GO TO 30
187         DFLOC(NP)=.707106781*DFLOC(NP)*RRBAR/U(NR,NP)
188         DFRAD(NP)=DFRAD(NP)*RRBAR
189      30 CONTINUE
```

Impedance Integrals

The integrals are computed from B1 and B2 in steps of S1. Initial values of B1, B2, and S1 are 0, 5, and 0.01, respectively. Value X represents α, the integration variable. (S1 is equivalent to $d\alpha$.) The integrands are evaluated at the midpoints of the steps, S1. The value of S1 is progressively increased to 0.5 as X increases toward 80, where the integration terminates.

All integrands are computed and summed by the subroutine RFCOLM (Appendix E, p. 178).

RCON = $\omega\mu\sigma\overline{r}^2$ with unit conversion (later multiplied by μ_r = relative permability and ρ = resistivity in $\mu\Omega$-cm).

$(4\pi \times 10^{-7}\text{H/m})(0.0254\ \text{m/in.})^2(2\pi\text{rad/Hz})(10^8\mu\Omega\text{-cm}/\Omega\text{-m})/\sqrt{2} = 0.36\ldots$

TMUTRE, TMUTIM = real and imaginary parts of M^T (without conversion factors) [Eq. (19)].

DRIV, PICK denote corresponding terms for Z_D, $Z_P^{(R)}$ [Eqs. (18), (22)].

DRVD/PICKD denote corresponding terms for defect impedances, Z_D', Z_P' [Eqs. (50), (53), (54)].

WUSR = $\omega\mu\sigma\overline{r}^2$ ($R_5 = \overline{r}$), and
A1R1, A1R2 = Z_D, $Z_P^{(R)}$ with $\gamma_D = 0$ [Eqs. (18), (22)].

```
C
C           THE INTEGRATION IS PERFORMED BY THE MIDPOINT METHOD,
C        EVALUATING AT THE CENTER OF THE INTERVAL; FOR X LARGE
C        THE INTEGRAL CONVERGES RAPIDLY,SO LARGER INTERVALS
C        ARE TAKEN.
C           IN THE INTEGRATION TMUT, DRIVER, PICKUP, AIR1, AND
C        AIR2 ARE CALCULATED.
C
      S1 = 0.01
      S2 = 5.0
      B1 = 0.0
      B2 = S2
C
C        INITIALIZE ALL SUMS TO ZERO AND CALCULATE THE VALUES OF
C        WUSR(NR,NP,NF).
C
      DO 50 NF=1,NFT
      RCON=.360198724*RBAR*RBAR*FREQ(NF)
      DO  50 NP=1,NPTT
      DO 40 NL=1,NLT
          TMUTRE(NL,NP,NF)=0.
          TMUTIM(NL,NP,NF)=0.
          DRIVRE(NL,NP,NF)=0.
          DRIVIM(NL,NP,NF)=0.
          PICKRE(NL,NP,NF)=0.
          PICKIM(NL,NP,NF)=0.
          DRVDR(NL,NP,NF)=0.
```

```
            DRVDI(NL,NP,NF)=0.
            PICKDR(NL,NP,NF)=0.
            PICKDI(NL,NP,NF)=0.
   40   CONTINUE
        DO 50 NR=1,NRT
            WUSR(NR,NP,NF)=RCON/(RHO(NR,NP)*U(NR,NP))
   50   CONTINUE
        AIR1=0.
        AIR2=0.
C
   60   I1 = (B2 - B1)/S1
        X = B1 - S1*0.5
        DO 70 I=1,I1
          X = X + S1
          XX = X*X
          XXXX = XX*XX
        CALL RFCOLM(S1,NLT,NPTT,NRT,NFT,RL1,RL2,GAMAR,GAMAI
       1,GAMADR,GAMADI,TMUTRE,TMUTIM,DRIVRE,DRIVIM,PICKRE,PICKIM
       2,DRVDR,DRVDI,PICKDR,PICKDI,BETAR,BETAI,COSF,SINF,XPON
       3,WUSR,U,TH,NFDF,DFRAD,DFLOC,FREQ,AIR1,AIR2)
C
   70   CONTINUE
        B1 = B2
        B2 = B2 + S2
        S1 = 0.05
        IF (X .LT. 9.) GO TO 60
        S1 = 0.1
        IF (X .LT. 29.) GO TO 60
        S1 = 0.2
        IF (X .LT. 39.) GO TO 60
        S1 = 0.5
        IF (X .LT. 79.) GO TO 60
```

Restoration of Property Values and Unit Conversion

Defect size = defect volume × ($\bar{r}$ × length conversion)^3.

The defect size array is generated at this point; it represents actual volume:

$\mu_n = 1/(\mu_n' \sqrt{2})$, and

$T_n = T_n' \times (\text{conversion})\bar{r}/\mu$.

Defect location $Z_7 = (\text{conversion}) Z_7' \bar{r}/\mu_n'$.

Defect radius $r_d = (\text{conversion}) r_d' \bar{r}$.

Loop 85 converts column headings for printouts

```
C
C       THE INTEGRATION ENDS HERE.THE VALUES OF THICKNESS,U,DFLOC,DFRAD
C       ARE RESTORED TO ORIGIONAL VALUES,THEN CONVERTED TO DESIRED UNITS.
C
```

```
253          DO 80 NP=1,NPTT
254          DFSIZ(NP)=DFV(NP)*((RBAR*CNVT)**3)
255          DO 80 NR=1,NRT
256          U(NR,NP)=.707106781/U(NR,NP)
257          TH(NR,NP)=CNVT*TH(NR,NP)/(U(NR,NP)*RRBAR)
258          IF(NR.NE.NRDF(NP)) GO TO  80
259          DFLOC(NP)=CNVT*DFLOC(NP)/(RRBAR*U(NR,NP))
260          DFRAD(NP)=CNVT*DFRAD(NP)/RRBAR
261       80 CONTINUE
262          DO 85 NU=1,6
263          UN(NU)=UNITS(NU,NUNIT)
264       85 CONTINUE
```

Computation and Printout of CPU Time for Calculation

```
265          CALL TIMER(STOPT)
266          ELAPST=STOPT-STARTT
267          WRITE (LOU,90) ELAPST
268       90 FORMAT(' ELAPSED TIME ',F9.2,/)
269          GO TO 640
```

Selection of Further Actions

At this point the program execution jumps from line 269 to 454 (statement 640). The action is the same as if line 452 followed 269. (See Table A3 for summary of interactive displays and responses.)

The options are described in the following numbered sections.

Print Properties (1)

Branch to statement 100 (line 273). If NPRINT $\neq$ 0 branch to subroutine PRFCOL (Appendix E, p. 166) to print dimensional coil properties then back to line 277 to print list of property sets for the conductor stack. (Must have NPRINT = 1 to reach this branch.)

```
270   C
271   C      THE COIL VALUES ARE NOW PRINTED OUT
272   C
273     100 IF (NPRINT.NE.0) CALL PRFCOL(LOU)
274   C
275   C      EACH SET OF PROPERTIES FOR EACH REGION IS NOW PRINTED OUT.
276   C
277          IF(NPRINT.NE.0)WRITE(LOU,110)UN(3),UN(1),UN(2),UN(5),UN(5),UN(6)
278     110 FORMAT('OPR.SET  REG   LAY TH       RESTVY       PERM    DEF LOC ',
279        *' DEF RAD   DEF SIZ',/,16X,A4,8X,A4,7X,A4,6X,A4,2(5X,A4))
280          DO 120 NS=1,NPT
281          IF (NPRINT.NE.0) WRITE(LOU,25)
282          DO 120 ND=1,NDPS1
```

```
283          NP=(NS-1)*NDPS1+ND
284          DO 120 NR=1,NRT
285          IF (NPRINT.EQ.0) GO TO 120
286          IF(NR.EQ.1)WRITE(LOU,130)NP,NR,TH(NR,NP),RHO(NR,NP),U(NR,NP)
287          IF(NR.GT.1)WRITE(LOU,140)NR,TH(NR,NP),RHO(NR,NP),U(NR,NP)
288          IF(DFV(NP).EQ.0.) GO TO 120
289          IF(NR.EQ.NRDF(NP))WRITE(LOU,150)DFLOC(NP),DFRAD(NP),DFSIZ(NP)
290      120 CONTINUE
291      130 FORMAT(1X,2(I5),2(1PE12.4),0PF9.3)
292      140 FORMAT(6X,I5,2(1PE12.4),0PF9.3)
293      150 FORMAT('+',44X,2(F9.4),,1PE12.4)
```

Fit Property (2)

This branch, starting at statement 160, will follow branch (1) above when NPRINT = 1 or branch directly if NPRINT = 2. Subroutine RFCIRK computes and returns magnitudes and phases of V_{out} [Eq. (14)]. Subroutine PRFCKT prints out coil and circuit electrical properties. Subroutine PRFVLT prints the magnitudes and phases of V_{out} calculated by RFCIRK. The return to statement 640 at line 311 completes option (1).

```
294   C
295   C        NEXT THE PROPERTIES OF THE COILS ARE DETERMINED AND PRINTED
296   C
297      160 CALL RFCIRK(TMAG,PHASE,TWOPI,RA[illegible],TMUTRE,TMUTIM,DRIVRE
298         1,DRIVIM,PICKRE,PICKIM,DRVDR,DR[illegible],PICKDR,PICKDI,NRDF,FREQ
299         2,U,WUSR,GAIN,AIR1,AIR2,NLT,NR[illegible]PTT,NFT,NODF,NPTT,DFV)
300   C
301   C       THE CIRCUIT PARAMETE[illegible]TED OUT FOR THE REFLECTION
302   C       COIL CIRCUIT.
303   C
304           IF (NPRINT.EQ.1) CALL [illegible](LOU)
305   C
306   C        THE VOLTAGE MAGNITUDE,PHASE,AND SHIFT ARE PRINTED FOR
307   C        THE VARIOUS LIFT-OFF VALUES AND DEFECTS.
308   C
309           IF(NPRINT.EQ.1) CALL PRFVLT(LOU,TMAG,PHASE,NRT,NPTT,NFT,NLT,NODF
310          1,NPTT,RL1,NRDF,FREQ,GAIN)
311           IF (NPRINT.NE.2) GO TO 640
```

When branching is direct to statement 160, NPRINT = 2, the circuit and voltage printouts are skipped, and least squares design calculations begin. The value of MSET1 becomes the READING array row dimension required.

The maximum filled section of PROP(M) will contain NPTT*NLT entries. Each step value of a conductor property appears (NPTT/NPT) times; each value of lift-off, (NL-1)*L_2, appears NPTT times; and each step value of a defect location appears NPT times.

As written the single value of defect size appears NODF times. Thus, the defect size is weighted differently than the other properties in the fitting process, unless NODF = NPT.

```
C
C       LEAST SQUARES DESIGN SECTION.
C
C       SELECT PROPERTY TO BE FITTED AND SET UP PROPERTY ARRAY.
C
        MSET=NLT*NPTT
        MSET1=MSET+1
    170 WRITE(LOU,180)
        WRITE(LOU,190)
    180 FORMAT(' SELECT NUMBER OF THE PROPERTY TO BE FITTED:'/
       *' 1=RHO 2=MU 3=THICKNESS 4=LIFT-OFF 5=DEFECT LOCAT 6=DEFECT SIZE')
    190 FORMAT(' SELECT REGION TO BE FITTED. LOWEST IS NUMBER 1.')
        READ(LI,*)NPROP,NREG
        DO 210 NP=1,NPTT
        DO 200 NL=1,NLT
        M=(NP-1)*NLT+NL
        IF(NPROP.EQ.1) PROP(M)=RHO(NREG,NP)
        IF(NPROP.EQ.2) PROP(M)=U(NREG,NP)
        IF(NPROP.EQ.3) PROP(M)=TH(NREG,NP)
        IF(NPROP.EQ.4) PROP(M)=(FLOAT(NL-1)*L2)*RBAR*CNVT
        IF(NPROP.EQ.5) PROP(M)=DFLOC(NP)
        IF(NPROP.EQ.6) PROP(M)=DFSIZ(NP)
    200 CONTINUE
    210 CONTINUE
```

Changing Fit Function (5)

Selecting OFFSET (1) means that an initial constant will be added to the function generated by the fitting process. Entering polynomial selections fills the NPOL array. (See Table A2 and Appendix G for array element definitions.)

```
    220 WRITE(LOU,230)
    230 FORMAT(' TYPE 1 IF THERE IS OFFSET, 0 IF NO OFFSET: ')
        READ(LI,*) IOFSET
        IRDPR=IOFSET
    240 WRITE(LOU,250)
    250 FORMAT(' SELECT THE NUMBER OF THE FUNCTION TYPE, POLYNOMIAL',
       1' DEGREE, & # OF '/,
       2' CROSS TERMS FOR EACH MAGNITUDE & PHASE')
        WRITE(LOU,260)
    260 FORMAT(' FUNCTION TYPE:1=LINEAR 2=LOG 3=EXP 4=INV ')
    300 WRITE(LOU,25)
        WRITE(LOU,310)
    310 FORMAT(25X,'FCTN POL # CROSS'/,25X'TYPE DEG TERMS')
        DO 370 NF=1,NFT
        DO 360 NC=1,2
        NCC=NC*3
        NCP=NCC-1
        NCF=NCP-1
        IF(NC.EQ.1) WRITE(LOU,320) FREQ(NF)
        IF(NC.EQ.2) WRITE(LOU,330) FREQ(NF)
```

```
356    320 FORMAT(' MAG AT ',1PE12.6,' HZ    ')
357    330 FORMAT(' PHA AT ',1PE12.6,' HZ    ')
358        READ(LI,*)NPOL(NCF,NF),NPOL(NCP,NF),NPOL(NCC,NF)
359        NPOL(3,NF)=0
360        IRDPR=IRDPR+NPOL(NCP,NF)+NPOL(NCC,NF)
361    360 CONTINUE
```

The term IRDPR is the number of terms in the function expansion selected; its value is increased by 1 to size the column dimension of READNG, allowing space for the property value to which the function is fitted. If the number of expansion terms selected exceeds the preset capacity of READNG, the error message (line 363) occurs. Subroutine POLTYP prints a representation of the function expansion (see Appendix E, p. 163).

```
362    370 CONTINUE
363        IRDPR1=IRDPR+1
364        IF(IRDPRM.LT.IRDPR)WRITE(LOU,400)
365        IF (IRDPRM.LT.IRDPR)GO TO 640
366    400 FORMAT(' ERROR: # OF TERMS IN POLARY EXCEEDS DIMENSION')
367        JROW=IRDPRM+1
```

The loop at statement 460 initializes arrays for storing values of simulated magnitude and phase response into which drifts are introduced. The loops at statements 470 and 480 substitute the successive simulated magnitude and phase readings into the function expansion chosen above in preparation for fitting. (See Appendix E, p. 169, for RDGEXP subroutine).

```
368          CALL POLTYP(POLARY,JROW,IRDPR,NPOL,6,NFT,2,IOFSET,LOU)
369   C
370   C       EXPAND THE RAW READINGS INTO IRDPR READINGS.
371   C
372     450 DO 460 NF=1,NFT
373          TMDFT(NF)=0.
374          PHDFT(NF)=0.
375     460 CONTINUE
376          DO 480 NP=1,NPTT
377          DO 470 NL=1,NLT
378          M=(NP-1)*NLT+NL
379          CALL RDGEXP(READNG,TMAG,PHASE,NPOL,IOFSET,TMDFT,PHDFT,MSET1
380         1,IRDPR1,M,NFT,NL,NLT,NP,NPTT)
381     470 CONTINUE
```

Subroutine ALSQS computes the least squares fit between the simulated response and the conductor or defect property (see Appendix E, p. 147). It returns coefficients of the function terms in the COEF array.

```
382     480 CONTINUE
383   C
384   C       DO THE LEAST SQUARES FIT OF THE READINGS TO THE PROPERTIES.
385   C
```

Tests of Fit Quality [Printout of Function Fit (3)]

Two tests are performed simultaneously within the loops of statements 570 and 580. (Details of the tests are described on pp. 39–41.)

The same loop set prints, when selected (NPRINT = 3), a table of the values computed from the fitted function, the properties, and the fit-testing parameters (see p. 75 for sample output). Regardless of the section selected (NPRINT value), the values of rms residual and drift are printed at the end of the computation.

```
      CALL ALSQS(READNG,PROP,COEF,RSQS,MSET,IRDPR,MSET1)
C
C     CALCULATE THE DIFFERENCES IN THE FIT AND THE MAXIMUM DRIFTS.
C
  490 SSDRIF=0.
      SSDIFF=0.
      UNIT=UN(NPROP)
      IF(NPRINT.EQ.3)WRITE(LOU,500)PROPTY(NPROP),NREG,PROPTY(NPROP),
     *UNIT,UNIT,UNIT,UNIT
  500 FORMAT('0 NP NL ',A4,' IN REG',I3,' CAL ',A4,7X,'DIFF',7X,'DRIFT'
     *,/,6X,4(8X,A4),/)
      DO 580 NP=1,NPTT
      DO 570 NL=1,NLT
      M=(NP-1)*NLT+NL
      DRIFT=0.
      DO 540 NF=1,NFT
      DO 530 NC=1,2
C
C     ONE MAGNITUDE OR PHASE DRIFT IS SET ON AT A TIME.
C
      IF(NC.EQ.1)TMDFT(NF)=.0001*TMAG(NL,NP,NF)
      IF(NC.EQ.2)PHDFT(NF)=.01
      CALL RDGEXP(READNG,TMAG,PHASE,NPOL,IOFSET,TMDFT,PHDFT,MSET1
     1,IRDPR1,M,NFT,NL,NLT,NP,NPTT)
C
C     THE POLYNOMIAL IS CALCULATED
C
      SUM=0.
      DO 520 IR=1,IRDPR
      SUM=SUM+COEF(IR)*READNG(M,IR)
  520 CONTINUE
      DRIFT=DRIFT+ABS(READNG(M,IRDPR1)-SUM)
      TMDFT(NF)=0.
      PHDFT(NF)=0.
  530 CONTINUE
  540 CONTINUE
      DIFF=PROP(M)-READNG(M,IRDPR1)
      SSDIFF=SSDIFF+DIFF*DIFF
      SSDRIF=SSDRIF+DRIFT*DRIFT
      IF(NPRINT.NE.3)GO TO 570
      WRITE(LOU,560)NP,NL,PROP(M),READNG(M,IRDPR1),DIFF,DRIFT
  560 FORMAT(1X,2I3,4F12.5)
  570 CONTINUE
  580 CONTINUE
      SDRIF=SQRT(SSDRIF/FLOAT(MSET))
```

```
431          SDIFF=SQRT(SSDIFF/FLOAT(MSET))
432          IF(NPROP.EQ.6) GO TO 585
433      585 WRITE(LOU,590)PROPTY(NPROP),UN(NPROP),NREG,SDIFF,SDRIF
434      590 FORMAT(' DIFF IN ',A4,' IN ',A4,' IN REG #',I3,'=',F12.5,
```

Printing Coefficients of Least Squares Fit

Subroutine CON8 converts decimal values to hexadecimal; both values are printed. Function terms were previously generated by the POLTYP subroutine.

```
435         *' DRIFT= ',F12.5)
436      600 IF (NPRINT.NE.4)GO TO 640
437          WRITE(LOU,25)
438          DO 630 I=1,IRDPR
439          CALL CON8(COEF(I),IHEX)
440          WRITE(LOU,620)I,COEF(I),IHEX,(POLARY(I,J),J=1,5)
441      620 FORMAT(' COEF(',I3,')=',1PE15.7,4X,Z8,1X,5A4)
442      630 CONTINUE
```

Writing Fit Coefficients and Identification on Disc File

The term PROPTY identifies the property fitted; IRDPR indicates the number of terms in the fit (function expansion); IOFSET indicates a constant term in the fit; COEF (IR) contains the coefficients of the expansion terms; NPOL carries the structure of the function expansion, including type of function, number of terms, and number of cross terms.

```
443   C
444   C      THE COEFFICIENT,OFFSET,NPOL AND IRDPR ARE WRITTEN ON THE RAD DISC
445   C      FILE #21.
446   C
447          WRITE(21'ICOEF)PROPTY(NPROP),IRDPR,IOFSET
448          WRITE(21'ICOEF)(COEF(IR),IR=1,IRDPR)
449          DO 635 NF=1,NFT
450          WRITE(21'ICOEF)(NPOL(I,NF),I=1,6)
451      635 CONTINUE
```

Selection of Next Action

This sequence is executed initially from a jump at line 269 and after the completion of the various selected sequences from jumps at lines 311, 364, 435, and 465.

```
452   C
453   C      THE NEXT ACTION IS SELECTED.
454   C
455      640 WRITE(LOU,650)
```

```
      READ(LI,*)NPRINT
      GO TO(100,160,490,600,220,700,700,660,800),NPRINT
  650 FORMAT(' TYPE NUMBER OF NEXT ACTION TO BE TAKEN:'/' 1=PRT PROPS;2=
     *FIT PROP;3=PRT ENTIRE FIT;4=PRT COEF;5=CHG FCTN/POL TYP  '/
```

Circuit Component Drift (8)

Option 8 from statement 640 calls subroutine RFCKDT, which calculates the effects of small drifts in circuit component values (see Appendix E, p. 176).

```
     *' 6=CHG COIL/CKT VAL;7=CHG W PRT;8=CAL COIL/CKT DRIFT;9=STOP')
  660 CALL RFCKDT(TMAG,PHASE,TWOPI,RAD,TMUTRE,TMUTIM,DRIVRE,
     1DRIVIM,PICKRE,PICKIM,DRVDR,DRVDI,PICKDR,PICKDI,NRDF,FREQ
     2,U,WUSR,GAIN,AIR1,AIR2,NLT,NRT,NPTT,NFT,NODF,NPTT,DFVOL,
     3READNG,NPOL,IOFSET,TMDFT,PHDFT,MSET1,IRDPR1,M,NL,NP,LOU,
     4MSET,IRDPR,CKTPAR,COEF,PROP,PROPTY(NPROP))
```

Change of Circuit or Coil Parameters (6, 7)

Options 6 and 7 call subroutine CHRFPM, which inserts changes. This subroutine duplicates subroutine COLFIL for inserted coil parameters (see Appendix E, p. 160).

```
      GO TO 640
  700 CALL CHRFPM(LOU,LI,INTG,NPRINT)
      IF(INTG.EQ.1)GO TO 15
```

Stop Execution (9)

Selection of statement 800 (option 9) stops execution.

```
      GO TO 160
  800 STOP JOB
```

A partial printout of a sample run is given below.

```
PROGRAM MULRFD DATE   9/26/78 TIME  12:59:26

TYPE COIL NAME (UP TO 6 LETTERS):150B
ELAPSED TIME    615.09
```

(See p. 21 for coil dimension table.)

PR.SET	REG	LAY TH MM	RESTVY MOCM	PERM REL.	DEF LOC MM	DEF RAD MM	DEF SIZ CUMM
1	1	2.5400E+11	1.0000E+10	1.000			
	2	1.3970E+00	3.9500E+00	1.000			
	3	2.5400E+11	1.0000E+10	1.000			
2	1	2.5400E+11	1.0000E+10	1.000			
	2	1.3970E+00	3.9500E+00	1.000	0.4064	2.1133	2.8116E-01
	3	2.5400E+11	1.0000E+10	1.000			
3	1	2.5400E+11	1.0000E+10	1.000			
	2	1.3970E+00	3.9500E+00	1.000	0.6985	2.1133	2.8116E-01
	3	2.5400E+11	1.0000E+10	1.000			
4	1	2.5400E+11	1.0000E+10	1.000			
	2	1.3970E+00	3.9500E+00	1.000	0.9906	2.1133	2.8116E-01
	3	2.5400E+11	1.0000E+10	1.000			

(Blocks 5—32 left out for brevity.)

33	1	2.5400E+11	1.0000E+10	1.000			
	2	1.8288E+00	6.0700E+00	1.000			
	3	2.5400E+11	1.0000E+10	1.000			
34	1	2.5400E+11	1.0000E+10	1.000			
	2	1.8288E+00	6.0700E+00	1.000	0.4064	2.1133	2.8116E-01
	3	2.5400E+11	1.0000E+10	1.000			
35	1	2.5400E+11	1.0000E+10	1.000			
	2	1.8288E+00	6.0700E+00	1.000	0.9144	2.1133	2.8116E-01
	3	2.5400E+11	1.0000E+10	1.000			
36	1	2.5400E+11	1.0000E+10	1.000			
	2	1.8288E+00	6.0700E+00	1.000	1.4224	2.1133	2.8116E-01
	3	2.5400E+11	1.0000E+10	1.000			

The following output is generated by the PRFVLT subroutine.

FREQUENCY 2.00000E+03 GAIN 3.30000E+02

PROP SET	LFT OF	0.0750	0.0883	0.1017
1	MAG	3.2476	3.1094	2.9771
	PHA	-34.531	-34.683	-34.843
2	MAG	3.2841	3.1435	3.0089
	PHA	-34.640	-34.790	-34.947
3	MAG	3.2710	3.1313	2.9976
	PHA	-34.658	-34.808	-34.965
4	MAG	3.2627	3.1236	2.9904
	PHA	-34.646	-34.796	-34.953
5	MAG	3.4340	3.2869	3.1462
	PHA	-38.354	-38.521	-38.696
6	MAG	3.4686	3.3192	3.1763
	PHA	-38.435	-38.600	-38.772
7	MAG	3.4498	3.3018	3.1601
	PHA	-38.465	-38.630	-38.802
8	MAG	3.4413	3.2938	3.1526
	PHA	-38.438	-38.604	-38.777

(Blocks 9–32 left out for brevity.)

33	MAG	2.5256	2.4200	2.3188
	PHA	-30.614	-30.774	-30.940
34	MAG	2.5513	2.4439	2.3411
	PHA	-30.661	-30.820	-30.984
35	MAG	2.5379	2.4314	2.3296
	PHA	-30.689	-30.847	-31.011
36	MAG	2.5316	2.4256	2.3241
	PHA	-30.673	-30.832	-30.997

FREQUENCY 2.00000E+04 GAIN 2.00000E+01

PROP SET	LFT OF	0.0750	0.0883	0.1017
1	MAG	3.6321	3.4651	3.3043
	PHA	-105.678	-106.123	-106.564
2	MAG	3.6462	3.4782	3.3165
	PHA	-105.986	-106.423	-106.856
3	MAG	3.6295	3.4627	3.3021
	PHA	-105.841	-106.281	-106.717
4	MAG	3.6272	3.4606	3.3001
	PHA	-105.723	-106.166	-106.605

(Blocks 5–36 left out for brevity.)

There are sample outputs from least squares curve fits on pp. 91–93.

APPENDIX B

PROGRAM MULTRU

MULTRU was designed and written to simulate the operation and response of an ideal through-transmission eddy-current instrument applied to a structure with multiple layers of conductors having parallel planar boundaries. It is used to perform design studies necessary to optimize the operational and electrical characteristics of through-transmission instruments. Simulations significantly reduce the number of laboratory experiments required and allow concentration on the more useful ones.

MULTRU accepts initializing inputs of conductor and circuit data by editing and operator interaction, computes electrical properties of the coils and responses of the circuit, constructs mathematical relationships between conductor properties and circuit responses, and computes the effects of drifts or errors in circuit response on the conductor properties indicated by these mathematical relationships.

The following initial inputs are required:

1. passive circuit component values;
2. values (or series of values) of magnetic permeability, electrical resistivity, and thickness for each conducting layer;
3. frequency and magnitude of each input voltage to the driver coil circuit;
4. amplifier gains (one for each frequency used);
5. a value for the normalized increment of lift-off (L_2);
6. various parameters to control array sizing, input, output, etc.

These inputs are itemized in Table B1. Table B2 lists step-by-step instructions for executing this program.

During execution the program interactively requests from the operator: identification of coils selected or manual (keyboard) insertion of coil dimensions and electrical parameters, selection of printed output tabulations, selection of actions to be performed by the program, and selection of property for fitting and function description for least squares fits.

Table B1. Summary of Editorial Changes Preparatory to Running MULTRU

Line(s)	Actions
42	(option) Remove initial "C" to insert double-precision variables.
58–70	Adjust array dimensions; follow key provided in lines 45–57 (see p. 34–35 for READNG array).
81	Select normalization option (NORM); adjust control parameters (NPT, NRT, NLT, NFT); enter air gap and lift-off parameters (THMAX, GAP).
82–84	Enter step values for conductor properties: resistivity, RES(NR,NP); permeability, PERM(NR,NP); and thickness, THICK(NR,NP). In each case NR = 2, ...(NRT-1), and NP = 1, ...(NPT-1).
85	Select coil spacing option (NOPT) and input/output devices (LI, LOU, LO2).
86	Enter frequencies of driver coil voltage, FREQ(NF), and bandpass amplifier gains, GAIN(NF). For each array, NF = 1, ...NFT.
147–152	Enter normalized lift-off increment (L2), passive circuit component values (R0, R9, CAPDR, CAPPU), and driver voltage magnitude (V0). Units specified in lines 141–145.

The program computes the following as intermediate results or output

1. coil inductances in air (including the mutual inductance for one specific air gap),
2. real and imaginary components of individual coil impedances and mutual coupling between coils for each distinguishable combination of conducting layer thicknesses and electrical properties and driver coil lift-off [Eqs. (18), (19), (21)],
3. amplitude and phase of pickup coil voltage [Eq. (14)] for each property and lift-off combination itemized in (2) above.

The least squares design section fits four properties: resistivity, permeability, thickness, and lift-off (driver side only). The fitting process is described earlier in this report (pp. 4–11). The property

Table B2. Interactive Program Queries and Operator Responses Occurring During Execution of MULTRU

Line	QUERY or Response
133	SELECT PRINTED TABLES: ENTER 1 TO PRINT, 0 TO OMIT, 1-PROPERTY SETS, 2-COIL DIMS, 3-COIL/CKT ELECT VALUES, 4-V/OUT MAG & PHA FOR EA PROPERTY SET
	Enter four integers, 1 or 0, separated by delimiting spaces (blanks) (sample outputs, p. 77–78, 167–168).
172 (S-868)[a]	From Subroutine TCOFIL TYPE COIL NAMES-DRIVER, PICKUP (UP TO 6 LETTERS):
	Enter two six-character alphanumeric coil designations with no separating delimiter; include blanks as necessary to make six characters.
(S-910)	COIL (name) NOT FOUND. TYPE FOLLOWING COIL PARAMETERS (INSERT ZEROES FOR UNUSED ONE): RBAR, R1, R2, L3, R3, R4, L4, L6, RDCDR, RDCPU, TNDR, TNPU
	Enter zeroes or free format numbers: RBAR in inches, R1 through L6 as normalized coil dimensions (p. 21), RDCDR and RDCPU in ohms, and TNDR and TNPU as integers.
	[This message occurs only if the coil names entered above (line S-868) are not found.]
393	TYPE NUMBER OF NEXT ACTION TO BE TAKEN: 1 = CHG FCN/POL TYP 2 = FIT PROP 3 = PRT ENTIRE FIT 4 = PRT COEF 5 = CAL COIL/CKT DRIFT 6 = CHG COIL CKT VAL 7 = CHG W PRT 8 = STOP
	Enter integer 1–8.

The following sections list ensuing interactions resulting from each of the option selections at line 393.

1. Change (Fitting) Function (or) Polynomial Type

Line	QUERY or Response
287	TYPE 1 IF THERE IS OFFSET, 0 IF NO OFFSET:
	Enter integer 0 or 1. (OFFSET (1) adds a constant term to the function expansion.)
291	SELECT THE NUMBER OF THE FUNCTION TYPE, POLYNOMIAL DEGREE, & # OF CROSS TERMS FOR EACH MAGNITUDE & PHASE
295	FUNCTION TYPE: 1 = LINEAR 2 = LOG 3 = EXP 4 = INVERSE
297	(blank line)

Table B2. (Continued)

Line		QUERY or Response			
298			FCN TYPE	POL DEG	#CROSS TERMS
305		MAG AT (number) HZ			Responses entered
306		PHA AT (number) HZ			on these lines
		Enter on each MAG line: integer 1—4, delimiting blank (space), integer 1—15, space, integer 0.			
		Enter on each PHA line: integer 1, 3, or 5, space, integer 1—15, space, integer 1—15. (Sum of POL DEG and # CROSS TERMS entries must be ≤15.) (See p. 78 for sample outputs.)			
314		ERROR: # OF TERMS IN POLARY EXCEEDS DIMENSION			
		No response — program returns to line 393 above. This error message occurs when sum of POL DEG and # CROSS TERMS entries exceeds 15.			
		Actions included in option 1 continue until line 393 is executed again. (See p. 78 for samples of sequentially occurring outputs.)			
	2.	Fit Property			
272		SELECT NUMBER OF THE PROPERTY TO BE FITTED: 1 = RHO 2 = MU 3 = THICKNESS 4 = LIFTOFF			
273		SELECT REGION TO BE FITTED. LOWEST IS NUMBER 1			
		Enter integer 1—4 (property), space, integer 1-NRT (region).			
		(The remaining actions following this option are those listed under option 1 above.)			
(342)	3.	Print Entire Fit			
		No operator input required. (See p. 78 for sample output.)			
(387)	4.	Print Coefficients			
		No operator input required. (See p. 78 for sample output.)			
(399)	5.	Calculate Coil or Circuit Drift			
		As now written a dead-end branch to a subroutine that has not been prepared. Selection returns control to line 393.			

Table B2. (Continued)

Line		QUERY or Response
(401)	6.	<u>Change Coil or Circuit Values</u>
		From subroutine CHRFPM:
(S-13) (S-16) (S-18)		COIL INN. RAD. OUT RAD. LENGTH SHT CAP PC RES TURNS EA DVR (number) (number) (number) (number) (number) (number) (blank line)
		Enter free format number or "0" for each parameter that is not changed and new values for those that are changed; separate entries by blank (space) or comma delimiter. [Variables are R1, R2, L3, C_6(CAPDR), R_6(RDCDR), N3.]
(S-20)		PIC (number) ... ...
		Use same entry format as for "DVR" line. [Variables are R3, R4, L4, C_7(CAPPU), R_7(RDCPU), N4.]
(S-24) (S-26) (S-28)		RBAR VOUT DVR AMP RES PIC AMP RES (number) ... ... (blank line)
		Enter numbers as above for "DVR" and "PIC." [Variables are RBAR(R_5 or $\overline{F}$), V_{out}, R_0, R_9.]
	7.	<u>Change (Coil or Circuit Values) with Printout</u>
		Interaction is identical to that shown for option 6 above. Additional hard-copy output is printed (line 257).
***		After execution of CHRFPM (options 6 or 7), control returns to line 172 if coil dimensions are changed or to line 248.
		As now written return of control to line 172 repeats the action of CHRFPM with respect to coil parameters by the action of TCOFIL.
		Return of control to line 248 provides recomputation of the output voltage magnitudes and phases. After either return sequence control eventually returns to line 393.
	8.	<u>Stop</u>
404		No operator input — stop job instruction executed.

[a]Indicates line number in subroutine.

arrays contain one extra entry not found in the other programs: a property set in which all conductors are given their maximum thickness and other properties are equal to those of air. This set is used to compute the air gap response used in normalization (see below). This property set must be omitted when the least squares fits are constructed.

When the least squares design function is selected, the program computes a least squares fit for the selected property according to Eq. (57). Upon selection it prints listings showing the form of the function terms, the independent coefficients of each term, the values of the function, the property value, and the error and drift for each distinguishable property set. The root-mean-square (rms) difference and drift [see Eqs. (60) and (62)] are also computed to provide an indication of fit quality.

The program can be used for either of two coil configuration options: (1) constant coil spacing — the impedance values are computed for each lift-off value increment as though the coils were fixed in position. To compensate for changes in the conductor stack thickness, an additional amount of lift-off is added to the value on the driver coil side of the stack. (2) Coil-specimen contact — impedance values for various conductor thicknesses are computed as though the coils (or probes) remained in contact with the conductor stack. (If a lift-off increment is included with this option, the driver coil-to-stack spacing is held constant and equal to the specified lift-off as conductor thickness varies.)

The readings may be computed as "absolute" or "normalized." The normalized mode is used most since normalized results are most easily compared to experimental measurements. In the normalized mode each computed magnitude of V_{out} is divided by the magnitude computed for a specific air gap, which is usually set equal to the maximum conductor stack thickness. Each computed phase of V_{out} has subtracted from it the phase shift computed for the same air gap. When normalization is used the initializing values inserted for the input voltages and amplifier gains become arbitrary because they cancel out in the division.

MULTRU — SAMPLE OUTPUTS

Standard Headings — No Table Selection

```
PROGRAM MULTRU DATE   9/ 7/78 TIME   13:53:18
COILS-DRIVER, PICKUP: 150B  150B
 CONSTANT COIL SPACING
ELAPSED TIME    204.35
NORMALIZED MAGNITUDES AND PHASES
```

Property Set Table (Selection 1, Line 137)

PR.SET	RG NO	LAY TH	RESTVY	PERM
1	2	5.5000E-02	3.9500E+00	1.000
2	2	7.1000E-02	3.9500E+00	1.000
3	2	7.7000E-02	3.9500E+00	1.000
4	2	5.2000E-02	5.0300E+00	1.000
5	2	6.6000E-02	5.0300E+00	1.000
6	2	7.6000E-02	5.0300E+00	1.000
7	2	5.4000E-02	6.0700E+00	1.000
8	2	6.7000E-02	6.0700E+00	1.000
9	2	7.2000E-02	6.0700E+00	1.000
10	2	7.7000E-02	1.0000E+10	1.000

Output Voltage Table (Selection 4, Line 137)

FREQUENCY 2.00000E+03 GAIN 3.50000E+02

PROP SET	LFT OF	0.0750	0.1000
1	MAG	0.7531	0.7508
	PHA	-39.989	-40.195
2	MAG	0.6921	0.6894
	PHA	-48.306	-48.532
3	MAG	0.6716	0.6689
	PHA	-51.310	-51.543
4	MAG	0.8203	0.8183
	PHA	-32.242	-32.428
5	MAG	0.7721	0.7699
	PHA	-38.708	-38.914
6	MAG	0.7410	0.7385
	PHA	-43.105	-43.323
7	MAG	0.8500	0.8482
	PHA	-28.831	-29.006
8	MAG	0.8115	0.8094
	PHA	-34.165	-34.359
9	MAG	0.7975	0.7954
	PHA	-36.140	-36.341
10	MAG	5.0802	4.9047
	PHA	89.991	89.991

For sample outputs from selections 2 and 3, line 137, see subroutines PRFCOL (p. 168) and PRFCKT (p. 167), respectively.

Least Squares Fitting

Print of Entire Fit (line 343) — Selection 3, line 398. (Property 3 selected at line 277.)

NP	NL	THIK IN REG	2 CAL THIK	DIFF	DRIFT
1	1	0.05500	0.05501	-0.00001	0.00027
1	2	0.05500	0.05499	0.00001	0.00027
2	1	0.07100	0.07097	0.00003	0.00029
2	2	0.07100	0.07102	-0.00002	0.00029
3	1	0.07700	0.07699	0.00001	0.00030
3	2	0.07700	0.07702	-0.00002	0.00030
4	1	0.05200	0.05200	-0.00000	0.00025
4	2	0.05200	0.05199	0.00001	0.00025
5	1	0.06600	0.06602	-0.00002	0.00026
5	2	0.06600	0.06599	0.00001	0.00026
6	1	0.07600	0.07604	-0.00004	0.00027
6	2	0.07600	0.07596	0.00004	0.00028
7	1	0.05400	0.05397	0.00003	0.00024
7	2	0.05400	0.05403	-0.00003	0.00024
8	1	0.06700	0.06700	-0.00000	0.00025
8	2	0.06700	0.06699	0.00001	0.00025
9	1	0.07200	0.07199	0.00001	0.00026
9	2	0.07200	0.07200	-0.00000	0.00026

Print of Coefficients (lines 380, 388) — Selection 4, line 398.

```
DIFF IN THIK IN REG #  2= 2.0245E-05 DRIFT=  2.6670E-04
COEF(  1)=  8.8687390E-02    3DB5A1BC CONSTANT
COEF(  2)=  2.0504755E-01    3ED1F7FC (LOG M1)1
COEF(  3)=  1.4278069E+00    41B6C260 (LOG M1)2
COEF(  4)= -2.3803741E-02    BBC30010 (LIN P1)1
COEF(  5)= -1.4198927E-04    B494E2F4 (LIN P1)2
COEF(  6)=  2.8382786E-02    3BE88304 (LOG M2)1
COEF(  7)= -1.4312267E-01    BE928EC0 (LOG M2)2
COEF(  8)=  1.6970341E-03    37DE6F04 (LIN P2)1
COEF(  9)=  9.1336296E-06    30993CA4 (LIN P2)2
COEF( 10)=  1.0580194E-01    3DD8AEB0 (LOG M3)1
COEF( 11)=  1.7173082E-02    3B8CAE90 (LOG M3)2
COEF( 12)=  5.6543486E-06    2FBDBA7C (LIN P3)1
COEF( 13)=  3.2359566E-08    288AFBB8 (LIN P3)2
```

This fit corresponds to three frequencies with the following selections: IOFSET = 1; M1, 2 2 0; P1, 1 2 0; M2, 2 2 0; P2, 1 2 0; M3, 2 2 0; P3, 1 2 0. (See interaction Table B2, lines 298–306.)

LISTING AND ANALYSIS OF MULTRU

Local File Alias and Date of Revision

```
C
C     MULTRU        (1 NOVEMBER 1978)
C
```

Description and Key Variable Definition

```
C     THROUGH TRANSMISSION PROGRAM
C     THIS PROGRAM CALCULATES MAGNITUDES AND PHASES FOR ALL NPT SETS
C     OF PROPERTIES(ONE MORE THAN THE NUMBER OF DISTINCT COMBINATIONS
C     OF DIFFERENT RESISTIVITIES,PERMEABILITIES AND THICKNESSES IN THE
C     VARIOUS CONDUCTING LAYERS). THE FREQUENCY CAN BE VARIED OVER NFT
C     VALUES AND THE LIFTOFF VARIED OVER NLT DIFFERENT VALUES.
C
C     GAP=CONDUCTOR-TO-PICKUP SPACING
C     IRDPRM+1=MAX.NO.OF TERMS IN POLYNOMIAL EXPANSION(INCL.CONSTANT)
C     LI=LOGICAL INPUT UNIT (1=TTY)
C     LOU=LOGICAL OUTPUT HARD COPY UNIT
C     LO2=LOGICAL OUTPUT, OPERATOR INTERFACE
C     NCOL=POSITION INDEX OF COIL DATA IN FILE 28
C     NFT=NUMBER OF FREQUENCIES
C     NLT=NUMBER OF LIFTOFF VALUES
C     NOPT=INDEX OF COIL SPACING OPTION
C     NORM =1,VALUES WILL BE NORMALIZED TO VALUES OF PROPERTY NPT,
C        OTHERWISE THE MAGNITUDES AND PHASES WILL BE ABSOLUTE.
C     NPRINT=PRINT AND TRANSFER INDEX
C     NPT=1+NUMBER OF DISTINCT SETS OF PROPERTIES
C     NRT=NUMBER OF CONDUCTORS+2
C     THMAX=SUM OF THE MAXIMUM THICKNESS VALUES FOR ALL OF THE
C                 CONDUCTING LAYERS (NR=2,NRT-1)
C     **THE LAST COLUMNS (NP=NPT) OF THE RHO, U, & TH ARRAYS WILL BE
C     FILLED AS FOLLOWS:
C        RHO(NR,NPT+1)= VALUE FOR AIR (=>1.E10)
C        U(NR,NPT+1)= VALUE FOR AIR (1.)
C        TH(NR,NPT+1)= MAXIMUM THICKNESS OF THE (NR) LAYER
C     ** OPTION ONE (NOPT=1) - CONSTANT COIL SPACING
C     FOR THE (NL) VALUE OF LIFTOFF, L2*(NL-1), COIL-TO-COIL
C     SPACING REMAINS CONSTANT, EQUAL TO THMAX+GAP+L5+L6+L2*(NL-1).
C     A CORRECTION FOR CONDUCTOR THICKNESS CHANGES IS ADDED TO THE
C     LIFTOFF ON THE DRIVER SIDE (L6+L2*(NL-1)).
C     ** OPTION TWO (NOPT=2) - COIL-SPECIMEN CONTACT
C     COIL-TO-COIL SPACING VARIES WITH TOTAL CONDUCTOR THICKNESS
C     AND EQUALS SUM(TH(NR),NR=2,NRT-1)+L5+L6+L2*(NL-1).
C
```

Program Identification

```
      PROGRAM MULTRU
```

Optional Double-Precision Variables

Remove Comment "C" during editing to select.

```
C     REAL*8 READNG,COEF,SUM
```

Real Variables

Conversions from normally integer variables.

```
      REAL L2,L3,L4,L5,L6
```

Fixed-Dimension Arrays

```
      DIMENSION HDR(2), HPU(2), ITIM(3), IDAY(3),PROPTY(6),CKTPAR(8)
      DIMENSION PRICON(4)
```

Listing and Identification of Adjustable Array Dimensions

These arrays may require adjustment for new problems.

```
C     DIMENSION READNG((NPT-1)*NLT+1,IRDPRM+1),POLARY(IRDPRM+1,5)
C     DIMENSION PROP((NPT-1)*NLT),RL1(NLT),WUSR(NRT,NPT,NFT)
C     DIMENSION TMDFT(NFT),PHDFT(NFT),NPOL(6,NFT),COEF(IRDPRM)
C     DIMENSION BETAR(NRT),BETAI(NRT),COSF(NRT),SINF(NRT),XPON(NRT)
C     DIMENSION RHO(NRT,NPT),U(NRT,NPT),TH(NRT,NPT)
C     DIMENSION RES(NRT-2,NPT-1),PERM(NRT-2,NPT-1),THICK(NRT-2,NPT-1)
C     DIMENSION FREQ(NFT),GAIN(NFT)
C     DIMENSION GAMAR(NPT),GAMAI(NPT),GAMADR(NPT),GAMADI(NPT)
C     DIMENSION GAMAPR(NPT),GAMAPI(NPT)
C     DIMENSION TMAG(NLT,NPT,NFT),PHASE(NLT,NPT,NFT)
C     DIMENSION TMUTRE(NLT,NPT,NFT),TMUTIM(NLT,NPT,NFT)
C     DIMENSION DRIVRE(NLT,NPT,NFT),DRIVIM(NLT,NPT,NFT)
C     DIMENSION PICKRE(NLT,NPT,NFT),PICKIM(NLT,NPT,NFT)
```

Adjustable Arrays

```
      DIMENSION READNG(19,16),POLARY(16,5)
      DIMENSION PROP(18),RL1(2),WUSR(3,10,3)
      DIMENSION TMDFT(3),PHDFT(3),NPOL(6,3),COEF(15)
```

```
      DIMENSION BETAR(3),BETAI(3),COSF(3),SINF(3),XPON(3)
      DIMENSION RHO(3,10),U(3,10),TH(3,10)
      DIMENSION RES(1,9),PERM(1,9),THICK(1,9)
      DIMENSION FREQ(3),GAIN(3)
      DIMENSION GAMAR(10),GAMAI(10),GAMADR(10),GAMADI(10)
      DIMENSION GAMAPR(10),GAMAPI(10)
      DIMENSION TMAG(2,10,3),PHASE(2,10,3)
      DIMENSION TMUTRE(2,10,3),TMUTIM(2,10,3)
      DIMENSION DRIVRE(2,10,3),DRIVIM(2,10,3)
      DIMENSION PICKRE(2,10,3),PICKIM(2,10,3)
```

Common Blocks

B1 — Powers of integration variable transferred to and from subroutines computing impedance integrals.

B2, B6 — Coil and circuit parameters transferred to and from subroutines computing instrument response and electrical characteristics.

```
      COMMON /B1/X,XX,XXXX
      COMMON /B2/R1,R2,R3,R4,L3,L4,RBAR,VO,ZLDR,ZLPU,
     *TNDR,TNPU,RDCDR,RDCPU,RO,R9,CAPDR,CAPPU
      COMMON /B6/ L2,L5,L6
```

RAD File Identification

28 — Coil Data File

```
      DEFINE FILE 28 (80, 32, U, NCOL)
```

Data Initialization

TWOPI = 2π; RAD = degrees/radian = $180/\pi$.

PROPTY and CKTPAR contain repeated title blocks for printouts.

See Table B1 for editorial changes.

```
      DATA TWOPI, RAD /6.28318531, 57.2957795/
      DATA PROPTY/4HREST,4HPERM,4HTHIK,4HL.O.,4HDFDF,4HDFSI/
      DATA CKTPAR/4HDR.T,4HPU.T,4HDR.R,4HPU.R,4HSE.R,4HSH.R,
     *4HDR.C,4HPU.C/
      DATA NPT/10/,NRT/3/,NLT/2/,NFT/3/,NORM/1/,THMAX/.077/,GAP/.002/
      DATA RES/3*3.95,3*5.03,3*6.07/
      DATA PERM/9*1./
      DATA THICK/.055,.071,.077,.052,.066,.076,.054,.067,.072/
      DATA NOPT/1/,LOU/6/,LI/1/,NCOL/1/,IRDPRM/15/,LO2/5/
      DATA FREQ/2.E3,2.E4,1.E5/,GAIN/350.,100.,20./
```

Property Array Expansion

Arrays storing separate step values are expanded to form arrays that contain all possible unique property sets used in the impedance integral computations. Large arrays contain values for air not stored in smaller arrays.

```
C
C   THE PROPERTIES ARE SET AND COUNTED.
C
      NPT1=NPT-1
      NRT1=NRT-1
      NRT2=NRT-2
      DO 5 NP=1,NPT1
      DO 5 NR=1,NRT2
      RHO(1,NP)=1.E10
      RHO(NRT,NP)=1.E10
      U(1,NP)=1.
      U(NRT,NP)=1.
      TH(1,NP)=1.E10
      TH(NRT,NP)=1.E10
C   THE CONDUCTOR PROPERTIES ARE SET.
      RHO(NR+1,NP)=RES(NR,NP)
      U(NR+1,NP)=PERM(NR,NP)
      TH(NR+1,NP)=THICK(NR,NP)
    5 CONTINUE
C   THE COIL IN AIR PROPERTIES ARE SET.
      DO 7 I=1,NRT
      TH(I,NPT)=0.
      RHO(I,NPT)=1.E10
      U(I,NPT)=1.
    7 CONTINUE
      TH(1,NPT)=1.E10
      TH(2,NPT)=THMAX
      TH(NRT,NPT)=1.E10
```

File Location on Disc Storage

Search is initiated early to have file ready when needed.

```
C
      FIND (28'NCOL)
C
```

Program Identification, Date, and Time

Elapsed time clock started for integral computations.

```
C       THE DATE AND TIME ARE PRINTED
C
      CALL TIME (ITIM)
      CALL DATE (IDAY)
      CALL TIMER(STARTT)
```

```
      IT = IDAY(1) - 1900
      IDAY(1) = IDAY(2)
      IDAY(2) = IDAY(3)
      IDAY(3) = IT
      WRITE (LOU,10) IDAY, ITIM
   10 FORMAT ('OPROGRAM MULTRU DATE   ', 2(I2,1H/), I2,
     1  ' TIME   ', 2(I2,1H:), I2,/)
```

Printed Table Selection

See Table B2 for queries and responses. See pp. 77–78, for sample outputs.

```
C
C     *** SELECT PRINTED TABLES ***
C
      WRITE(LO2,15)
   15 FORMAT(' SELECT PRINTED TABLES: ENTER 1 TO PRINT, 0 TO OMIT'/
     1' 1-PROPERTY SETS; 2-COIL DIMS; 3-COIL/CKT ELECT VALUES;'/
     2' 4-V/OUT MAG & PHA FOR EA PROPERTY SET')
      READ(LI,*) (PRICON(I), I=1,4)
```

Description and Initialization of Circuit Parameters and Lift-off

```
C
C     THE CIRCUIT PARAMETERS ARE NOW GIVEN.
C     L2=NORMALIZED LIFT-OFF INCREMENT.
C     R0=DRIVER SERIES RESISTANCE(OHMS)
C     R9=PICKUP SHUNT RESISTANCE(OHMS)
C     CAPDR=DRIVER SHUNT CAPACITANCE(FARADS)
C     CAPPU=PICKUP SHUNT CAPACITANCE(FARADS)
C     V0=DRIVER OUTPUT VOLTAGE(RMS VOLTS)
C
      L2=.025
      R0=2000.
      R9=1.0E6
      CAPDR=8.47E-11
      CAPPU=8.45E-11
      V0=10.
```

Optional Printout of Conductor Property Sets

See pp. 77–78 for sample output.

```
C
C     EACH SET OF PROPERTIES FOR EACH REGION IS NOW PRINTED OUT
C
      IF (PRICON(1).EQ.0) GO TO 85
      WRITE (LOU,20)
   20 FORMAT(' PR.SET  RG NO  LAY TH      RESTVY       PERM')
      WRITE(LOU,30)
   30 FORMAT (1X)
      DO  50 NP=1,NPT
      DO  50 NR=2,NRT1
```

```
      IF(NR.EQ.2)WRITE(LOU,60)NP,NR,TH(NR,NP),RHO(NR,NP),U(NR,NP)
      IF(NR.GT.2)WRITE(LOU,70)NR,TH(NR,NP),RHO(NR,NP),U(NR,NP)
   50 CONTINUE
   60 FORMAT(1X,2(I5),2(1PE12.4),0PF9.3)
   70 FORMAT(1X,5X,I5,2(1PE12.4),0PF9.3)
   80 FORMAT('+',44X,2(F9.4),1PE12.4)
```

Coil Data Selection or Insertion

See p. 180 for subroutine TCOFIL. Coil data returned in common blocks B2 and B6.

```
C
C     *** COIL DATA READ FROM FILE 28 OR MANUALLY INSERTED ***
C
   85 CALL TCOFIL (LI,LOU)
```

Optional Printout of Coil Dimensions

All dimensions normalized except RBAR (= $\bar{r}$ or $R5$ in equations). GAP is added to the insulation thickness of (L5 of the pickup coil to simulate the situation likely to be found in most inspections; some spacing on both sides of the conductor stack).

```
C
C     THE COIL VALUES ARE NOW PRINTED OUT
C
      IF (NOPT.EQ.1) L5=L5+GAP
      IF (PRICON(2).NE.0) CALL PRFCOL(LOU)
```

Computation of Impedance Integrals

Initial limits and increments of integration are specified: from B1 to B2 in steps of S1; initially B1 = 0 and B2 = 5; both are changed in steps S2.

$$\omega\mu\sigma\bar{r}^2 \text{ (dimensionless)} = 2\pi(\text{rad/s-Hz})\cdot 4\pi \times 10^{-7}(\text{H/m})\cdot 10^8(\mu\Omega\text{-cm}/\Omega\text{-m})$$

$$\cdot(0.0254)^2(\text{m}^2/\text{in.}^2)\cdot\mu_{rel}\cdot\bar{r}^2(\text{in.}^2)$$

$$\cdot f(\text{Hz})/\rho(\mu\Omega\text{-cm})$$

$$[0.50939 \ldots = 80\pi^2(0.0254)^2] \ .$$

Arrays to hold sums corresponding to real and imaginary parts of the impedance integrals are initialized to zero. Constants corresponding to the air impedance ($\omega \times$ inductance) are initialized to zero. The array containing $\omega\mu\sigma\bar{r}^2$, WUSR(NR,NP,NF), is constructed.

```
C
C             THE INTEGRATION IS PERFORMED BY THE MIDPOINT METHOD,
C          EVALUATING AT THE CENTER OF THE INTERVAL; FOR X LARGE
C          THE INTEGRAL CONVERGES RAPIDLY,SO LARGER INTERVALS
C          ARE TAKEN.
C             IN THE INTEGRATION TMUT, DRIVER, PICKUP, AIR1, AND
C          AIR2 ARE CALCULATED.
C
        RRBAR=1./RBAR
     90 S1 = 0.01
        S2 = 5.0
        B1 = 0.0
        B2 = S2
C
C          INITIALIZE ALL SUMS TO ZERO AND CALCULATE THE VALUES OF
C          WUSR(NR,NP,NF).
C
        DO 130 NF=1,NFT
        RCON=.509397921*RBAR*RBAR*FREQ(NF)
        DO  120 NP=1,NPT
        DO 100 NL=1,NLT
            TMUTRE(NL,NP,NF)=0.
            TMUTIM(NL,NP,NF)=0.
            DRIVRE(NL,NP,NF)=0.
            DRIVIM(NL,NP,NF)=0.
            PICKRE(NL,NP,NF)=0.
            PICKIM(NL,NP,NF)=0.
    100 CONTINUE
        DO 110 NR=1,NRT
            WUSR(NR,NP,NF)=RCON*U(NR,NP)/RHO(NR,NP)
    110 CONTINUE
    120 CONTINUE
    130 CONTINUE
        AIR1=0.
        AIR2=0.
C
```

Integration step sizes vary from 0.01 to 0.5 as X (i.e., α) increases. Subroutine TRCOLM computes and returns the six impedance array values and two constants for each lift off value, frequency, and conductor property set. Elapsed time is measured and written out at the end of the integrations. (See p. 182 for TRCOLM.)

```
        IF(NOPT.EQ.1) WRITE(LOU,132)
        IF(NOPT.EQ.2) WRITE(LOU,135)
    132 FORMAT(1X,' CONSTANT COIL SPACING')
    135 FORMAT(1X,' COIL-SPECIMEN CONTACT')
C
    140 I1 = (B2 - B1)/S1
        X = B1 - S1*0.5
        DO 150 I=1,I1
          X = X + S1
```

```
          XX = X*X
          XXXX = XX*XX
        CALL TRCOLM(S1,NLT,NPT,NRT,NFT,RL1,GAMAR,GAMAI
       1,GAMADR,GAMADI,TMUTRE,TMUTIM,DRIVRE,DRIVIM,PICKRE,PICKIM
       2,GAMAPR,GAMAPI,BETAR,BETAI,COSF,SINF,XPON
       3,WUSR,U,TH,RRBAR,FREQ,AIR1,AIR2,THMAX,NOPT)
C
  150   CONTINUE
        B1 = B2
        B2 = B2 + S2
        S1 = 0.05
        IF (X .LT. 9.) GO TO 140
        S1 = 0.1
        IF (X .LT. 29.) GO TO 140
        S1 = 0.2
        IF (X .LT. 39.) GO TO 140
        S1 = 0.5
        IF (X .LT. 79.) GO TO 140
        CALL TIMER(STOPT)
        ELAPST=STOPT-STARTT
        WRITE(LOU,160)ELAPST
  160   FORMAT( ' ELAPSED TIME ',F9.2)
```

Magnitude and Phase of the Output Voltages

Subroutine TRUCIR computes and returns the magnitudes and phases of output voltages corresponding to each coil impedance value computed above by using the equivalent of Eq. (14). (See p. 184 for TRUCIR.) If the normalization option is selected TRUCIR performs it, and statement 167 is printed.

If selected electrical properties of the coils and circuit will be printed by subroutine PRFCKT. (See p. 167 for both sample output and PRFCKT.)

If selected initially the tabulations of magnitudes and phases will be printed by frequencies with use of subroutine PRFVLT. (See p. 70 for sample output and p. 169 for PRFVLT.)

```
C
C         NEXT THE PROPERTIES OF THE COILS ARE DETERMINED AND PRINTED
C
  165 CALL TRUCIR(TMAG,PHASE,TWOPI,RAD,TMUTRE,TMUTIM,DRIVRE
     1,DRIVIM,PICKRE,PICKIM,FREQ
     2,U,WUSR,GAIN,AIR1,AIR2,NLT,NRT,NPT,NFT,NORM)
C
C       THE CIRCUIT PARAMETERS ARE PRINTED OUT FOR THE REFLECTION
C       COIL CIRCUIT.
C
        IF(NORM.EQ.1)WRITE(LOU,167)
  167 FORMAT(' NORMALIZED MAGNITUDES AND PHASES')
```

```
257          IF (PRICON(3).NE.0) CALL PRECKT(LOU)
258   C
259   C         THE VOLTAGE MAGNITUDE,PHASE,AND SHIFT ARE PRINTED FOR
260   C         THE VARIOUS LIFT-OFF VALUES
261   C
262          IF (PRICON(4).NE.0) CALL PTRVLT(LOU,TMAG,PHASE,NRT,NPT,NFT,NLT,RL1
263         1,FREQ,GAIN)
```

Least Squares Fitting of Output Voltages to Properties

MSET1 is the array row size required to hold the inputs and results of the fit. The program branches to line 394 (p. 91) to select the desired step in the fitting process; several steps require repetition.

```
264   C
265   C      LEAST SQUARES DESIGN SECTION.
266   C
267   C      SELECT PROPERTY TO BE FITTED AND SET UP PROPERTY ARRAY.
268   C
269          MSET=NLT*NPT1
270          MSET1=MSET+1
271          GO TO 640
```

To perform a fit it is always necessary to select the property to be fitted and the region for which it occurs. (See Table B2 for interactive responses.)

```
272      170 WRITE(LO2,180)
273          WRITE(LO2,190)
274      180 FORMAT(' SELECT NUMBER OF THE PROPERTY TO BE FITTED:'/
275         *' 1=RHO 2=MU 3=THICKNESS 4=LIFT-OFF')
276      190 FORMAT(' SELECT REGION TO BE FITTED. LOWEST IS NUMBER 1.')
277          READ(LI,*)NPROP,NREG
```

The sequence of values for the selected property is loaded into a special array for transfer to the fitting subroutine (ALSQS). (The PROP array holds a single row corresponding to one region of the large property arrays, RHO, etc.)

```
278          DO 210 NP=1,NPT1
279          DO 200 NL=1,NLT
280          M=(NP-1)*NLT+NL
281          IF(NPROP.EQ.1)PROP(M)=RHO(NREG,NP)
282          IF(NPROP.EQ.2)PROP(M)=U(NREG,NP)
283          IF(NPROP.EQ.3)PROP(M)=TH(NREG,NP)
284          IF(NPROP.EQ.4)PROP(M)=(L6+FLOAT(NL-1)*L2)*RBAR
285      200 CONTINUE
286      210 CONTINUE
```

The fitting function is selected. The NPOL array (see Table 3) carries the parameters describing the nonconstant functions and their powers. The term IOFSET determines whether a constant is added to the fitting function. If only a function change is desired selective execution branches to statement 220. (See Table B2.)

```
  220 WRITE(LO2,230)
  230 FORMAT(' TYPE 1 IF THERE IS OFFSET, 0 IF NO OFFSET: ')
      READ(LI,*) IOFSET
      IRDPR=IOFSET
      WRITE(LO2,240)
  240 FORMAT(' SELECT THE NUMBER OF THE FUNCTION TYPE, POLYNOMIAL',
     1' DEGREE, & # OF '/,
     2' CROSS TERMS FOR EACH MAGNITUDE & PHASE')
      WRITE(LO2,250)
  250 FORMAT(' FUNCTION TYPE:1=LINEAR 2=LOG 3=EXP 4=INV ')
  300 WRITE(LO2,30)
      WRITE(LO2,310)
  310 FORMAT(25X,'FCTN POL # CROSS'/,25X'TYPE DEG TERMS')
      DO 370 NF=1,NFT
      DO 360 NC=1,2
      NCC=NC*3
      NCP=NCC-1
      NCF=NCP-1
      IF(NC.EQ.1) WRITE(LO2,320) FREQ(NF)
      IF(NC.EQ.2) WRITE(LO2,330) FREQ(NF)
  320 FORMAT(' MAG AT ',1PE12.6,' HZ    ')
  330 FORMAT(' PHA AT ',1PE12.6,' HZ    ')
      READ(LI,*)NPOL(NCF,NF),NPOL(NCP,NF),NPOL(NCC,NF)
      NPOL(3,NF)=0
      IRDPR=IRDPR+NPOL(NCP,NF)+NPOL(NCC,NF)
  360 CONTINUE
```

The size of the selected function is checked to assure that it does not contain more terms than the subroutine ALSQS can handle. If it does an error message is printed (see Table B2).

```
  370 CONTINUE
      IRDPR1=IRDPR+1
      IF(IRDPRM.LT.IRDPR)WRITE(LO2,400)
      IF (IRDPRM.LT.IRDPR)GO TO 640
```

Subroutine POLYTYP generates a "printable" representation of the selected fitting function. (See p. 78 for sample output.)

```
  400 FORMAT(' ERROR: # OF TERMS IN POLARY EXCEEDS DIMENSION')
      JROW=IRDPRM+1
```

When functions of the computed output voltages (e.g. log, exponential, powers >1) are used, the function terms are generated by subroutine RDGEXP before the fit is performed. Small arrays TMDFT and PHDFT are used to store values of TMAG and PHASE to which known drift errors have been added. (See p. 171 for RDGEXP.)

```
          CALL POLTYP(POLARY,JROW,IRDPR,NPOL,6,NFT,2,IOFSET,LO2)
C
C          EXPAND THE RAW READINGS INTO IRDPR READINGS.
C
      450 DO 460 NF=1,NFT
          TMDFT(NF)=0.
          PHDFT(NF)=0.
      460 CONTINUE
          DO 480 NP=1,NPT1
          DO 470 NL=1,NLT
          M=(NP-1)*NLT+NL
          CALL RDGEXP(READNG,TMAG,PHASE,NPOL,IOFSET,TMDFT,PHDFT,MSET1
         1,IRDPR1,M,NFT,NL,NLT,NP,NPT)
      470 CONTINUE
```

Subroutine ALSQS performs the least squares fit, returning the fit coefficients in the COEF array and the property values computed from the fit in the IRDPR1 column of the READNG array. (See p. 151 for ALSQS.)

```
      480 CONTINUE
C
C          DO THE LEAST SQUARES FIT OF THE READINGS TO THE PROPERTIES.
C
```

Tests of Fit Quality

These tests are described earlier. (See p. 77 for sample output.)

```
          CALL ALSQS(READNG,PROP,COEF,RSQS,MSET,IRDPR,MSET1)
C
C          CALCULATE THE DIFFERENCES IN THE FIT AND THE MAXIMUM DRIFTS.
C
      490 SSDRIF=0.
          SSDIFF=0.
          IF(NPRINT.EQ.3)WRITE(LOU,500)PROPTY(NPROP),NREG,PROPTY(NPROP)
      500 FORMAT('0 NP NL ',A4,' IN REG',I3,' CAL ',A4,7X,'DIFF',7X,'DRIFT')
          DO 580 NP=1,NPT1
          DO 570 NL=1,NLT
          M=(NP-1)*NLT+NL
          DRIFT=0.
          DO 540 NF=1,NFT
          DO 530 NC=1,2
C
C          ONE MAGNITUDE OR PHASE DRIFT IS SET ON AT A TIME.
C
```

```
354          IF(NC.EQ.1)TMDFT(NF)=.0001*TMAG(NL,NP,NF)
355          IF(NC.EQ.2)PHDFT(NF)=.01
356          CALL RDGEXP(READNG,TMAG,PHASE,NPOL,IOFSET,TMDFT,PHDFT,MSET1
357         1,IRDPR1,M,NFT,NL,NLT,NP,NPT)
358   C
359   C      THE POLYNOMIAL IS CALCULATED
360   C
361          SUM=0.
362          DO 520 IR=1,IRDPR
363          SUM=SUM+COEF(IR)*READNG(M,IR)
364      520 CONTINUE
365          DRIFT=DRIFT+ABS(READNG(M,IRDPR1)-SUM)
366          TMDFT(NF)=0.
367          PHDFT(NF)=0.
368      530 CONTINUE
369      540 CONTINUE
370          DIFF=PROP(M)-READNG(M,IRDPR1)
371          SSDIFF=SSDIFF+DIFF*DIFF
372          SSDRIF=SSDRIF+DRIFT*DRIFT
373          IF(NPRINT.NE.3)GO TO 570
374          WRITE(LOU,560)NP,NL,PROP(M),READNG(M,IRDPR1),DIFF,DRIFT
375      560 FORMAT(1X,2I3,4F12.5)
376      570 CONTINUE
377      580 CONTINUE
378          SDRIF=SQRT(SSDRIF/FLOAT(MSET))
```

Printout of Fit Quality Parameters

```
379          SDIFF=SQRT(SSDIFF/FLOAT(MSET))
380          WRITE(LO2,590)PROPTY(NPROP),NREG,SDIFF,SDRIF
381      590 FORMAT(' DIFF IN ',A4,' IN REG #',I3,'=',1PE11.4,
```

Optional Printout of Fit Function and Coefficients

See Table B2 for selection.

```
382         1' DRIFT= ',1PE11.4)
383      600 IF (NPRINT.NE.4)GO TO 640
384          WRITE(LOU,30)
385          WRITE(LOU,590) PROPTY(NPROP), NREG, SDIFF, SDRIF
386          DO 630 I=1,IRDPR
387          CALL CON8(COEF(I),IHEX)
388          WRITE(LOU,620)I,COEF(I),IHEX,(POLARY(I,J),J=1,5)
389      620 FORMAT(' COEF(',I3,')=',1PE15.7,4X,Z8,1X,5A4)
```

Selection of Next Action

See Table B2 for query and responses. This sequence is initially executed from a jump at line 271 and, after later selected sequences, from jumps at lines 316 (conditional), 383, and 401.

```
390      630 CONTINUE
391   C
```

```
C       THE NEXT ACTION IS SELECTED.
C
    640 WRITE(LO2,650)
    650 FORMAT(' TYPE NUMBER OF NEXT ACTION TO BE TAKEN:'/
       1' 1=CHG FUN/POL TYP 2=FIT PROP 3=PRT ENTIRE FIT 4=PRT COEF '/
       2' 5=CAL COIL/CKT DRIFT 6=CHG COIL/CKT VAL 7=CHG W PRT 8=STOP')
        READ(LI,*) NPRINT
```

Noninstalled Option for Circuit Drift

Option 5 is not implemented at present and returns control to statement 640.

```
        GO TO(220,170,490,600,660,700,700,800),NPRINT
    660 CONTINUE
        GO TO 640
```

Change of Circuit or Coil Parameters

Options 6 and 7 call subroutine CHRFPM (see p. 159), which can be used to change circuit parameters. Changing coils with CHRFPM does not work because the main program jumps back to TCOFIL (see p. 180) at statement 85.

```
    700 CALL CHRFPM(LOU,LI,INTG,NPRINT)
        IF(INTG.EQ.1)GO TO 85
```

Stop

Selection of option 8 (statement 800) stops execution.

```
        GO TO 165
    800 STOP JOB
```

A partial printout of a sample run is given below.

```
PROGRAM MULTRU DATE   9/ 7/78 TIME  13:53:18

PR.SET  RG NO  LAY TH      RESTVY       PERM

    1     2   5.5000E-02  3.9500E+00    1.000
    2     2   7.1000E-02  3.9500E+00    1.000
    3     2   7.7000E-02  3.9500E+00    1.000
    4     2   5.2000E-02  5.0300E+00    1.000
    5     2   6.6000E-02  5.0300E+00    1.000
    6     2   7.6000E-02  5.0300E+00    1.000
    7     2   5.4000E-02  6.0700E+00    1.000
    8     2   6.7000E-02  6.0700E+00    1.000
    9     2   7.2000E-02  6.0700E+00    1.000
   10     2   7.7000E-02  1.0000E+10    1.000
```

COILS-DRIVER, PICKUP: 150B 150B
MEAN RADIUS 0.15000 INCHES

COIL	INN. RAD	OUT. RAD	LENGTH	TURNS EA	O L.D./RCES
DRIVER	0.7500	1.2500	0.3600	513.0	0.0750
PICK-UP	0.7500	1.2500	0.3600	513.0	0.0770

CONSTANT COIL SPACING
ELAPSED TIME 204.35
NORMALIZED MAGNITUDES AND PHASES

	SER/SHT RES	COIL DC RES	SHUNT CAP.	COIL INDUCT	RES.FREQ	DRIV. VOLT
DVR CKT	2000.000	67.300	8.4700E-11	2.2136E-03	2.3095E+06	10.0000
PICK CKT	1000000.000	67.300	8.4500E-11	2.2136E-03	2.3122E+06	

FREQUENCY 2.00000E+03 GAIN 3.50000E+02

PROP SET	LFT OF	0.0750	0.1000
1	MAG	0.7531	0.7508
	PHA	-39.989	-40.195
2	MAG	0.6921	0.6894
	PHA	-48.306	-48.532
3	MAG	0.6716	0.6689
	PHA	-51.310	-51.543
4	MAG	0.8203	0.8183
	PHA	-32.242	-32.428

(Blocks 5–10 left out for brevity.)

NP	NL	REST IN REG	2 CAL REST	DIFF	DRIFT
1	1	3.95000	3.94128	0.00872	0.04644
1	2	3.95000	3.95733	-0.00733	0.04639
2	1	3.95000	3.93946	0.01054	0.05697
2	2	3.95000	3.95689	-0.00689	0.05726
3	1	3.95000	3.95253	-0.00253	0.06072
3	2	3.95000	3.94979	0.00021	0.06124
4	1	5.03000	5.02888	0.00111	0.03627
4	2	5.03000	5.02893	0.00107	0.03660
5	1	5.03000	5.03568	-0.00568	0.04439
5	2	5.03000	5.03494	-0.00494	0.04473
6	1	5.03000	5.03515	-0.00516	0.05014
6	2	5.03000	5.02251	0.00748	0.05041
7	1	6.07000	6.07165	-0.00165	0.03196
7	2	6.07000	6.06990	0.00010	0.03212
8	1	6.07000	6.07067	-0.00067	0.03871
8	2	6.07000	6.06635	0.00364	0.03889
9	1	6.07000	6.07256	-0.00256	0.04113
9	2	6.07000	6.06559	0.00441	0.04147

DIFF IN REST IN REG # 2= 5.1580E-03 DRIFT= 4.6203E-02

```
COEF(  1)=  2.8454575E+01     45E3A2F8 CONSTANT
COEF(  2)= -1.4015057E+02     C88C268C (LOG M1)1
COEF(  3)=  1.1102585E+02     47DE0D3C (LOG M1)2
COEF(  4)= -2.2013702E+00     C28CE340 (LIN P1)1
COEF(  5)= -5.0143704E-02     BCCD637C (LIN P1)2
COEF(  6)=  4.4451462E+01     46B1CE4C (LOG M2)1
COEF(  7)= -2.5134802E-01     BF80B0B0 (LIN P2)1
COEF(  8)= -4.5070046E-05     B28D0998 (LIN P2)2
COEF(  9)=  1.6449745E+01     45839914 (LOG M3)1
COEF( 10)=  2.3310022E+00     42952F24 (LOG M3)2
COEF( 11)=  6.9642346E-04     36B69030 (LIN P3)1
COEF( 12)=  3.9434108E-06     2F8451A4 (LIN P3)2
```

NP	NL	THIK IN REG 2	CAL THIK	DIFF	DRIFT
1	1	0.05500	0.05501	-0.00001	0.00027
1	2	0.05500	0.05499	0.00001	0.00027
2	1	0.07100	0.07097	0.00003	0.00029
2	2	0.07100	0.07102	-0.00002	0.00029
3	1	0.07700	0.07699	0.00001	0.00030
3	2	0.07700	0.07702	-0.00002	0.00030
4	1	0.05200	0.05200	-0.00000	0.00025
4	2	0.05200	0.05199	0.00001	0.00025
5	1	0.06600	0.06602	-0.00002	0.00026
5	2	0.06600	0.06599	0.00001	0.00026
6	1	0.07600	0.07604	-0.00004	0.00027
6	2	0.07600	0.07596	0.00004	0.00028
7	1	0.05400	0.05397	0.00003	0.00024
7	2	0.05400	0.05403	-0.00003	0.00024
8	1	0.06700	0.06700	-0.00000	0.00025
8	2	0.06700	0.06699	0.00001	0.00025
9	1	0.07200	0.07199	0.00001	0.00026
9	2	0.07200	0.07200	-0.00000	0.00026

```
DIFF IN THIK IN REG #  2= 2.0245E-05 DRIFT=  2.6670E-04
COEF(  1)=  8.8687390E-02     3DB5A1BC CONSTANT
COEF(  2)=  2.0504755E-01     3ED1F7FC (LOG M1)1
COEF(  3)=  1.4278069E+00     41B6C260 (LOG M1)2
COEF(  4)= -2.3803741E-02     BBC30010 (LIN P1)1
COEF(  5)= -1.4198927E-04     B494E2F4 (LIN P1)2
COEF(  6)=  2.8382786E-02     3BE88304 (LOG M2)1
COEF(  7)= -1.4312267E-01     BE928EC0 (LOG M2)2
COEF(  8)=  1.6970341E-03     37DE6F04 (LIN P2)1
COEF(  9)=  9.1336296E-06     30993CA4 (LIN P2)2
COEF( 10)=  1.0580194E-01     3DD8AEB0 (LOG M3)1
COEF( 11)=  1.7173082E-02     3B8CAE90 (LOG M3)2
COEF( 12)=  5.6543486E-06     2FBDBA7C (LIN P3)1
COEF( 13)=  3.2359566E-08     288AFBB8 (LIN P3)2
```

NP	NL	L.O. IN REG 3	CAL L.O.	DIFF	DRIFT
1	1	0.01125	0.01172	-0.00047	0.00054
1	2	0.01500	0.01380	0.00120	0.00053
2	1	0.01125	0.01156	-0.00031	0.00053
2	2	0.01500	0.01491	0.00009	0.00053
3	1	0.01125	0.01101	0.00024	0.00053
3	2	0.01500	0.01518	-0.00018	0.00053
4	1	0.01125	0.01191	-0.00066	0.00053
4	2	0.01500	0.01412	0.00088	0.00053
5	1	0.01125	0.01260	-0.00135	0.00053
5	2	0.01500	0.01577	-0.00077	0.00053
6	1	0.01125	0.01074	0.00051	0.00053
6	2	0.01500	0.01507	-0.00007	0.00053
7	1	0.01125	0.01197	-0.00072	0.00053
7	2	0.01500	0.01441	0.00059	0.00053
8	1	0.01125	0.01134	-0.00009	0.00053
8	2	0.01500	0.01527	-0.00027	0.00053
9	1	0.01125	0.01044	0.00081	0.00053
9	2	0.01500	0.01444	0.00056	0.00053

APPENDIX C

PROGRAM RFLRDG

The functions of this program are described in earlier sections (pp. 3, 32). Table C1 describes editorial changes required before running the program. Note that the RAD file size must be adjusted separately from the program and must be compatible with the program storage requirements. Table C2 contains step-by-step instructions for execution of the program. The final section explains the construction and operation of the program.

Table C1. Editorial Changes Preparatory to Running RFLRDG

Line(s)	Action
40	Dimensions of NARRAY and VOLTS (NCHS, line 24) must be compatible with numbers and assignment of parallel input lines of the minicomputer data acquisition interface.
41–51	Adjust array dimensions according to guide, lines 25–38. (See p. 34 for critical size parameters of READNG array.) Except for READNG the array size guide gives minimum array dimensions.
60	Set file size according to guide, lines 53–58.
62	Input or output device parameters will depend on resource identification and allocation in the computer system used. The value of NCHS must match value in array dimensions (line 40).
63	Stored title — no change required.

Table C2. Interactive Program Queries, Displays, and Operator Responses Occurring During Execution of RLFRDG

Line	QUERY[a] or DISPLAY or Response
73	CALIBRATION DATA
82	DATE (numeric) TIME (numeric)
84	TYPE IN THE FOLLOWING DATA AS REQUESTED
	(Preparatory instruction — no direct response required.)
90	PROBE #:
	Enter six-character alphanumeric[b] for new probe in use or "OLD" plus three blanks (spaces) if no change is desired.
***	If "OLD" is entered queries from here to line 190 are skipped.
95	SERIAL #:
	Enter integer serial number of probe in use.
98	DRIVER SERIES RESISTANCE:
	Enter free format number for driver coil power oscillator series resistance (R_0) in ohms.
101	DRIVER SHUNT CAPACITANCE:
	Enter free format number for C_6 in farads.
104	PICK-UP SHUNT RESISTANCE:
	Enter free format number for pickup amplifier input resistance (R_9) in ohms.
107	PICK-UP SHUNT CAP:
	Enter free format number for C_7 in farads.
110	CABLE I.D. #:
	Enter six-character alphanumeric probe cable identification number.
113	LENGTH OF CABLE:
	Enter free format number for cable length in inches.

Table C2. (Continued)

Line	QUERY[a] or DISPLAY or Response
116	CAPACITANCE OF CABLE:
	Enter free format number for capacitance of probe cable in farads.
119	EDDY-CURRENT INSTRUMENT #:
	Enter six-character alphanumeric for instrument serial number.
122	POWER OSC I.D.:
	Enter six-character alphanumeric oscillator serial number.
128	NO. OF FREQUENCIES
	Enter integer for number of discrete frequencies to be used (NFT).
131	INPUT THE VALUE OF EACH FREQUENCY SEPARATED BY A SPACE:
	Enter free format numbers for frequencies; separate with a space or comma delimiter.
	The number of entries must equal NFT (line 129).
134	INPUT THE FOLLOWING DATA FOR EACH OF FREQUENCIES:
	FREQUENCY: (number)........(up to ten numbers)
137	PICK-UP AMP (EACH 6 CHARACTERS, NO SPACING):
	Enter one six-character alphanumeric identifier for each frequency (NFT total); no delimiters.
141	PHASE DETECTOR (EACH 6 CHARACTERS, NO SPACING):
	Enter one six-character alphanumeric identifier for each frequency (NFT total); no delimiters.
144	180-DEG SW (OFF/ON) (EACH 3 CHARACTERS, NO SPACING):
	Enter one three-character phase switch position for each frequency (NFT total), "OFF" or "ON"; no delimiters. [b = blank (space).]
151	TYPE # OF LIFT-OFF VALUES:
	Enter integer (NLT).

Table C2. (Continued)

Line	QUERY[a] or DISPLAY or Response
154	TYPE EACH LIFT-OFF VALUE SEPARATED BY A SPACE:
	Enter free format numbers (total of NLT) equal to lift-off values in inches; space delimiters.
157	# OF RESISTIVITY VALUES:
	Enter integer (NRES).
160	TYPE EACH RESISTIVITY VALUE SEPARATED BY A SPACE:
	Enter free format numbers (total NRES) equal to resistivity values in μΩ-cm; space delimiters.
164	TYPE # OF THICKNESS VALUES FOR RES (number)
	Enter integer [NTH(I)].
168	TYPE (integer) THICKNESS VALUES, EACH SEPARATED BY A SPACE, FOR RHO (integer) = (number)
	Enter free format numbers, as many as specified [NTH(I)], equal to thickness values in inches.
172	# OF DEFECTS FOR EACH THICKNESS VALUE:
	(Number) (Number) .... (total of nine, maximum)
	Enter an integer equal to the number of defects in each thickness specimen for each (number) written; space delimiters.
***	The sequence from line 164 through line 172 will be repeated for each resistivity value, total NRES (line 158).
182	FOR RES (number) THICK (number) TYPE DEPTH, RADIUS, AND SIZE DEFECT # (integer):
	Enter free format numbers equal to required lengths in inches and size in cubic inches. (See pp. 35–36 for definitions.)
***	If "OLD" was entered at line 91 (above), the display will now show all of the old data on the instruments and calibration standards that are stored in the RAD file.[c] (No response required.)

Table C2. (Continued)

Line	QUERY[a] or DISPLAY or Response
	If new data were entered control jumps to line 255 below.
	...
233	LIFT-OFF VALUES (number) (number) ... (total of ten, maximum)
	At this point the display pauses to allow the operator time to read it. Entry of any real number, free format, will start the display of the remaining data.
256	GAIN ADJUSTMENT:
258	TYPE SAMPLE #:
	Enter integer assigned to calibration standard for gain adjustment.
261	SET ON SAMPLE
	# (integer) FQ (integer) = (number)...(Repeats NFT times)
268	(blank line)
	The operator places the probe on the designated sample; the CAL/OP switch on the instrument must be in the OPERATE position. The reading is taken by depressing and releasing the foot pedal.
271	(number) (number)... (NFT numbers)
	These numbers are overprinted on the blank line inserted at 268.
285	SET IN AIR
	FQ (integer) = (number)... (repeats NFT times)
268	(blank line)
	The operator removes the probe from any specimen (conductor) and suspends it in air. The reading is taken by depressing and releasing the foot pedal.
271	(number) (number)... (NFT numbers)
293	POWER AMP FQ (integer) = (number)... (repeats NFT times)

Table C2. (Continued)

Line	QUERY[a] or DISPLAY or Response
268	(blank line)
	With the probe still in air a third set of readings is taken by using the foot pedal as before.
271	(number) (number)... (NFT numbers)
302	TYPE # OF PHASE CALIBRATIONS
	Enter integer equal to number of phase steps to be used from instrument's internal phase calibration network, typically 3.
305	# OF MAG CALIBRATIONS
	Enter integer equal to the number of settings of the magnitude control on the instrument's internal phase calibrator to be used with each phase step, typically 2.
315	CALIBRATION READINGS
	This message cues the operator to commence setting up the calibration inputs. (No response.)
	The data to be collected are the internal calibration data for the instrument, generated by substituting the phase calibration networks in place of the probe. Calibration must start with the frequency listed as FREQ(1) and proceed through FREQ(NPT); the order of magnitude and phase steps is arbitrary but must be kept consistent. Readings are taken by depressing and releasing the foot pedal.
327	+FQ (integer) = (number) PHA (integer) = (number) MAG (integer) = (number)
	This display appears after each reading is made. (No response.) The program loops are arranged so that magnitude settings are cycled through their range first, then through phase settings and frequencies last.
343	Control shifts to statement with the current value of NSTATE.
(403)	The program branches to the sequence for clearing the terminal screen.

Table C2. (Continued)

Line	QUERY[a] or DISPLAY or Response
406	SAMPLE # = (integer) LIFT-OFF = (number) THICK = (number) RHO = (number)
412	DEFECT # (integer) DEPTH = (number) RADIUS = (number) SIZE = (number)
416	PH (integer) MAG (integer)... (NFT repetitions)
420	(number) (number)...
	After the heading data are displayed (through line 237), readings are made by depressing and releasing the foot pedal. Following release the readings (line 432) are displayed.
432	PHASE (integer, integer, integer) = (number)
	TMAG (integer, integer, integer) = (number)
	The integer indices indicate lift-off increment (I), the sequence number in the property set (LL), and the frequency (KK), respectively. The line 432 display repeats for each frequency.
	The sequence 387–440 repeats through all the possible specimen and defect property and lift-off combinations.
***	After the specimen readings are finished, execution returns to line 313 (above) to repeat the internal calibration sequence, 313–343 inclusive, then jumps to 448 (below).
448	(integer) SETS OF READINGS HAVE BEEN MADE.
	DO YOU WANT TO DELETE THE PREVIOUS SET OF READINGS? Type Y or N.
	Enter "Y" or "N".
	If "N" is entered execution branches to line 460 and adds the new data to the earlier data (no display or response). If "Y" is entered execution jumps directly to 480. In either case the next interactive display occurs at 482.
482	DO YOU WANT TO (1) RE-DO THE CALIBRATIONS, (2) MAKE ANOTHER SET OF READINGS, OR (3) STOP? TYPE 1, 2, OR 3:

Table C2. (Continued)

Line	QUERY[a] or DISPLAY or Response
	Enter 1, 2, or 3.
	If "1" is entered the program branches to the interaction at line 313 (above).
	If "2" is entered the branch is to the sequence leading to the display at line 382 (above).
	If "3" is entered the branch is to a test for data acquisition, which gives the warning at line 495 if no datum has been acquired.
	If any other character is entered the error message at line 487 occurs.
487	ERROR ON INPUT, ENTER AGAIN
	After this message execution returns to 480 (above) and repeats.
495	NO READINGS HAVE BEEN SAVED; THEREFORE, PROGRAM WILL STOP WITHOUT SAVING OR PRINTING ANYTHING.
	If this warning occurs the program will branch to line 692 and stop execution. If this warning does not occur printout of data both displayed earlier and read by the program will occur. Then the program branches to line 692 and stops. (See p. 122 for sample printouts.)

[a]Queries shown in all caps simulating the computer output display.

[b]All alphanumerics may include blanks (spaces).

[c]Because of its length the representation of this display has been omitted from the table. It contains all of the information entered in lines 89–184 of this table, identified by labels.

LISTING AND ANALYSIS OF RFLRDG

Program Identification

```
      PROGRAM RFLRDG
```

Effective Revision Date

```
C     (13 DECEMBER 1978)
```

Description

```
C
C THIS PROGRAM READS MAGNITUDE AND PHASE DATA AT DIFFERENT
C     FREQUENCIES FOR REFLECTION TYPE COILS, USING THE PHASE
C     SENSITIVE INSTRUMENT.
C
```

Variable Type Specifications

Normal integer-to-real conversion; double-precision array.

```
      REAL LIFT
      REAL*8 TITLE(2)
      REAL*6 NPROBE,NCABLE,INSTNO,POWOSC,PICKAM,PHADET
```

Arrays with Nonadjusted Dimensions

```
      DIMENSION IDAY(3),ITIM(3),IDY(3),ITM(3)
```

Variable Definitions

```
C
C     MNDEF=MAXIMUM NUMBER OF DEFECTS FOR ANY SAMPLE
C     MNT=MAXIMUM NUMBER OF THICKNESSES FOR ANY RESISTIVITY
C     NCHS=NUMBER OF DATA CHANNELS IN DATA ACQUISITION SYSTEM
C     NFILIM=FILE LIMIT ON LINES IN FILE 25, THE DATA FILE
C     NFT=NUMBER OF FREQUENCIES
C     NLT=NUMBER OF LIFTOFFS
C     NMGCAL=NUMBER OF MAGNITUDE CALIBRATIONS
C     NPHCAL=NUMBER OF PHASE CALIBRATIONS
C     NPT=NUMBER OF PROPERTIES=TOTAL NUMBER OF SAMPLES+DEFECTS
C     NRES=NUMBER OF RESISTIVITIES
```

Array Dimension Guide

```
C
C       DIMENSION VOLTS(NCHS)
C       DIMENSION FREQ(NFT),PICKAM(NFT),PHADET(NFT),PHASW(NFT)
C       DIMENSION DCMGCB(NFT),DCMGAR(NFT),DCPHAR(NFT),POWAMP(NFT)
C       DIMENSION LIFT(NLT),RHO(NRES),THICK(NRES,MNT),NTH(MNT)
C       DIMENSION DFDEPT(NRES,MNT,MNDEF),DFRAD(NRES,MNT,MNDEF)
C       DIMENSION DFSIZE(NRES,MNT,MNDEF),NDEF(NRES,MNT)
C       DIMENSION PHASE(NLT,NPT,NFT),TMAG(NLT,NPT,NFT)
C       DIMENSION SUMPHA(NLT,NPT,NFT),SUMMAG(NLT,NPT,NFT)
C       DIMENSION SSPHA(NLT,NPT,NFT),SSMAG(NLT,NPT,NFT)
C       DIMENSION SDVPHA(NLT,NPT,NFT),SDVMAG(NLT,NPT,NFT)
C       DIMENSION SUMCPH(NFT,NPHCAL,NMGCAL),SUMCMG(NFT,NPHCAL,NMGCAL)
C       DIMENSION SSCPH(NFT,NPHCAL,NMGCAL),SSCMG(NFT,NPHCAL,NMGCAL)
C       DIMENSION SDVCPH(NFT,NPHCAL,NMGCAL),SDVCMG(NFT,NPHCAL,NMGCAL)
C       DIMENSION CALPHA(NFT,NPHCAL,NMGCAL),CALMAG(NFT,NPHCAL,NMGCAL)
C       DIMENSION OLDPHC(NFT,NPHCAL,NMGCAL),OLDMGC(NFT,NPHCAL,NMGCAL)
C
```

Array Dimension Declarations

```
        DIMENSION VOLTS(12)
        DIMENSION FREQ(4),PICKAM(4),PHADET(4),PHASW(4)
        DIMENSION DCMGCB(4),DCMGAR(4),DCPHAR(4),POWAMP(4)
        DIMENSION LIFT(5),RHO(5),THICK(5,8),NTH(8)
        DIMENSION DFDEPT(5,8,4),DFRAD(5,8,4),DFSIZE(5,8,4),NDEF(5,8)
        DIMENSION PHASE(5,40,4),TMAG(5,40,4)
        DIMENSION SUMPHA(5,40,4),SUMMAG(5,40,4)
        DIMENSION SSPHA(5,40,4),SSMAG(5,40,4)
        DIMENSION SDVPHA(5,40,4),SDVMAG(5,40,4)
        DIMENSION SUMCPH(4,3,3),SUMCMG(4,3,3)
        DIMENSION SSCPH(4,3,3),SSCMG(4,3,3),SDVCPH(4,3,3),SDVCMG(4,3,3)
        DIMENSION CALPHA(4,3,3),CALMAG(4,3,3),OLDPHC(4,3,3),OLDMGC(4,3,3)
```

File Sizing Guide

```
C
C***CAUTION***BE SURE ENOUGH SPACE HAS BEEN ALLOCATED ON RAD FILE 25
C       FOR WRITING ALL THE RECORDS OUTPUT BY THIS PROGRAM. NUMBER OF
C       RECORDS IN THIS DEFINE STATEMENT(NFILIM=FIRST # IN PARENTHESES)
C       SHOULD EQUAL AT LEAST NFT(NPHCAL+NPTT)+NRES(NTHI*MNDEF+2)+5.
C       THE SECOND SHOULD BE AT LEAST 2 MORE THAN THE GREATEST OF THE
C       FOLLOWING: 26,14*NFT,8*NLT,2*NRES*NTHI,6*NTHI*MNDEF,8*NMGCAL.
C
```

RAD File Assignment

```
        DEFINE FILE 25(200,40,U,IREC)
C
```

Data Initialization

See Table C1 for requirements.

```
62          DATA LOT/12/,LI/1/,NCHS/12/,LPT/6/,NFILIM/200/
63          DATA TITLE/'CALIBRATION DATA'/,BLANKS/'      '/
64    C
```

Clearing Screen on CRT Terminal

The step sequence (lines 678–688) to which control jumps is unique to the minicomputer in use.

```
65    C     GO CLEAR TERMINAL SCREEN & RETURN HERE
66    C
67          ASSIGN 10 TO ISTATE
68          GO TO 1800
69    C
```

Printing of Output Heading

DATE and TIME subroutines are part of the minicomputer operating system.

```
70    C FORM FEED ON LPT FOR SUMMARY REPORT.
71    C PRINT TITLE AND DATE
72    C
73    10    WRITE (LOT,15) TITLE
74    15    FORMAT(1X,2A8)
75          CALL DATE(IDAY)
76          CALL TIME(ITIM)
77          IT=IDAY(1)-1900
78          IDAY(1)=IDAY(2)
79          IDAY(2)=IDAY(3)
80          IDAY(3)=IT
81          WRITE(LOT,20 ) IDAY,ITIM
82    20     FORMAT(' DATE   ',2(I2,1H/),I2,'     TIME   ',2(I2,1H:),I2)
```

Manual Entry of New Instrument and Calibration Standard Data

See Table C2 for interactive responses.

Instrument Identification

```
83          WRITE(LOT,25)
84    25    FORMAT(' TYPE IN THE FOLLOWING DATA AS REQUESTED.')
85          ANSWER=BLANKS
86    C
87    C INPUT DESCRIPTION OF EXPERIMENTAL APPARATUS
88    C
89          WRITE(LOT,30)
90    30    FORMAT(' PROBE #:  ')
91          READ(LI,35)NPROBE
92    35    FORMAT(A6)
93          IF(NPROBE.EQ.'OLD   ') GO TO 200
```

```
      WRITE(LOT,40)
40    FORMAT(' SERIAL #:  ')
      READ(LI,*)NSER
      WRITE(LOT,45)
45    FORMAT(' DRIVER SERIES RESISTANCE:  ')
      READ(LI,*)R0
      WRITE(LOT,50)
50    FORMAT(' DRIVER SHUNT CAP:  ')
      READ(LI,*)CAPDR
      WRITE(LOT,55)
55    FORMAT(' PICK-UP SHUNT RESISTANCE:  ')
      READ(LI,*)R9
      WRITE(LOT,60)
60    FORMAT(' PICK-UP SHUNT CAP:  ')
      READ(LI,*)CAPPU
      WRITE(LOT,65)
65    FORMAT(' CABLE I.D. #:  ')
      READ(LI,35)NCABLE
      WRITE(LOT,70)
70    FORMAT(' LENGTH OF CABLE:  ')
      READ(LI,*)CABLEL
      WRITE(LOT,75)
75    FORMAT(' CAPACITANCE OF CABLE:  ')
      READ(LI,*)CCABLE
      WRITE(LOT,80)
80    FORMAT(' EDDY CURRENT INSTRUMENT #:  ')
      READ(LI,35) INSTNO
      WRITE(LOT,85)
85    FORMAT(' POWER OSC I.D.:  ')
      READ(LI,35)POWOSC
```

Frequency Channel Data

```
C
C INPUT FREQUENCY VALUES
C
      WRITE(LOT,90)
90    FORMAT(' NO. OF FREQUENCIES:  ')
      READ(LI,*)NFT
      WRITE(LOT,100)
100   FORMAT(' INPUT THE VALUE OF EACH FREQ SEPARATED BY A SPACE:',/)
      READ(LI,*)(FREQ(I), I=1,NFT)
      WRITE(LOT,105)(FREQ(I),I=1,NFT)
105   FORMAT(' INPUT THE FOLLOWING DATA FOR EACH OF FREQUENCIES',/,
     * ' FREQUENCY:',13X,10(1PE12.3))
      WRITE(LOT,110)
110   FORMAT(' PICK-UP AMP (EACH 6 CHARACTERS, NO SPACE BETWEEN):',/)
      READ(LI,115)(PICKAM(I),I=1,NFT)
115   FORMAT(10(A6))
      WRITE(LOT,120)
120   FORMAT(' PHASE DETECTOR (EACH 6 CHARACTERS,NO SPACING):',/)
      READ(LI,115)(PHADET(I),I=1,NFT)
      WRITE(LOT,125)
125   FORMAT(' 180-DEG SW(OFF/ON)(EACH 3 CHARACTERS,NO SPACING):',/)
      READ(LI,130)(PHASW(I),I=1,NFT)
130   FORMAT(10(A3))
```

Calibration Standard Properties

```
C
C INPUT PROPERTY VALUES
```

```
C
      WRITE(LOT,140)
140   FORMAT(' TYPE # OF LIFT-OFF VALUES:',/)
      READ(LI,*)NLT
      WRITE(LOT,150)
150   FORMAT(' TYPE EACH LIFT-OFF VALUE SEPARATED BY A SPACE:',/)
      READ(LI,*)(LIFT(I),I=1,NLT)
      WRITE(LOT,155)
155   FORMAT(' # OF RESISTIVITY VALUES: ')
      READ(LI,*)NRES
      WRITE(LOT,160)
160   FORMAT(' TYPE EACH RESISTIVITY VALUE SEPARATED BY A SPACE:',/)
      READ(LI,*)(RHO(I),I=1,NRES)
      DO 180 I=1,NRES
      WRITE(LOT,165)RHO(I)
165   FORMAT(' TYPE # OF THICKNESS VALUES FOR RES ',/,1X,F10.4,2X)
      READ(LI,*) NTH(I)
      NTHI=NTH(I)
      WRITE(LOT,170)NTH(I),I,RHO(I)
170   FORMAT(' TYPE',I3,' THICKNESS VALUES, EACH SEPARATED BY A SPACE,'
     1       ,/,' FOR RHO(',I1,')=',F9.4,2X)
      READ(LI,*)(THICK(I,J),J=1,NTHI)
      WRITE(LOT,175)(THICK(I,J),J=1,NTHI)
175   FORMAT(' # OF DEFECTS FOR EACH THICKNESS VALUE:',/,9F10.4,/)
      READ(LI,*)(NDEF(I,J),J=1,NTHI)
180   CONTINUE
      DO 190 I=1,NRES
      NTHI=NTH(I)
      DO 190 J=1,NTHI
      IF(NDEF(I,J).EQ.0) GO TO 190
      NDIJ=NDEF(I,J)
      DO 190 K=1,NDIJ
      WRITE(LOT,185) RHO(I),THICK(I,J),K
185   FORMAT(' FOR RES ',F9.4,' THICK ',F9.4,/,' TYPE    LOCAT,
     *'RADIUS $ SIZE',/,' DEFECT #',I2,': ')
     READ(LI,*)DFDEPT(I,J,K),DFRAD(I,J,K),DFSIZE(I,J,K)
190   CONTINUE
      GO TO 400
```

Recovery of Existing Data from RAD File

If new data have been entered this section is skipped.

```
C
C     SECTION TO READ OLD DATA FROM FILE 25.
C
200   IREC=1
      READ(25'IREC) TITLE,IDY,ITM
      READ(25'IREC) NPROBE,NSER,RO,CAPDR,R9,CAPPU,NCABLE,
     *              CABLEL,CCABLE,INSTNO,POWOSC
      READ(25'IREC) NFT,NLT,NRES,NPT,NSAMPL,NPHCAL,NMGCAL
      READ(25'IREC)((FREQ(I),PICKAM(I),PHADET(I),PHASW(I),
     *              DCMGCB(I),DCMGAR(I),POWAMP(I)),I=1,NFT)
      READ(25'IREC) (LIFT(I),I=1,NLT)
      DO 210 I=1,NRES
      READ(25'IREC) NTHI,RHO(I)
      NTH(I)=NTHI
      READ(25'IREC) ((THICK(I,J),NDEF(I,J)),J=1,NTHI)
      DO 210 J=1,NTHI
      NDIJ=NDEF(I,J)
      IF(NDIJ.EQ.0) GO TO 210
      READ(25'IREC) ((DFDEPT(I,J,K),DFRAD(I,J,K),DFSIZE(I,J,K)),
     * K=1,NDIJ)
210   CONTINUE
```

Display of Recovered Data

The READ instruction (line 234) allows a pause for the operator to read the full screen.

```
208            IF(IREC.EQ.NFILIM) GO TO 1900
209            WRITE(LOT,220) TITLE,IDY,ITM
210   220      FORMAT(1X,2A8,10X,'DATE   ',2(I2,1H/),I2,10X,'TIME   ',2(I2,1H:),
211        *          I2///)
212            WRITE(LOT,230) NPROBE,NSER
213   230      FORMAT(' PROBE NO.:',A6,5X,' SERIAL NO.:',I5)
214            WRITE(LOT,235) RO,CAPDR
215   235      FORMAT(' DRIVER SERIES RES.:',F10.1,5X,'DRIVER SHUNT CAP.:',
216        *            E12.4)
217            WRITE(LOT,240) R9,CAPPU
218   240      FORMAT(' PICK-UP SHUNT RES.:',F10.1,5X,'PICK-UP SHUNT CAP.:',
219        *            E12.4)
220            WRITE(LOT,245) NCABLE,CABLEL,CCABLE
221   245      FORMAT(' CABLE I.D.NO.:',A6,5X,'LENGTH:',F10.1,5X,'CAP.:',E12.4)
222            WRITE(LOT,250) INSTNO,POWOSC
223   250      FORMAT(' EDDY CURRENT INST.NO.:',A6,' POWER OSC.I.D.:',A6)
224            WRITE(LOT,255) (FREQ(I),I=1,NFT)
225   255      FORMAT(1X,/,10X,'FREQUENCY:',10(1PE12.4,5X))
226            WRITE(LOT,260) (PICKAM(I),I=1,NFT)
227   260      FORMAT(' PICK-UP AMP.I.D.:',7X,10(A6,9X))
228            WRITE(LOT,265) (PHADET(I),I=1,NFT)
229   265      FORMAT(' PHASE DETECTOR I.D.:',4X,10(A6,9X))
230            WRITE(LOT,270) (PHASW(I),I=1,NFT)
231   270      FORMAT(' 180 PHASE SWITCH:',8X,10(A3,12X))
232            WRITE(LOT,275) (LIFT(I),I=1,NLT)
233   275      FORMAT(1X,/,' LIFT-OFF VALUES',3X,10(F10.5,5X))
234            READ(LI,*) XYZ
```

Entering any real number rolls the screen and display continues.

```
235            WRITE (LOT,280)
236   280      FORMAT(' PROP.NO.',6X,'RESISTIVITY   THICKNESS',5X,'DEF.LOCAT',
237        *   5X,'DEF.RADIUS',4X,'DEF. SIZE')
238            NP=1
239            DO 300 I=1,NRES
240            NTHI=NTH(I)
241            DO 300 J=1,NTHI
242            WRITE(LOT,285) NP,RHO(I),THICK(I,J)
243   285      FORMAT(3X,I3,4X,2(F14.5))
244            NDIJ=NDEF(I,J)
245            IF(NDIJ.EQ.0) GO TO 300
246            DO 295 K=1,NDIJ
247            WRITE(LOT,290) NP,RHO(I),THICK(I,J),DFDEPT(I,J,K),DFRAD(I,J,K),
248        *   DFSIZE(I,J,K)
249   290      FORMAT(3X,I3,4X,5(F14.5))
250   295      CONTINUE
251   300      NP=NP+1
```

Gain Adjustment

This is the first step in the calibration procedure (details on p. 43).

1. The sequence starting at line 262 is entered with the foot pedal up, and execution of 269—275 repeats until it is depressed. During pedal depression execution progresses from 269 to 277, setting LAST = 1 and repeating READAT until pedal release. After release READAT repeats one last time and sets NRET = 1, and execution jumps from 274 to 278, resetting NRET = 0 and then to 279, incrementing NTIMES to 1.

```
252  C
253  C GAIN ADJUSTMENT
254  C
255  400    WRITE(LOT,420)
256  420    FORMAT(' GAIN ADJUSTMENT:')
257         WRITE(LOT,430)
258  430    FORMAT(' TYPE SAMPLE #:')
259         READ(LI,*)NSAMPL
260         WRITE(LOT,440)NSAMPL,(BLANKS,I,FREQ(I),I=1,NFT)
261  440    FORMAT(' SET ON SAMPLE'/'   #',I10,10(A1,'FQ',I1,'=',1PE12.4))
262         NTIMES=0
263         ISTART=2
264         ITIMS=512
265         NRET=0
266  450    LAST=0
267         WRITE(LOT,460)
268  460    FORMAT(1H )
269  470    CALL READAT(VOLTS,NCHS,ITIMS,NRET)
270         MAXFRE=4*(NFT-1)+ISTART
271         WRITE(LOT,490)(VOLTS(I),I=ISTART,MAXFRE,4)
272  490    FORMAT(1H+,19X,10(F9.4,5X))
273         IF(NTIMES.EQ.2) GO TO 510
274         IF(LAST.EQ.1) GO TO 500
275         IF(VOLTS(12).GE.2.) GO TO 470
276         LAST=1
277         GO TO 470
278  500    NRET=0
279  510    NTIMES=NTIMES+1
```

2. Execution returns from 286 to 266, resetting LAST = 0, and the sequence (1) above is repeated, incrementing NTIMES to 2. Then execution jumps to 294, changing ISTART to 3, which, in turn, changes the voltages displayed. [When ISTART = 2 those channels (2,6,10) carrying the output (pickup) voltage magnitude are printed or displayed; when ISTART = 3 channels (3,7,11) carrying the power oscillator voltage amplitudes are displayed.]

```
280         IF(NTIMES.GE.2) GO TO 540
281         DO 520 I=1,NFT
282         J=4*(I-1)+2
283  520    DCMGCB(I)=VOLTS(J)
284         WRITE(LOT,530)(BLANKS,I,FREQ(I),I=1,NFT)
```

```
530     FORMAT(' SET IN AIR ',2X,10(A1,'FQ',I1,'=',1PE12.4,))
        GO TO 450
540     IF(NTIMES.GE.3) GO TO 570
        DO 550 I=1,NFT
        J=4*(I-1)+2
        DCPHAR(I)=VOLTS(J-1)
550     DCMGAR(I)=VOLTS(J)
        WRITE(LOT,560) (BLANKS,I,FREQ(I),I=1,NFT)
560     FORMAT(' POWER AMP ',3X,10(A1,'FQ',I1,'=',1PE12.4))
        ISTART=3
```

The value of NTIMES increments each time the pedal is depressed and released to control the sequential use of the readings taken. Values of NRET and LAST change state to ensure that only one execution of READAT occurs, and only one complete set of readings is taken and returned to the main program each time the pedal is depressed and released.

The DCMGCB stores voltages from the output magnitude channels taken with the probe on the specimen producing the largest magnitude (step 1 of calibration). The DCPHAR and DCMGAR store voltages from output magnitude and phase channels taken with the probe in air. The POWAMP stores voltages from channels carrying the power oscillator voltages at each of the frequencies.

Here, and each time the READAT subroutine is used, it is followed by a loop that converts the analog-digital converter counts in the NARRAY array to a voltage. For the acquisition system used with this program, 8192 counts equal 10.000 V.

(See p. 172 for READAT subroutine; see Table C2 for operator instructions.)

```
        GO TO 450
570     CONTINUE
        DO 580 I=1,NFT
        J=4*(I-1)+3
        POWAMP(I)=VOLTS(J)
580     CONTINUE
```

Internal Calibration Readings

The numbers of phase and magnitude calibration points to be used are selected. The ASSIGN statement sets up branching for other parts of the program (line 343).

```
301          WRITE(LOT,590)
302   590    FORMAT(' TYPE # OF PHASE CALIBRATIONS:')
303          READ(LI,*) NPHCAL
304          WRITE(LOT,600)
305   600    FORMAT(' # OF MAG CALIBRATIONS:')
306          READ(LI,*)NMGCAL
307          ASSIGN 720 TO NSTATE
```

The phase and magnitude calibration readings are loaded into CALPHA and CALMAG arrays, the older contents of which were first loaded into the OLDPHC and OLDMGC arrays. (Each time this internal calibration section is used, the transfer from CAL to OLD is made. Thus, the calibrations preceding and following any set of external readings are always available.) For all these arrays the indices are: I = frequency, J = the phase calibration point, and K = the magnitude calibration point.

```
308   C
309   C THIS SECTION TAKES THE CALIBRATION READINGS. CONTROL IS RETURNED
310   C      TO SOME SECTION OF THE PROGRAM ACCORDING TO THE CURRENT
311   C      VALUE OF THE VARIABLE, NSTATE (STATEMENT NO. VALUE).
312   C
313   610    ANSWER=BLANKS
314   630    WRITE(LOT,640)
315   640    FORMAT(' CALIBRATION READINGS:')
316          NPHCH=1
317          NMGCH=NPHCH+1
318          ITIMS=512
319          NRET=0
320          DO 710 I=1,NFT
321          DO 700 J=1,NPHCAL
322          DO 690 K=1,NMGCAL
323          LAST=0
324          WRITE(LOT,460)
325   650    CALL READAT(VOLTS,NCHS,ITIMS,NRET)
326          WRITE(LOT,670) I,FREQ(I),J,VOLTS(NPHCH),K,VOLTS(NMGCH)
327   670    FORMAT('+FQ ',I1,'=',1PE12.4,' PHA ',I1,'=',0PF9.4,' MAG ',I1,
328         1        '=',F9.4)
329          IF(LAST.EQ.1) GO TO 680
330          IF(VOLTS(12).GE.2.) GO TO 650
331          LAST=1
332          GO TO 650
333   680    NRET=0
334          OLDPHC(I,J,K)=CALPHA(I,J,K)
335          OLDMGC(I,J,K)=CALMAG(I,J,K)
336          CALPHA(I,J,K)=VOLTS(NPHCH)
```

```
      CALMAG(I,J,K)=VOLTS(NMGCH)
690   CONTINUE
700   CONTINUE
      NPHCH=NPHCH+4
      NMGCH=NPHCH+1
710   CONTINUE
      GO TO NSTATE
C
C END OF SECTION WHICH TAKES CALIBRATION READINGS
```

Readings from Standards

The total number of property sets equals the number of conductor property sets, NRES*NTHI, plus the number of combinations of conductor properties with defect properties,

$$\text{NRES*NTHI} + \sum_{i=1}^{\text{NRES}} \sum_{j=1}^{\text{NTHI}} \text{NDEF}(i,j) \ .$$

(Additional array dimensions are added below to accommodate lift-off.)

```
C
C ZERO ARRAYS THAT WILL CONTAIN SUMS OF THE PHASE & MAG READINGS,
C       SUMS OF CALIBRATION READINGS, AND SUMS OF SQUARES OF EACH
C
720   MSET=0
      NPT=0
      DO 730 I=1,NRES
      NTHI=NTH(I)
      DO 730 J=1,NTHI
      NPT=NPT+1+NDEF(I,J)
730   CONTINUE
```

The storage arrays are initialized. The SUM arrays contain sums of the phase or magnitude readings. The SS arrays contain the sums of squares of these same readings.

```
      DO 750 KK=1,NPT
      DO 750 LL=1,NPT
      DO 750 I=1,NLT
      SUMPHA(I,LL,KK)=0.
      SUMMAG(I,LL,KK)=0.
      SSPHA(I,LL,KK)=0.
      SSMAG(I,LL,KK)=0.
750   CONTINUE
```

Arrays for the internal calibration readings are also provided.

```
          DO 760 I=1,NFT
          DO 760 J=1,NPHCAL
          DO 760 K=1,NMGCAL
          SUMCPH(I,J,K)=0.
          SUMCMG(I,J,K)=0.
          SSCPH(I,J,K)=0.
          SSCMG(I,J,K)=0.
  760     CONTINUE
```

The operator selects the next operation (see Table C2).

```
          WRITE(LOT,770)
  770     FORMAT(' SWITCH TO OPERATE AND ANSWER Y IF YOU ARE READY TO',/,
         *'TAKE THE FIRST SET OF READINGS,N OTHERWISE ??')
          ANSWER=BLANKS
          READ(LI,1020) ANSWER
          IF(ANSWER.EQ.'N   ') GO TO 1140
```

Loops are constructed to number specimens according to resistivity (NRES), thickness (NTHL), and lift-off (NLT) steps. For those conductors containing defects, the basic loops are expanded to provide additional numbers (NDLK).

```
C
C THIS SECTION TAKES THE ACTUAL PHASE & MAGNITUDE DATA READINGS
C
  780     ANSWER=BLANKS
  800     ITIMS=512
          NCOUNT=0
          NRET=0
          LL=1
          DO 990 L=1,NRES
          NTHL=NTH(L)
          DO 980 K=1,NTHL
          J=1
          IF (NDEF(L,K).EQ.0) GO TO 805
          NDLK=1+NDEF(L,K)
          DO 970 M=1,NDLK
  805     DO 960 I=1,NLT
```

A sample number is generated from the loop indices.

```
          ISAMPL=I*1000+J*100+K*10+L
          IF (NDEF(L,K).EQ.0) ISAMPL=I*1000+K*10+L
```

The CRT screen is cleared. The NCOUNT value test prevents clearing the screen on each pass through the loop.

```
397          IF(NCOUNT.NE.0) GO TO 810
398   C
399   C GO CLEAR TERMINAL SCREEN & RETURN HERE
400   C
401          NCOUNT=1
402          ASSIGN 810 TO ISTATE
403          GO TO 1800
404   C
```

Conductor properties and sample (standard) identification are displayed first.

```
405   810    WRITE(LOT,820)ISAMPL,LIFT(I),THICK(L,K),RHO(L)
406   820    FORMAT(1H1,'SAMPLE #=',I4,' LIFT-OFF=',F9.4,' THICK=',
407         * F9.4,' RHO=',F9.4)
```

If the conductors contain defects, their properties are displayed.

```
408          IF(NDEF(L,K).EQ.0) GO TO 850
409          IF(J.EQ.1) GO TO 850
410          MDF=M-1
411          WRITE(LOT,830)MDF,DFDEPT(L,K,MDF),DFRAD(L,K,MDF),DFSIZE(L,K,MDF)
412   830    FORMAT(3X,'DEFECT #',I2,3X,'LOCAT=',F8.5,3X,'RADIUS=',F8.5,
413         * 3X,'SIZE=',F8.5)
```

If an automatic calibration module is not used, the instrument must be switched from CALIBRATE to OPERATE mode. The foot switch is depressed and released to take each set of readings. [Limits and controls on the subroutine are as before (see lines 318–325).] The readings are displayed, as taken, below the headings generated at line 416 (see Table C2).

```
414   850    MAXCH=4*(NFT-1)+1
415          WRITE(LOT,860)(BLANKS,II,II, II=1,NFT)
416   860    FORMAT(10(A1,'PH(',I1,')',6X,' MAG(',I1,')',5X))
417          WRITE(LOT,460)
418          LAST=0
419   870    CALL READAT(VOLTS,NCHS,ITIMS,NRET)
420          WRITE(LOT,890)((VOLTS(JJ),VOLTS(JJ+1)),JJ=1,MAXCH,4)
421   890    FORMAT(1H+,10(F9.4,1X))
422          IF(LAST.EQ.1) GO TO 900
423          IF(VOLTS(12).GE.2.) GO TO 870
424          LAST=1
425          GO TO 870
426   900    NRET=0
```

The readings are redisplayed, with identification of lift-off increment, combined property set sequence number, and frequency (see Table C2).

```
427         DO 950 KK=1,NFT
428         JJ=4*(KK-1)+1
429         PHASE(I,LL,KK)=VOLTS(JJ)
430         TMAG(I,LL,KK)=VOLTS(JJ+1)
431         WRITE(LOT,930)I,LL,KK,VOLTS(JJ),I,LL,KK,VOLTS(JJ+1)
432   930   FORMAT(' PHASE(',I1,',',I2,',',I1,')=',F10.5,2X,
433      1        'TMAG(',I1,',',I2,',',I1,')=',F10.5)
434   950   CONTINUE
435   960   CONTINUE
```

The combined property set sequence (LL) and conductor property set sequence (J) numbers are incremented and the loops closed.

```
436         J=J+1
437         LL=LL+1
438   970   CONTINUE
439   980   CONTINUE
440   990   CONTINUE
```

Internal Calibration Recheck

The number of readings from the standards is counted and a flag (NSTATE) is set to jump execution out of the reading loop after the second set of internal calibration readings is made (see line 343). Execution returns to line 313.

```
441         MSET=MSET+1
442         ASSIGN 1000 TO NSTATE
443         GO TO 610
```

The user is queried to determine whether to save or delete the readings just taken. If deletion is selected the count of readings sets taken is decremented and execution jumps to an action selection (see line 480 below). If retention is selected execution jumps to filing steps (immediately below).

```
444   C
445   C END OF A SET OF READINGS. ASK USER IF HE WANTS TO DELETE THIS SET.
446   C
447   1000  WRITE(LOT,1010)MSET
448   1010  FORMAT(1H ,I5,' SET(S) OF READINGS HAVE BEEN MADE.'/
449      1   ' DO YOU WANT TO DELETE THE PREVIOUS SET OF READINGS? ',
450      2   'TYPE Y OR N:')
```

```
451          ANSWER=BLANKS
452          READ(LI,1020)ANSWER
453   1020   FORMAT(A1)
454          IF(ANSWER.EQ.'N   ') GO TO 1030
455          MSET=MSET-1
456          GO TO 1140
```

File Storage of Collected Data

When the first complete set of readings, internal and external, is completed and selected for retention, the first set of internal calibration readings, in OLD__C arrays, is added to the storage arrays SUMC and SSC, and the external standardization readings, in PHASE and TMAG arrays, are added to the storage arrays SUM and SS. The second set of internal calibration readings is held in the CAL arrays and is either transferred to the OLD arrays if another set of external readings is made or added to the storage arrays if the STOP sequence is selected (line 483).

```
457   C
458   C ADD NEW CURRENT SET OF READINGS TO SUM OF PREVIOUS SETS
459   C
460   1030   ANSWER=BLANKS
461   1050   DO 1080 I=1,NFT
462          DO 1080 J=1,NPHCAL
463          DO 1080 K=1,NMGCAL
464          SUMCPH(I,J,K)=SUMCPH(I,J,K)+OLDPHC(I,J,K)
465          SUMCMG(I,J,K)=SUMCMG(I,J,K)+OLDMGC(I,J,K)
466          SSCPH(I,J,K)=SSCPH(I,J,K)+OLDPHC(I,J,K)*OLDPHC(I,J,K)
467          SSCMG(I,J,K)=SSCMG(I,J,K)+OLDMGC(I,J,K)*OLDMGC(I,J,K)
468   1080   CONTINUE
469   1100   DO 1130 KK=1,NFT
470          DO 1130 LL=1,NPT
471          DO 1130 I=1,NLT
472          SUMPHA(I,LL,KK)=SUMPHA(I,LL,KK)+PHASE(I,LL,KK)
473          SUMMAG(I,LL,KK)=SUMMAG(I,LL,KK)+TMAG(I,LL,KK)
474          SSPHA(I,LL,KK)=SSPHA(I,LL,KK)+PHASE(I,LL,KK)*PHASE(I,LL,KK)
475          SSMAG(I,LL,KK)=SSMAG(I,LL,KK)+TMAG(I,LL,KK)*TMAG(I,LL,KK)
476   1130   CONTINUE
```

Action Selection

The operator can repeat the calibration, take another set of external readings, or stop. Unrecognized entries do not jump execution, and an error message results.

```
477   C
478   C PROMPT USER ABOUT WHAT HE WANTS TO DO NEXT
479   C
480   1140   ANSWER=BLANKS
481          WRITE(LOT,1150)
```

```
1150    FORMAT(' DO YOU WANT TO (1) RE-DO THE CALIBRATIONS, '/
       *' (2) MAKE ANOTHER SET OF READINGS, OR (3) STOP? TYPE 1,2,OR 3:')
        READ(LI,*)NEXT
        GO TO (610,780,1170),NEXT
        WRITE(LOT,1160)
1160    FORMAT(' ERROR ON INPUT. ENTER AGAIN.')
        GO TO 1140
```

Statement 1170 determines whether external readings were taken and selected for retention. If so, execution jumps to the next retention step. Otherwise, a status message is displayed and execution jumps to the STOP instruction.

```
C
C BEFORE STOPPING, ADD FINAL SET OF CALIBRATION READINGS TO CUMULATIVE
C        SUM & CALCULATE AVERAGES & STANDARD DEVIATIONS
C
1170    IF(MSET.GE.1) GO TO 1190
        WRITE(LOT,1180)
1180     FORMAT(' NO READINGS HAVE BEEN SAVED; THEREFORE, PROGRAM WILL '/
     1   ' STOP WITHOUT SAVING OR PRINTING ANYTHING.')
        GO TO 2000
```

Continuation of File Storage

The final sets of internal calibration data are added to the storage arrays (SUMC and SSC) from the CAL arrays.

```
1190    DO 1200 I=1,NFT
        DO 1200 J=1,NPHCAL
        DO 1200 K=1,NMGCAL
        SUMCPH(I,J,K)=SUMCPH(I,J,K)+CALPHA(I,J,K)
        SUMCMG(I,J,K)=SUMCMG(I,J,K)+CALMAG(I,J,K)
        SSCPH(I,J,K)=SSCPH(I,J,K)+CALPHA(I,J,K)*CALPHA(I,J,K)
        SSCMG(I,J,K)=SSCMG(I,J,K)+CALMAG(I,J,K)*CALMAG(I,J,K)
1200    CONTINUE
```

The means of the internal calibration readings are computed and stored in the SUMC arrays. Standard deviations are computed and stored in the SDVC arrays. (Note that there is one more set of internal calibration readings than external standardization readings because of the initial set taken.)

```
        DO 1250 I=1,NFT
        DO 1250 J=1,NPHCAL
        DO 1250 K=1,NMGCAL
        SUMCPH(I,J,K)=SUMCPH(I,J,K)/(FLOAT(MSET+1))
        SUMCMG(I,J,K)=SUMCMG(I,J,K)/(FLOAT(MSET+1))
        SDVCPH(I,J,K)=SQRT((SSCPH(I,J,K)-SUMCPH(I,J,K)*SUMCPH(I,J,K)
```

```
     1                *FLOAT(MSET+1))/FLOAT(MSET))
      SDVCMG(I,J,K)=SQRT((SSCMG(I,J,K)-SUMCMG(I,J,K)*SUMCMG(I,J,K)
     1                *FLOAT(MSET+1))/FLOAT(MSET))
 1250 CONTINUE
```

Means of the external standardization readings are computed and stored back in the SUM arrays. Standard deviations are computed and stored in the SDV arrays.

```
C
C CALCULATE AVERAGES & STANDARD DEVIATIONS OF READINGS
C
      DO 1290 KK=1,NFT
      DO 1290 LL=1,NPT
      DO 1290 I=1,NLT
      SUMPHA(I,LL,KK)=SUMPHA(I,LL,KK)/(FLOAT(MSET))
      SUMMAG(I,LL,KK)=SUMMAG(I,LL,KK)/(FLOAT(MSET))
      SDVPHA(I,LL,KK)=SQRT((SSPHA(I,LL,KK)-SUMPHA(I,LL,KK)
     1        *SUMPHA(I,LL,KK)*FLOAT(MSET))/FLOAT(MSET-1))
      SDVMAG(I,LL,KK)=SQRT((SSMAG(I,LL,KK)-SUMMAG(I,LL,KK)
     1        *SUMMAG(I,LL,KK)*FLOAT(MSET))/FLOAT(MSET-1))
 1290 CONTINUE
```

Instrument identification, control parameters, conductor and defect properties, and all calibration and standardization data are stored on the file disc.

```
C
C  STORE ALL INFORMATION IN DIRECT ACCESS FILE #25 ON DISK
C
      IREC=1
      WRITE(25'IREC) TITLE,IDAY,ITIM
      WRITE(25'IREC)NPROBE,NSER,RO,CAPDR,R9,CAPPU,NCABLE,CABLEL,
     *           CCABLE,INSTNO,POWOSC
      WRITE(25'IREC) NFT,NLT,NRES,NPT,NSAMPL,NPHCAL,NMGCAL
      WRITE(25'IREC) ((FREQ(I),PICKAM(I),PHADET(I),PHASW(I),
     *            DCMGCB(I),DCMGAR(I),POWAMP(I)),I=1,NFT)
      WRITE(25'IREC) (LIFT(I),I=1,NLT)
      DO 1300 I=1,NRES
      NTHI=NTH(I)
      WRITE(25'IREC) NTHI,RHO(I)
      WRITE(25'IREC)((THICK(I,J),NDEF(I,J)),J=1,NTHI)
      DO 1300 J=1,NTHI
      NDIJ=NDEF(I,J)
      IF(NDIJ.EQ.0) GO TO 1300
      WRITE(25'IREC)((DFDEPT(I,J,K),DFRAD(I,J,K),DFSIZE(I,J,K)),
     *K=1,NDIJ)
 1300 CONTINUE
      DO 1310 I=1,NFT
      DO 1310 J=1,NPHCAL
      WRITE(25'IREC)((SUMCPH(I,J,K),SUMCMG(I,J,K),SDVCPH(I,J,K),
     *SDVCMG(I,J,K)),K=1,NMGCAL)
 1310 CONTINUE
      DO 1320 K=1,NFT
      DO 1320 J=1,NPT
      WRITE(25'IREC)((SUMPHA(I,J,K),SUMMAG(I,J,K),SDVPHA(I,J,K),
```

```
       *SDVMAG(I,J,K)),I=1,NLT)
 1320   CONTINUE
        IF(IREC.EQ.NFILIM) GO TO 1900
C
C END OF SECTION WHICH WRITES DIRECT ACCESS FILE
```

Printout of Final Results of Program Run

This summary duplicates data stored on the disc, except for explanatory headings that are inserted. (See p. 122 for sample outputs.)

Spacing and Heading

```
C PRINT SUMMARY OF JOB STATISTICS ON LPT
C
        WRITE(LPT,1335)
 1335   FORMAT(1H1,20X)
        WRITE(LPT,1340) TITLE,IDAY,ITIM
 1340   FORMAT(1X,2A8,10X,'DATE  ',2(I2,1H/),I2,10X,'TIME  ',2(I2,1H:),
       1        I2///)
```

Probe and Instrument Identifiers and Characteristics

```
        WRITE(LPT,1350) NPROBE,NSER
 1350   FORMAT(' PROBE NO.:',A6,5X,' SERIAL NO.:',I5)
        WRITE(LPT,1360) RO,CAPDR
 1360   FORMAT(' DRIVER SERIES RESISTANCE:',F10.1,5X,'DRIVER SHUNT CAP.:',
       1        E12.4)
        WRITE(LPT,1370) R9,CAPPU
 1370   FORMAT(' PICK-UP SHUNT RESISTANCE:',F10.1,5X,'PICK-UP SHUNT CAP.:'
       1         ,E12.4)
        WRITE(LPT,1330) NCABLE,CABLEL,CCABLE
 1380   FORMAT(' CABLE I.D. NO.:',A6,5X,'LENGTH:',F10.1,5X,'CAP.:',
       1         E12.4)
        WRITE(LPT,1390) INSTNO
 1390   FORMAT(' EDDY CURRENT INSTRUMENT NO.:',A6)
        WRITE(LPT,1400) POWOSC
 1400   FORMAT(' POWER OSC I.D.',A6)
        WRITE(LPT,1410) (FREQ(I),I=1,NFT)
 1410   FORMAT(1H0,10X,'FREQUENCY:',10(1PE12.4,5X))
        WRITE(LPT,1420) (PICKAM(I),I=1,NFT)
 1420   FORMAT(' PICK-UP AMP I.D.:',7X,10(A6,9X))
        WRITE(LPT,1430) (PHADET(I),I=1,NFT)
 1430   FORMAT(' PHASE DETECTOR I.D.:',4X,10(A6,9X))
        WRITE(LPT,1440) (PHASW(I),I=1,NFT)
 1440   FORMAT(' 180  PHASE SWITCH:',6X,10(A3,12X))
```

Lift-Off Values and Conductor Properties

```
        WRITE(LPT,1450) (LIFT(I),I=1,NLT)
 1450   FORMAT(1H0,'LIFT-OFF VALUES',3X,10(F10.5,5X))
        WRITE(LPT,1460)
 1460   FORMAT(' PROP.NO.',6X,'RESISTIVITY   THICKNESS',4X,'DEF.LOCATION',
```

```
     *3X,'DEF.RADIUS',4X,'DEF. SIZE')
      NP=1
      DO 1500 I=1,NRES
      NTHI=NTH(I)
      DO 1500 J=1,NTHI
      WRITE(LPT,1470) NP,RHO(I),THICK(I,J)
1470  FORMAT(3X,I3,4X,2(F14.5))
```

Defect Properties

```
      IF(NDEF(I,J).EQ.0) GO TO 1500
      NDIJ=NDEF(I,J)
      DO 1500 K=1,NDIJ
      WRITE(LPT,1480)NP,RHO(I),THICK(I,J),DFDEPT(I,J,K),DFRAD(I,J,K),
     *DFSIZE(I,J,K)
1480  FORMAT(3X,I3,4X,5(F14.5))
```

Internal Calibration

```
1500  NP=NP+1
      WRITE(LPT,1510)
1510  FORMAT(1H0,'GAIN ADJUSTMENT:')
      WRITE(LPT,1520) NSAMPL,(DCMGCB(I),I=1,NFT)
1520  FORMAT(' SET ON SAMPLE #:',I10,5X,'D.C. MAG:',10(F10.5,5X))
      WRITE(LPT,1530) (DCMGAR(I),I=1,NFT)
1530  FORMAT(' SET IN AIR',21X,'D.C. MAG:',10(F10.5,5X))
      WRITE(LPT,1540) (POWAMP(I),I=1,NFT)
1540  FORMAT(' POWER AMP',31X,10(F10.5,5X))
      WRITE(LPT,1550) NPHCAL,NMGCAL
1550  FORMAT(1H0,' AVERAGES & STANDARD DEVIATIONS OF CALIBRATION ',
     1      'READINGS:',I3,' PHA',I3,' MAG'/)
      WRITE(LPT,1560) (FREQ(I),I=1,NFT)
1560  FORMAT(3(10X,1PE12.4))
      WRITE(LPT,1570)((BLANKS),KK=1,NFT)
1570  FORMAT(8X,3(A1,'PHA',8X,'MAG',7X))
      DO 1600 J=1,NPHCAL
      DO 1600 K=1,NMGCAL
      WRITE(LPT,460)
      WRITE(LPT,1580)((BLANKS,SUMCPH(I,J,K),SUMCMG(I,J,K)),I=1,NFT)
1580  FORMAT(5X,3(A1,F10.5 ,F10.5))
      WRITE(LPT,1590)((BLANKS,SDVCPH(I,J,K),SDVCMG(I,J,K)),I=1,NFT)
1590  FORMAT(' S.D.',3(A1,F10.5, F10.5))
```

External Standardization

The jump statement bypasses execution of the following section (at statement 1800), which functions as a subroutine.

```
1600  CONTINUE
      WRITE(LPT,1610) MSET
1610  FORMAT(1H0,' AVERAGES & STANDARD DEVIATIONS OF ',I5,' SETS OF ',
     1      'READINGS'/)
      WRITE(LPT,1620) (FREQ(I),I=1,NFT)
1620  FORMAT(3X,3(9X,1PE12.4))
```

```
      WRITE(LPT,460)
      LL=1
      DO 1740 L=1,NRES
      WRITE(LPT,1630) RHO(L)
1630  FORMAT(' RESISTIVITY',2X,F10.5)
      NTHL=NTH(L)
      DO 1730 K=1,NTHL
      WRITE(LPT,1640) THICK(L,K)
1640  FORMAT(' THICKNESS',4X,F10.5)
      IF(NDEF(L,K).EQ.0) GO TO 1680
      NDLK=1+NDEF(L,K)
      DO 1730 M=1,NDLK
      IF(M.GT.1) GO TO 1660
      WRITE(LPT,1650)
1650  FORMAT(' NO DEFECT')
      GO TO 1680
1660  WRITE(LPT,1670)DFDEPT(L,K,M-1),DFRAD(L,K,M-1),DFSIZE(L,K,M-1)
1670  FORMAT(' DFLOCAT=',F11.5,3X,'DFRADIUS=',F8.5,'   DFSIZE=',F8.5)
1680  WRITE(LPT,1690) ((BLANKS),KK=1,NFT)
1690  FORMAT(' LIFT-OFF',4X,3(A1,'PHA',7X,'MAG',6X))
      DO 1720 I=1,NLT
      WRITE(LPT,1700)LIFT(I),((SUMPHA(I,LL,KK),SUMMAG(I,LL,KK)),
     1                 KK=1,NFT)
1700  FORMAT( F10.5,3( F10.5, F10.5))
      WRITE(LPT,1710)((SDVPHA(I,LL,KK),SDVMAG(I,LL,KK)),KK=1,NFT)
1710  FORMAT(6X,'S.D.',3( F10.5, F10.5))
1720  CONTINUE
      LL=LL+1
      WRITE(LPT,460)
1730  CONTINUE
1740  CONTINUE
      GO TO 2000
C
C END OF SECTION WHICH WRITES SUMMARY REPORT ON LPT
C
```

Clearing the CRT Screen

The assembler code used has some features unique to the minicomputer on which the programs were run.

```
C
C THIS SECTION OF ASSEMBLER CODE IS NECESSARY IN ORDER TO CLEAR
C      THE SCREEN OF THE TERMINAL.
C
1800  CONTINUE
      INLINE
      LDI,4       #001A       LOAD SUB(ERASE SCREEN) IN REG 4
      LDI,1       #4002       LOAD 4000 + CHANNEL OFFSET INTO REG 1.
      OCB,1,8                   SELECT CH 02
      LDI,3       #801A         LOAD REG 3 WITH NO PARITY,8DATA BITS 1 STOP
      OCB,3,9                   OUTPUT REG 3
READY ISB,2,9                   INPUT STATUS IN REG 2
      TBRB,2,8    READY         TEST BIT 8 IN REG 2.STAY IN LOOP UNTIL RDY
      ODB,4,9                   OUTPUT BITS 8-15 OF REG 4 TO CH A02
      FINI
      GO TO ISTATE
1900  WRITE(LOT,1950)
1950  FORMAT(' LIMIT OF FILE 25 IS EXCEEDED.')
```

Stopping the Job

```
692  2000  STOP JOB
693        END
694  C
```

A partial printout of a sample run is given below.

```
CALIBRATION DATA          DATE   7/ 5/78          TIME   9:28:49


PROBE NO.:150B          SERIAL NO.:34953
DRIVER SERIES RESISTANCE:     9090.0      DRIVER SHUNT CAP.:  0.0000E+00
PICK-UP SHUNT RESISTANCE: 1000000.0      PICK-UP SHUNT CAP.:  0.0000E+00
CABLE I.D..NO.:6-3          LENGTH:       18.0      CAP.:  0.8400E-10
EDDY CURRENT INSTRUMENT NO.:1502
POWER OSC I.D.15A-4


          FREQUENCY:  5.0337E+03        5.0162E+04
PICK-UP AMP I.D.:       15A-35          15A-10
PHASE DETECTOR I.D.:    15A-7           15A-2
180  PHASE SWITCH:      OFF             OFF


LIFT-OFF VALUES       0.00000        0.00200        0.00400
PROP.NO.        RESISTIVITY    THICKNESS    DEF.LOCATION    DEF.RADIUS    DEF. SIZE
    1             3.95000       0.05491
    1             3.95000       0.05491       0.00550        0.75000       0.01400
    2             3.95000       0.07098
    2             3.95000       0.07098       0.00700        0.75000       0.01400
    3             3.95000       0.07657
    3             3.95000       0.07657       0.00760        0.75000       0.01400


GAIN ADJUSTMENT:
SET ON SAMPLE #:          3      D.C. MAG:   5.99290        5.20892
SET IN AIR                       D.C. MAG:   0.19660        0.00598
POWER AMP                                    2.87402        3.28163


 AVERAGES & STANDARD DEVIATIONS OF CALIBRATION READINGS:  3 PHA  2 MAG

          5.0337E+03           5.0162E+04
        PHA       MAG        PHA       MAG

        0.00356   6.30107    0.51024   5.78233
S.D.    0.00123   0.00000    0.00000   0.00000

        0.00211   4.80119    0.50985   4.79293
S.D.    0.00062   0.01105    0.00000   0.00000

        0.97334   6.20991    0.98934   5.76981
S.D.    0.00000   0.00000    0.00000   0.00000

        0.96735   4.80443    0.98901   4.81191
S.D.    0.00000   0.01296    0.00000   0.01033

        1.94187   6.14363    1.47052   5.76176
S.D.    0.00000   0.00000    0.00000   0.00000

        1.93094   4.80064    1.46996   4.79987
S.D.    0.00195   0.01235    0.00000   0.00552
```

AVERAGES & STANDARD DEVIATIONS OF 1 SETS OF READINGS

5.0337E+03 5.0162E+04

RESISTIVITY 3.95000
THICKNESS 0.05491
NO DEFECT

LIFT-OFF	PHA	MAG	PHA	MAG
0.00000	0.66741	6.00526	0.60018	5.18818
S.D.	0.00000	0.00000	0.00000	0.00000
0.00200	0.65469	5.69178	0.61776	4.94105
S.D.	0.00000	0.00000	0.00000	0.00000
0.00400	0.63583	5.40200	0.63172	4.70475
S.D.	0.00000	0.00000	0.00000	0.00000

DFDEPTH= 0.00550 DFRADIUS= 0.75000 DFSIZE= 0.01400

LIFT-OFF	PHA	MAG	PHA	MAG
0.00000	0.67207	6.05842	0.59668	5.22871
S.D.	0.00000	0.00000	0.00000	0.00000
0.00200	0.65817	5.70182	0.61696	4.95032
S.D.	0.00000	0.00000	0.00000	0.00000
0.00400	0.64286	5.42219	0.62998	4.72176
S.D.	0.00000	0.00000	0.00000	0.00000

THICKNESS 0.07098
NO DEFECT

LIFT-OFF	PHA	MAG	PHA	MAG
0.00000	0.32584	6.03053	0.59426	5.21299
S.D.	0.00000	0.00000	0.00000	0.00000
0.00200	0.32408	5.69421	0.61326	4.94778
S.D.	0.00000	0.00000	0.00000	0.00000
0.00400	0.29891	5.42240	0.62601	4.73225
S.D.	0.00000	0.00000	0.00000	0.00000

DFDEPTH= 0.00700 DFRADIUS= 0.75000 DFSIZE= 0.01400

LIFT-OFF	PHA	MAG	PHA	MAG
0.00000	0.33398	6.04703	0.59479	5.23951
S.D.	0.00000	0.00000	0.00000	0.00000
0.00200	0.32065	5.71642	0.61356	4.97899
S.D.	0.00000	0.00000	0.00000	0.00000
0.00400	0.30180	5.43998	0.62499	4.74860
S.D.	0.00000	0.00000	0.00000	0.00000

THICKNESS 0.07657
NO DEFECT

LIFT-OFF	PHA	MAG	PHA	MAG
0.00000	0.25146	6.00352	0.59336	5.21676
S.D.	0.00000	0.00000	0.00000	0.00000
0.00200	0.23711	5.66189	0.61231	4.94805
S.D.	0.00000	0.00000	0.00000	0.00000
0.00400	0.22571	5.38150	0.62596	4.72260
S.D.	0.00000	0.00000	0.00000	0.00000

DFDEPTH= 0.00760 DFRADIUS= 0.75000 DFSIZE= 0.01400

LIFT-OFF	PHA	MAG	PHA	MAG
0.00000	0.25840	6.01485	0.59311	5.22444
S.D.	0.00000	0.00000	0.00000	0.00000
0.00200	0.24540	5.67555	0.61249	4.95859
S.D.	0.00000	0.00000	0.00000	0.00000
0.00400	0.23607	5.36708	0.62740	4.71398
S.D.	0.00000	0.00000	0.00000	0.00000

APPENDIX D

PROGRAM RFLFIT

The functions of this program are described on pp. 3 and 33. Table D1 lists editorial changes required preparatory to running the program. Table D2 lists step-by-step instructions for use during execution. The last section lists the program and explains its construction and function.

Table D1. Editorial Changes Preparatory to Running Program RFLFIT

Line(s)	Entry
65–78	Adjust array dimensions according to guide, lines 43–60. See p. 35 for calculation of NPTT. Guide provides minimum dimensions for all except READNG array.
103–105	Initialize control parameters. See lines 10–37 and Appendix G for definitions and limits.
115	Select control parameter for units. (1 for English lengths, 2 for metric).
116–118	Initialize conductor properties. These arrays can be initialized with the properties of air since they are later replaced by data from the disc files. The number of "air values" must match the array dimensions in line 70.
119	Initialize defect properties. All should be "0.0", and some are later replaced by file data. The number of values must match the array dimensions in line 71.
120	Initialize amplifier gains for each frequency used. The number of values must match the array dimension in line 69.

Table D2. Interactive Program Queries[a] and Operator Responses During Execution of RFLFIT

Line	QUERY[b] or Response
211	THERE IS AN ERROR IN THE NUMBER OF PROPERTIES
	If this message appears the program will halt execution by a STOP instruction (line 510) so that array dimensions can be adjusted (by editing) for consistency.
442	1 FIT PROP 2 PRT ENTIRE FIT 3 PRT/SV COEF 4 CHG FCTN/POL TYP 5 RUN TEST
	Enter integer 1—5 to select desired action.
	Definitions of optional actions are listed in terms that follow. Program branches and responds as though options were sequentially listed.
	Note: To stop execution option 1 or 5 must be selected so that execution will jump to line 503.
	1. Fit Property
502	PROP ARRAY IS FILLED
	This message occurs if the operator attempts to fit more than six properties during a single program run. Display at line 504 follows. (No response.)
	Otherwise, the sequence below occurs.
286	SELECT NUMBER OF THE PROPERTY TO BE FITTED: 1=RHO 2=MU 3=THICKNESS 4=LIFT-OFF 5=DEFECT LOCAT 6=DEFECT SIZE
289	SELECT REGION TO BE FITTED. LOWEST IS NUMBER 1.
	Enter integer 1—6 (property), delimiting blank (space), integer 1-NRT (region).
301	TYPE 1 IF THERE IS OFFSET, 0 IF NO OFFSET:
	Enter integer 0 or 1. [OFFSET (1) introduces a leading constant term into the least squares function expansion.]
307	SELECT THE NUMBER OF THE FUNCTION TYPE, POLYNOMIAL, DEGREE, & # OF CROSS TERMS FOR EACH MAGNITUDE & PHASE
311	FUNCTION TYPE; 1=LINEAR 2=LOG 3=EXP 4=INV
312	(blank line)

Table D2. (Continued)

Line	QUERY[b] or Response
314	FCTN POL # CROSS TYPE DEG TERMS
321	MAG AT (number) HZ
323	PHA AT (number) HZ (repeats for each frequency) On MAG lines enter integer 1–4 (function type), blank, integer 1–15 (polynomial degree), blank, integer 0. On PHA lines enter integer 1, 3, or 4 (function type), blank, integer 1–15 (polynomial degree), blank, integer 1–15 (no. of cross terms). The sum of POLDEG and # CROSS TERMS for each pair of MAG and PHA lines cannot exceed 15.
337	ERROR: # OF TERMS IN POLARY EXCEEDS DIMENSION This error message occurs if the sum of POLDEG and # CROSS TERMS exceeds 15 for a MAG and PHA pair. After this error occurs program execution returns to line 442 (above).
(362)	2. <u>Print Entire Fit</u> The fit is printed in hard copy and displayed on the CRT in the same format. See sample printout (p. 78) for display format.
(414)	3. <u>Print and Save Coefficients</u> No query or response. Table of coefficients is displayed on CRT, printed on hard-copy device, and written on disc file. See sample printout (p. 78) for display format.
(301)	4. <u>Change Function or Polynomial</u> Program execution jumps to line 301 (above) and continues through the remainder of the sequence for option 1.
(447)	5. <u>Run Test</u> This selection shifts program actions from the fitting of functions to data from standards to the active acquisition of new data from specimens and simulation of an instrument's microcomputer.

Table D2. (Continued)

Line	QUERY[b] or Response
504	PRINT OUT: 1. PARAMETERS 2. MEAS VOLT 3. CAL & DISPLAY PROPS 4. RAW RDGS 5. STOP
	Enter integer 1–5.
***	Definitions of the optional actions are listed in the items that follow, with letters A–E substituted for numbers 1–5 to avoid confusion with the above list of options.
(240)	A. Parameters
	No response; execution jumps to line 240 and prints a table of the conductor and defect property sets based on data read from the disc file (25). Then a table of circuit parameters is printed. See p. 145 for sample outputs.
(240)	B. Measured Voltages
	No response; execution jumps to line 240, skips table, and jumps again to line 260, skips second table, and resumes execution at line 270. Then a table of measured voltages is printed. (These are the voltages measured on the standard specimens read from file 25.)
(453)	C. Calculate and Display Properties
	No keyboard response. Program enters a loop to read instrument output and calculate properties by using stored bits. It stays in this loop until the foot pedal is depressed. A heading listing property titles is displayed. When the pedal is released the property values are computed and displayed.
(488)	D. Raw Readings
	No keyboard response. Execution jumps to same point in program as option C but skips property heading and displays voltages read from appropriate instrument channels.
(510)	E. Stop
	No response. Fortran statement halts execution.

[a]This table shows queries and other displays appearing on the CRT or other output terminal used by the operator (LI, LOTEK); hard-copy output appears on a printer (LOU).

[b]Queries in all caps simulate CRT presentation or computer printout.

LISTING AND ANALYSIS OF RFLFIT

Program Identification

```
1   C
2         PROGRAM RFLFIT
```

Date of Current Revision

```
3   C     (14 DECEMBER 1978)
```

Description and Variable Definitions

Lines 10–37 define various quantities used in the program. The term NRT is the total number of parallel layers, the first being farthest from the coil and the last being the air region where the coil is located. If a conducting sheet of finite thickness is used, the first region of "infinite" thickness and resistivity is air and the second region is the conducting sheet.

```
4   C     PROGRAM TO PERFORM A LEAST SQUARES FIT TO DATA READ INTO A DISK
5   C     FILE BY RFLRDG PROGRAM AND THEN PERFORM CONTINOUS BACK CALCULATIONS
6   C     USING INSTRUMENT READINGS THAT ARE MADE BETWEEN EACH DISPLAYED SET.
7   C     THE COEFFICIENTS ARE CONVERTED TO THE NDT-COMP8 OR COMP9 FORM
8   C     AND SAVED IN THE PROM DATA FILE IN THE CORRECT ORDER.
9   C
10  C     DFLOC=NORMALIZED DISTANCE OF DEFECT BELOW TOP OF THE REGION
11  C     DFRAD=NORMALIZED DISTANCE OF DEFECT FROM AXIS
12  C     DFVOL=NORMALIZED VOLUME OF DEFECT
13  C     IRDPRM=MAXIMUM NUMBER OF COEFFICIENTS IN EXPANSION
14  C     LI=LOGICAL INPUT UNIT (1=TTY)
15  C     LITEK=LSI INPUT UNIT FOR TAKING READINGS
16  C     LOTEK=LSI OUTPUT UNIT FOR PROMPT AND DISPLAY
17  C     LOU=LOGICAL OUTPUT UNIT FOR PERMANENT RECORD
18  C     MNDF=MAXIMUM NUMBER OF DEFECTS IN ANY REGION
19  C     MNT=MAXIMUM NUMBER OF THICKNESSES FOR ANY ONE RESISTIVITY
20  C     NCOED=STARTING ADDRESS IN PROM FOR COEFFICIENT DATA;SEE 8080 ASSEM.
21  C     NDEF=NUMBER OF DEFECTS IN A GIVEN SAMPLE: NDEF(IRHO,JTH)
22  C     NFILIM=FILE LIMIT ON NUMBER OF LINES IN FILE 25
23  C     NFT=NUMBER OF FREQUENCIES
24  C     NLT=NUMBER OF LIFTOFF INCREMENTS
25  C     NMGCAL=NUMBER OF MAGNITUDE CALIBRATIONS
26  C     NODF=TOTAL NUMBER OF DEFECTS
27  C     NPHCAL=NUMBER OF PHASE CALIBRATIONS
28  C     NPRINT=PRINT AND TRANSFER INDEX
29  C     NPROPM=MAXIMUM NUMBER OF PROPERTIES THAT WILL BE CALCULATED
30  C     NPROPT=INDEX FOR PROPERTY BEING STUDIED (1 TO NPROPM)
31  C     NPT=TOTAL NUMBER OF PROPERTIES WITHOUT DEFECTS
32  C     NPTT=TOTAL NUMBER OF PROPERTIES + NUMBER OF DEFECTS
33  C     NRDF=NUMBER OF THE REGION WHERE DEFECT IS FOUND(=NRT IF NO DEFECT)
```

```
34  C      NRDG=NPTT*NLT+1=FIRST DIMENSION OF READNG ARRAY
35  C      NRREP=NUMBER OF REGION BEING STUDIED(1=LOWEST)
36  C      NRES=NUMBER OF RESISTIVITIES
37  C      NRT=NUMBER OF CONDUCTORS + 1
```

Fixed-Dimension Arrays

```
38  C
39         DIMENSION HDR(2),HPU(2),ITIM(3),IDAY(3),PROPTY(6),PRONAM(6)
```

Guide for Adjustable-Dimension Arrays

The dimensions in lines 43–61 are listed symbolically in terms of the quantities defined above, and the appropriate numbers should be entered in the actual dimension statements in lines 65–79. Actually, the dimensions can be larger than required (within storage limits of the computer) without causing any difficulty, *except for* READNG *in line 65, which must have its first dimension exactly equal to* NPTT*NLT+1. See p. 35 for computation of NPTT.

```
40         DIMENSION NCONV(3),CONVRT(2),UNITS(6,2),VOLTS(12)
41  C
42  C                    DIMENSIONS THAT ARE CHANGED.
43  C *** DIMENSION READNG(NRDG,IRDPRM+1),NPOL(6,NFT)
44  C     *         ,JOFSET(NPROPM),JRDPR(NPROPM)
45  C *** DIMENSION TMAGM(NLT,NPTT,NFT),PHASEM(NLT,NPTT,NFT)
46  C      DIMENSION POLARY(IRDPRM+1,5),PRO(NPTT*NLT),PROP(NPTT*NLT,NPROPM)
47  C     *          ,RL1(NLT),RL2(NLT)
48  C      DIMENSION TMDFT(NFT),PHDFT(NFT),JPOL(6,NFT,NPROPM),
49  C     *          COE(IRDPRM),COEF(IRDPRM,NPROPM)
50  C      DIMENSION NRDF(NPT),FREQ(NFT),GAIN(NFT)
51  C      DIMENSION RHO(NRT,NPTT),U(NRT,NPTT),TH(NRT,NPTT)
52  C      DIMENSION XLOC(NRT,NPTT),XRAD(NRT,NPTT),XVOL(NRT,NPTT)
53  C      DIMENSION SDVPHA(NLT,NPTT,NFT),SDVMAG(NLT,NPTT,NFT)
54  C      DIMENSION PICKAM(NFT),PHADET(NFT),PHASW(NFT)
55  C      DIMENSION DCMGCB(NFT),DCMGAR(NFT),POWAMP(NFT)
56  C      DIMENSION LIFT(NLT),THICK(NRES,MNT),XRHO(NRES)
57  C      DIMENSION NTH(NRES),NDEF(NRES,MNT)
58  C      DIMENSION DFLOC(NRES,MNT,MNDF),DFRAD(NRES,MNT,MNDF),
59  C     *          DFVOL(NRES,MNT,MNDF)
60  C      DIMENSION SUMCPH(NFT,NPHCAL,NMGCAL),SUMCMG(NFT,NPHCAL,NMGCAL)
61  C      DIMENSION SDVCPH(NFT,NPHCAL,NMGCAL),SDVCMG(NFT,NPHCAL,NMGCAL)
```

Adjusted-Dimension Arrays

(The starred arrays are found in the guide above.)

```
62  C      THESE DIMENSIONS USUALLY MUST BE CHANGED,WHEN THE PROBLEM IS
63  C      CHANGED.THE STARRED(***) ARRAYS MUST BE EXACT,AND THE OTHERS MUST
64  C      AT LEAST BE LARGE ENOUGH.
65         DIMENSION READNG(82,16),NPOL(6,2),JOFSET(6),JRDPR(6)
```

```
      DIMENSION TMAGM(3,27,3),PHASEM(3,27,3)
      DIMENSION POLARY(16,5),PRO(81),PROP(81,6),RL1(3),RL2(3)
      DIMENSION TMDFT(3),PHDFT(3),JPOL(6,3,6),COE(15),COEF(15,6)
      DIMENSION FREQ(4),GAIN(4)
      DIMENSION RHO(3,27),U(3,27),TH(3,27)
      DIMENSION XLOC(3,27),XRAD(3,27),XVOL(3,27)
      DIMENSION SDVPHA(3,27,3),SDVMAG(3,27,3)
      DIMENSION PICKAM(4),PHADET(4),PHASW(4)
      DIMENSION DCMGCB(4),DCMGAR(4),POWAMP(4)
      DIMENSION LIFT(5),THICK(5,5),XRHO(5)
      DIMENSION NTH(5),NDEF(5,5)
      DIMENSION DFLOC(5,5,4),DFRAD(5,5,4),DFVOL(5,5,4)
      DIMENSION SUMCPH(3,3,3),SUMCMG(3,3,3)
      DIMENSION SDVCPH(3,3,3),SDVCMG(3,3,3)
```

Real and Double-Precision Variables

The L-variables correspond to prior local definition and usage. Higher space allocation is required for alphanumeric variable storage.

```
      REAL L2,L4,L3,L5,L6,LIFT
      REAL*6 NPROBE,NCABLE,INSTNO,POWOSC,PICKAM,PHADET
      REAL*8 TITLE(2)
```

Common Block Storage

Block B1 transfers coil electrical data to subroutines computing electrical response. Blocks B1 and B2 are used when complete coil data are transferred.

```
      COMMON /B2/R1,R2,R3,R4,L3,L4,RBAR,V0,ZLDR,ZLPU,
     *TNDR,TNPU,RDCDR,RDCPU,R0,R9,CAPDR,CAPPU
      COMMON /B6/ L2,L5,L6
```

Random Access Data File Definition

```
C
C     FILE 28 CONTAINS THE COIL DATA.
      DEFINE FILE 28(80,32,U,NCOIL)
C
C     FILE 25 CONTAINS THE EXPERIMENTAL MEASUREMENTS MADE USING RFLRDG.
      DEFINE FILE 25(200,40,U,IREC)
C
C     FILE 31 WILL STORE THE COEFFICIENT DATA FOR THE MICROCOMPUTER.
      DEFINE FILE 31(8192,4,U,NCOED)
C
C     FILE 21 WILL CONTAIN THE COEFFICIENTS OF THE FITS.
      DEFINE FILE 21(30,40,U,ICOEF)
```

Data Initialization

The data in lines 99—102 will not need to be changed unless the properties being fitted or their units are changed, but the data in lines 103—105 may need changing, especially if the designations of the input and output terminals are changed.

See Table Cl and Appendix G for definitions and limits.

```
 98   C
 99          DATA TWOPI/6.28318531/,RAD/57.2957795/,CONVRT/1.,25.4/,NLABL/1/
100          DATA PROPTY/4HREST,4HPERM,4HTHIK,4HL.O.,4HDFLC,4HDFSI/
101          DATA UNITS/4HMOCM,4HREL.,4HINCH,4HINCH,4HINCH,4HCUIN,
102         *          4HMOCM,4HREL.,4H MM ,4H MM ,4H MM ,4HCUMM/
103          DATA NRT/3/,NRREP/2/,NCHS/12/,ITIMS/1024/,NPROPM/6/,NPROPT/1/
104          DATA NPRINT/0/,LOU/6/,LITEK/1/,LOTEK/12/,LI/7/,IR/1/,IRDPRM/15/
105          DATA NCOED/1680/,NRDG/82/,NFILIM/200/
106   C
107   C        THE MATERIAL PROPERTIES NOW FOLLOW,STARTING WITH THE FIRST
108   C        PROPERTY SET FOR THE LOWERMOST REGION. (RHO=RESISTIVITY IN
109   C        MICROHM-CM;U=RELATIVE PERMEABILITY;TH=THICKNESS IN INCHES)
110   C        SOME OF THESE VALUES ARE CHANGED TO THE VALUES IN THE DATA SET,
111   C        READ FROM FILE 25. THERE ARE NRT*NPTT VALUES FOR RHO,U,,TH&XVOL.
112   C        IF NUNIT=1, LENGTHS WILL BE PRINTED OUT IN INCHES; IF NUNIT=2,
113   C        LENGTHS WILL BE PRINTED OUT IN MILLIMETERS.
114   C
115          DATA NUNIT/2/
116          DATA RHO/81*1.E12/
117          DATA U/81*1./
118          DATA TH/81*1.E10/
119          DATA XVOL/81*0.0/
```

Output Heading

```
120          DATA GAIN/5.8025E3,3.1094E2,2*3.E3/
121          CALL TIME(ITIM)
122          CALL DATE(IDAY)
123          IT=IDAY(1)-1900
124          IDAY(1)=IDAY(2)
125          IDAY(2)=IDAY(3)
126          IDAY(3)=IT
127          WRITE (LOU,5)
128     5    FORMAT (1H1,20X)
129          WRITE (LOU,10) IDAY, ITIM
130    10    FORMAT (10X,'PROGRAM RFLFIT DATE   ', 2(I2,1H/), I2,
131         1  ' TIME   ', 2(I2,1H:), I2)
```

Data Entry from RAD Files

The data written in RAD file 25 during the running of the RFLRDG program are read.

```
132          NCOED=NCOED+1
133   C  READ INFORMATION IN DIRECT ACCESS FILE #25 ON DISK
134   C
```

```
135           IREC=1
136           READ(25'IREC) TITLE,IDAY,ITIM
137           READ(25'IREC)NPROBE,NSER,RO,CAPDR,R9,CAPPU,NCABLE,CABLEL,
138          *                 CCABLE,INSTNO,POWOSC
139           READ(25'IREC) NFT,NLT,NRES,NPTT,NSAMPL,NPHCAL,NMGCAL
140           READ(25'IREC) ((FREQ(I),PICKAM(I),PHADET(I),PHASW(I),
141          *                 DCMGCB(I),DCMGAR(I),POWAMP(I)),I=1,NFT)
142           READ(25'IREC) (LIFT(I),I=1,NLT)
143           DO 20 I=1,NRES
144           READ(25'IREC) NTHI,XRHO(I)
145           NTH(I)=NTHI
146           READ(25'IREC) ((THICK(I,J),NDEF(I,J)),J=1,NTHI)
147           DO 20 J=1,NTHI
148           NDIJ=NDEF(I,J)
149           IF(NDIJ.EQ.0) GO TO 20
150           READ(25'IREC)((DFLOC(I,J,K),DFRAD(I,J,K),DFVOL(I,J,K)),
151          *K=1,NDIJ)
152     20    CONTINUE
153           DO 30 I=1,NFT
154           DO 30 J=1,NPHCAL
155           READ(25'IREC)((SUMCPH(I,J,K),SUMCMG(I,J,K),SDVCPH(I,J,K),
156          *SDVCMG(I,J,K)),K=1,NMGCAL)
157     30    CONTINUE
158           DO 40 K=1,NFT
159           DO 40 J=1,NPTT
160           READ(25'IREC)((PHASEM(I,J,K),TMAGM(I,J,K),SDVPHA(I,J,K),
161          *SDVMAG(I,J,K)),I=1,NLT)
162     40    CONTINUE
163           IF(IREC.EQ.NFILIM) GO TO 960
```

Coil Data Entry

The subroutine COLFIL, stored in the UL, is now called up to obtain the coil parameters. It will hunt in File 28 for a coil with exactly the same name as NPROBE, read from File 25. If such a coil is found in File 28, the parameters of that coil will be transferred back to the main program in the COMMON BLOCKS B2 and B6. If no such coil name is found in the file, the user will be prompted to type in the necessary parameters, which will then be transferred back in the same way.

```
164   C
165   C       THE INPUT DATA FOR THE COIL PARAMETERS ARE READ FROM FILE 28
166   C       USING THE COIL NAME READ FROM FILE 25.SEE FIG.2,P.4
167   C       AND FIG.4, P.7, ORNL-TM-4107, FOR DEFINITIONS.
168   C
169           CALL COLFIL(LITEK,LOTEK,NPROBE)
```

Construction of Property Arrays

The region of study, with index NR, is set equal to the value of NRREP set in the DATA statement in line 103. The property index NP, defect number NODF, and total number of properties NPT are initialized in lines 174–177.

The DO loops over all NRES resistivities and the NTHI thickness fill the resistivity (RHO) and thickness (TH) arrays by using the values from XRHO and THICK read from file 25. The first time through the loops, for a given sample resistivity and thickness, the defect volume (XVOL) is left at the value 0.0 given in the DATA statements, and the defect location (XLOC) and radial location (XRAD) are set at the nominal values of half the sample thickness and three-fourths of the mean coil radius, respectively. Then the property index NP and the number of samples NPT are incremented, and the DO loops are continued to the next sample if there are no defects in the given sample. The length dimensions TH, XLOC, and XRAD are converted to the desired set of units by the conversion factor CONVRT, chosen by the index NUNIT set in the first DATA statement (line 115).

```
170  C
171  C      RESISTIVITIES, THICKNESSES AND DEFECTS ARE SET FROM THE
172  C      DATA READ FROM FILE #25.
173  C
174         NR=NRREP
175         NP=1
176         NODF=0
177         NPT=0
178  C
179  C      SET THE VALUES OF RHO AND THICKNESS IN THE REGION OF STUDY,NRREP.
180  C      LOOP OVER ALL NRES RESISTIVITIES AND NTHI THICKNESSES.
181  C
182         DO 60 NRE=1,NRES
183         NTHI=NTH(NRE)
184         DO 60 NT =1,NTHI
185         RHO(NR,NP)=XRHO(NRE)
186         TH(NR,NP)=THICK(NRE,NT)*CONVRT(NUNIT)
187         XLOC(NR,NP)=.5*THICK(NRE,NT)*CONVRT(NUNIT)
188         XRAD(NR,NP)=.75*RBAR*CONVRT(NUNIT)
189         NPT=NPT+1
190         NP=NP+1
191         IF(NDEF(NRE,NT).EQ.0) GO TO 60
```

If the number of defects NDEF in the given sample is not zero, then a DO loop is started to cycle over all NDRT defects in the given sample. The values of RHO and TH are set to be the same as for the undefective sample, but the values of XLOC, XRAD, and XVOL are set for the actual values as read from File 25 and converted to the desired units by the factor CONVRT. Then the number of defects NODF and the property index NP are incremented, and the DO loops are continued. Lines 209–211 test that the final property index NP is in agreement with the total number of properties NPTT read from file 25 and print an error message if not.

```
C
C      BRANCH AROUND IF THERE ARE NO DEFECTS,SET LOC.IN WALL,RADIAL LOC.
C      AND SIZE IF THERE ARE DEFECTS.
C
C      CYCLE OVER THE NUMBER OF DEFECTS PER SAMPLE
       NDRT=NDEF(NRE,NT)
       DO 50 NDR=1,NDRT
       RHO(NR,NP)=XRHO(NRE)
       TH(NR,NP)=THICK(NRE,NT)*CONVRT(NUNIT)
       XLOC(NR,NP)=DFLOC(NRE,NT,NDR)*CONVRT(NUNIT)
       XRAD(NR,NP)=DFRAD(NRE,NT,NDR)*CONVRT(NUNIT)
C       THE DEFECT VOLUME (SIZE) IS CALCULATED IN CUBIC MILLIMETERS.
       XVOL(NR,NP)=DFVOL(NRE,NT,NDR)*((CONVRT(NUNIT))**3)
       NODF=NODF+1
       NP=NP+1
    50 CONTINUE
    60 CONTINUE
       IF(NP.EQ.NRDG)GO TO 75
       WRITE(LOTEK,70)
    70 FORMAT(' THERE IS AN ERROR IN THE NUMBER OF PROPERTIES.')
```

Phase Data Conversion

The phases read from file 25 are all multiplied by 10 to convert from instrument voltage (0.1 V/deg) to degrees.

```
       GO TO 1000
    75 DO 80 NF=1,NFT
       DO 80 NP=1,NPTT
       DO 80 NL=1,NLT
       PHASEM(NL,NP,NF)=10.*PHASEM(NL,NP,NF)
    80 CONTINUE
```

Date and Time of File Data Recording

```
C
C       THE DATE AND TIME ARE PRINTED
C
       WRITE (LOU,90) IDAY,ITIM
    90 FORMAT (' CALIBRATION DATA TAKEN   ',2(I2,1H/),I2,' TIME  ',
     *2(I2,1H:),I2)
```

Lift-Off Increment and Circuit Value Corrections

The lift-off increment L2 and some circuit parameters are calculated from data read from file 25.

```
C
C      THE CIRCUIT PARAMETERS ARE NOW CALCULATED FROM FILE 25 DATA
C      L2=NORMALIZED LIFTOFF INCREMENT
C      CAPDR=DRIVER SHUNT CAPACITANCE(FARADS)
C      CAPPU=PICKUP SHUNT CAPACITANCE(FARADS)
C      THE CABLE CAPACITANCE IS ASSUMED TO BE THE SAME FOR EACH LEG
C      AND ADDED TO BOTH DRIVER AND PICKUP CIRCUITS.
C      VO=DRIVER OUTPUT VOLTAGE(AC/DC CONVERTOR OUTPUT VOLTS)
C
       L2=(LIFT(2)-LIFT(1))/RBAR
       CAPDR=CAPDR+CCABLE
       CAPPU=CAPPU+CCABLE
       VO=POWAMP(1)
```

Property Table Printout

If the print index NPRINT equals 1, the properties for each region and samples are printed on the LOU output unit.

```
C
C      EACH SET OF PROPERTIES FOR EACH REGION IS NOW PRINTED OUT
C
  100  IF (NPRINT.NE.1) GO TO 160
       WRITE(LOU,110)
  110  FORMAT(' PR.SET  REG  THICKNESS    RESTVY      PERM   DEF LOC
      1DEF RAD  DEF SIZ')
       WRITE(LOU,115)UNITS(3,NUNIT),UNITS(1,NUNIT),UNITS(2,NUNIT),
      *UNITS(5,NUNIT),UNITS(5,NUNIT),UNITS(6,NUNIT)
  115  FORMAT(16X,A4,8X,A4,7X,A4,5X,A4,5X,A4,5X,A4)
C
       NPP=1
       DO  160 NP=1,NPTT
       WRITE(LOU,120)
  120  FORMAT (1X)
       DO 150 NR=1,NRT
       WRITE(LOU,130)NPP,NR,TH(NR,NPP),RHO(NR,NPP),U(NR,NPP)
  130  FORMAT(1X,2(I5),2(1PE12.4),0PF9.3)
       IF(XVOL(NR,NPP).NE.0.)WRITE(LOU,140)XLOC(NR,NPP),XRAD(NR,NPP)
      *,XVOL(NR,NPP)
  140  FORMAT('+',44X,2(F9.4),,1PE12.4)
  150  CONTINUE
       NPP=NPP+1
  160  CONTINUE
```

Circuit Parameter Printout

Also if NPRINT=1 the subroutine PRFCKT is called to print the circuit parameters.

```
C
C      THE CIRCUIT PARAMETERS ARE PRINTED OUT FOR THE REFLECTION
C      COIL CIRCUIT.
C
       IF (NPRINT.EQ.1) CALL PRFCKT(LOU)
```

Output Voltage (VOUT) Magnitude and Phase Printout

If NPRINT=2 subroutine PRFVLT is called to print the voltage magnitudes and phases; then control is returned to the choices at Fortran label number 900 or 540, depending upon the value of NPRINT.

```
266  C
267  C        THE VOLTAGE MAGNITUDE,PHASE,AND SHIFT ARE PRINTED FOR
268  C        THE VARIOUS LIFT-OFF VALUES
269  C
270          IF(NPRINT.EQ.2)WRITE(LOU,170)
271    170   FORMAT('0  MEASURED VOLTAGES')
272          IF (NPRINT.EQ.2) CALL PRFVLT(LOU,TMAGM,PHASEM,NRT,NPT,NFT,NLT,NODF
273         1,NPTT,RL1,NRDF,FREQ,GAIN)
274          NLABL=1
275          IF(NPRINT.NE.0)GO TO 900
276          GO TO 540
```

Least Squares Function Fitting

Before executing this section the program branches to statement 540 (line 442) to select the next action. Execution proceeds as though the action selection preceded this section. In this section a user-selected function is fitted to the particular property being studied.

Property Selection

First, the user chooses the property and the region where the conductor or defect having the property is located. (See Table D2 for interactive responses.)

```
277  C
278  C      LEAST SQUARES DESIGN SECTION.
279  C
280  C      SELECT PROPERTY TO BE FITTED AND SET UP PROPERTY ARRAY.
281  C
282   200   MSET=NLT*NPTT
283         MSET1=MSET+1
284         IF(NPROPT.GT.NPROPM) GO TO 850
285         WRITE(LOTEK,210)
286   210   FORMAT(' SELECT NUMBER OF THE PROPERTY TO BE FITTED:'/
287        *' 1=RHO 2=MU 3=THICKNESS 4=LIFT-OFF 5=DEFECT LOCAT 6=DEFECT SIZE')
288         WRITE(LOTEK,220)
289   220   FORMAT(' SELECT REGION TO BE FITTED. LOWEST IS NUMBER 1.')
290         READ(LITEK,*)NPROP,NREG
```

The array PROP is now set up with data in the first column, since NPROPT was initialized at 1 in line 103.

The element in the *M*th row will be the NPROP property just selected and located in the NREG region.

```
291          DO 230 NP=1,NPTT
292          DO 230 NL=1,NLT
293          M=(NP-1)*NLT+NL
294          IF(NPROP.EQ.1)PROP(M,NPROPT)=RHO(NREG,NP)
295          IF(NPROP.EQ.2)PROP(M,NPROPT)=U(NREG,NP)
296          IF(NPROP.EQ.3)PROP(M,NPROPT)=TH(NREG,NP)
297          IF(NPROP.EQ.4)PROP(M,NPROPT)=(FLOAT(NL-1)*L2)*RBAR
298          IF(NPROP.EQ.5)PROP(M,NPROPT)=XLOC(NREG,NP)
299          IF(NPROP.EQ.6)PROP(M,NPROPT)=XVOL(NREG,NP)
300     230  CONTINUE
```

<u>Fitting Function Selection</u>

Selection of OFFSET adds a constant term to the fitting function. See Table D2 for prompts and responses in selecting the function. The accumulation of the number of terms in the polynomial is started in line 305.

```
301     240  WRITE(LOTEK,250)
302     250  FORMAT(' TYPE 1 IF THERE IS OFFSET, 0 IF NO OFFSET:')
303          READ(LITEK,*)JOFSET(NPROPT)
304          IOFSET=JOFSET(NPROPT)
305          IRDPR=IOFSET
```

If more than IRDPRM terms (now set at 15 in the DATA statement) are selected for the polynomial (excluding the OFFSET), the error message in statement 330 (line 337) will be displayed.

The subroutine POLTYP constructs an array, POLARY, for printing a representation of the polynomial expansion. (See p. 166 for POLYTP subroutine.)

```
306          WRITE(LOTEK,260)
307     260  FORMAT(' SELECT THE NUMBER OF THE FUNCTION TYPE, POLYNOMIAL',
308         1' DEGREE, & # OF '/,
309         2' CROSS TERMS FOR EACH MAGNITUDE & PHASE')
310          WRITE(LOTEK,270)
311     270  FORMAT(' FUNCTION TYPE:1=LINEAR 2=LOG 3=EXP 4=INV ')
312          WRITE(LOTEK,120)
313          WRITE(LOTEK,280)
314     280  FORMAT(25X,'FCTN POL # CROSS'/,25X'TYPE DEG TERMS')
315          DO 320 NF=1,NFT
316          DO 310 NC=1,2
```

```
317          NCC=NC*3
318          NCP=NCC-1
319          NCF=NCP-1
320          IF(NC.EQ.1) WRITE(LOTEK,290) FREQ(NF)
321     290  FORMAT(' MAG AT ',1PE12.6,' HZ    ')
322          IF(NC.EQ.2) WRITE(LOTEK,300) FREQ(NF)
323     300  FORMAT(' PHA AT ',1PE12.6,' HZ    ')
324          READ(LITEK,*)
325         *JPOL(NCF,NF,NPROPT),JPOL(NCP,NF,NPROPT),JPOL(NCC,NF,NPROPT)
326          JPOL(3,NF,NPROPT)=0
327          IRDPR=IRDPR+JPOL(NCP,NF,NPROPT)+JPOL(NCC,NF,NPROPT)
328          JRDPR(NPROPT)=IRDPR
329          NPOL(NCF,NF)=JPOL(NCF,NF,NPROPT)
330          NPOL(NCP,NF)=JPOL(NCP,NF,NPROPT)
331          NPOL(NCC,NF)=JPOL(NCC,NF,NPROPT)
332     310  CONTINUE
333     320  CONTINUE
334          IRDPR1=IRDPR+1
335          IF(IRDPRM.LT.JRDPR(NPROPT))WRITE(LOTEK,330)
336          IF (IRDPRM.LT.JRDPR(NPROPT))GO TO 540
337     330  FORMAT(' ERROR: # OF TERMS IN POLARY EXCEEDS DIMENSION')
338          JROW=IRDPRM+1
```

Expansion into Functions of the Readings

Next, the magnitude and phase drifts, TMDFT and PHDFT, are initialized to zero, and then the sets of measured readings of magnitude and phase, TMAGM and PHASEM, for each set of properties are expanded according to the chosen polynomial into the array READNG. Subroutine RDGEXP constructs the polynomial expansions, and a total of NPTT*NLT rows of the READNG array will be determined in this way. The corresponding column matrix, PRO, for the chosen property is also set up at the same time in line 352. Actually, the READNG matrix has one more row and column than are determined by the DO loops in lines 347–348. The extra row and column are used for the calculations in the ALSQS least squares program. In fact, the last column eventually contains the property values calculated by using the least squares coefficients upon return from the ALSQS subroutine.

```
339          CALL POLTYP(POLARY,JROW,IRDPR,NPOL,6,NFT,2,IOFSET,LOTEK)
340   C
341   C       EXPAND THE RAW READINGS INTO IRDPR READINGS.
342   C
343          DO 340 NF=1,NFT
344          TMDFT(NF)=0.
345          PHDFT(NF)=0.
346     340  CONTINUE
347          DO 360 NP=1,NPTT
348          DO 350 NL=1,NLT
349          M=(NP-1)*NLT+NL
350          CALL RDGEXP(READNG,TMAGM,PHASEM,NPOL,IOFSET,TMDFT,PHDFT,
351         1MSET1,IRDPR1,M,NFT,NL,NLT,NP,NPTT)
352          PRO(M)=PROP(M,NPROPT)
353     350  CONTINUE
354     360  CONTINUE
```

Calculation of Coefficients and Approximating Vector

Upon return from ALSQS the column matrix COE contains the least squares coefficients that give the "best" fit to the property column matrix PRO by using the polynomial terms written into READNG. Mathematically, in matrix notation ALSQS minimizes the determinant of (READNG)(COE) — (PRO). Also, upon return the quantity RSOS contains the residual sum of squares of the differences between the properties and the values calculated by using the least squares coefficient. These calculated values are the elements in the last column of READNG upon return from ALSQS. (See p. 151 for ALSQS subroutine.)

```
355  C
356  C      DO THE LEAST SQUARES FIT OF THE READINGS TO THE PROPERTIES.
357  C
358         CALL ALSQS(READNG,PRO,COE,RSOS,MSET,IRDPR,MSET1)
```

Tests of Fit Quality

Two properties of the fit are calculated: the rms difference between the property value from which the fit was constructed and the value computed from the fit and the effect of small errors in the instrument readings on the values computed from the fit function. Details of these facts are discussed on pp. 39—41.

The sums of the squares of the drifts and differences are initialized in lines 362 and 363. Then the Property Fit table heading is printed and displayed if NPRINT=2. (Selection occurs in line 445.)

```
359  C
360  C      CALCULATE THE DIFFERENCES IN THE FIT AND THE MAXIMUM DRIFTS.
361  C
362   400   SSDRIF=0.
363         SSDIFF=0.
364         UNIT=UNITS(NPROP,NUNIT)
365         IF(NPRINT.EQ.2)WRITE(LOU,410)PROPTY(NPROP),NREG,PROPTY(NPROP),
366        *UNIT,UNIT,UNIT,UNIT
367         IF(NPRINT.EQ.2)WRITE(LOTEK,410)PROPTY(NPROP),NREG,PROPTY(NPROP),
368        *UNIT,UNIT,UNIT,UNIT
369   410   FORMAT('0 NP NL ',A4,' IN REG',I3,' CAL ',A4,7X,'DIFF',7X,'DRIFT'
370        *,/,4X,4(8X,A4),/)
```

Now the drift calculation is looped over all NPTT property sets and NLT lift-offs. The drift is initialized to zero for each set of conditions. Then the calculation is looped over all frequencies and over NC. When NC=1 a 0.01% error in the magnitude is introduced into each reading at that frequency; when NC=2 a 0.01 degree error in the phase is introduced. In each case a polynomial using the erroneous magnitude or phase is constructed by RDGEXP; all the other magnitudes and phases retain their correct values.

```
371           DO 470 NP=1,NPTT
372           DO 460 NL=1,NLT
373           M=(NP-1)*NLT+NL
374           DRIFT=0.
375           DO 440 NF=1,NFT
376           DO 430 NC=1,2
377   C
378   C       ONE MAGNITUDE OR PHASE DRIFT IS SET ON AT A TIME.
379   C
380           IF(NC.EQ.1)TMDFT(NF)=.0001*TMAGM(NL,NP,NF)
381           IF(NC.EQ.2)PHDFT(NF)=.01
382           CALL RDGEXP(READNG,TMAGM,PHASEM,NPOL,IOFSET,TMDFT,PHDFT
383          1,MSET1,IRDPR1,M,NFT,NL,NLT,NP,NPTT)
384   C
385   C       THE POLYNOMIAL IS CALCULATED
386   C
387           SUM=0.
388           DO 420 IR=1,IRDPR
389           SUM=SUM+COE(IR)*READNG(M,IR)
390     420   CONTINUE
```

The sums of the squares of the differences and drifts are accumulated in SSDIFF and SSDRIF (lines 397—398).

```
391           DRIFT=DRIFT+ABS(READNG(M,IRDPR1)-SUM)
392           TMDFT(NF)=0.
393           PHDFT(NF)=0.
394     430   CONTINUE
395     440   CONTINUE
396           DIFF=PRD(M)-READNG(M,IRDPR1)
397           SSDIFF=SSDIFF+DIFF*DIFF
398           SSDRIF=SSDRIF+DRIFT*DRIFT
```

If NPRINT=2 the entire fit is printed (selection made at line 442). This print option is included in the SDIFF/SDRIF calculation loop because the same calculation steps are required to generate the printout of the entire fit for the selected function.

```
399           IF(NPRINT.NE.2)GO TO 460
400   C
401   C        THE ENTIRE FIT IS PRINTED OUT
```

```
402  C
403           WRITE(LOU,450)NP,NL,PRO(M),READNG(M,IRDPR1),DIFF,DRIFT
404           WRITE(LOTEK,450)NP,NL,PRO(M),READNG(M,IRDPR1),DIFF,DRIFT
405     450   FORMAT(1X,2I3,4F12.5)
406     460   CONTINUE
407     470   CONTINUE
```

The standard deviations of the drift and differences, SDRIF and SDIFF, are calculated in lines 408 and 409 and printed out.

```
408           SDRIF=SQRT(SSDRIF/FLOAT(MSET))
409           SDIFF=SQRT(SSDIFF/FLOAT(MSET))
410           WRITE(LOU,480)PROPTY(NPROP),UNIT,NREG,SDIFF,SDRIF
411           WRITE(LOTEK,480)PROPTY(NPROP),UNIT,NREG,SDIFF,SDRIF
412     480   FORMAT(' DIFF IN ',A4,' IN ',A4,' IN REG #',I3,'=',F12.5,' DRIFT
413          *= ',F12.5)
```

Printing, HEX Conversion, and "Saving" of Coefficients

If NPRINT=3 (selected at line 442 the coefficients are: converted to binary hex notation for microcomputer use by the subroutine CONVR8, printed at both output terminals, and written in hexadecimal on RAD file 31 and in decimal on RAD file 21. Then the property index NPROPT is incremented. PRONAM stores the names of the properties in the order that the coefficients are saved.

```
414     500   IF (NPRINT.NE.3)GO TO 540
415           WRITE(LOU,120)
416           WRITE(LOTEK,120)
417           NCOED1=NCOED+3*IRDPRM+1
418           WRITE(31'NCOED)IRDPR
419           DO 520 I=1,IRDPR
420           CALL CONVR8(COE(I),NCONV)
421           WRITE(LOU,510)I,COE(I),(NCONV(II),II=1,3),(POLARY(I,J),J=1,5)
422           WRITE(LOTEK,510)I,COE(I),(NCONV(II),II=1,3),(POLARY(I,J),J=1,5)
423     510   FORMAT(' COEF(',I3,')=',1PE15.7,4X,3(Z2,1X),1X,5A4)
424           COEF(I,NPROPT)=COE(I)
425           WRITE(31'NCOED)NCONV(2)
426           WRITE(31'NCOED)NCONV(3)
427           WRITE(31'NCOED)NCONV(1)
428     520   CONTINUE
429           NCOED=NCOED1
430           PRONAM(NPROPT)=PROPTY(NPROP)
431  C
432  C        THE COEFFICIENT,OFFSET,NPOL AND IRDPR ARE WRITTEN ON THE RAD DISC
433  C        FILE #21.
434  C
435           ICOEF=1
436           WRITE(21'ICOEF)PROPTY(NPROP),IRDPR,JOFSET(NPROPT)
437           WRITE(21'ICOEF)(COEF(IR,NPROPT),IR=1,IRDPR)
438           DO 530 NF=1,NFT
439           WRITE(21'ICOEF)(NPOL(I,NF),I=1,6)
440     530   CONTINUE
441           NPROPT=NPROPT+1
```

Selection of the Next Action

Although listed later this section is executed earlier in the operating sequence by jumps from lines 276, 336 (conditional), and 414 (conditional).

The first option starts the fitting of a new property; the second provides a complete printout of the fit of the property just completed; the third causes the coefficients of the fit to be printed and also saved in files 21 and 31; the fourth offers a chance to change the types of functions and polynomials to be used in fitting the same property as before; and the last option leads to running a test by using the coefficients already determined (or to stop).

```
442    540  WRITE(LOTEK,550)
443    550  FORMAT(' 1 FIT PROP 2 PRT ENTIRE FIT 3 PRT/SV COEF 4 CHG FCTN/POL'
444        1,'TYP 5 RUN TEST')
445         READ(LITEK,*)NPRINT
446         GO TO(200,400,500,240,600),NPRINT
```

The choice to RUN TEST shifts execution to statement number 600, which sets the value of NPROPT back to its last value since it had been incremented in line 441.

```
447    600  NPROPT=NPROPT-1
448         WRITE(LOU,120).
449         GO TO 900
```

At statement 900 further choices are offered: to print out (1) the parameters (the material properties and circuit constants), (2) the magnitudes and phases of the voltages measured for each of the properties, (3) calculated values of newly measured properties, (4) the current instrument readings in V, or to (5) stop. The first two choices are useful for a permanent record; they both return control to statement 100 but with different values of NPRINT. Option 3 allows measurement of an unknown sample; it transfers control to statement 700, which displays headings to identify properties for which fits are available. The following steps take instrument readings of the magnitudes and phases at the NFT frequencies, substitute them into the polynomials by using RDGEXP (line 470), and then calculate each of the NPROPT properties (lines 472–475).

```
450  C      CALCULATES PROPERTIES FROM MAGNITUDES AND PHASES AND CONTINOUSLY
451  C      THE VALUES ON THE LSI TERMINAL.
452  C
453   700   IF(NPRINT.EQ.3)WRITE(LOTEK,710)(PRONAM(NPRO),NPRO=1,NPROPT)
454   710   FORMAT(1X,6(6X,A4,3X))
455         WRITE(LOTEK,120)
456  C
457  C      EXPANSION OF TMAGM(NL,NP,NF) AND PHASEM(NL,NP,NF) INTO READNG(1,IRDPR
458  C
459   720   DO 760 NPRO=1,NPROPT
460         DO 740 NF=1,NFT
461         DO 730 I=1,6
462         NPOL(I,NF)=JPOL(I,NF,NPRO)
463   730   CONTINUE
464         TMDFT(NF)=0.
465         PHDFT(NF)=0.
466   740   CONTINUE
467         IOFSET=JOFSET(NPRO)
468         IRDPR=JRDPR(NPRO)
469         IRDPR1=IRDPR+1
470         CALL RDGEXP(READNG,TMAGM,PHASEM,NPOL,IOFSET,TMDFT,PHDFT,
471        1MSET1,IRDPR1,1,NFT,1,NLT,1,NPTT)
472         PRO(NPRO)=0.
473         DO 750 IR=1,IRDPR
474         PRO(NPRO)=PRO(NPRO)+COEF(IR,NPRO)*READNG(1,IR)
475   750   CONTINUE
476   760   CONTINUE
477         NSTART=1
478         IF(NPRINT.EQ.3)WRITE(LOTEK,770)(PRO(NPRO),NPRO=1,NPROPT)
479   770   FORMAT(1H+,6(F13.5))
480  C
```

The instrument readings are taken by subroutine READAT (line 483), terminated when the operator presses and releases the foot pedal. If the NPRINT=3 option was taken (line 496), the calculated properties PRO will be written on output unit LOU at lines 496–497. If the NPRINT=4 option was selected, the raw instrument readings (voltages) will be printed. Then the operator is again offered the choices in lines 504–505.

```
481  C      NEW READINGS ARE MADE FROM THE EDDY CURRENT INSTRUMENT.
482  C
483         CALL READAT(VOLTS,NCHS,ITIMS,1)
484         DO 790 NF=1,NFT
485         PHASEM(1,1,NF)=10*VOLTS(4*NF-3)
486         TMAGM(1,1,NF)=VOLTS(4*NF-2)
487   790   CONTINUE
488         IF(NPRINT.EQ.4)WRITE(LOTEK,770)(VOLTS(4*NF-3),VOLTS(4*NF-2)
489        1,NF=1,NFT)
490         IF(VOLTS(12).GT.2.0)GO TO 720
491  C
492  C      PROGRAM WILL STAY IN THIS LOOP UNTIL THE FOOT PEDAL IS PRESSED.
493  C
494         IF(NLABL+NPRINT.EQ.4)WRITE(LOU,710)(PRONAM(NPRO),NPRO=1,NPROPT)
495         NLABL=NPRINT
496         IF(NPRINT.EQ.3)WRITE(LOU,800)(PRO(NPRO),NPRO=1,NPROPT)
497   800   FORMAT(1H ,6(F13.5))
498         IF(NPRINT.EQ.4)WRITE(LOU,800)(VOLTS(4*NF-3),VOLTS(4*NF-2)
```

```
      1,NF=1,NFT)
       GO TO 900
  850  WRITE(LOTEK,860)
  860  FORMAT(' PROP ARRAY IS FILLED.')
  900  WRITE(LOTEK,950)
  950  FORMAT(' PRINT OUT :1.PARAMETERS 2.MEAS VOLT 3.CAL&DISPLAY PROPS'
      1,'4.RAW RDGS.5.STOP')
       READ(LITEK,*)NPRINT
       GO TO (100,100,700,700,1000),NPRINT
  960  WRITE(LOTEK,970)
  970  FORMAT(' LIMIT OF FILE 25 IS EXCEEDED.')
 1000  STOP JOB
       END
```

The subroutine READAT has already been described in connection with the RFLRDG program. It allows data to be read in from an eddy-current instrument by using the MODAC. When the operator presses and releases a foot switch, the DO loop for the readings is interrupted, and the readings are passed back to the main program in the array called NARRAY.

A partial printout of a sample run is given below.

```
 NF NL THIK IN REG  2 CAL THIK        DIFF        DRIFT
  1  1      0.05491      0.05475     0.00016     0.00079
  1  2      0.05491      0.05492    -0.00001     0.00037
  1  3      0.05491      0.05495    -0.00004     0.00015
  2  1      0.05491      0.05504    -0.00013     0.00044
  2  2      0.05491      0.05494    -0.00003     0.00034
  2  3      0.05491      0.05487     0.00004     0.00014
  3  1      0.07098      0.07119    -0.00021     0.00033
  3  2      0.07098      0.07048     0.00050     0.00025
  3  3      0.07098      0.07102    -0.00004     0.00017
  4  1      0.07098      0.07100    -0.00002     0.00038
  4  2      0.07098      0.07119    -0.00021     0.00031
  4  3      0.07098      0.07115    -0.00017     0.00015
  5  1      0.07657      0.07671    -0.00014     0.00027
  5  2      0.07657      0.07697    -0.00040     0.00020
  5  3      0.07657      0.07664    -0.00007     0.00020
  6  1      0.07657      0.07622     0.00035     0.00029
  6  2      0.07657      0.07641     0.00016     0.00027
  6  3      0.07657      0.07633     0.00024     0.00019
DIFF IN THIK IN REG #  2=      0.00021 DRIFT=        0.00033

COEF(  1)= -2.8852402E+01    C5E6D1B8 CONSTANT
COEF(  2)= -1.0624557E+00    C187FE8C (LOG M1)1
COEF(  3)=  2.1322972E-01    3EDA58E4 (LOG M1)2
COEF(  4)= -7.6059904E-03    B9F93BAC (LIN F1)1
COEF(  5)=  2.0804506E-04    34DA26AC (LIN F1)2
COEF(  6)=  7.0635532E-04    36B92AB4 (LIN F1)1 (LOG M1)1
COEF(  7)=  1.9930824E+01    459F7254 (LOG M2)1
COEF(  8)= -3.2884159E+00    C2B27568 (LOG M2)2
COEF(  9)=  4.5667820E+00    43922314 (LIN F2)1
COEF( 10)= -1.8003631E-01    BEB85B70 (LIN F2)2
COEF( 11)= -1.4840379E+00    C1BDF4F4 (LIN F2)1 (LOG M2)1
DIFF IN L.O. IN REG #  3=      0.00008 DRIFT=        0.00013
```

```
 NP NL L.O. IN REG  3 CAL L.O.       DIFF        DRIFT
  1  1      0.00000      0.00017    -0.00017     0.00009
  1  2      0.00200      0.00200     0.00000     0.00014
  1  3      0.00400      0.00405    -0.00005     0.00018
  2  1      0.00000     -0.00016     0.00016     0.00008
  2  2      0.00200      0.00196     0.00004     0.00014
  2  3      0.00400      0.00399     0.00001     0.00018
  3  1      0.00000     -0.00001     0.00001     0.00006
  3  2      0.00200      0.00212    -0.00012     0.00012
  3  3      0.00400      0.00400     0.00000     0.00017
  4  1      0.00000      0.00005    -0.00005     0.00007
  4  2      0.00200      0.00189     0.00011     0.00012
  4  3      0.00400      0.00390     0.00010     0.00016
  5  1      0.00000      0.00001    -0.00001     0.00006
  5  2      0.00200      0.00206    -0.00006     0.00012
  5  3      0.00400      0.00397     0.00003     0.00017
  6  1      0.00000     -0.00003     0.00003     0.00006
  6  2      0.00200      0.00196     0.00004     0.00012
  6  3      0.00400      0.00407    -0.00007     0.00017
DIFF IN L.O. IN REG #  3=     0.00008 DRIFT=       0.00013

COEF(  1)= -4.7020391E-02    BCC09874 CONSTANT
COEF(  2)=  4.9068952E-01    3FFB3BA8 (LOG M1)1
COEF(  3)= -1.6810232E-01    BEAC2304 (LOG M1)2
COEF(  4)=  1.6929968E-03    37DDE78C (LIN P1)1
COEF(  5)= -1.6164233E-04    B4A97E88 (LIN P1)2
COEF(  6)= -1.1057153E+00    C18D8814 (LOG M2)1
COEF(  7)=  3.5483551E-01    3FB5AD00 (LOG M2)2
COEF(  8)=  1.9319773E-01    3EC5D5A0 (LIN P2)1
COEF(  9)= -1.6580246E-02    BB87D34C (LIN P2)2
DIFF IN DFLC IN REG #  2=     0.01018 DRIFT=       0.02568

 NP NL DFLC IN REG  2 CAL DFLC       DIFF        DRIFT
  1  1      0.02746      0.01729     0.01017     0.02184
  1  2      0.02746      0.01945     0.00800     0.02610
  1  3      0.02746      0.02448     0.00298     0.02943
  2  1      0.00550      0.01993    -0.01443     0.02089
  2  2      0.00550      0.01080    -0.00530     0.02594
  2  3      0.00550      0.00633    -0.00083     0.02910
  3  1      0.03549      0.03505     0.00044     0.02092
  3  2      0.03549      0.02893     0.00656     0.02571
  3  3      0.03549      0.03514     0.00035     0.02907
  4  1      0.00700     -0.00216     0.00916     0.02104
  4  2      0.00700      0.01179    -0.00479     0.02586
  4  3      0.00700      0.01847    -0.01147     0.02876
  5  1      0.03829      0.02503     0.01326     0.02118
  5  2      0.03829      0.02345     0.01483     0.02603
  5  3      0.03829      0.02746     0.01083     0.02950
  6  1      0.00760      0.02389    -0.01629     0.02109
  6  2      0.00760      0.02701    -0.01941     0.02600
  6  3      0.00760      0.01166    -0.00406     0.02985
DIFF IN DFLC IN REG #  2=     0.01018 DRIFT=       0.02568

COEF(  1)=  1.6516670E+01    45842224 CONSTANT
COEF(  2)=  2.3896156E+01    45BF2B54 (LOG M1)1
COEF(  3)= -1.6142899E-01    BEA54DA4 (LOG M1)2
COEF(  4)= -3.4869254E-01    BFB287D4 (LIN P1)1
COEF(  5)=  3.1984657E-02    3C830258 (LIN P1)2
COEF(  6)= -8.1253052E+00    C4820140 (LOG M2)1
COEF(  7)= -3.3701324E+00    C2D7B040 (LOG M2)2
COEF(  8)= -1.3464085E+01    C4D76CE4 (LIN P2)1
COEF(  9)=  1.2605195E+00    41A158B4 (LIN P2)2
DIFF IN DFSI IN REG #  2=     0.00421 DRIFT=       0.01928
```

NP	NL	DFSI IN REG 2	CAL DFSI	DIFF	DRIFT
1	1	0.00000	0.00204	-0.00204	0.02986
1	2	0.00000	0.00282	-0.00282	0.02065
1	3	0.00000	0.00324	-0.00324	0.01563
2	1	0.01400	0.01331	0.00069	0.02283
2	2	0.01400	0.00874	0.00526	0.02114
2	3	0.01400	0.01198	0.00202	0.01567
3	1	0.00000	-0.00236	0.00236	0.01978
3	2	0.00000	0.00473	-0.00473	0.01807
3	3	0.00000	-0.00077	0.00077	0.01331
4	1	0.01400	0.01547	-0.00147	0.02283
4	2	0.01400	0.01369	0.00031	0.02124
4	3	0.01400	0.01137	0.00263	0.01343
5	1	0.00000	0.00680	-0.00680	0.02014
5	2	0.00000	0.00736	-0.00736	0.01831
5	3	0.00000	-0.00105	0.00105	0.01282
6	1	0.01400	0.00712	0.00688	0.02063
6	2	0.01400	0.00543	0.00857	0.01868
6	3	0.01400	0.01610	-0.00210	0.01379

DIFF IN DFSI IN REG # 2= 0.00421 DRIFT= 0.01928

COEF(1)=	-7.5439087E+02	CABC9904	CONSTANT
COEF(2)=	1.8964050E+01	4597B660	(LOG M1)1
COEF(3)=	-1.0463875E+01	C4A76C08	(LOG M1)2
COEF(4)=	-3.0119598E-02	BBF6BD60	(LIN P1)1
COEF(5)=	-2.3847945E-02	BBC35CC4	(LIN P1)2
COEF(6)=	1.6520500E-01	3EA92B80	(LIN P1)1 (LOG M1)1
COEF(7)=	4.7432214E+02	49ED293C	(LOG M2)1
COEF(8)=	-6.9833405E+01	C78BAAB4	(LOG M2)2
COEF(9)=	1.1953238E+02	47EF1094	(LIN P2)1
COEF(10)=	-4.8128529E+00	C39A02E4	(LIN P2)2
COEF(11)=	-3.8694794E+01	C69AC778	(LIN P2)1 (LOG M2)1

PR.SET	REG	LAYER TH	RESTVY	PERM	DEF BLW	DEF RAD	DEF SIZ
1	1	1.0000E+10	1.0000E+12	1.000			
1	2	5.4910E-02	3.9500E+00	1.000	0.0275	0.1125	0.0000E-01
1	3	1.0000E+10	1.0000E+12	1.000			
2	1	1.0000E+10	1.0000E+12	1.000			
2	2	5.4910E-02	3.9500E+00	1.000	0.0055	0.7500	1.4000E-02
2	3	1.0000E+10	1.0000E+12	1.000			
3	1	1.0000E+10	1.0000E+12	1.000			
3	2	7.0980E-02	3.9500E+00	1.000	0.0355	0.1125	0.0000E-01
3	3	1.0000E+10	1.0000E+12	1.000			
4	1	1.0000E+10	1.0000E+12	1.000			
4	2	7.0980E-02	3.9500E+00	1.000			
4	3	1.0000E+10	1.0000E+12	1.000			
5	1	1.0000E+10	1.0000E+12	1.000			
5	2	7.6570E-02	3.9500E+00	1.000			
5	3	1.0000E+10	1.0000E+12	1.000			
6	1	1.0000E+10	1.0000E+12	1.000			
6	2	7.6570E-02	3.9500E+00	1.000			
6	3	1.0000E+10	1.0000E+12	1.000			

	SER/SHT RES	COIL DC RES	SHUNT CAP.	COIL INDUCT	RES.FREQ	DRIV. VOLT
DVR CKT	9090.000	67.300	8.4000E-11	0.9458E-78	5.7896E+76	2.8746
PICK CKT	1000000.000	531.000	8.4000E-11	8.1425E-67	1.2092E+38	

```
MEASURED VOLTAGES

FREQUENCY   5.03370E+03    GAIN    5.80250E+03
 PROP  LFT OF     0.0750    0.0883    0.1017
  SET
   1     MAG      6.0053    5.6918    5.4020
         PHA       6.674     6.547     6.358

 WITH    MAG      6.0584    5.7018    5.4222
  DEF    PHA       6.721     6.582     6.429

   2     MAG      6.0305    5.6942    5.4224
         PHA       3.258     3.241     2.989

 WITH    MAG      6.0470    5.7164    5.4400
  DEF    PHA       3.340     3.207     3.018

   3     MAG      6.0035    5.6619    5.3815
         PHA       2.515     2.371     2.257

 WITH    MAG      6.0148    5.6756    5.3671
  DEF    PHA       2.584     2.454     2.361


FREQUENCY   5.01620E+04    GAIN    3.10940E+02
 PROP  LFT OF     0.0750    0.0883    0.1017
  SET
   1     MAG      5.1882    4.9411    4.7048
         PHA       6.002     6.178     6.317

 WITH    MAG      5.2287    4.9503    4.7218
  DEF    PHA       5.967     6.170     6.300

   2     MAG      5.2130    4.9478    4.7322
         PHA       5.943     6.133     6.260

 WITH    MAG      5.2395    4.9790    4.7486
  DEF    PHA       5.948     6.136     6.250

   3     MAG      5.2168    4.9480    4.7226
         PHA       5.934     6.123     6.260

 WITH    MAG      5.2244    4.9586    4.7140
  DEF    PHA       5.931     6.125     6.274
```

REFERENCES

1. W. A. Simpson, Jr., J. W. Luquire, C. V. Dodd, and W. G. Spoeri, *Computer Programs for Some Eddy-Current Problems — 1970,* ORNL-TM-3295 (June 1971), pp. 258–261.

2. Modular Computer Systems, Inc., Ft. Lauderdale, Fla., *Reference Manual, MAX II/III/IV System Processors, Assemblers,* Publication 210-600500-008400/B00, 1976.

APPENDIX E

THE SUBROUTINES

Subroutines called by the four main programs are listed alphabetically in Table E1. They can be categorized by functions, such as integral calculation or input or output, and by their application to the reflection or the transmission problem. Those associated with least squares function fitting apply generally to either problem. All are written in Fortran, except the data acquisition subroutine, READAT, which is in assembly language.

Table E1. Subroutines and Calling Programs

Subroutine Name	Calling Programs			
	MULFRD	MULTRU	RFLRDG	RFLFIT
ALSQS	X	X		X
BESSEL	X	X		
BESEL1	X			
BETAM	X			
BETAMT		X		
CHRFPM	X	X		
COLFIL	X			X
CON8	X	X		X
CONVR8				X
CPXQOT	X	X		
GAMAML	X			
GAMAMT		X		
POLTYP	X	X		X
PRFCKT	X	X		X
PRFCOL	X	X		
PRFVLT	X			X
PTRVLT		X		
RDGEXP	X	X		X
READAT			X	X
RFCIRK	X			
RFCKDT	X			
RFCOLM	X			
TRCOLM		X		
TRUCIR		X		
TCOFIL		X		
VINITM	X			
VMATRM	X			
VMATRT		X		

ALSQS

Subroutine ALSQS takes the READNG array constructed by subroutine RDGEXP and finds the array of coefficients, COEF, that when multiplied by the READNG array will give the best least squares fit to the property array PRO. Upon return the best fitting property values are in the last column of the READNG array, and RSOS is the residual sum of squares of the differences from the actual property values.

```
      SUBROUTINE ALSQS(A,Y,B,R2,NN,MM,NA)
C
C     ALSQS IS A FORTRAN IV SUBROUTINE TO SOLVE THE LINEAR LEAST
C     SQUARES PROBLEM  NORM(AB - Y) = MIN.   CALLING SEQUENCE IS
C              CALL ALSQS(A,Y,B,R2,N,M,NA)
C     WHERE
C        A    IS AN ARRAY CONTAINING THE LEAST SQUARES MATRIX.
C             UPON RETURN THE (M+1)-TH COLUMN CONTAINS THE
C             APPROXIMATING VECTOR AB.
C        Y    IS THE VECTOR TO BE FIT
C        B    CONTAINS UPON RETURN THE COEFFICIENTS OF THE FIT.
C        R2   CONTAINS UPON RETURN THE RESIDUAL SUM OF SQUARES.
C        N    IS THE NUMBER OF ROWS IN THE LEAST SQUARES MATRIX.
C        M    IS THE NUMBER OF COLUMNS IN THE LEAST SQUARES
C             MATRIX.
C        NA   IS THE FIRST DIMENSION OF THE ARRAY A.
C
C     IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(NA,1),Y(1),B(1)
      N = NN
      N1 = N+1
      M = MM
      M1 = M+1
      MM1 = M-1
C
C     REDUCE THE LEAST SQUARES MATRIX TO UPPER TRIANGULAR FORM
C
      DO 60 L=1,M
         SS = 0.
         DO 10 I=L,N
   10       SS = SS + A(I,L)**2
         S2 = SS
         S =SQRT(S2)
         IF (A(L,L).LT.0.) S=-S
         D = S2 + S*A(L,L)
         A(L,L) = A(L,L) + S
         IF (L.EQ.M) GO TO 50
         L1 = L+1
         DO 30 J=L1,M
            PP = 0.
            DO 20 I=L,N
   20          PP = PP + A(I,L)*A(I,J)
   30       A(N1,J) = PP/D
         DO 40 J=L1,M
            DO 40 I=L,N
   40          A(I,J) = A(I,J) - A(I,L)*A(N1,J)
   50    A(N1,L) = -S
   60    CONTINUE
C
```

```
C       REDUCE THE VECTOR Y
C
        DO 80 I=1,N
   80      A(I,M1) = Y(I)
        DO 100 L=1,M
           PP = 0.
           DO 90 I=L,N
   90         PP = PP + A(I,L)*A(I,M1)
           D = PP/(-A(L,L)*A(N1,L))
           DO 100 I=L,N
  100         A(I,M1) = A(I,M1) - D*A(I,L)
C
C       CALCULATE THE COEFFICIENT VECTOR B
C
        B(M) = A(M,M1)/A(N1,M)
        IF (M.EQ.1) GO TO 130
        DO 120 LL=1,MM1
           L = M-LL
           L1 = L+1
           PP = A(L,M1)
           DO 110 I=L1,M
  110         PP = PP - A(L,I)*B(I)
  120      B(L) = PP/A(N1,L)
C
C       CALCULATE R2
C
  130   SS = 0.
        MP1 = M+1
        DO 140 I=MP1,N
           SS = SS + A(I,M1)**2
  140      A(I,M1) = 0.
        R2 = SS
C
C       PERFORM THE BACK CALCULATIONS
C
        DO 170 LL=1,M
           L = M-LL+1
           PP = 0.
           DO 150 I=L,N
  150         PP = PP + A(I,L)*A(I,M1)
           D = PP/(-A(L,L)*A(N1,L))
           DO 160 I=L,N
  160         A(I,M1) = A(I,M1) - D*A(I,L)
  170      CONTINUE
        RETURN
        END
```

BESEL1

Subroutine BESEL1 calculates the Bessel function J_1. It is called by the RFCOLM subroutine. Approximations are discussed in earlier work. (See Bessel for reference and notes on calculations.)

```
C
C     SUBROUTINE BESEL1     (19 JULY 1977)
C
      SUBROUTINE BESEL1(Q1,RJ1)
C
C     CALCULATES J1(Q1) .
C
      IF(Q1.GT.3) GO TO 20
      Q1S=Q1*Q1
      Q2S=((2.1E-11*Q1S-5.38E-9)*Q1S+6.757E-7)*Q1S-5.42443E-5
      Q2S=((Q2S*Q1S+2.60415E-3)*Q1S-6.25E-2)*Q1S+.5
      RJ1=Q1*Q2S
      RETURN
   20 Q3S=(((-.14604057/Q1+.27617679)/Q1-.20210391)/Q1+4.61835E-3)/Q1
      Q3S=((Q3S+.14937)/Q1+4.68E-6)/Q1+.79788456
      Q4S=(((-.21262014/Q1+.19397232)/Q1+6.022188E-2)/Q1-.17222733)/Q1
      Q4S=((Q4S+5.085E-4)/Q1+.37498836)/Q1-2.35619449+Q1
      RJ1=Q3S*COS(Q4S)/SQRT(Q1)
      RETURN
      END
```

BESSEL

Subroutine BESSEL (XJ1,X,R) calculates $\int_0^{RX} xJ_1(x)\ dx$ by using series approximations and returns the integral as XJ1. The approximations are discussed in earlier work.[1] Note that approximations of the form

$$A_0 + A_1x + A_2x^2 + \ldots + A_nx^n$$

are generally computed as

$$\{\ldots\ [(A_nx + A_{n-1})x + A_{n-2}]x\ \ldots + A_1\}x + A_0$$

to improve speed and accuracy.

```
C
C         BESSEL.MOD     (17 FEB. 1977)
C
      SUBROUTINE BESSEL(XJ1,X,R)
C
C       COMPUTES THE INTEGRAL OF THE BESSEL FUNCTION J1(X)
C       TIMES X FROM ZERO TO Z. USES POWER SERIES SUMMATION FOR
C       Z .LE. 5, AND AN ASYMPTOTIC SERIES FOR Z .GT. 5.
C       SEE ORNL-TM-3295, PP. 258-261.
C
      DATA PIO4 /.785398163/
      Z=X*R
      IF (Z .GT. 5.0) GO TO 1010
      K = Z + Z
      L = K + 3
      F1 = 0.5*R*R*R
      XJ1 = F1/3.0
      T = -.25*Z*Z
      D1 = 2.
      D2 = 5.
      E = 4.
      DO 1000 I=1,L
        F1 = F1*T/D1
        XJ1 = XJ1 + F1/D2
        D1 = D1 + E
        E = E + 2.
        D2 = D2 + 2.
 1000 CONTINUE
      GO TO 1020
C
C       ASYMPTOTIC SERIES FOR Z .GT. 5
C
 1010 T = 1./Z
      X0=(((((-188.1357*T + 109.1142)*T - 23.79333)*T + 2.050931)*T
     1      - 0.1730503)*T + 0.7034845)*T - 0.064109E-3
      X1 = ((((-5.817517*T + 2.105874)*T - .6896196)*T +.4952024)*T
     1     - 0.187344E-2)*T + 0.7979095
      ARG = Z - PIO4
      XJ1 = (1. - SQRT(Z)*(X1*COS(ARG) - X0*SIN(ARG)))/(X**3)
 1020 RETURN
      END
```

BETAM

This subroutine computes the β_n and $\exp(\beta_n T_n)$ factors [see Eqs. (26)—(31)] used to compute the γ-factors appearing in the impedance integrals. When the coil dimensions are normalized, the factors $\alpha_n l_i$ become

$$\alpha_n \bar{r}(l_i/\bar{r}) = \alpha_n L_i \ ,$$

when $\bar{r}$ is absorbed in α_n. For conductors, terms including ε_n (the permittivity) are dropped so that in normalized form

$$\alpha_n^2 = \alpha^2 + j\omega\mu_n\sigma_n\bar{r}^{\,2} \ .$$

[For the uppermost layer (air), $\alpha_n = \alpha$; there is no imaginary part.] If we let

$$a = \alpha^2 \ ,$$

$$b = \omega\mu_n\sigma_n\bar{r}^{\,2} \ ,$$

and

$$\rho = (a^2 + b^2)^{\frac{1}{2}} \ ,$$

then

$$\mathrm{Re}(\alpha_n) = \left(\frac{\rho + a}{2}\right)^{\frac{1}{2}} \ ,$$

and

$$\mathrm{Im}(\alpha_n) = \left(\frac{\rho - a}{2}\right)^{\frac{1}{2}} \ .$$

Note also that

$$\mathrm{Im}(\alpha_n) = \frac{b}{2} \frac{1}{\mathrm{Re}(\alpha_n)} .$$

In the main program, MULRFD, we set

$$\mu_n' = \frac{1}{\mu_n\sqrt{2}} ,$$

so that

$$\mathrm{BETAR(NR)} = \mathrm{Re}(\beta_n) = \mu_n'(\rho + a)^{\frac{1}{2}}$$

and

$$\mathrm{BETAI(NR)} = \mathrm{Im}(\beta_n) = \mu_n'(\rho - a)^{\frac{1}{2}} = \mu_n'^2 b/\mathrm{Re}(\beta_n) .$$

Since β_n is complex we compute real and imaginary parts of $\exp(\beta_n T_n)$ as

$$\mathrm{XPON} = \exp[\mathrm{Re}(\beta_n)T_n] ,$$

$$\mathrm{COSF} = \cos[\mathrm{Im}(\beta_n)T_n] ,$$

and

$$\mathrm{SINF} = \sin[\mathrm{Im}(\beta_n)T_n] .$$

When a defect is present in a region, the routine computes the function

$$\exp[\beta_n(Z_{n+1} - Z)] ,$$

where Z is the axial location of the defect. (DFT corresponds to DFLOC in the main program.)

```
C
C     SUBROUTINE BETAM (13 JULY 1977)
C
      SUBROUTINE BETAM(NR,NRT,NP,NPT,NF,NFT,BETAR,BETAI,COSF,SINF,XPON
     1,WUSR,U,TH,COSD,SIND,XPOND,DFT,NRDF)
      DIMENSION BETAR(NRT),BETAI(NRT),COSF(NRT),SINF(NRT),XPON(NRT)
      DIMENSION WUSR(NRT,NPT,NFT),U(NRT,NPT),TH(NRT,NPT)
      DIMENSION DFT(NPT),NRDF(NPT)
      COMMON /B1/X,XX,XXXX
C     START BETA CALCULATIONS AT UPPERMOST REGION AND WORK DOWN
C     AWAY FROM COIL.
      NR=NRT
      BETAR(NR)=X
      BETAI(NR)=0.
C     SET EXPONENT SUM TO ZERO AND START CALCULATING BETAS.
      SUMEXP=0.
   20 NR=NR-1
      BETAR(NR)=U(NR,NP)*SQRT(XX+SQRT(XXXX+WUSR(NR,NP,NF)*WUSR(NR,NP,NF
     1)))
      BETAI(NR)=U(NR,NP)*U(NR,NP)*WUSR(NR,NP,NF)/BETAR(NR)
      IF (NR.EQ.1) GO TO 50
      XTH=BETAR(NR)*TH(NR,NP)
      SUMEXP=SUMEXP+XTH
      IF (SUMEXP.GT.20.) GO TO 40
      XPON(NR)=EXP(XTH)
      COSF(NR)=COS(BETAI(NR)*TH(NR,NP))
      SINF(NR)=SIN(BETAI(NR)*TH(NR,NP))
      IF(NRDF(NP).NE.NR) GO TO 20
   25 XPOND=EXP(BETAR(NR)*DFT(NP))
      COSD=COS(BETAI(NR)*DFT(NP))
      SIND=SIN(BETAI(NR)*DFT(NP))
      IF(NR.NE.1) GO TO 20
   40 RETURN
C     CALCULATE THE DEFECT VALUES IF IT IS IN REGION 1.
   50 IF(NRDF(NP).EQ.1)GO TO 25
      RETURN
      END
```

BETAMT

See BETAM (p. 155) for listing of the equations used.

This subroutine performs calculations of the β-factors and $\exp(\beta_n T_n)$ identical to those calculated by BETAM for defect-free conductors. It does not include calculations for defects and is therefore a shorter version. The main program (MULTRU) does not convert μ_n to $1/(\sqrt{2}\mu_n)$ so

$$\text{BETAR} = \text{Re}(\beta_n) = \frac{\text{Re}(\alpha_n)}{\sqrt{2}\mu_n} ,$$

and

$$\text{BETAI} = \text{Im}(\beta_n) = \frac{\text{Im}(\alpha_n)}{\sqrt{2}\mu_n} = \frac{b}{2\mu_n^2 \text{Re}(\alpha_n)} .$$

This routine contains no test for the exponent size, as in BETAM.

```
C
C       SUBROUTINE BETAMT (29 NOVEMBER 1977)
C
        SUBROUTINE BETAMT(NRT,NP,NPT,NF,NFT,BETAR,BETAI,COSF,SINF,XPON
      1,WUSR,U,TH,RRBAR)
        DIMENSION BETAR(NRT),BETAI(NRT),COSF(NRT),SINF(NRT),XPON(NRT)
        DIMENSION WUSR(NRT,NPT,NFT),U(NRT,NPT),TH(NRT,NPT)
        COMMON /B1/X,XX,XXXX
C       START BETA CALCULATIONS AT UPPERMOST REGION AND WORK DOWN
C       AWAY FROM COIL.
        NR=NRT
        BETAR(NR)=X
        BETAI(NR)=0.
     20 NR=NR-1
        BETAR(NR)=.70710678*SQRT(XX+SQRT(XXXX+WUSR(NR,NP,NF)
      1*WUSR(NR,NP,NF)))/U(NR,NP)
        BETAI(NR)=.5*WUSR(NR,NP,NF)/(BETAR(NR)*U(NR,NP)*U(NR,NP))
        IF (NR.EQ.1) GO TO 50
        XTH=TH(NR,NP)*RRBAR*U(NR,NP)
        XPON(NR)=EXP(BETAR(NR)*XTH)
        COSF(NR)=COS(BETAI(NR)*XTH)
        SINF(NR)=SIN(BETAI(NR)*XTH)
        GO TO 20
     50 RETURN
        END
```

CHRFPM

Subroutine CHRFPM prompts the user to type in changes in the coil and circuit parameters for a reflection-type coil and transfers the changed values back to the main program in COMMON block B2. The coil parameter functions are now duplicated by the COLFIL and TCOFIL subroutines (see pp. 160 and 180).

```
      SUBROUTINE CHRFPM(LOU,LI,INTG,NPRINT)
C
C     SUBROUTINE TO CHANGE THE COIL AND CIRCUIT PRAMETERS FOR A
C     REFLECTION TYPE COIL.
C
      DIMENSION CKTVAL(18),B(18)
      COMMON /B2/R1,R2,R3,R4,XL3,XL4,RBAR,VO,ZLDR,ZLPU,
     *TNDR,TNPU,RDCDR,RDCPU,RO,R9,CAPDR,CAPPU
      EQUIVALENCE(R1,CKTVAL)
      DO 10 NB=1,18
      B(NB)=0.
   10 CONTINUE
      WRITE(LOU,20)
   20 FORMAT(' COIL     INN. RAD.    OUT. RAD.',7X,'LENGTH',6X,'SHT CAP',
     17X,'DC RES',5X,'TURNS EA')
      WRITE(LOU,30)R1,R2,XL3,CAPDR,RDCDR,TNDR
   30 FORMAT(' DVR ',3(F13.5),1PE13.4,2(0PF13.2))
      WRITE(LOU,90)
      READ(LI,*)B(1),B(2),B(5),B(17),B(13),B(11)
      WRITE(LOU,40)R3,R4,XL4,CAPPU,RDCPU,TNPU
   40 FORMAT(' PIC',3(F13.5),1PE13.4,2(0PF13.2))
      WRITE(LOU,90)
      READ(LI,*)B(3),B(4),B(6),B(18),B(14),B(12)
      WRITE(LOU,50)
   50 FORMAT(8X,'RBAR     VOUT        DVR AMP RES      PIC AMP RES')
      WRITE(LOU,60)RBAR,VO,RO,R9
   60 FORMAT(1X,2F14.5,2F14.2)
      WRITE(LOU,90)
      READ(LI,*)B(7),B(8),B(15),B(16)
      DO 70 NB=1,18
      IF(B(NB).NE.0.)CKTVAL(NB)=B(NB)
   70 CONTINUE
      INTG=0
      DO 80 NB=1,7
      IF(B(NB).NE.0.)INTG=1
   80 CONTINUE
   90 FORMAT(1X)
      IF(NPRINT.EQ.6)NPRINT=0
      RETURN
      END
```

COLFIL

Subroutine COLFIL (LI, LOU CNAM) reads a coil name CNAM from the main program. If none or "0." the user will be asked to type in a coil name (up to six characters). If there is a coil in file 28 with the name typed into input terminal LI, its parameters are fed back to the main program in COMMON blocks B2 and B6. If not the user is requested (on output unit LOU) to type the appropriate coil parameters on terminal LI, and the parameters are transmitted to the main program in the common blocks. (See Tables A3 and D2 for operator interactive responses.)

```
C
      SUBROUTINE COLFIL(LI,LOU,CNAM)
C *** SUBROUTINE TO OBTAIN COIL DATA *** (3 APRIL 1978)
      REAL*6 COIL,CNAM
      REAL L2,L3,L4,L5,L6
      COMMON /B2/R1,R2,R3,R4,L3,L4,RBAR,V0,ZLDR,ZLPU,
     *TNDR,TNPU,RDCDR,RDCPU,R0,R9,CAPDR,CAPPU
      COMMON /B6/L2,L5,L6
      DEFINE FILE 28(80,32,U,NCOL)
      NCOL=1
      IF(CNAM.NE.0.)GO TO 40
   10 WRITE (LOU,20)
   20 FORMAT (' TYPE COIL NAME (UP TO 6 LETTERS):')
      READ (LI,30) CNAM
   30 FORMAT (A6)
   40 READ (28'NCOL) COIL,RBAR,R1,R2,L3,R3,R4,L4,L5,
     *TNDR,TNPU,L6,RDCDR,RDCPU
      IF (COIL.EQ.CNAM) GO TO 70
      IF (COIL.EQ.'END   ') GO TO 50
      GO TO 40
   50 WRITE (LOU,60) CNAM
   60 FORMAT(' COIL ',A6,' NOT FOUND. TYPE FOLLOWING COIL PARAMETERS:',
     */,' RBAR,R1,R2,L3,R3,R4,L4,L5,L6,RDCDR,RDCPU,TNDR,TNPU:')
      READ (LI,*) RBAR,R1,R2,L3,R3,R4,L4,L5,L6,RDCDR,RDCPU,TNDR,TNPU
   70 RETURN
      END
```

CONVR8

Floating-point numbers from the ModComp minicomputer are converted to the hexadecimal representation of the floating-point binary used by the NDT-COMP8 microcomputer, which has an exponent and a high- and low-order mantissa. (The numbers converted here are not used directly by the microcomputer but rather by the minicomputer assembler program, which generates the program for the microcomputer.) The three parts of the hex number are returned in the NCONV array. It is written in the ModComp IV assembly language.[2]

```
      SUBROUTINE CONVR8(XNUM,NCONV)
      INTEGER*2 NEXP1,NHI1,NLO1
      DIMENSION NCONV(3)
C
C     SUBROUTINE TO CONVERT MODCOMP FLOATING POINT NUMBERS TO THE
C     HEX REPRESENTATION OF THE FLOATING POINT BINARY THAT THE NDT-COMP8
C     USES.THE FORM IS EXPONENT,FRACTION WITH THE FIRST TWO NUMBERS
C     THE EXPONENT AND THE LAST SIX THE FRACTION.  VERSION 16 11 77
C
      ZNUM=XNUM
      INLINE
      LDMD,6        ZNUM          LOAD ZNUM INTO REG 6&7
      LLQ,4,1                     SHIFT QUAD REG 4-7 TO LEFT
      LAD,6,2                     ARTH SHIFT DOUB REG TO LEFT
      RLQ,4,1                     SHIFT QUAD REG TO RT,DUMP 2 EXPT BIT
      TBRB,6,0      SKIP          SKIP AROUND IF ZNUM WAS NEG
      BRU           RETN          GO TO RETURN IF NO WAS POS
SKIP  ABR,6,0                     INCREMENT BIT 0,SIGN BIT
      TTRD,6,6                    TWOS COMPLIMENT OF 6,7-CHANGS TO POS
RETN  TBRB,6,8      NZRO          TESTS BIT 8,XFERS IF NONZERO
      LDI,6         #B000         LOADS B000 IN REG 6 IF IT WAS 0
NZRO  MBR,2,6                     THE EXPONENT IS TRANSFERED TO 2
      STM,2         NEXP1         AND STORED IN MEMORY.
      MLR,2,6                     THE HI ORDER MANTESSA IS XFERED TO 2
      STM,2         NHI1          AND STORED.
      MBR,2,7                     THE LOW ORDER MANTESSA IS MOVED TO 2
      STM,2         NLO1          AND STORED.
      FINI
      NCONV(1)=NEXP1
      NCONV(2)=NHI1
      NCONV(3)=NLO1
      RETURN
      END
```

CON8

Subroutine CON8 converts MODCOMP floating-point numbers to the hexadeciaml representation of floating-point binary numbers used by the NDT-COMP8 microcomputer. It is written in the ModComp IV assembly language.[2]

```
           SUBROUTINE CON8(XNUM,JNUM)
      C
      C    SUBROUTINE TO CONVERT MODCOMP FLOATING POINT NUMBERS TO THE
      C    HEX REPRESENTATION OF THE FLOATING POINT BINARY THAT THE NDT-COMP8
      C    USES.THE FORM IS EXPONENT,FRACTION WITH THE FIRST TWO NUMBERS
      C    THE EXPONENT AND THE LAST SIX THE FRACTION.  VERSION 16 11 77
      C
           ZNUM=XNUM
           INLINE
           LDMD,6        ZNUM            LOAD ZNUM INTO REG 6&7
           LLQ,4,1                       SHIFT QUAD REG 4-7 TO LEFT
           LAD,6,2                       ARTH SHIFT DOUB REG TO LEFT
           RLQ,4,1                       SHIFT QUAD REG TO RT,DUMP 2 EXPT BIT
           TBRB,6,0      SKIP           SKIP AROUND IF ZNUM WAS NEG
           BRU            RETN           GO TO RETURN IF NO WAS POS
      SKIP ABR,6,0                       INCREMENT BIT 0,SIGN BIT
           TTRD,6,6                     TWOS COMPLIMENT OF 6,7-CHANGS TO POS
      RETN TBRB,6,8      NZRO           TESTS BIT 8,XFERS IF NONZERO
           LDI,6         #B000           LOADS B000 IN REG 6 IF IT WAS 0
      NZRO STMD,6        IZ               STORE RESULTS IN IZ
           FINI
           JNUM=IZ
           RETURN
           END
```

CPXQOT

Subroutine CPXQOT calculates the gamma function, which is the complex quotient of V_{12}/V_{22}. In the computer notation V22I = $\text{Im}(V_{22})$, V_{22} = V22R + j(V22I), etc.

$$\frac{V_{12}}{V_{22}} = \frac{\text{Re}(V_{12})\text{Re}(V_{22}) - \text{Im}(V_{12})\text{Im}(V_{22}) + j[\text{Re}(V_{12})\text{Im}(V_{22}) + \text{Im}(V_{12})\text{Re}(V_{22})]}{[\text{Re}(V_{22})]^2 + [\text{Im}(V_{22})]^2}$$

```
C
C       CPXQOT SUBROUTINE         (18 MAY 1977)
C
      SUBROUTINE CPXQOT(GAMR,GAMI,V12R,V12I,V22R,V22I)
C     CALCULATES THE COMPLEX QUOTIENT OF V12/V22,WHICH IS THE GAMMA FACTOR
      Q4=V22I/V22R
      Q3=1./(V22R+V22I*Q4)
      Q4=-Q3*Q4
      GAMR=V12R*Q3-V12I*Q4
      GAMI=V12R*Q4+V12I*Q3
      RETURN
      END
```

GAMAML

Subroutine GAMAML directs calculation of the real and imaginary parts of the multilayer gamma factor γ_D and GAMAD factor γ' by calling subroutines BETAM, VINITM, VMATRM, and CPXQOT.

```
C
C     SUBROUTINE GAMAML    (13 JULY 1977)
C
      SUBROUTINE GAMAML(NRT,NPT,NP,NFT,NF,BETAR,BETAI,COSF,SINF,XPON
     1,WUSR,U,TH,GAMAR,GAMAI,GAMADR,GAMADI,DFT,DFR,NRDF)
C
C     CALCULATES THE GAMMA FACTOR AND THE GAMAD FACTOR FOR ANY GIVEN SET
C     OF MATERIAL PROPERTIES CONSISTING FOR PLANER LAYERS WITH ARBITRARY
C     RESISTIVITIES,THICKNESSES,AND DEFECTS.
C
      DIMENSION BETAR(NRT),BETAI(NRT),COSF(NRT),SINF(NRT),XPON(NRT)
      DIMENSION WUSR(NRT,NPT,NFT),U(NRT,NPT),TH(NRT,NPT)
      DIMENSION GAMAR(NPT),GAMAI(NPT),GAMADR(NPT),GAMADI(NPT),DFT(NPT)
     1,DFR(NPT), NRDF(NPT)
C
C     A MATERIAL WITHOUT A DEFECT IS CALCULATED
C     AT THE SAME TIME AS THE MATERIAL WITH DEFECTS.
C
C      CALCULATE THE BETA VALUES THAT WILL BE USED AND THE LOWERMOST
C      REGION THAT WILL BE SEEN BY THE COIL.
C
      CALL BETAM(NSRT,NRT,NP,NPT,NF,NFT,BETAR,BETAI,COSF,SINF,XPON
     1,WUSR,U,TH,COSD,SIND,XPOND,DFT,NRDF)
C
C     CALCULATE THE INITIAL VALUES OF V1&V2 MATRICES AND THE DEFECT
C     MATRIX ,VD,IF THE DEFECT IS IN THE INITIAL REGION.IT IS SET TO 0
C     OTHERWISE.
C
      CALL VINITM(NSRT,NRT,NPT,NP,BETAR,BETAI,V1R,V1I,V2R,V2I
     1,VDR,VDI,NRDF,COSD,SIND,XPOND)
      NSRT=NSRT+1
      IF(NSRT.GE.NRT) GO TO 40
C
C     TRANSFORM FROM THE INITIAL REGION +1 TO THE LAST REGION,NRT.
C
      CALL VMATRM(NRT,NSRT,NRT,NPT,NP,BETAR,BETAI,COSF,SINF,XPON,
     1V1R,V1I,V2R,V2I,VDR,VDI,NRDF,COSD,SIND,XPOND)
C     THE GAMMA FACTORS ARE CALCULATED FROM THE V-MATRICES.
C
   40 CALL CPXQOT(GAMAR(NP),GAMAI(NP),V1R,V1I,V2R,V2I)
      CALL CPXQOT(GAMADR(NP),GAMADI(NP),VDR,VDI,V2R,V2I)
      RETURN
      END
```

GAMAMT

This routine is similar in function to GAMAML (p. 164) with the following differences:

1. defects are not included, hence, BETAMT is called instead of BETAM; and
2. three γ-factors for the defect-free conductor stack are computed, the third of which, γ_p, (GAMAPR, GAMAPI) requires inversion of the conductor stack.

```
C
C      SUBROUTINE GAMAML    (29 NOVEMBER 1977)
C
       SUBROUTINE GAMAMT(NRT,NPT,NP,NFT,NF,BETAR,BETAI,COSF,SINF,XPON
      1,WUSR,U,TH,RRBAR,GAMAR,GAMAI,GAMADR,GAMADI,GAMAPR,GAMAPI)
C
C      CALCULATES THE GAMA,GAMAD,AND GAMAP FACTORS FOR ANY GIVEN SET
C      OF MATERIAL PROPERTIES CONSISTING OF PLANAR LAYERS WITH ARBITRARY
C      RESISTIVITIES AND THICKNESSES,FOR THROUGH TRANSMISSION COILS.
C
       DIMENSION BETAR(NRT),BETAI(NRT),COSF(NRT),SINF(NRT),XPON(NRT)
       DIMENSION WUSR(NRT,NPT,NFT),U(NRT,NPT),TH(NRT,NPT)
       DIMENSION GAMAR(NPT),GAMAI(NPT),GAMADR(NPT),GAMADI(NPT)
       DIMENSION GAMAPR(NPT),GAMAPI(NPT)
C
C      CALCULATE THE BETA VALUES AND THE EXPONENTIAL FUNCTIONS THAT
C      WILL BE USED.
C
       CALL BETAMT(NRT,NP,NPT,NF,NFT,BETAR,BETAI,COSF,SINF,XPON
      1,WUSR,U,TH,RRBAR)
C
C      TRANSFORM FROM THE INITIAL REGION +1 TO THE LAST REGION,NRT.
C
       NSTP=NRT
       NSRT=1
       CALL VMATRT(NSTP,NSRT,NRT,NPT,BETAR,BETAI,COSF,SINF,XPON,
      1V1R,V1I,V2R,V2I,VI1R,VI1I,BTR,BTI)
C      THE GAMMA FACTORS ARE CALCULATED FROM THE V-MATRICES.
C
    40 CALL CPXQOT(GAMADR(NP),GAMADI(NP),V1R,V1I,V2R,V2I)
       CALL CPXQOT(GAMAPR(NP),GAMAPI(NP),VI1R,VI1I,V2R,V2I)
       CALL CPXQOT(GAMAR(NP),GAMAI(NP),BTR,BTI,V2R,V2I)
       RETURN
       END
```

POLTYP

Subroutine POLTYP constructs the POLARY array for printing a representation of the polynomial expansion. (See p. 78 for sample output.) The ENCODE instruction (Cf, line 619) is a ModComp addition to standard Fortran. It is the equivalent of a WRITE statement, which performs an internal character transfer to specific positions to construct output "words."

```
588  C
589        SUBROUTINE POLTYP(POLARY,JROW,MROW,NPOL,IROW,NFT,NC,IOFSET,IDEV)
590  C
591  C THIS SUBROUTINE CONSTRUCTS AN ARRAY FOR PRINTING
592  C
593        INTEGER*2 RDGTYP(2),INUM(9),IROLD(4)
594        REAL*8 ROLD
595        DIMENSION POLARY(JROW,5),NPOL(IROW,NFT),FUNTYP(4),CONSTA(2)
596        EQUIVALENCE (ROLD,IROLD(1))
597        DATA  RDGTYP/' M',' P'/,CONSTA/'CONS','TANT'/,
598       2    FUNTYP/'(LIN','(LOG','(EXP','(INV'/,
599       3    INUM/'1)','2)','3)','4)','5)','6)','7)','8)','9)'/,
600       4    BLANKS/'    '/
601  C
602  C     BLANK OUT POLARY ARRAY
603  C
604        DO 71 J=1,5
605        DO 71 I=1,JROW
606  71    POLARY(I,J)=BLANKS
607  C
608        NROW=1
609        IF(IOFSET.NE.1) GO TO 70
610        POLARY(NROW,1)=CONSTA(1)
611        POLARY(NROW,2)=CONSTA(2)
612        NROW=NROW+1
613  C
614  70    DO 200 NF=1,NFT
615        DO 300 NTYP=1,NC
616        NCC=NTYP*3
617        NCP=NCC-1
618        NCF=NCP-1
619        ENCODE(8,10,ROLD)POLARY(NROW-1,1),POLARY(NROW-1,2)
620  10    FORMAT(2A4)
621        NDEG=NPOL(NCP,NF)
622        IF(NDEG.EQ.0) GO TO 300
623        DO 400 I=1,NDEG
624        POLARY(NROW,1)=FUNTYP(NPOL(NCF,NF))
625        ENCODE(4,20,POLARY(NROW,2)) RDGTYP(NTYP),INUM(NF)
626  20    FORMAT(2A2)
627        ENCODE(1,30,POLARY(NROW,3)) INUM(I)
628  30    FORMAT(A1)
629        NROW=NROW+1
630  400   CONTINUE
631  C
632  C    CREATE CROSS TERMS
633  C
634        NCTERM=NPOL(NCC,NF)
635        IF(NCTERM.EQ.0) GO TO 300
636        IF(NF.EQ.1.AND.NTYP.EQ.1) GO TO 99
637        J=NCTERM
638        DO 500 I=1,NCTERM
```

```
      POLARY(NROW,1)=POLARY(NROW-1,1)
      POLARY(NROW,2)=POLARY(NROW-1,2)
      ENCODE(4,40,POLARY(NROW,3)) INUM(I),ROLD
 40   FORMAT(A1,1X,A2)
      ENCODE(4,50,POLARY(NROW,4)) IROLD(2),IROLD(3)
 50   FORMAT(2A2)
      ENCODE(4,60,POLARY(NROW,5))IROLD(4),INUM(J)
 60   FORMAT(A2,A1,1X)
      J=J-1
      NROW=NROW+1
 500  CONTINUE
 300  CONTINUE
 200  CONTINUE
C
C  PRINT RESULTS
C
      WRITE(IDEV,21)((POLARY(I,J),J=1,5),I=1,MROW)
 21   FORMAT(' POLARY='/(1X,5A4))
      GO TO 1000
C
C  PRINT ERROR MESSAGES
C
 99   WRITE(IDEV,31)
 31   FORMAT(' ERROR: CANNOT HAVE A CROSS TERM ON 1ST ITERATION')
C
 1000 RETURN
      END
```

PRFCKT

Subroutine PRFCKT (LOU) prints the circuit parameters on the LOU output unit.

```
C
C     SUBROUTINE PRFCKT   (20 JULY 1977)
C
      SUBROUTINE PRFCKT(LOU)
      REAL L3,L4
      COMMON /B2/R1,R2,R3,R4,L3,L4,RBAR,V0,ZLDR,ZLPU,
     1   TNDR,TNPU,RDCDR,RDCPU,R0,R9,CAPDR,CAPPU
      WLCDR=1./SQRT(ZLDR*CAPDR)
      WLCPU=1./SQRT(ZLPU*CAPPU)
      WRITE(LOU,50)
      WRITE(LOU,60)R0,RDCDR,CAPDR,ZLDR,WLCDR,V0
      WRITE(LOU,70)R9,RDCPU,CAPPU,ZLPU,WLCPU
      WRITE(LOU,80)
   50 FORMAT('             SER/SHT RES COIL DC RES  SHUNT CAP. COIL INDUCT.
     1  RES.FREQ   DRIV. VOLT ')
   60 FORMAT(' DVR CKT ',2(F12.3),3(1PE12.4),0PF12.4)
   70 FORMAT(' PICK CKT',2(F12.3),3(1PE12.4),0PF12.4)
   80 FORMAT(1X)
      RETURN
      END
```

Sample Output

```
          SER/SHT RES COIL DC RES  SHUNT CAP. COIL INDUCT  RES.FREQ   DRIV. VOLT
DVR CKT     2000.000      67.300  8.4700E-11  2.2136E-03  2.3095E+06    10.0000
PICK CKT 1000000.000      67.300  8.4500E-11  2.2136E-03  2.3122E+06
```

PRFCOL

Subroutine PRFCOL (LOU) prints the coil parameters on the LOU output unit.

```
C
C      SUBROUTINE PRFCOL          (20 JULY 1977)
C
       SUBROUTINE PRFCOL(LOU)
C
C       SUBROUTINE TO PRINT OUT THE REFLECTION COIL PARAMETERS.(20 JULY 77)
C
       REAL L3,L4,L2,L5,L6
       COMMON /B2/R1,R2,R3,R4,L3,L4,RBAR,V0,ZLDR,ZLPU,
      1   TNDR,TNPU,RDCDR,RDCPU,R0,R9,CAPDR,CAPPU
       COMMON /B6/ L2,L5,L6
       WRITE(LOU,50)RBAR
       WRITE(LOU,60)
       WRITE(LOU,70)R1,R2,L3,TNDR,L6
       WRITE(LOU,80)R3,R4,L4,TNPU,L5
       WRITE(LOU,90)
    50 FORMAT(' MEAN RADIUS ',F12.5,' INCHES')
    60 FORMAT('    COIL      INN. RAD    OUT. RAD      LENGTH     TURNS EA
      1   0 L.O./RCES ')
    70 FORMAT(' DRIVER   ',3(F12.4),F12.1,F14.4)
    80 FORMAT(' PICK-UP ',3(F12.4),F12.1,F14.4)
    90 FORMAT(1X)
       RETURN
       END
```

<u>Sample Output</u>

```
MEAN RADIUS        0.15000 INCHES
   COIL      INN. RAD    OUT. RAD      LENGTH     TURNS EA    0 L.O./RCES
DRIVER         0.7500      1.2500      0.3600       513.0         0.0750
PICK-UP        0.7500      1.2500      0.3600       513.0         0.0770
```

PRFVLT

Subroutine PRFVLT prints the output voltage magnitudes and phases for the different property sets and frequencies. (See p. 70 for sample output.)

```
C
C      SUBROUTINE PRFVLT     (20 JULY 1977)
C
       SUBROUTINE PRFVLT(LOU,TMAG,PHASE,NRT,NPT,NFT,NLT,NODF,NPTT,RL1
      1,NRDF,FREQ,GAIN)
C
C      PRINTS OUT THE MAGNITUDES AND PHASES FOR THE DIFFERENT PROPERTY
C      SETS,FREQUENCIES AND WITH AND WITHOUT DEFECTS.
C
       REAL L2,L5,L6
       DIMENSION RL1(NLT),TMAG(NLT,NPTT,NFT),PHASE(NLT,NPTT,NFT)
       DIMENSION NRDF(NPT),FREQ(NFT),GAIN(NFT)
       COMMON /B6/ L2,L5,L6
       RL1(1)=L6
       DO 10 NL=2,NLT
       RL1(NL)=RL1(NL-1)+L2
    10 CONTINUE
       DO 110 NF=1,NFT
       WRITE(LOU,190)
       WRITE(LOU,150)FREQ(NF),GAIN(NF)
       WRITE(LOU,160)(RL1(NL),NL=1,NLT)
       WRITE(LOU,165)
       NPP=1
       DO 100 NP=1,NPT
       WRITE(LOU,170)NP,(TMAG(NL,NPP,NF),NL=1,NLT)
       WRITE(LOU,180)(PHASE(NL,NPP,NF),NL=1,NLT)
       WRITE(LOU,190)
       NPP=1+NPP
       IF(NRDF(NP).GE.NRT) GO TO 20
       WRITE(LOU,200)(TMAG(NL,NPP,NF),NL=1,NLT)
       WRITE(LOU,210)(PHASE(NL,NPP,NF),NL=1,NLT)
       WRITE(LOU,190)
       NPP=1+NPP
    20 CONTINUE
   100 CONTINUE
   110 CONTINUE
   150 FORMAT(' FREQUENCY ',1PE13.5,'   GAIN ',1PE13.5)
   160 FORMAT('  PROP  LFT OF ',9(F9.4))
   165 FORMAT('   SET')
   170 FORMAT(1X,I5,'     MAG  ',9(F9.4))
   180 FORMAT('              PHA  ',9(F9.3))
   190 FORMAT(1X)
   200 FORMAT('  WITH     MAG  ',9(F9.4))
   210 FORMAT('   DEF     PHA  ',9(F9.3))
       RETURN
       END
```

PTRVLT

This routine is a shortened version of PRFVLT with the capability of printing defect values removed for use with the through-transmission program (MULTRU).

```
C
C       SUBROUTINE PTRVLT       (1 DECEMBER 1977)
C
        SUBROUTINE PTRVLT(LOU,TMAG,PHASE,NRT,NPT,NFT,NLT,RL1
       1,FREQ,GAIN)
C
C       PRINTS OUT THE MAGNITUDES AND PHASES FOR THE DIFFERENT PROPERTY
C       SETS AND FREQUENCIES.
C
        REAL L2,L5,L6
        DIMENSION RL1(NLT),TMAG(NLT,NPT,NFT),PHASE(NLT,NPT,NFT)
        DIMENSION FREQ(NFT),GAIN(NFT)
        COMMON /B6/ L2,L5,L6
        RL1(1)=L6
        DO 10 NL=2,NLT
        RL1(NL)=RL1(NL-1)+L2
     10 CONTINUE
        DO 110 NF=1,NFT
        WRITE(LOU,190)
        WRITE(LOU,150)FREQ(NF),GAIN(NF)
        WRITE(LOU,160)(RL1(NL),NL=1,NLT)
        WRITE(LOU,165)
        DO 100 NP=1,NPT
        WRITE(LOU,170)NP,(TMAG(NL,NP,NF),NL=1,NLT)
        WRITE(LOU,180)(PHASE(NL,NP,NF),NL=1,NLT)
     20 CONTINUE
    100 CONTINUE
    110 CONTINUE
    150 FORMAT(' FREQUENCY ',1PE13.5,'    GAIN ',1PE13.5)
    160 FORMAT('   PROP   LFT OF ',9(F9.4))
    165 FORMAT('    SET')
    170 FORMAT(1X,I5,'     MAG   ',9(F9.4))
    180 FORMAT('            PHA   ',9(F9.3))
    190 FORMAT(1X)
        RETURN
        END
C
```

RDGEXP

Subroutine RDGEXP constructs the READNG array from a polynomial expansion of various powers of various functions of the magnitude and phase readings.

The "recipe" for the expansion is contained in the value of IOFSET and in the NPOL array (see Table 3). Subroutine RDGEXP constructs the Mth row of the READNG array, which will have the following elements (total IRDPRM + 1): $(1, f_1(M_1), \ldots, [f_1(M_1)]^{i_1}, g_1(P_1), \ldots, [g_1(P_1)]^{j_1}$, $f_1(M_1)[g_1(P_1)]^{k_1-2} \ldots, [f_1(M_1)]^{k_1-1}g_1(P_1), \ldots [f_n(M_n)]^{k_n-1}g_n(P_n), 0,$ $\ldots, 0$, where the number of zeroes will equal (IRDPRM + 1) − IRDPR. If IOFSET = 0 the leading 1 is omitted and all other terms move one position left, adding another 0 at the right or making another function term available if needed.

```
C
      SUBROUTINE RDGEXP(READNG,TMAG,PHASE,NPOL,IOFSET,TMDFT,PHDFT,MSET1
     1,IRDPR1,M,NFT,NL,NLT,NP,NPTT)
C     REAL*8 READNG,RDG
      DIMENSION READNG(MSET1,IRDPR1),NPOL(6,NFT),TMDFT(NFT),PHDFT(NFT)
      DIMENSION TMAG(NLT,NPTT,NFT),PHASE(NLT,NPTT,NFT)
C
C     NPOL CONTAINS A NUMBER FOR THE FUNCTION TYPE, THE POLYNOMIAL
C     DEGREE, AND THE NUMBER OF CROSS TERMS
C     FOR THE MAGNITUDE AND PHASE AT EACH FREQUENCY,STORED AS NPOL
C     (NF;1-MAG FUN, 2-MAG POL,3-MAG #CROSS TERMS,4-PH FUN,5-PH POL,
C     6-PH # CROSS TERMS).  IF IOFSET=0, NO OFF-SET
C     WILL BE INCLUDED,=1 OFF-SET IS INCLUDED.THE VALUES OF TMDFT(NF)&
C     PHDFT(NF) GIVE THE AMOUNT OF DRIFT IN THE MAGNITUDE AND PHASE.IF
C      NPOL(NCP,NF) =0,THAT PARTICULAR MAGNITUDE AND PHASE FOR THAT
C     FREQUENCY WILL BE SKIPPED.
C
      READNG(M,1)=1.
      N=1
      IF(IOFSET.EQ.1)N=2
      DO 210 NF=1,NFT
      DO 200 NC=1,2
      NCC=NC*3
      NCP=NCC-1
      NCF=NCP-1
      ROLD=RDG
      IF(NPOL(NCP,NF).EQ.0) GO TO 200
      IF(NC.EQ.1) RDG=TMAG(NL,NP,NF)+TMDFT(NF)
      IF(NC.EQ.2) RDG=PHASE(NL,NP,NF)+PHDFT(NF)
C
C     THE TYPE OF FUNCTION IS SELECTED
C
      IF(NPOL(NCF,NF).EQ.1)RDG=RDG
      IF(NPOL(NCF,NF).EQ.2)RDG=ALOG(RDG)
      IF(NPOL(NCF,NF).EQ.3)RDG=EXP(RDG)
      IF(NPOL(NCF,NF).EQ.4)RDG=1./RDG
C
C     THE TYPE OF POLYNOMIAL IS SELECTED
C     AND THE POLYNOMIAL VALUES ARE CONSTRUCTED.
```

```
C
      READNG(M,N)=RDG
      N=N+1
      NDEG=NPOL(NCP,NF)-1
      IF(NDEG.LT.1) GO TO 15
      DO 10 I=1,NDEG
      READNG(M,N)=RDG*READNG(M,N-1)
      N=N+1
   10 CONTINUE
C
C     CROSS TERMS ARE CONSTRUCTED
C
15    IF(NPOL(NCC,NF).EQ.0) GO TO 200
      RDY=ROLD
      NCTERM=NPOL(NCC,NF)
      DO 20 I=1,NCTERM
      RDY=ROLD*RDY
   20 CONTINUE
      IF(RDY.NE.0) RINV=RDG/ROLD
      IF(RDY.EQ.0) RINV=0.
      DO 30 I=1,NCTERM
      RDY=RDY*RINV
      READNG(M,N)=RDY
      N=N+1
   30 CONTINUE
C
  200 CONTINUE
  210 CONTINUE
      RETURN
      END
```

READAT

This subroutine controls the MODACS to read signal voltages from the eddy-current instrument. The section between the INLINE (line 1043) and FINI (line 1063) instructions is written in the ModComp IV Assembly language;[2] therefore, it is unique to the minicomputer used for this work. For other machines this section could be replaced by other appropriate instructions. The ITIMS is currently set at 512 in the main programs. The NRET flag assures that one complete set is read into the zeroed MARRAY array and returned to the main program in the VOLTS array.

```
1020  C
1021        SUBROUTINE READAT(VOLTS,NCHS,ITIMS,NRET)
1022  C
1023  C     SUBROUTINE TO READ DATA FROM AN EDDY CURRENT TEST USING THE MODAC.
1024  C     PROGRAM SENDS PROMPT SIGNALS OVER THE AO2 CHANNEL AND DISPLAYS THE
1025  C     CURRENT READING VALUES OF THE DESIGNATED CHANNELS OVER AO2.
1026  C     NOTE: SYSTEM PROTECT SWITCH MUST BE OFF FOR PROGRAM TO RUN.
1027  C     AFTER CH#12 HAS GONE LOW(FOOT PEDAL DEPRESSED) AND ROUTINE HAS
1028  C        RETURNED ONCE TO SIGNAL PROGRAM TO MAKE READING (NRET=1),
1029  C        THEN HANG IN HERE (LOOP NRET=0) UNTIL CH#12 GOES HIGH
1030  C        (>=2.0 VOLTS) AGAIN IMPLYING THAT FOOT PEDAL HAS BEEN
1031  C        RELEASED.
1032  C
```

```
      INTEGER*2 ITIM,NCH
      DIMENSION VOLTS(12),MARRAY(12)
      ITIM=ITIMS
      CONFAC=819.2*FLOAT(ITIMS)
      NCH=2*(NCHS-1)
C     ZERO ARRAY INITIALLY.
30    DO 10 N=1,NCHS
      MARRAY(N)=0
   10 CONTINUE
C     LOOP THROUGH THE CHANNELS,NCHS,ITIMS TIMES AND SUM THE READINGS.
      INLINE
      LDM,6           ITIM
      LDI,7           0
ITLP  LDI,3           1
      LDI,2           0
CHLP  ABR,3,0
      ODD,3,2
      SBR,3,0
      ODD,3,2
      LAS,3,1
BUSY  IDD,5,0
      TBRB,5,14        BUSY
      ODD,7,2
      RAS,5,2
      ESS,4,5
      ADMD,4,2         MARRAY
      STMD,4,2         MARRAY
      ABR,2,14
      CRMB,2           NCH,CHLP,CHLP
      SBRB,6,15        ITLP
      FINI
      DO 20 N=1,NCHS
      VOLTS(N)=FLOAT(MARRAY(N))/CONFAC
   20 CONTINUE
      IF(VOLTS(12).LT.2.0.AND.NRET.EQ.0) GO TO 30
      NRET=1
      RETURN
      END
```

RFCIRK

Subroutine RFCIRK calculates the magnitudes and phases of the output voltage [Eq. (14)] for the different property sets, frequencies, and liftoffs. (The circuit analysis begins on p. 11.) Real and imaginary parts of Z_D, $Z_P^{(R)}$, and $M^{(R)}$ are substituted into Eq. (14) for V_{out}, and the real and imaginary parts of V_{out} are separated. Then, the magnitude of the output voltage is

$$|V_{\text{out}}| = \left\{[\text{Re}(V_{\text{out}})]^2 + [\text{Im}(V_{\text{out}})]^2\right\}^{\frac{1}{2}} ,$$

and the phase is

$$\phi = \tan^{-1}\left[\frac{\text{Im}(V_{\text{out}})}{\text{Re}(V_{\text{out}})}\right] .$$

Conversion factors and leading constants include:

1. 1.0027518 E-7 = π(4π E-7 H/m)(0.0254 m/in.), and
2. 0.16880931 = $3\sqrt{2}/8\pi$, which corrects for value of $\mu^* = \mu/\sqrt{2}$ divided out of $\omega\mu\sigma\bar{r}^2$ and $3/8\pi$ difference between defect and defect-free impedance equations.

The loop structure is so arranged that the first pass computes $|V_{\text{out}}|$ and ϕ without defects (DEF = 0), and the second pass includes the defects (DEF = DEFF).

Coil impedance variables are designated ZXXR or ZXXI for real or imaginary parts, where XX is MU for mutual, DR for driver, and PU for pickup. Term TMAG = $|V_{\text{out}}|$ and PHASE = ϕ.

```
C
C       RFCIRK.MOD     (20 JULY 1977)
C
      SUBROUTINE RFCIRK(TMAG,PHASE,TWOPI,RAD,TMUTRE,TMUTIM,DRIVRE
     1,DRIVIM,PICKRE,PICKIM,DRVDR,DRVDI,PICKDR,PICKDI,NRDF,FREQ
     2,U,WUSR,GAIN,AIR1,AIR2,NLT,NRT,NPT,NFT,NODF,NPTT,DFSIZE)
C
C       COMPUTES MAGNITUDES AND PHASES OF OUTPUT VOLTAGE
C       FOR VARIOUS PROPERTIES,FREQUENCIES AND LIFTOFFS.
C       SEE ORNL-TM-4107, PP. 3-5 FOR THE CIRCUIT ANALYSIS.
      DIMENSION TMAG(NLT,NPTT,NFT),PHASE(NLT,NPTT,NFT)
     1,U(NRT,NPT),WUSR(NRT,NPT,NFT),NRDF(NPT),FREQ(NFT),GAIN(NFT)
     2,TMUTRE(NLT,NPT,NFT),TMUTIM(NLT,NPT,NFT),DRIVRE(NLT,NPT,NFT)
     3,DRIVIM(NLT,NPT,NFT),PICKRE(NLT,NPT,NFT),PICKIM(NLT,NPT,NFT)
      DIMENSION DRVDR(NLT,NPT,NFT),PICKDR(NLT,NPT,NFT)
      DIMENSION DRVDI(NLT,NPT,NFT),PICKDI(NLT,NPT,NFT)
```

```
      COMMON /B2/R1,R2,R3,R4,RL3,RL4,RBAR,VO,ZLDR,ZLPU,
     1   TNDR,TNPU,RDCDR,RDCPU,RO,R9,CAPDR,CAPPU
      T1 = TNDR/((R2 - R1)*RL3)
      T2 = TNPU/((R4 - R3)*RL4)
      COLFAC=1.0027518E-7*RBAR
      DEFAC=.16880931*DFSIZE
      DVRFAC=COLFAC*T1*T1
      PICFAC=COLFAC*T2*T2
      ZMUTFC=COLFAC*T1*T2
      ZLDR=DVRFAC*AIR1
      ZLPU=PICFAC*AIR2
      DO 100 NF=1,NFT
      W=TWOPI*FREQ(NF)
      DVRF=DVRFAC*W
      PICF=PICFAC*W
      ZMUTF=ZMUTFC*W
      QT = VO*R9*GAIN(NF)
      X1 = W*RO*CAPDR
      X2 = W*R9*CAPPU
      Z1Z2RE = X1*X2 - 1.
      Z1Z2IM = -X1 - X2
      NPP=1
      DO 60  NP=1,NPT
      DEFF=DEFAC*WUSR(NRDF(NP),NP,NF)*U(NRDF(NP),NP)
      DEF=0.
   20   DO 50 NL=1,NLT
          ZMUR = ZMUTF*(TMUTRE(NL,NP,NF)+DEF*(DRVDR(NL,NP,NF)
     1 *PICKDI(NL,NP,NF)+PICKDR(NL,NP,NF)*DRVDI(NL,NP,NF)))
          ZMUI = ZMUTF*(TMUTIM(NL,NP,NF)-DEF*(DRVDR(NL,NP,NF)
     1 *PICKDR(NL,NP,NF)-DRVDI(NL,NP,NF)*PICKDI(NL,NP,NF)))
          ZDRR = -DVRF*(DRIVIM(NL,NP,NF)-DEF*(DRVDR(NL,NP,NF)
     1 *DRVDR(NL,NP,NF)-DRVDI(NL,NP,NF)*DRVDI(NL,NP,NF)))
          ZDRI = DVRF*(DRIVRE(NL,NP,NF) + AIR1+
     1 2*DEF*(DRVDR(NL,NP,NF)*DRVDI(NL,NP,NF)))
          ZPUR = -PICF*(PICKIM(NL,NP,NF)-DEF*(PICKDR(NL,NP,NF)
     1 *PICKDR(NL,NP,NF)-PICKDI(NL,NP,NF)*PICKDI(NL,NP,NF)))
          ZPUI = PICF*(PICKRE(NL,NP,NF) + AIR2+
     1 2*DEF*(PICKDR(NL,NP,NF)*PICKDI(NL,NP,NF)))
          ZPR = ZDRR + RDCDR
          C1 = ZPR*X1 + ZDRI
          C2 = X1*ZDRI - RO - ZPR
          ZPR = ZPUR + RDCPU
          C3 = ZPR*X2 + ZPUI
          C4 = X2*ZPUI - R9 - ZPR
          Z5SQRE = (ZMUR + ZMUI)*(ZMUR - ZMUI)
          ZPR = ZMUR*ZMUI
          Z5SQIM = ZPR + ZPR
          DENRE = Z1Z2RE*Z5SQRE - Z1Z2IM*Z5SQIM + C1*C3 - C2*C4
          DENIM = Z1Z2RE*Z5SQIM + Z1Z2IM*Z5SQRE + C1*C4 + C2*C3
          TNMRE = QT*ZMUI
          TNMIM = -QT*ZMUR
          TMAG(NL,NPP,NF) =
     1       SQRT((TNMRE**2 + TNMIM**2)/(DENRE**2 + DENIM**2))
          PHASE(NL,NPP,NF) =RAD*(
     1       ATAN2(TNMIM*DENRE - TNMRE*DENIM,
     2             TNMRE*DENRE + TNMIM*DENIM))
   50   CONTINUE
      IF(NRDF(NP).GE.NRT)GO TO 55
      IF(DEF.GT.O.)GO TO 55
      DEF=DEFF
      NPP=NPP+1
      GO TO 20
   55   NPP=NPP+1
   60 CONTINUE
  100 CONTINUE
      RETURN
      END
```

The following is a sample output:

```
          SER/SHT RES COIL DC RES  SHUNT CAP. COIL INDUCT  RES.FREQ   DRIV. VOLT
DVR CKT      200.000       67.300  8.4700E-11  2.2136E-03  2.3095E+06      3.5350
PICK CKT 1000000.000      531.000  8.4500E-11  1.7222E-03  2.6214E+06
```

RFCKDT

Subroutine RFCKDT adds a 1% drift to the value of one circuit parameter at a time and calculates the resultant change in the calculated property, which was preselected in the main program. Its function is similar to that of the sections of main programs that do DIFF and DRIFT calculations (e.g., MULTRU, lines 337–382), except that only DRIFT is computed, and the variation is added to circuit parameters instead of magnitude or phase readings at a particular frequency.

```
666  C
667         SUBROUTINE RFCKDT(TMAG,PHASE,TWOPI,RAD,TMUTRE,TMUTIM,DRIVRE,
668        1DRIVIM,PICKRE,PICKIM,DRVDR,DRVDI,PICKDR,PICKDI,NRDF,FREQ
669        2,U,WUSR,GAIN,AIR1,AIR2,NLT,NRT,NPT,NFT,NODF,NPTT,DFSIZE,
670        3READNG,NPOL,IOFSET,TMDFT,PHDFT,MSET1,IRDPR1,M,NL,NP,LOU,
671        4MSET,IRDPR,CKTPAR,COEF,PROP,PROPT)
672  C
673  C      SUBROUTINE TO ADD A SMALL DRIFT IN A CIRCUIT OR COIL PARAMETER
674  C      AND CALCULATE THE RESULTANT CHANGE IN THE CALCULATED PROPERTY.
675  C
676         DIMENSION READNG(MSET1,IRDPR1),PROP(MSET),COEF(IRDPR)
677         DIMENSION TMDFT(NFT),PHDFT(NFT),NPOL(6,NFT)
678         DIMENSION CKTPAR(8)
679         DIMENSION TMAG(NLT,NPTT,NFT),PHASE(NLT,NPTT,NFT)
680         DIMENSION WUSR(NRT,NPT,NFT),U(NRT,NPT)
681         DIMENSION NRDF(NPT),FREQ(NFT),GAIN(NFT)
682        *, TMUTRE(NLT,NPT,NFT),TMUTIM(NLT,NPT,NFT),DRIVRE(NLT,NPT,NFT)
683        *, DRIVIM(NLT,NPT,NFT),PICKRE(NLT,NPT,NFT),PICKIM(NLT,NPT,NFT)
684         DIMENSION DRVDR(NLT,NPT,NFT),PICKDR(NLT,NPT,NFT)
685         DIMENSION DRVDI(NLT,NPT,NFT),PICKDI(NLT,NPT,NFT)
686         DIMENSION CKTVAL(18)
687         COMMON /B2/R1,R2,R3,R4,RL3,RL4,RBAR,VO,ZLDR,ZLPU,
688        *TNDR,TNPU,RDCDR,RDCPU,RO,R9,CAPDR,CAPPU
689         EQUIVALENCE(R1,CKTVAL)
690         WRITE(LOU,10)PROPT
691      10 FORMAT(' 1%CHANGE       ',A4,' CHA')
692         DO 800 NDRIFT=1,8
693         CKTVAL(NDRIFT+10)=CKTVAL(NDRIFT+10)*1.01
694         CALL RFCIRK(TMAG,PHASE,TWOPI,RAD,TMUTRE,TMUTIM,DRIVRE
695        1,DRIVIM,PICKRE,PICKIM,DRVDR,DRVDI,PICKDR,PICKDI,NRDF,FREQ
696        2,U,WUSR,GAIN,AIR1,AIR2,NLT,NRT,NPT,NFT,NODF,NPTT,DFSIZE)
697         SSDRIF=0.
698         DO  80 NP=1,NPT
699         DO  70 NL=1,NLT
700         M=(NP-1)*NLT+NL
701         CALL RDGEXP(READNG,TMAG,PHASE,NPOL,IOFSET,TMDFT,PHDFT,MSET1
702        1,IRDPR1,M,NFT,NL,NLT,NP,NPTT)
703  C
```

```
C       THE POLYNOMIAL IS CALCULATED
C
        SUM=0.
        DO  20 IR=1,IRDPR
        SUM=SUM+COEF(IR)*READNG(M,IR)
     20 CONTINUE
        DRIFT=ABS(READNG(M,IRDPR1)-SUM)
        SSDRIF=SSDRIF+DRIFT*DRIFT
     70 CONTINUE
     80 CONTINUE
        SDRIF=SQRT(SSDRIF/FLOAT(MSET))
        WRITE(LOU,60)CKTPAR(NDRIFT),SDRIF
     60 FORMAT(5X,A4,F15.5)
        CKTVAL(NDRIFT+10)=CKTVAL(NDRIFT+10)/1.01
    800 CONTINUE
        CALL RFCIRK(TMAG,PHASE,TWOPI,RAD,TMUTRE,TMUTIM,DRIVRE
      1,DRIVIM,PICKRE,PICKIM,DRVDR,DRVDI,PICKDR,PICKDI,NRDF,FREQ
      2,U,WUSR,GAIN,AIR1,AIR2,NLT,NRT,NPT,NFT,NODF,NPTT,DFSIZE)
        RETURN
        END
```

RFCOLM

Subroutine RFCOLM calculates the coil properties — the real and imaginary parts of the integrands for Z_D, $M^{(R)}$, and $Z_P^{(R)}$ [Eqs. (18), (20), and (22)] and Z_D' and Z_P' [Eqs. (51), (53), and (54)]. It calls the other subroutines BESSEL, GAMAML, and BESEL1. The terms AIR1 and AIR2 are the driver and pickup coil impedances with no conductors present.

The BESSEL subroutine returns

$$\text{XJR1} = \int_0^{\alpha R_1} x J_1(x)\ dx \ ,$$

so that

$$\text{D21} = \int_{\alpha R_1}^{\alpha R_2} x J_1(x)\ dx = J(R_2, R_1) \ .$$

Similarly,

$$\text{D43} = J(R_4, R_3) \ .$$

Then

$$S_6 = \alpha J(R_4,R_3) \ ,$$

and S3 through S7 equal similar expressions. So,

$$EX3 = \exp(-\alpha L_3) \ ,$$

and

$$W1 = 1 - \exp(-\alpha L_3) \ ,$$

with similar expressions for W2, EX1, EX4, EX5, and EX7.

The terms TMUT, etc., represent the factors multiplied by γ's, which enter the integrands of Eqs. (18), (20), and (22). Note that the variables positioned as DRF and DFT in the call have been replaced by DFRAD and DFLOC in the main program.

Subroutine BESEL1 returns $J_1(\alpha r)$ where r is the radial location of the defect (Fig. 7). The remaining equations combine the various factors to generate the integrands returned to the main program. Terms TMUTRE/IM, DRIVRE/IM, and PICKRE/IM go to Eqs. (18), (20), and (22), respectively; DRVDR/I and PICKDR/I go to Eqs. (53) and (54).

```
C
C        RFCOLM (13 JULY 1977)
C
      SUBROUTINE RFCOLM(S1,NLT,NPT,NRT,NFT,RL1,RL2,GAMAR,GAMAI
     1,GAMADR,GAMADI,TMUTRE,TMUTIM,DRIVRE,DRIVIM,PICKRE,PICKIM
     2,DRVDR,DRVDI,PICKDR,PICKDI,BETAR,BETAI,COSF,SINF,XPON
     3,WUSR,U,TH,NRDF,DFR,DFT,FREQ,AIR1,AIR2)
      DIMENSION BETAR(NRT),BETAI(NRT),COSF(NRT),SINF(NRT),XPON(NRT)
      DIMENSION WUSR(NRT,NPT,NFT),U(NRT,NPT),TH(NRT,NPT)
      DIMENSION RL1(NLT),RL2(NLT),NRDF(NPT),DFR(NPT),DFT(NPT),FREQ(NFT)
      DIMENSION GAMAR(NPT),GAMAI(NPT),GAMADR(NPT),GAMADI(NPT)
     1,TMUTRE(NLT,NPT,NFT),TMUTIM(NLT,NPT,NFT),DRIVRE(NLT,NPT,NFT)
     2,DRIVIM(NLT,NPT,NFT),PICKRE(NLT,NPT,NFT),PICKIM(NLT,NPT,NFT)
      DIMENSION DRVDR(NLT,NPT,NFT),PICKDR(NLT,NPT,NFT)
      DIMENSION DRVDI(NLT,NPT,NFT),PICKDI(NLT,NPT,NFT)
      REAL L3,L4,L2,L5,L6
      COMMON /B1/X,XX,XXXX
      COMMON /B2/R1,R2,R3,R4,L3,L4,RBAR,V0,ZLDR,ZLPU,
     1   TNDR,TNPU,RDCDR,RDCPU,RO,R9,CAPDR,CAPPU
      COMMON /B6/ L2,L5,L6
C
C       CALCULATION OF THE COIL PART OF THE INTEGRAND. SUBPROGRAM FOR
C       FOR REFLECTION TYPE COILS.
C
C
```

```
C          SUBROUTINE BESSEL EVALUATES THE INTEGRAL OF
C          THE PRODUCT OF THE BESSEL FUNCTION J1(X) AND ITS
C          ARGUMENT, X.
C
        CALL BESSEL(XJR2,X,R2)
        CALL BESSEL(XJR1,X,R1)
        CALL BESSEL(XJR4,X,R4)
        CALL BESSEL(XJR3,X,R3)
        D21 = XJR2 - XJR1
        D43 = XJR4 - XJR3
        S6 = S1*D43
        S3 = S6*D21
        S7=S1*D21
        S4=S7*D21
        S5 = S6*D43
        EX3 = EXP(-X*L3)
        W1 = 1. - EX3
        EX4 = EXP(-X*L4)
        W2 = 1. - EX4
        EX5 = EXP(-X*L5)
        W3 = EX3/(EX4*EX4*EX5*EX5)
C
C       UPDATE OF AIR VALUES
C
        AIR1=AIR1+(S4+S4)*(X*L3-W1)
        Q3=X*L4-W2
        AIR2=AIR2+(S5+S5)*(Q3+Q3-W2*W2*W3)
C
C          BYPASS THE UPDATE OF MUTUAL INDUCTANCE QUANTITIES
C          FOR LARGE X.
C
        IF (X .GT. 30.0) GO TO 200
        EX1 = EXP(-X*L2)
        EX7 = EXP(-X*L6)
        EX2=EX1*EX1
        EX6=EX7*EX7
        W4 = 1. - EX4*W3
        W6 = EX6*W1
        W7 = EX5*W2*W4
        W8=W7*EX7
        W9=W1*EX7
        W5 = W7*W6
        W6 = W6*W1
        W7 = EX6*W7*W7
        RL1(1)=1.0
        RL2(1) = 1.0
        DO  50 NL=2,NLT
        LMINUS=NL-1
        RL1(NL)=EX1*RL1(LMINUS)
        RL2(NL)=EX2*RL2(LMINUS)
     50 CONTINUE
        TMUT = S3*W5
        DVR = S4*W6
        PIC = S5*W7
        DVRD=S7*W9
        PICD=S6*W8
C
C
C       LOOP OVER ALL NFT FREQUENCIES.
C
        DO 160 NF=1,NFT
C
C        LOOP OVER ALL THE NPT DIFFERENT SETS OF PROPERTIES,CALCULATING THE
C        VARIOUS GAMMA AND DEFECT FACTORS AS WE GO.
C
        DO 155 NP=1,NPT
        CALL GAMAML(NRT,NPT,NP,NFT,NF,BETAR,BETAI,COSF,SINF,XPON
       1,WUSR,U,TH,GAMAR,GAMAI,GAMADR,GAMADI,DFT,DFR,NRDF)
```

```
          TMUR = TMUT*GAMAR(NP)
          TMUI = TMUT*GAMAI(NP)
          DVRR = DVR*GAMAR(NP)
          DVRI = DVR*GAMAI(NP)
          PICR = PIC*GAMAR(NP)
          PICI = PIC*GAMAI(NP)
        IF(NRDF(NP).GT.NRT) GO TO 100
        Q1=X*DFR(NP)
        CALL BESEL1(Q1,RJ1)
        DRDR=DVRD*RJ1*GAMADR(NP)
        DRDI=DVRD*RJ1*GAMADI(NP)
        PICDR=PICD*RJ1*GAMADR(NP)
        PICDI=PICD*RJ1*GAMADI(NP)
    100 CONTINUE
          DO 150 NL=1,NLT
            TMUTRE(NL,NP,NF) = TMUTRE(NL,NP,NF) + TMUR*RL2(NL)
            TMUTIM(NL,NP,NF) = TMUTIM(NL,NP,NF) + TMUI*RL2(NL)
            DRIVRE(NL,NP,NF) = DRIVRE(NL,NP,NF) + DVRR*RL2(NL)
            DRIVIM(NL,NP,NF) = DRIVIM(NL,NP,NF) + DVRI*RL2(NL)
            PICKRE(NL,NP,NF) = PICKRE(NL,NP,NF) + PICR*RL2(NL)
            PICKIM(NL,NP,NF) = PICKIM(NL,NP,NF) + PICI*RL2(NL)
           IF(NRDF(NP).GT.NRT) GO TO 150
            DRVDR(NL,NP,NF)=DRVDR(NL,NP,NF)+DRDR*RL1(NL)
            DRVDI(NL,NP,NF)=DRVDI(NL,NP,NF)+DRDI*RL1(NL)
            PICKDR(NL,NP,NF)=PICKDR(NL,NP,NF)+PICDR*RL1(NL)
            PICKDI(NL,NP,NF)=PICKDI(NL,NP,NF)+PICDI*RL1(NL)
   150    CONTINUE
   155  CONTINUE
    160 CONTINUE
  C
   200  CONTINUE
        RETURN
        END
```

TCOFIL

This routine is similar in function to COLFIL (p. 160) but specialized to the through-transmission case. It allows the selection of coil names for the driver and the pickup. Since all coils for which data are filed are reflection types, the driver data for the second coil (except the mean radius) are normalized with respect to the driver (first selected coil) mean radius; they are then assigned to variables representing pickup coil parameters and thus returned to the main program.

```
C
      SUBROUTINE TCOFIL (LI,LOU)
C     *** SUBROUTINE TO LOOK UP/ENTER SEPARATE DRIVER & PICKUP COIL DATA
C         FOR THROUGH-TRANSMISSION PROGRAMS ***
C      ** USES DRIVER OF 2ND REFLECTION PROBE FOR PICKUP **
      REAL*6 COIL, CNAM
      REAL L2,L3,L4,L5,L6
      DIMENSION CNAM(2)
      COMMON/B2/R1,R2,R3,R4,L3,L4,RBAR,V0,ZLDR,ZLPU,TNDR,TNPU,RDCDR,
     *RDCPU,R0,R9,CAPDR,CAPPU
```

```
      COMMON/B6/L2,L5,L6
      DEFINE FILE 28(80,32,U,NCOL)
C
      DATA LO2/5/
      WRITE (LO2,20)
   20 FORMAT( ' TYPE COIL NAMES-DRIVER, PICKUP (UP TO 6 LETTERS): ')
      READ (LI,25) (CNAM(I), I=1,2)
   25 FORMAT (A6,A6)
      WRITE (LOU,30) (CNAM(I), I=1,2)
   30 FORMAT (' COILS-DRIVER, PICKUP: ',A6,A6)
C
C     *** SEARCH FILE 28 FOR DATA ON SELECTED COILS ***
C
   40 I=1
   50 NCOL=1
   60 READ (28'NCOL) COIL, X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,X11,
     1 X12,X13
   80 IF (COIL.EQ.CNAM(I)) GO TO (120,140),I
      IF (COIL.EQ.'END   ') GO TO 150
      GO TO 60
C
C      *** COPY DESIRED DATA FROM FILE ***
C
  120 RBAR=X1
      R1=X2
      L3=X4
      R2=X3
      L6=X11
C     ZLDR=X14
      TNDR=X9
      RDCDR=X12
      I=2
      GO TO 50
C
  140 F=X1/RBAR
      R3=X2*F
      R4=X3*F
      L4=X4*F
      L5=X11*F
C     ZLPU=X14
      TNPU=X9
      RDCPU=X12
      GO TO 200
C
C     *** MANUAL INSERTION OF DATA NOT ON FILE ***
C
  150 WRITE (LO2,160) CNAM(I)
  160 FORMAT (' COIL ',A6,' NOT FOUND. TYPE FOLLOWING COIL PARAMETERS',
     1'(INSERT ',/,'  ZEROES FOR UNUSED ONES): RBAR,R1,R2,L3,R3,R4,L4,',
     2'L6,RDCDR,RDCPU,'/'  TNDR,TNPU')
      READ (LI,*) X1,X2,X3,X4,X5,X6,X7,X11,X12,X13,X9,X10
      GO TO (120,140),I
C
  200 RETURN
      END
```

TRCOLM

This routine computes the real and imaginary parts of the coil impedance integrands given by Eqs. (18), (19), and (21) for Z_D, $M^{(T)}$, and $Z_P^{(T)}$. The defect cases are not included (as in RFCOLM, p. 177). Coil spacing options are selected by the initialized value of NOPT, which determines the value of THTT used in the calculations. (See p. 73 for description of options.)

```
C
C        TRCOLM (12 MAY 78)
C
      SUBROUTINE TRCOLM(S1,NLT,NPT,NRT,NFT,RL1,GAMAR,GAMAI
     1,GAMADR,GAMADI,TMUTRE,TMUTIM,DRIVRE,DRIVIM,PICKRE,PICKIM
     2,GAMAPR,GAMAPI,BETAR,BETAI,COSF,SINF,XPON
     3,WUSR,U,TH,RRBAR,FREQ,AIR1,AIR2,XMAX,NOPT)
      DIMENSION BETAR(NRT),BETAI(NRT),COSF(NRT),SINF(NRT),XPON(NRT)
      DIMENSION WUSR(NRT,NPT,NFT),U(NRT,NPT),TH(NRT,NPT)
      DIMENSION RL1(NLT),FREQ(NFT)
      DIMENSION GAMAR(NPT),GAMAI(NPT),GAMADR(NPT),GAMADI(NPT)
      DIMENSION GAMAPR(NPT),GAMAPI(NPT)
     1,TMUTRE(NLT,NPT,NFT),TMUTIM(NLT,NPT,NFT),DRIVRE(NLT,NPT,NFT)
     2,DRIVIM(NLT,NPT,NFT),PICKRE(NLT,NPT,NFT),PICKIM(NLT,NPT,NFT)
      REAL L3,L4,L2,L5,L6
      COMMON /B1/X,XX,XXXX
      COMMON /B2/R1,R2,R3,R4,L3,L4,RBAR,V0,ZLDR,ZLPU,
     1    TNDR,TNPU,RDCDR,RDCPU,R0,R9,CAPDR,CAPPU
      COMMON /B6/ L2,L5,L6
C
C       CALCULATION OF THE COIL PART OF THE INTEGRAND. SUBPROGRAM FOR
C       THROUGH TRANSMISSION TYPE COILS.
C
C
C        SUBROUTINE BESSEL EVALUATES THE INTEGRAL OF
C        THE PRODUCT OF THE BESSEL FUNCTION J1(X) AND ITS
C        ARGUMENT, X.
C
      CALL BESSEL(XJR2,X,R2)
      CALL BESSEL(XJR1,X,R1)
      CALL BESSEL(XJR4,X,R4)
      CALL BESSEL(XJR3,X,R3)
      D21 = XJR2 - XJR1
      D43 = XJR4 - XJR3
      S3=S1*D21*D43
      S4=S1*D21*D21
      S5=S1*D43*D43
      WD=-1.
      IF(X*L3.GT.20.)GO TO 20
      WD=EXP(-X*L3)-1.
   20 WP=-1.
      IF(X*L4.GT.20)GO TO 30
      WP=EXP(-X*L4)-1.
C
C     UPDATE OF AIR VALUES
C
   30 AIR1=AIR1+(S4+S4)*(X*L3+WD)
      AIR2=AIR2+(S5+S5)*(X*L4+WP)
C
C        BYPASS THE UPDATE OF MUTUAL INDUCTANCE QUANTITIES
```

```
C         FOR LARGE X.
C
       IF (X .GT. 30.0) GO TO 200
       NRT1=NRT-1
       EX6=EXP(-X*L6)
       EX5=EXP(-X*L5)
       TMUT = S3*EX5*EX6*WD*WP
       DVR = S4*EX6*EX6*WD*WD
       PIC = S5*EX5*EX5*WP*WP
C
C
C      LOOP OVER ALL NFT FREQUENCIES.
C
       DO 160 NF=1,NFT
C
C       LOOP OVER ALL THE NPT DIFFERENT SETS OF PROPERTIES,CALCULATING THE
C      VARIOUS GAMMA FACTORS AS WE GO.
C
       DO 155 NP=1,NPT
       CALL GAMAMT(NRT,NPT,NP,NFT,NF,BETAR,BETAI,COSF,SINF,XPON
      1,WUSR,U,TH,RRBAR,GAMAR,GAMAI,GAMADR,GAMADI,GAMAPR,GAMAPI)
         TMUR = TMUT*GAMAR(NP)
         TMUI = TMUT*GAMAI(NP)
         DVRR = DVR*GAMADR(NP)
         DVRI = DVR*GAMADI(NP)
         PICR = PIC*GAMAPR(NP)
         PICI = PIC*GAMAPI(NP)
       THTT=THMAX*RRBAR
       DO 100 NR=2,NRT1
       THTT=THTT-TH(NR,NP)*RRBAR
  100  CONTINUE
       IF(NOPT.EQ.2) THTT=0.
        DO 150 NL=1,NLT
       XNL=FLOAT(NL-1)
        TMUTRE(NL,NP,NF)=TMUTRE(NL,NP,NF)+TMUR*EXP(-X*(L2*XNL+THTT))
        TMUTIM(NL,NP,NF)=TMUTIM(NL,NP,NF)+TMUI*EXP(-X*(L2*XNL+THTT))
        DRIVRE(NL,NP,NF)=DRIVRE(NL,NP,NF)+DVRR*EXP(-X*2.*(L2*XNL+THTT))
        DRIVIM(NL,NP,NF)=DRIVIM(NL,NP,NF)+DVRI*EXP(-X*2.*(L2*XNL+THTT))
        PICKRE(NL,NP,NF)=PICKRE(NL,NP,NF)+PICR
        PICKIM(NL,NP,NF)=PICKIM(NL,NP,NF)+PICI
 150     CONTINUE
 155   CONTINUE
  160  CONTINUE
 200   CONTINUE
       RETURN
       END
```

TRUCIR

This routine performs the same voltage calculations [Eq. (14)] as subroutine RFCIRK. It differs from RFCIRK as follows:

1. The arrays, dimensions, and indices to handle defects have been removed; certain equations that include defects have been modified.

2. Voltages for air gaps (between coils) are calculated.

3. Selected voltages for conductor combinations are normalized with respect to the appropriate air gap voltage (see p. 73 for discussion of normalization).

```
C
C         TRUCIR.MOD      (1 DECEMBER 1977)
C
      SUBROUTINE TRUCIR(TMAG,PHASE,TWOPI,RAD,TMUTRE,TMUTIM,DRIVRE
     1,DRIVIM,PICKRE,PICKIM,FREQ
     2,U,WUSR,GAIN,AIR1,AIR2,NLT,NRT,NPT,NFT,NORM)
C
C         COMPUTES MAGNITUDES AND PHASES OF OUTPUT VOLTAGE
C         FOR VARIOUS PROPERTIES,FREQUENCIES AND LIFTOFFS.
C         SEE ORNL-TM-4107, PP. 3-5 FOR THE CIRCUIT ANALYSIS.
      DIMENSION TMAG(NLT,NPT,NFT),PHASE(NLT,NPT,NFT)
     1,U(NRT,NPT),WUSR(NRT,NPT,NFT),FREQ(NFT),GAIN(NFT)
     2,TMUTRE(NLT,NPT,NFT),TMUTIM(NLT,NPT,NFT),DRIVRE(NLT,NPT,NFT)
     3,DRIVIM(NLT,NPT,NFT),PICKRE(NLT,NPT,NFT),PICKIM(NLT,NPT,NFT)
      COMMON /B2/R1,R2,R3,R4,RL3,RL4,RBAR,VO,ZLDR,ZLPU,
     1    TNDR,TNPU,RDCDR,RDCPU,RO,R9,CAPDR,CAPPU
      T1 = TNDR/((R2 - R1)*RL3)
      T2 = TNPU/((R4 - R3)*RL4)
      COLFAC=1.0027518E-7*RBAR
      DVRFAC=COLFAC*T1*T1
      PICFAC=COLFAC*T2*T2
      ZMUTFC=COLFAC*T1*T2
      ZLDR=DVRFAC*AIR1
      ZLPU=PICFAC*AIR2
      DO 100 NF=1,NFT
      W=TWOPI*FREQ(NF)
      DVRF=DVRFAC*W
      PICF=PICFAC*W
      ZMUTF=ZMUTFC*W
      QT = VO*R9*GAIN(NF)
      X1 = W*RO*CAPDR
      X2 = W*R9*CAPPU
      Z1Z2RE = X1*X2 - 1.
      Z1Z2IM = -X1 - X2
      NP=NPT
C     CALCULATION OF NLT AIR VALUES FOR EACH FREQUENCY
C
   20   DO 50 NL=1,NLT
          ZMUR = ZMUTF*TMUTRE(NL,NP,NF)
          ZMUI = ZMUTF*TMUTIM(NL,NP,NF)
          ZDRR = -DVRF*DRIVIM(NL,NP,NF)
          ZDRI = DVRF*DRIVRE(NL,NP,NF) + AIR1
          ZPUR = -PICF*PICKIM(NL,NP,NF)
          ZPUI = PICF*PICKRE(NL,NP,NF) + AIR2
```

```
              ZPR = ZDRR + RDCDR
              C1 = ZPR*X1 + ZDRI
              C2 = X1*ZDRI - RO - ZPR
              ZPR = ZPUR + RDCPU
              C3 = ZPR*X2 + ZPUI
              C4 = X2*ZPUI - R9 - ZPR
              Z5SQRE = (ZMUR + ZMUI)*(ZMUR - ZMUI)
              ZPR = ZMUR*ZMUI
              Z5SQIM = ZPR + ZPR
              DENRE = Z1Z2RE*Z5SQRE - Z1Z2IM*Z5SQIM + C1*C3 - C2*C4
              DENIM = Z1Z2RE*Z5SQIM + Z1Z2IM*Z5SQRE + C1*C4 + C2*C3
              TNMRE = QT*ZMUI
              TNMIM = -QT*ZMUR
              TMAG(NL,NP,NF) =
     1          SQRT((TNMRE**2 + TNMIM**2)/(DENRE**2 + DENIM**2))
              PHASE(NL,NP,NF) =RAD*(
     1          ATAN2(TNMIM*DENRE - TNMRE*DENIM,
     2                TNMRE*DENRE + TNMIM*DENIM))
   50       CONTINUE
          NPTT=NPT-1
          DO 60  NP=1,NPTT
            DO 55 NL=1,NLT
              ZMUR = ZMUTF*TMUTRE(NL,NP,NF)
              ZMUI = ZMUTF*TMUTIM(NL,NP,NF)
              ZDRR = -DVRF*DRIVIM(NL,NP,NF)
              ZDRI = DVRF*DRIVRE(NL,NP,NF) + AIR1
              ZPUR = -PICF*PICKIM(NL,NP,NF)
              ZPUI = PICF*PICKRE(NL,NP,NF) + AIR2
              ZPR = ZDRR + RDCDR
              C1 = ZPR*X1 + ZDRI
              C2 = X1*ZDRI - RO - ZPR
              ZPR = ZPUR + RDCPU
              C3 = ZPR*X2 + ZPUI
              C4 = X2*ZPUI - R9 - ZPR
              Z5SQRE = (ZMUR + ZMUI)*(ZMUR - ZMUI)
              ZPR = ZMUR*ZMUI
              Z5SQIM = ZPR + ZPR
              DENRE = Z1Z2RE*Z5SQRE - Z1Z2IM*Z5SQIM + C1*C3 - C2*C4
              DENIM = Z1Z2RE*Z5SQIM + Z1Z2IM*Z5SQRE + C1*C4 + C2*C3
              TNMRE = QT*ZMUI
              TNMIM = -QT*ZMUR
              TMAG(NL,NP,NF) =
     1          SQRT((TNMRE**2 + TNMIM**2)/(DENRE**2 + DENIM**2))
          IF(NORM.EQ.1)TMAG(NL,NP,NF)=TMAG(NL,NP,NF)/TMAG(NL,NPT,NF)
              PHASE(NL,NP,NF) =RAD*(
     1          ATAN2(TNMIM*DENRE - TNMRE*DENIM,
     2                TNMRE*DENRE + TNMIM*DENIM))
          IF(NORM.EQ.1)PHASE(NL,NP,NF)=PHASE(NL,NP,NF)-PHASE(NL,NPT,NF)
   55       CONTINUE
   60     CONTINUE
  100     CONTINUE
          RETURN
          END
```

VINITM

Subroutine VINITM is called by the GAMAML subroutine to calculate the initial $V_{ij}[V(2,1)]$ matrix in the multilayer theory and transform V_{ij} to the next region (NSRT + 1). The routine calculates the $V_{i2}'(2,1)$ matrix elements from the equation[1]

$$V_{i2}'(2,1) = [\beta_2 + (-1)^i \beta_1]$$

for $i = 1,2$. The term V1R = $\mathrm{Re}[V_{12}'(2,1)]$, etc., in the computer notation.

```
C
C       VINITM SUBROUTINE    (13 JULY 1977)
C
      SUBROUTINE VINITM(NSRT,NRT,NPT,NP,BETAR,BETAI,V1R,V1I,V2R,V2I
     1,VDR,VDI,NRDF,COSD,SIND,XPOND)
C     SUBROUTINE TO CALCULATE THE INITIAL VIJ MATRIX
C     AND TRANSFORM TO REGION NSRT+1.
      DIMENSION NRDF(NPT),BETAR(NRT),BETAI(NRT)
      NR=NSRT
      NSC=NR+1
      V1R=BETAR(NSC)-BETAR(NR)
      V1I=BETAI(NSC)-BETAI(NR)
      V2R=BETAR(NSC)+BETAR(NR)
      V2I=BETAI(NSC)+BETAI(NR)
      IF(NR.EQ.NRDF(NP)) GO TO 50
      VDR=0.0
      VDI=0.0
      RETURN
   50 VDR=(BETAR(NSC)*COSD+BETAI(NSC)*SIND)*2/XPOND
      VDI=(BETAI(NSC)*COSD-BETAR(NSC)*SIND)*2/XPOND
      RETURN
      END
```

VMATRM

Subroutine VMATRM calculates the transformation from the NSRT region to NSTP. It is called by the GAMAML subroutine. This routine computes the real and imaginary parts of $V_{12}'(k,1)$ and $V_{22}'(k,1)$, starting from the $\bar{V}(2,1)$ elements (computed by VINITM) and by using the equations[1]

$$V_{12}'(n,1) = t_{11}(n,n-1)V_{12}'(n-1,1) + t_{12}(n,n-1)V_{22}'(n-1,1) \ , \qquad (1)$$

and

$$V'_{22}(n,1) = t_{21}(n,n-1)V'_{12}(n-1,1) + t_{22}(n,n-1)V'_{22}(n-1,1) \ , \qquad (2)$$

where

$$t_{ij}(n,n-1) = \left[\beta_n + (-1)^{i+j}\beta_{n-1}\right] \exp\left[(-1)^{i}\alpha_{n-1}T_{n-1}\right] \ ,$$

$$\text{V1RO} = \text{Re}[V'_{12}(n-1,1)] \ , \text{ etc., and}$$

$$\text{V1R} = \text{Re}[V'_{12}(n,1)] \ , \text{ etc.,}$$

in the computer notation. (The BETA factors are computed by BETAM.)

For a single defect located in an arbitrary conductor, n, the routine starts from that layer and computes

$$V'_{12}(n,1)\left(\beta_{n+1}\beta_{n+2} \ \cdots \ \beta_k\right) \exp[\beta_n(Z_{n+1} - Z)] \qquad (3)$$

and the corresponding terms for $V'_{12}(n,1)$ up through the kth layer or outer conductor. These matrix elements are computed along with those for conductors only. (Only one defect can be treated with this version of the subroutines because BETAM returns only single values of COSD, SIND, and XPOND.)

```
C          VMATRM SUBROUTINE   (13 JULY 1977)
C
      SUBROUTINE VMATRM(NSTP,NSRT,NRT,NPT,NP,BETAR,BETAI,COSF,SINF,XPON,
     1V1R,V1I,V2R,V2I,VDR,VDI,NRDF,COSD,SIND,XPOND)
C     TRANSFORMATION FROM REGION NSRT TO REGION NSTP
      DIMENSION BETAR(NRT),BETAI(NRT),COSF(NRT),SINF(NRT),XPON(NRT)
      DIMENSION NRDF(NPT)
      NR=NSRT
C     MAIN LOOP
C     DEFINE OLD VALUES AS THE CURRENT VALUE OF VJ2(NP)
   20 V1RO=V1R
      V1IO=V1I
      V2RO=V2R
      V2IO=V2I
      NSC=NR+1
C     DEFINE THE BETA FUNCTIONS AND EXPONENTIAL FUNCTIONS USED IN THE
C     TRANSFORMATION CALCULATIONS BETWEEN NR AND NR+1.
      B1=BETAR(NSC)+BETAR(NR)
      B2=BETAI(NSC)+BETAI(NR)
```

```
      B3=BETAR(NSC)-BETAR(NR)
      B4=BETAI(NSC)-BETAI(NR)
      XP1=COSF(NR)/XPON(NR)
      XP2=SINF(NR)/XPON(NR)
      XP3=COSF(NR)*XPON(NR)
      XP4=SINF(NR)*XPON(NR)
C     THE REAL & IM PARTS OF THE TRANSFORMATION MATRIX,TIJ,ARE NOW
C     CALCULATED.
      T11R=B1*XP1+B2*XP2
      T11I=B2*XP1-B1*XP2
      T12R=B3*XP3-B4*XP4
      T12I=B4*XP3+B3*XP4
      T21R=B3*XP1+B4*XP2
      T21I=B4*XP1-B3*XP2
      T22R=B1*XP3-B2*XP4
      T22I=B2*XP3+B1*XP4
C     TRANSFORM FROM VJ2(NR) TO VI2(NR+1)
      V1R=T11R*V1RO-T11I*V1IO+T12R*V2RO-T12I*V2IO
      V1I=T11I*V1RO+T11R*V1IO+T12I*V2RO+T12R*V2IO
      V2R=T21R*V1RO-T21I*V1IO+T22R*V2RO-T22I*V2IO
      V2I=T21I*V1RO+T21R*V1IO+T22I*V2RO+T22R*V2IO
      IF(NR.LT.NRDF(NP)) GO TO 50
      IF(NR.GT.NRDF(NP)) GO TO 40
C     INITIAL VDR,VDI CALCULATION IN THE DEFECT REGION
      TGF1=COSF(NR)*COSD+SINF(NR)*SIND
      TGF2=COSF(NR)*SIND-SINF(NR)*COSD
      XP1=XPOND/XPON(NR)
      VDR=(V1RO*TGF1-V1IO*TGF2)*XP1+(V2RO*TGF1+V2IO*TGF2)/XP1
      VDI=(V1RO*TGF2+V1IO*TGF1)*XP1-(V2RO*TGF2-V2IO*TGF1)/XP1
C     CALCULATIONS FOR REGIONS ABOVE THE DEFECT.
   40 VDRO=VDR
      VDIO=VDI
      VDR=2*(BETAR(NSC)*VDRO-BETAI(NSC)*VDIO)
      VDI=2*(BETAI(NSC)*VDRO+BETAR(NSC)*VDIO)
C     INCREMENT REGION COUNT &EXIT IF WE HAVE REACHED THE STOP (NSTP) REG.
   50 NR=NR+1
      IF (NR.LT.NSTP) GO TO 20
      RETURN
      END
```

VMATRT

This routine computes the $V_{12}'(k,1)$ and $V_{22}'(k,1)$ [Eqs. (1) and (2), pp. 186–187] by following the formalism outlined below (subroutines VINITM and VMATRM) and described in additional detail elsewhere.[1,3–4] Then the loop is reversed to compute $V_{12}'(k',1')$ and $V_{22}'(k',1')$. In the program V1R/I and V2R/I correspond to V_{12}' and V_{22}' for unprimed variables, and VI1R/I and VI2R/I correspond to V_{12}' and V_{22}' for primed variables. (R and I indicate real and imaginary parts.) Note that this routine is the through-transmission equivalent of VINITM and VMATRM combined, except that no defect calculations are included.

```
C
C          VMATRT SUBROUTINE    (29 NOVEMBER 1977)
C
      SUBROUTINE VMATRT(NSTP,NSRT,NRT,NPT,BETAR,BETAI,COSF,SINF,XPON,
     1V1R,V1I,V2R,V2I,VI1R,VI1I,BTR,BTI)
C     TRANSFORMATION FROM REGION NSRT TO REGION NSTP
      DIMENSION BETAR(NRT),BETAI(NRT),COSF(NRT),SINF(NRT),XPON(NRT)
C
C      CALCULATE THE INITIAL VIJ MATRIX
C     AND TRANSFORM TO REGION NSRT+1.ALSO INITIAL BETA
      NR=NSRT
      NSC=NR+1
      V1R=BETAR(NSC)-BETAR(NR)
      V1I=BETAI(NSC)-BETAI(NR)
      V2R=BETAR(NSC)+BETAR(NR)
      V2I=BETAI(NSC)+BETAI(NR)
      BTR=BETAR(NSC)
      BTI=BETAI(NSC)
      NR=NR+1
C     MAIN LOOP
C     DEFINE OLD VALUES AS THE CURRENT VALUE OF VJ2(NR)
   20 V1RO=V1R
      V1IO=V1I
      V2RO=V2R
      V2IO=V2I
      NSC=NR+1
C     DEFINE THE BETA FUNCTIONS AND EXPONENTIAL FUNCTIONS USED IN THE
C     TRANSFORMATION CALCULATIONS BETWEEN NR AND NR+1.
      B1=BETAR(NSC)+BETAR(NR)
      B2=BETAI(NSC)+BETAI(NR)
      B3=BETAR(NSC)-BETAR(NR)
      B4=BETAI(NSC)-BETAI(NR)
      XP1=COSF(NR)/XPON(NR)
      XP2=SINF(NR)/XPON(NR)
      XP3=COSF(NR)*XPON(NR)
      XP4=SINF(NR)*XPON(NR)
C     THE REAL & IM PARTS OF THE TRANSFORMATION MATRIX,TIJ,ARE NOW
C     CALCULATED.
      T11R=B1*XP1+B2*XP2
      T11I=B2*XP1-B1*XP2
      T12R=B3*XP3-B4*XP4
      T12I=B4*XP3+B3*XP4
      T21R=B3*XP1+B4*XP2
      T21I=B4*XP1-B3*XP2
      T22R=B1*XP3-B2*XP4
      T22I=B2*XP3+B1*XP4
C     TRANSFORM FROM VJ2(NR) TO VI2(NR+1)
      V1R=T11R*V1RO-T11I*V1IO+T12R*V2RO-T12I*V2IO
      V1I=T11I*V1RO+T11R*V1IO+T12I*V2RO+T12R*V2IO
      V2R=T21R*V1RO-T21I*V1IO+T22R*V2RO-T22I*V2IO
      V2I=T21I*V1RO+T21R*V1IO+T22I*V2RO+T22R*V2IO
C
C     CALCULATE THE PRODUCT OF THE BETAS.
C
   40 BTRO=BTR
      BTIO=BTI
      BTR=2*(BETAR(NSC)*BTRO-BETAI(NSC)*BTIO)
      BTI=2*(BETAI(NSC)*BTRO+BETAR(NSC)*BTIO)
C     INCREMENT REGION COUNT &EXIT IF WE HAVE REACHED THE STOP (NSTP) REG.
   50 NR=NR+1
      IF (NR.LT.NSTP) GO TO 20
C     REVERSE TRANSFORM LOOP
C     SET UP INITIAL TRANSFORMATION
      NSC=NR-1
      VI1R=BETAR(NSC)-BETAR(NR)
```

```
      VI1I=BETAI(NSC)-BETAI(NR)
      VI2R=BETAR(NSC)+BETAR(NR)
      VI2I=BETAI(NSC)+BETAI(NR)
      NR=NR-1
C     DEFINE OLD VALUES AS THE CURRENT VALUE OF VJ2(NR)
   70 VI1RO=VI1R
      VI1IO=VI1I
      VI2RO=VI2R
      VI2IO=VI2I
      NSC=NR-1
C     DEFINE THE BETA FUNCTIONS AND EXPONENTIAL FUNCTIONS USED IN THE
C     TRANSFORMATION CALCULATIONS BETWEEN NR AND NR+1.
      B1=BETAR(NSC)+BETAR(NR)
      B2=BETAI(NSC)+BETAI(NR)
      B3=BETAR(NSC)-BETAR(NR)
      B4=BETAI(NSC)-BETAI(NR)
      XP1=COSF(NR)/XPON(NR)
      XP2=SINF(NR)/XPON(NR)
      XP3=COSF(NR)*XPON(NR)
      XP4=SINF(NR)*XPON(NR)
C     THE REAL & IM PARTS OF THE TRANSFORMATION MATRIX,TIJ,ARE NOW
C     CALCULATED.
      T11R=B1*XP1+B2*XP2
      T11I=B2*XP1-B1*XP2
      T12R=B3*XP3-B4*XP4
      T12I=B4*XP3+B3*XP4
      T21R=B3*XP1+B4*XP2
      T21I=B4*XP1-B3*XP2
      T22R=B1*XP3-B2*XP4
      T22I=B2*XP3+B1*XP4
C     TRANSFORM FROM VIJ2(NR) TO VII2(NR+1)
      VI1R=T11R*VI1RO-T11I*VI1IO+T12R*VI2RO-T12I*VI2IO
      VI1I=T11I*VI1RO+T11R*VI1IO+T12I*VI2RO+T12R*VI2IO
      VI2R=T21R*VI1RO-T21I*VI1IO+T22R*VI2RO-T22I*VI2IO
      VI2I=T21I*VI1RO+T21R*VI1IO+T22I*VI2RO+T22R*VI2IO
      NR=NR-1
      IF (NR.GT.1)GO TO 70
      RETURN
      END
```

REFERENCES

1. C. V. Dodd, C. C. Cheng, W. A. Simpson, D. A. Deeds, and J. H. Smith, *The Analysis of Reflection Type Coils for Eddy-Current Testing*, ORNL/TM-4107 (April 1973).

2. Modular Computer Systems, Inc., Ft. Lauderdale, Fla., *Reference Manual, MAX II/III/IV System Processors, Assemblers*, Publication 210-0600500-008H00/B00, 1976.

3. C. C. Cheng, C. V. Dodd, and W. E. Deeds, "General Analysis of Probe Coils Near Stratified Conductors," *Int. J. Nondestr. Test.* 3(2): 109–30 (1971).

4. C. V. Dodd and W. E. Deeds, *Calculations of Magnetic Fields from Time-Varying Currents in the Presence of Conductors*, ORNL/TM-4958 (July 1975).

APPENDIX F

Compiler Features

COMPILER FEATURES

The MODCOMP FORTRAN IV Compiler[1] complies with the standard approved by the American Standards Institute (X3.9-1966)[2] with the following modifications and extensions:

General Features

- Identifiers may be of any length but only the first six characters are used by the processor.
- The receiving variable identifier and the equals character, "=", must appear on the first line of the statement.
- The first comma of a DO statement must appear on the first line of the statement.
- The complete statement name of statements other than DO, assignment, and statement function definitions must appear on the first line of the statement.
- Expressions may contain variables of mixed mode.
- Subscripts may be made up of any expression.
- A main program may be named.
- A variable number of arguments may be passed to a subroutine.
- The character string following the STOP and PAUSE statements may consist of any character in the FORTRAN Character Set.
- Hexadecimal constants are allowed in DATA statements.
- Inline coding of MODCOMP instructions is allowed.
- The number of continuation cards allowed is limited by the compiler work space.
- IMPLICIT statement.
- Array initialization in a DATA statement.
- Omission of statement number(s) in arithmetic IF.
- Variables may occupy various fractional multiples of a "storage unit" (2 MODCOMP words).
- Array identifier allowed in EQUIVALENCE statement.

Input and Output Features

- END and ERR options in READ and WRITE.
- Unformatted WRITE with no list.
- A or D format descriptors with Complex (either part).
- d greater than w for D, E, F, or G format descriptors on input.
- D format descriptor with Real.
- E, F, or G format descriptors with Double Precision.
- Comma after comma, slash, or P in format string.
- Slash after P in format string.
- Slash after any comma (rather than just after the final comma) in a format string.
- Omission of comma after H and X field descriptors, after groups, and after last descriptor in format string.
- Plus sign before scale factor constant in format string.
- Minus sign before constant for X field descriptor.
- G field descriptor with Logical or Integer.
- Prohibition on the use of the scale factor with D, E, or exponential form G output conversion if it will cause the generation of an exponent field with a magnitude greater than 89.
- T or F not required to be first non-blank character in Logical input fields.
- Repeat count with slash in format string.
- Z field descriptor for transmitting machine words.
- H field descriptors allowed in format strings stored in arrays if no extra spaces are introduced.
- Computation overlapped with WRITE.
- DEFINE FILE.
- FIND.
- Direct access READ and WRITE.
- Field beginning designator T allowed in format strings.
- Character strings delimited by apostrophes or quotation marks allowed in format strings.
- ENCODE and DECODE.
- List-directed READ and WRITE.
- BUFFER IN and BUFFER OUT.

REFERENCES

1. Modular Computer Systems, Inc., Ft. Lauderdale, Fla., *MAX IV ModComp IV Fortran IV System Processors Reference Manual,* Rev. 1, Appendix A, Publication 210-610500-010, 1976.
2. *ANSI X3.9-1966,* American National Standards Institute, New York, 1966.

APPENDIX G

Program Variables and Definitions

COMPOSITE PROGRAM VARIABLE LIST

All variables, arrays, and function names are listed below, including functions assigned by the Fortran compiler (SIN, COS, etc.) and used in the four main programs and their subroutine clusters.

When R or RE (real parts) and I or IM (imaginary parts) are at the end of a variable name, the two forms are listed together and alphabetized according to the form preceding these characters. (Example: DVRR/DVRI precedes DVRF.)

The forms MG or MAG (magnitude) and PH or PHA (phase) are grouped together and alphabetized according to the characters preceding these groups. (Example: SSMAG/SSPHA precedes SSCMG/SSCPH.)

In a few cases the real-imaginary or magnitude-phase groups fall in the middle of a name. Alphabetizing is applied to the characters preceding these groups.

Names with numerals follow in numerical order after equivalent combinations of letters. (Example: NPT1 follows NPTT.)

A
ABS
AIR1
AIR2
ALOG
ANSWER
ARG

B
BETAR/BETAI
BLANKS
BTR/BTI
BTRO/BTIO
B1—B4

CABLEL
CALMAG/CALPHA
CAPDR
CAPPU
CCABLE
CKTPAR
CKTVAL
CNAM
CNVT
COE
COEF
COIL
COLFAC
CONSTA
CONVRT
COS
COSD
COSF
C1—C4

DCMGAR/DCPHAR
DCMGCB/DCPHCB
DENRE/DENIM
DFDEPT
DFDIAM
DFDP
DFLDC
DFR
DFRAD
DFSIZ
DFSIZE
DFT
DFV
DFVOL
DIFF
DINORM
DRIFT
DRIVRE/DRIVIM
DRVDR/DRVDI
DVR
DVRR/DVRI
DVRF
DVRFAC
D1,D2
D21
D43

E
ELAPST
ENCODE
EXP
EX1—EX7

F
FLOAT
FREQ
FUNTYP
F1

GAIN
GAMR/GAMI
GAMAR/GAMAI
GAMADR/GAMADI
GAMAPR/GAMAPI
GAP

HDR
HPU

ICOEF
IDAY
IDEV
IHEX
INSTNO
INTG
INUM

IOFSET
IR
IRDPR
IRDPRM
IRDPR1
IREC
IROLD
ISAMPL
ISTART
IT
ITIM
ITIMS
I1

J
JJ
JOFSET
JPOL
JRDPR
JROW

K
KK

L
LAST
LI
LIFT
LITEK
LL
LOT
LOTEK
LOU
LO2
LPT
L1—L6

M
MM
MAXCH
MDF
MM1
MNDEF
MNDF
MNT
MP1
MROW
MSET
MSET1
M1

N
NA
NARRAY
NB
NC
NCABLE
NCC
NCF
NCHS
NCOED
NCOED1
NCOIL
NCOL
NCOUNT
NCP
NCTERM
ND
NDEF
NDEG
NDFLOC
NDFSIZ
NDIJ
NDLK
NDPS
NDPS1
NDRT
NEXT
NF
NFT
NL
NLABL
NLT
NMGCAL/NPHCAL
NMGCH/NPHCH
NN
NODF
NOPT
NORM
NP
NPOL
NPP
NPRINT
NPRO
NPROBE
NPROPM
NPROPT
NPROPT1
NPT
NPTT
NPT1

NR
NRDF
NRDG
NREG
NRES
NRET
NROW
NRREP
NRT
NRT1
NRT2
NS
NSC
NSRT
NSER
NSTART
NSTP
NT
NTH
NTHI
NTHL
NTIMES
NTYP
NUNIT
N1

OLDMGC/OLDPHC

PERM
PHADET
PHASE
PHASEM
PHDFT
PIC
PICR/PICI
PICF
PICFAC
PICKRE/PICKIM
PICKAM
PICKDR/PICKDI
PIO4
POLARY
POWAMP
POWOSC
PP
PRO
PRONAM
PROP
PROPT
PROPTY

QT
Q1—Q4
Q1S—Q4S

R
RAD
RBAR
RCON
RDCDR
RDCPU
RDG,RDGTYP,RDY
READNG
RES
RHO
RINV
RJ1
RL1
RL2
ROLD
RRBAR
RSOS
R0
R1—R5
R9

SDIFF
SDRIFT
SDVCMG/SDVCPH
SDVMAG/SDVPHA
SIN
SIND
SINF
SQRT
SS
SSMAG/SSPHA
SSCMG/SSCPH
SSDIFF
SSDRIF
STARTT
STOPT
SUM
SUMMAG/SUMPHA
SUMCMG/SUMCPH
SUMEXP
S1
S2
S3—S7

T
TITLE
TH
THICK
THMAX
THTT
TMAG
TMAGM
TMDFT
TMUR/TMUI
TMUTRE/TMUTIM
TNDR
TNMRE/TNMIM
TNPU
TWOPI
T1—T2
T11R/T11I
T12R/T12I
T21R/T21I
T22R/T22I

U
UN
UNIT
UNITS

VI1R/VI1I
VI1RO/VI1IO
VI2RO/VI2IO
VOLFAC
VOLTS
VO
V1R/V1I
V2R/V2I

WD
WLCDR
WLCPU
WP
WUSR

X
XJR1—XJR4
XLOC
XNL
XPON
XPOND
XP1—XP4
XRAD
XRHO
XTH
XVOL
XX
XXXX
XYZ
XO—X14

Y

Z
ZDRR/ZDRI
ZLDR
ZLPU
ZMUR/ZMUI
ZMUTF
ZMUTFC
ZPR
ZPUR/ZPUI
Z1Z2RE/Z1Z2IM
Z5SQRE/Z5SQIM

Definitions of Principle Variables

AIR1, AIR2 — Inductance integral of driver and pickup coils, respectively, in air (away from conductors).

BETAR, BETAI(NR) — Real, imaginary parts of beta (β) for region (layer) NR.

CABLEL — Probe cable length.

CALMAG, CALPHA(I,J,K) — Arrays holding latest values of calibration voltages, typically following standardization readings. (I = frequency, J = ordinal for phase values, K = ordinal for magnitude values.)

CAPDR, CAPPU — Shunt capacitances across driver and pickup coils, respectively (C_6,C_7).

CCABLE — Driver probe cable capacitance.

CKTVAL(I) — Array storing values of circuit components.

COE(NP,IR), COEF(IR) — Arrays storing coefficients computed in the least squares fits.

CONVRT — Array storing length conversion factors; 1 for inches, 25.4 for mm.

DCMGAR, DCPHAR(NF) — Arrays storing magnitude and phase voltages recorded with the probe in air.

DCMGCB(NF) — Array storing magnitude voltages recorded when the gain adjustment is made.

DFLOC(I) — The z-direction location of a defect within its layer.

DFSIZ/DFSIZE(I) — The normalized volume of a spherical defect.

DRIVRE, DRIVIM(NL,NP,NF) — Real, imaginary parts of the defect component of the driver coil impedance integral.

ELAPST — Difference between starting clock time and stopping clock time for the calculation of impedance integrals expressed in seconds.

FREQ(NF) — The operating frequency of the driver coil voltage.

GAIN(NF) — The gain of the pickup amplifier at frequency NF.

GAMAR, GAMAI(NP) — Real, imaginary parts of the gamma factor for mutual impedance in the transmission case (γ_m).

GAMADR, GAMADI(NP) — Real, imaginary parts of the driver-side gamma factor (γ_D).

GAMAPR, GAMAPI(NP) — Real, imaginary parts of the pickup-side gamma factor (γ_P) (transmission case).

GAP — Constant coil-conductor spacing parameter added on the pickup side in the transmission case.

INSTNO — Instrument serial number.

IOFSET, JOFSET(I) — Control parameter indicating that a leading constant should be included in the least squares fitting functions.

IRDPR, JRDPR(I) — The number of terms in a selected least squares fitting function.

IRDPRM — The maximum allowed size of IRDPR, usually 15.

ITIM(I) — Array containing hours, minutes, and seconds of current clock time (24-h scale).

LI, LITEK — Numerical designator for logical (operator) input unit (e.g., keyboard device).

LOT, LOTEK, LOU, LO2 — Numerical designator for nonpermanent readout or display device (e.g., CRT terminal).

LPT — Numerical designator for hard-copy output device (e.g., line printer); LOU occasionally used.

L2–L6 — Normalized coil dimension parameters (L_2–L_6).

MSET — Maximum row dimension of property arrays (PRO, PROP); includes conductor and defect properties and lift-off values.

NARRAY(I) — Array storing the analog-digital converter counts from instrument readings by the MODACS; I = channel number.

NCABLE — Probe cable designation number.

NCHS — The number of A-D converter channels available in the MODACS (12 for the system described here).

NDFLOC — The maximum number of locations (step values) for a defect.

NDPS — The number of defects per specimen.

NFT — The maximum number of discrete operating frequencies in a program run (NF = running index).

NLT — The maximum number of different lift-off values in a program run (NL = running index).

NOPT — Control parameter selecting coil-spacing mode for MULTRU.

NORM — Control parameter selecting normalization method for MULTRU.

NPOL(I,NF), JPOL(I,NF,NPROPT) — Arrays storing the identification of least squares fitting function terms.

NPROPM — The maximum number of properties that can be fitted by a program (NPROPT = index selecting specific property).

NPT — The maximum number of conductor property sets used in a program (NP = running index).

NPTT — The maximum total of conductor and combined (conductor-defect) property sets used in a program.

NREG — The number of a selected region (conductor or air layer).

NRT — The maximum number of regions used in a program run; includes conductors and air layers (NR = running index).

PHASE(NL,NP,NF) — The computed phase of the instrument output voltage (V_{out}).

PICKRE, PICKIM(NP) — Real, imaginary components of pickup coil impedance integral due to conductors.

PICKDR, PICKDI(NP) — Real, imaginary components of defect component of pickup coil impedance integral.

POLARY(I,J) — Array storing alphanumeric representation of least squares fitting function terms.

PRO(M), PROP(M), PROP(M,NPROP) — Arrays storing values of a selected property for least squares fitting.

RAD — Constant; 180/π (deg/rad).

RBAR — Mean driver coil radius (R_5,$\bar{r}$).

RDCDR, RDCPU — dc resistance of driver and pickup coils, respectively (R_6,R_7).

READNG(M,IR) — Array storing the computed functional values of real or simulated instrument readings for least squares function fitting.

RES(NR,NS), RHO(NR,NP) — Arrays storing resistivity values. (Note: NS is a substitute index for NP.)

R0 — Driver coil series (source) resistance (R_0).

R1—R4 — Normalized inner and outer radii of driver and pickup coils (R_1—R_4).

R9 — Pickup coil shunt resistance (R_9) in ohms.

SDIFF — The rms difference between assumed values of selected properties and values computed by the least squares fitting functions.

SDRIFT — The rms error in a property value generated by sequentially introducing 0.01% magnitude and 0.01° phase errors into each of the readings used by the fitting function.

S1 — Step size for changing the value of the integration variable, α.

S2 — Upper limit for integration; advanced with changes in step size.

TH(NR,NP), THICK(NR,NS) — Arrays for storing conductor thickness values.

THMAX — The maximum total conductor stack thickness.

TMAG(NL,NP,NR) — The magnitude of instrument output voltage (V_{out}).

TMUTRE, TMUTIM(NL,NP,NF) — Real, imaginary components of mutual (driver-pickup) coil imedpance.

TNDR, TNPU — Numbers of turns of the driver and pickup coils, respectively (N_3 and N_4).

TWOPI — Constant, 2π.

U(NR,NP), PERM(NR,NS) — Arrays storing permeability values.

VOLTS(I) — Array storing voltages converted from MODACS readings.

V0 — Input voltage applied to driver coil (constant for all frequencies).

WUSR(NR,NP,NR) — Product $\omega\mu\sigma\bar{r}^2$.

X — Integration variable, α(XX = α^2, etc.).

XJR1—XJR4 — Values of Bessel integrals, $\int_0^{\alpha R_1} xJ_1(x)\,dx$, etc.

ZLDR, ZLPU — Air values of driver and pickup coil impedances, respectively.

ZDRR, ZDRI — Real, imaginary components of the driver coil impedance (Z_D).

ZMUR, ZMUI — Real, imaginary components of the mutual coil impedance $\left(M^{(T)},\ M^{(R)}\right)$.

ZPUR, ZPUI — Real, imaginary components of the pickup coil impedance $\left(Z_P^{(T)},\ Z_P^{(R)}\right)$.

APPENDIX H

FILE MANAGEMENT PROGRAMS

Three programs have been used to build and maintain the coil data (RAD) file (28). The newest, COILDATA, includes all required editing functions: reading and printing, changing, and adding to the file. It has the capacity to select either of two coil files now available in our minicomputer system (28,29). (The second file stores data describing coils used inside or encircling cylindrical conductors; its use is outside the scope of this report.)

By using the older programs one can obtain a printout of the file (28) from the REDFIL program. The CHGDAT program allows the user to change a line by writing over the existing one or to add a line by writing a new record.

None of these programs requires editing to enter initializing data. Table H1 lists interactive directions for running COILDATA. The other programs are much simpler; thus, no instructions were considered necessary.

A sample printout of our current reflection coil data file is shown on p. 212. Subroutines for transferring coil data from file 28 to executing programs are listed on pp. 160 and 180.

Table H1. Interactive Program Queries and Operator Responses Occurring During Execution of COILDATA

Line	QUERY or Response
10	FOR REFLECTION COILS — REF, FOR CIRCULAR COILS—CIR?
	Enter REF to select file 28, CIR to select file 29.
16	1. LIST 2. CHANGE 3. ADD 4. STOP
	Enter integer 1—4.
	The program branches according to the options listed below.
	<u>1. List (file contents)</u>
28	No query or response; listing similar to sample on p. 212 is printed out on the hard-copy device. Execution returns to line 16.

Table H1. (Continued)

Line	QUERY or Response
	2. Change (coil data)
49	CHANGE COIL IN LINE NO: (TYPE LINE NO. IN DATA FILE)
	Enter integer number of line to be changed. (Line numbers listed in extreme left column of file printout.)
52	NEW COIL NAME (UP TO 6 CHARACTERS)
	Enter six-character alphanumeric coil name. (Fill unused character positions with blanks.)
	For REF coils program branches to line 58; for CIR coils it branches to line 67.
58	TYPE: RBAR, R1, R2, L3, R3, R4, L4, L5, L6, RDCDR, RDCPU, TNDR, TNPU
	Enter in free format R_5 or $(\overline{r})$, R_1, R_2, L_3, R_3, R_4, L_4, L_5, L_6, R_6, R_7, N_3, N_4.
	(R_5 in inches, R_6 and R_7 in ohms, N_3 and N_4 dimensionless, remainder normalized with respect to R_5.)
	Execution returns to line 16.
67	TYPE: RBAR, R1, R2, XL, ZL, RLIM, RDCDR, RDCPU, TNDR, TNPU, ZLDR, ZLPU, X
	Enter in free format R_5 or $(\overline{r})$, R_1, R_2, L_3, L_4, R_{lim}, R_6, R_7, N_3, N_4, (no entry).
	Execution returns to line 16.
	3. Add (coil data)
84	NEW COIL NAME (UP TO 6 CHARACTERS)
85	TYPE END TO STOP
	Enter six-character alphanumeric coil name. (Fill unused character positions with blanks.)
	Enter END if all coil data have been entered.

Table H1. (Continued)

Line	QUERY or Response
	If a new coil name is entered the queries and responses for line 58 or 67 will be repeated.
	If END is entered execution completes the END line in the file as shown on the sample printout and returns to line 16.
	<u>4. Stop</u>
110	No response. Execution branches to STOP instruction and halts.

```
1          PROGRAM COILDATA
2    C
3    C     COILDATA     VERSION 9/12/78
4    C     PROGRAM TO LIST,CHANGE OR ADD COIL DATA ON FILES 28 OR 29.
5    C
6          REAL*6 COIL,COILN,FILTYP
7          DEFINE FILE 28(80,32,U,NCOIL)
8          DEFINE FILE 29(80,32,U,NCOIL)
9          DATA NFILE/0/,LI/1/,LOU/5/,LPT/6/
10       5 WRITE (LOU,10)
11      10 FORMAT (' FOR REFLECTION COILS-REF,FOR CIRCULAR COILS-CIR ? ')
12         READ (LI,30) FILTYP
13         IF(FILTYP.EQ.'REF   ') NFILE=28
14         IF(FILTYP.EQ.'CIR   ') NFILE=29
15         IF(NFILE.EQ.0)GO TO 5
16      16 WRITE(LOU,17)
17      17 FORMAT(' 1.LIST 2.CHANGE 3.ADD 4.STOP  ?  ')
18      30 FORMAT(A6)
19      40 FORMAT(1X,I2,1X,A6,13(F5.1))
20      50 FORMAT(1X,I2,1X,A6,6F8.4,2F10.4,2F8.1,2E11.3,F8.1)
21      60 FORMAT(1X,I2,1X,A6,9F8.4,F10.4,F11.4,2F8.1)
22         READ(LI,*)NEXT
23         GO TO(100,200,300,400),NEXT
24     100 NCOIL = 1
25   C
26   C     LIST SECTION OF PROGRAM
27   C
28         IF(NFILE.EQ.28)WRITE(LPT,130)
29         IF(NFILE.EQ.29)WRITE(LPT,140)
30     110 IF(NFILE.EQ.28)READ(NFILE'NCOIL)COIL,RBAR,R1,R2,XL3,R3,R4,XL4,XL5
31        *,TNDR,TNPU,XL6,RDCDR,RDCPU
32         IF(NFILE.EQ.29)READ(NFILE'NCOIL)COIL,RBAR,R1,R2,XL,ZL,RLIM,RDCDR
33        *,RDCPU,TNDR,TNPU,ZLDR,ZLPU,X
34         IF(COIL.EQ.'END   ')GO TO 16
35         NC=NCOIL-1
36         IF(NFILE.EQ.28)WRITE(LPT,60)NC,COIL,RBAR,R1,R2,XL3,R3,R4,XL4,XL5
37        *,XL6,RDCDR,RDCPU,TNDR,TNPU
38         IF(NFILE.EQ.29)WRITE(LPT,50)NC,COIL,RBAR,R1,R2,XL,ZL,RLIM,
39        *RDCDR,RDCPU,TNDR,TNPU,ZLDR,ZLPU,X
40     130 FORMAT('  N COIL',5X,'RBAR',5X,'R1',6X,'R2',5X,'XL3',6X,'R3',6X,
41        *'R4',5X,'XL4',5X,'XL5',5X,'XL6',7X,'RDCDR',5X,'RDCPU',5X,
42        *'TNDR',4X,'TNPU',/)
43     140  FORMAT('  N COIL',5X,'RBAR',5X,'R1',6X,'R2',5X,'XL',6X,'ZL',6X,
44        *'RLIM',5X,'RDCDR',5X,'RDCPU',5X,'TNDR',4X,'TNPU',4X,'ZLDR',7X,
45        *'ZLPU',9X,'X',/)
46          GO TO 110
47   C
```

```
C       SECTION TO CHANGE THE COIL DATA
  200 WRITE(LOU,210)
  210 FORMAT(' CHANGE COIL IN LINE NO:(TYPE LINE NO IN DATA FILE) ')
  220 READ(LI,*)NCOIL
      WRITE(LOU,225)
  225 FORMAT(' NEW COIL NAME (UP TO 6 CHARACTERS) ')
      READ(LI,30)COILN
      IF(NFILE.EQ.28)GO TO 227
      IF(NFILE.EQ.29)GO TO 250
      GO TO 5
  227 WRITE(LOU,230)
  230 FORMAT(' TYPE:RBAR,R1,R2,L3,R3,R4,L4,L5,L6,RDCDR,RDCPU'
     *,',TNDR,TNPU',/)
      READ(LI,*)RBAR,R1,R2,XL3,R3,R4,XL4,XL5,XL6,RDCDR,RDCPU,TNDR,TNPU
      WRITE(LOU,40)NCOIL,COILN,RBAR,R1,R2,XL3,R3,R4,XL4,XL5,XL6
     *,RDCDR,RDCPU,TNDR,TNPU
      WRITE(28'NCOIL)COILN,RBAR,R1,R2,XL3,R3,R4,XL4,XL5,TNDR,TNPU
     *,XL6,RDCDR,RDCPU
      GO TO 16
  250 WRITE (LOU,260)
  260 FORMAT(' TYPE:RBAR,R1,R2,XL,ZL,RLIM,RDCDR,RDCPU,TNDR,TNPU,',
     *'ZLDR,ZLPU,X',/)
      READ(LI,*) RBAR,R1,R2,XL,ZL,RLIM,RDCDR,RDCPU,TNDR,
     *TNPU,ZLDR,ZLPU,X
      WRITE(LOU,40)NCOIL,COILN,RBAR,R1,R2,XL,ZL,RLIM,
     *RDCDR,RDCPU,TNDR,TNPU,ZLDR,ZLPU,X
      WRITE(29'NCOIL)COILN,RBAR,R1,R2,XL,ZL,RLIM,
     *RDCDR,RDCPU,TNDR,TNPU,ZLDR,ZLPU,X
      GO TO 16
C
C       SECTION TO ADD TO COIL DATA FILE
C
  300 NCOIL =1
  310 READ(NFILE'NCOIL)COILN
      IF(COILN.NE.'END   ')GO TO 310
      NCOIL=NCOIL-1
  315 WRITE(LOU,225)
      WRITE(LOU,317)
  317 FORMAT('  TYPE END TO STOP ')
      READ(LI,30)COILN
      IF(COILN.EQ.'END   ') GO TO 390
      IF(NFILE.EQ.28)GO TO 320
      IF(NFILE.EQ.29)GO TO 350
      GO TO 5
  320 WRITE(LOU,230)
      READ(LI,*)RBAR,R1,R2,XL3,R3,R4,XL4,XL5,XL6,RDCDR,RDCPU,TNDR,TNPU
      WRITE(28'NCOIL)COILN,RBAR,R1,R2,XL3,R3,R4,XL4,XL5,TNDR,TNPU
     *,XL6,RDCDR,RDCPU
      WRITE(LOU,40)NCOIL,COILN,RBAR,R1,R2,XL3,R3,R4,XL4,XL5,XL6
     *,RDCDR,RDCPU,TNDR,TNPU
      GO TO 315
  350 WRITE (LOU,260)
      READ(LI,*) RBAR,R1,R2,XL,ZL,RLIM,RDCDR,RDCPU,TNDR,TNPU,
     *ZLDR,ZLPU,X
      WRITE(LOU,40)NCOIL,COILN,RBAR,R1,R2,XL,ZL,RLIM,RDCDR,RDCPU,
     *TNDR,TNPU,ZLDR,ZLPU,X
      WRITE(29'NCOIL)COILN,RBAR,R1,R2,XL,ZL,RLIM,RDCDR,RDCPU,TNDR,TNPU,
     *ZLDR,ZLPU,X
      GO TO 315
  390 ZZ=0.
      WRITE (NFILE'NCOIL)COILN,ZZ,ZZ,ZZ,ZZ,ZZ,ZZ,ZZ,ZZ,ZZ,ZZ,ZZ,ZZ,ZZ
      GO TO 16
  400 STOP JOB
      END
TOTAL RECORDS WRITTEN =    112
EXIT
```

```
      PROGRAM REDFIL
C *** PROGRAM TO READ AND PRINT COIL DATA FROM FILE 28 ***
      REAL*6 COIL
      DEFINE FILE 28(80,32,U,NCOL)
      DATA LI,LOU/1,11/
      FIND (28'NCOL)
      WRITE (LOU,10)
   10 FORMAT ('  N COIL',5X,'RBAR',5X,'R1',6X,'R2',5X,'XL3',6X,'R3',6X,
     *'R4',5X,'XL4',5X,'XL5',5X,'XL6',8X,'R6',8X,'R7',7X,
     *'TNDR    TNPU',/)
      NCOL=1
   30 READ (28'NCOL) COIL,RBAR,R1,R2,XL3,R3,R4,XL4,XL5,
     *TNDR,TNPU,XL6,R6,R7
      NC=NCOL-1
      WRITE (LOU,40) NC,COIL,RBAR,R1,R2,XL3,R3,R4,XL4,XL5,
     *XL6,R6,R7,TNDR,TNPU
   40 FORMAT (1H,I2,1X,A6,9F8.4,F10.4,F11.4,2F8.1)
      IF (COIL.EQ.'END   ') GO TO 80
      GO TO 30
   80 STOP JOB
      END
TOTAL RECORDS WRITTEN =    22
EXIT
```

```
      PROGRAM CHGDAT
C *** PROGRAM TO WRITE ONE SET OF COIL DATA ON FILE 28 ***
      REAL*6 COIL
      DEFINE FILE 28(80,32,U,NCOL)
      DATA LI,LOU/1,5/
      WRITE (LOU,10)
   10 FORMAT (' WRITE LINE NUMBER OF DATA:')
      READ (LI,*) NCOL
      WRITE(LOU,20)
   20 FORMAT(' WRITE  COIL NAME(UP TO 6 LETTERS)')
      READ (LI,30) COIL
   30 FORMAT(A6)
      IF(COIL.EQ.'END   ') GO TO 50
      WRITE (LOU,40)
   40 FORMAT(' TYPE:RBAR,R1,R2,XL3,R3,R4,XL4,XL5,XL6,RDCDR,RDCPU,',
     *'TNDR,TNPU')
      READ(LI,*) RBAR,R1,R2,XL3,R3,R4,XL4,XL5,XL6,RDCDR,RDCPU,TNDR,TNPU
      WRITE(28'NCOL)COIL,RBAR,R1,R2,XL3,R3,R4,XL4,XL5,TNDR,TNPU,
     *XL6,RDCDR,RDCPU
   50 STOP JOB
      END
TOTAL RECORDS WRITTEN =    22
EXIT
```

N	COIL	RBAR	R1	R2	XL3	R3	R4	XL4	XL5	XL6	RDCDR	RDCPU	TNDR	TNPU
1	60-F	0.0600	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.1000	3.1470	31.6700	60.0	90.0
2	20-A	0.0200	0.7500	1.2500	1.5000	0.3500	0.7000	0.5000	0.0000	0.3000	3.7600	13.8000	85.0	56.0
3	20-B	0.0200	0.7500	1.2500	0.9000	0.3500	0.7000	0.3000	0.0000	0.3000	6.9737	128.4000	84.0	160.0
4	20-C	0.0200	0.7500	1.2500	0.9000	0.3500	0.7000	0.3000	0.0000	0.3000	2.3424	24.3053	53.0	65.0
5	20-D	0.0200	0.7500	1.2500	0.9000	0.3600	0.6960	0.3000	0.0000	0.3000	2.9200	23.1000	50.0	72.0
6	20-F	0.0200	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.3000	0.2100	2.1000	20.0	25.0
7	20-G	0.0200	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.2500	3.9351	24.6792	56.0	66.0
8	R20G	0.0200	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.2500	3.7000	26.6000	50.0	54.0
9	20-N	0.0200	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.1500	0.0660	0.3720	6.0	8.0
10	30-B	0.0300	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.1800	34.8600	74.2300	208.0	144.0
11	30-E	0.0300	0.8000	1.2000	0.5000	0.3500	0.7000	0.1670	0.0000	0.2000	0.8000	2.5780	24.0	21.0
12	40-A	0.0400	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.1000	5.4700	205.2000	90.0	264.0
13	40-B	0.0400	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.1000	15.7400	79.2100	112.0	240.0
14	40-C	0.0400	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.2000	84.8500	659.5000	378.0	510.0
15	40-D	0.0400	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.1000	9.7440	114.5000	99.0	154.0
16	40-E	0.0400	0.7500	1.2500	0.4000	0.3500	0.7000	0.1250	0.0000	0.1500	1.3720	11.2150	35.0	46.0
17	60-A	0.0600	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.1200	59.0600	1067.3999	249.0	450.0
18	60-B	0.0600	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.0500	52.7700	865.6001	252.0	427.5
19	60-D	0.0600	0.7500	1.2500	0.4000	0.3530	0.7050	0.1000	0.0000	0.0400	0.1970	1.0490	12.0	13.5
20	60-E	0.0600	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.0400	23.6050	190.7000	170.0	223.0
21	60-F	0.0600	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.1000	3.1470	31.6700	60.0	90.0
22	60-G	0.0600	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.1000	3.2565	30.3660	42.0	50.0
23	R60H	0.0600	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.0400	8.6700	77.2000	104.0	138.0
24	83-A	0.0832	0.7508	1.2490	0.3604	0.3604	0.6967	0.0961	0.0000	0.1000	27.2000	728.8999	234.0	528.0
25	83-B	0.0832	0.7500	1.2500	0.3600	0.3600	0.6960	0.0960	0.0000	0.1000	79.3600	530.5000	360.0	410.0
26	83-C	0.0830	0.7530	1.2530	0.3614	0.3614	0.6990	0.0964	0.0000	0.0360	10.9000	85.9000	150.0	175.0
27	100A	0.1000	0.7900	1.2100	0.4200	0.4700	0.7000	0.1000	0.0000	0.0800	0.0310	0.4050	8.8	10.0
28	120A	0.1200	0.7500	1.2500	0.3000	0.3500	0.7000	0.0667	0.0000	0.0500	13.0600	182.3000	185.0	245.0
29	140A	0.1400	0.7500	1.2500	0.3930	0.3140	0.7000	0.0643	0.0000	0.0643	0.0950	1.1800	20.0	24.0
30	150A	0.1500	0.7500	1.2500	0.4000	0.3600	0.7000	0.1000	0.0000	0.0750	182.2000	3335.0000	900.0	1400.0
31	150B	0.1500	0.7500	1.2500	0.3600	0.3600	0.7000	0.1000	0.0000	0.0750	67.3000	531.0000	513.0	590.0
32	200A	0.2000	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.0450	100.1600	7023.1406	860.0	3275.0
33	200B	0.2000	0.7500	1.2500	0.4000	0.3500	0.7000	0.1250	0.0000	0.0450	174.6000	1656.0000	1008.0	1375.0
34	300A	0.3000	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.1000	57.5550	3443.0098	810.0	2925.0
35	300B	0.3000	0.7500	1.2500	0.4000	0.3500	0.7000	0.0670	0.0000	0.2500	35.0950	472.5500	616.0	660.0
36	400A	0.4000	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.0400	3.2378	99.0500	238.0	858.0
37	400B	0.4000	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.0125	3.4130	190.3800	238.0	825.0
38	400C	0.4000	0.7500	1.2500	0.4000	0.3500	0.7000	0.0650	0.0000	0.0400	13.0070	161.2700	414.0	450.0
39	183B	0.1830	0.9618	1.0382	0.3825	0.7650	0.9180	0.1530	0.0000	0.0546	18.9840	25.7200	116.0	93.0

ORNL/TM-6858
Distribution
Category UC-79b, -h, -k

INTERNAL DISTRIBUTION

1–2.	Central Research Library	47.	C. E. Pugh
3.	Document Reference Section	48.	G. W. Scott
4–8.	Laboratory Records Department	49.	G. M. Slaughter
9.	Laboratory Records, ORNL RC	50.	J. H. Smith
10.	ORNL Patent Section	51.	R. W. Balluffi (Consultant)
11.	W. E. Deeds	52.	A. L. Bement, Jr. (Consultant)
12.	J. R. DiStefano	53.	W. R. Hibbard, Jr. (Consultant)
13–42.	C. V. Dodd	54.	E. H. Kottcamp, Jr. (Consultant)
43–45.	M. R. Hill	55.	M. J. Mayfield (Consultant)
46.	R. W. McClung	56.	J. T. Stringer (Consultant)

EXTERNAL DISTRIBUTION

57–58. DOE DIVISION OF REACTOR RESEARCH AND TECHNOLOGY, Washington, DC 20545
Director

59. DOE OAK RIDGE OPERATIONS OFFICE, P.O. Box E, Oak Ridge, TN 37830
Assistant Manager, Energy Research and Development

60–269. DOE TECHNICAL INFORMATION CENTER, Office of Information Services,
P. O. Box 62, Oak Ridge, TN 37830

For distribution as shown in TID-4500 Distribution Category,
UC-79b (Fuels and Materials Engineering Development);
UC-79h (Structural Materials Design Engineering);
UC-79k (Components).

☆ U. S. GOVERNMENT PRINTING OFFICE: 1980-640-245-261