NUREG/CR-0967
(ORNL/NUREG/TM-335)
cy. 3

ornl

OAK RIDGE NATIONAL LABORATORY

UNION CARBIDE

# Multiple Property Variations in Coaxial Cylindrical Conductors Determined with Multiple-Frequency Eddy Currents

W. E. Deeds
C. V. Dodd

ORNL TECHNICAL INFORMATION DIVISION
Y-12 TECHNICAL LIBRARY
*Document Reference Section*
**LOAN COPY ONLY**

**Do NOT transfer this document to any other person. If you want others to see it, attach their names, return the document, and the Library will arrange the loan as requested.**
UCN-1624
(3 5-60)

Prepared for the U.S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Under Interagency Agreements DOE 40-551-75

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

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161

[illegible]

NUREG/CR-0967
ORNL/NUREG/TM-335
Distribution
Category R5

Contract No. W-7405-eng-26

METALS AND CERAMICS DIVISION

# MULTIPLE PROPERTY VARIATIONS IN COAXIAL CYLINDRICAL CONDUCTORS DETERMINED WITH MULTIPLE-FREQUENCY EDDY CURRENTS

W. E. Deeds C. V. Dodd

Manuscript Completed - October 11, 1979
Date Published - November 1979

**NOTICE** This document contains information of a preliminary nature. It is subject to revision or correction and therefore does not represent a final report.

Prepared for the
U.S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Washington, DC 20555
Under Interagency Agreement DOE 40-551-75
NRC FIN No. B0417

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

## CONTENTS


Abstract ..... 1
Introduction ..... 1
Measurement of Multiple Properties ..... 3
Experiment Design ..... 6
Instrumentation ..... 6
Acquisition of Data from Standards with TUBRDG ..... 7
Fitting of Properties with Data from Standards by Using TUBFIT ..... 7
Finding Properties of Unknown Samples with TUBFIT ..... 13
Acknowledgment ..... 13
References ..... 13
Appendices
- A. The TUBRDG Program ..... 15
- B. The TUBFIT Program ..... 29
- C. The Subroutines
  - CIRCOL ..... 43
  - READAT ..... 43
  - CORRDG ..... 44
  - CALIB ..... 45
  - RESET ..... 45
  - SATMAG ..... 46
  - ALSQS ..... 47
  - CONVR8 ..... 48
  - CONVR9 ..... 49
  - POLTYP ..... 50
  - RDGEXP ..... 51
- D. The COILDATA Program ..... 55

# MULTIPLE PROPERTY VARIATIONS IN COAXIAL CYLINDRICAL CONDUCTORS DETERMINED WITH MULTIPLE-FREQUENCY EDDY CURRENTS

W. E. Deeds C. V. Dodd


## ABSTRACT

Since many factors can influence eddy-current readings, one must generally make readings on a given sample at several different frequencies to determine a particular factor unambiguously. To expedite and automate the taking of readings, we have written and tested computer programs that take simultaneous measurements of the magnitude and phase of the output voltage from an eddy-current pickup at several different frequencies on standards with known properties. Empirical formulas are then constructed from powers of various possible functions of the magnitude and phase readings, and the formulas are fitted to the desired properties of the material by a least-squares process. Finally, the empirical formulas are used with readings on unknown samples to determine their properties.

Our computer programs, written in FORTRAN IV, are especially designed to be used with coils coaxial with multiple layers of cylindrical conductors, possibly containing defects. However, the instrument calibrating, data taking, and formula fitting parts of the programs are actually applicable to any configuration of coils and conductors.

The programs for the calibration, acquisition, and reduction of data have been used to take measurements at three different frequencies on tubular samples. Up to six different properties have been obtained simultaneously and accurately for each sample.


## INTRODUCTION

Since many parameters influence eddy-current measurements, the unique determination of a given sample property when other properties may also be varying may be difficult or impossible unless sufficient data are obtained to eliminate the effects of unwanted variables. In general, at least as many independent eddy-current readings must be made on a given specimen as there are properties whose variations may affect the readings. By using sinusoidal eddy currents, one can determine no more than two quantities, such as the magnitude and phase, at the fixed frequency. Therefore, if more than two properties are to be determined, multiple-frequency, pulsed, or swept-frequency eddy currents must be used.

The use of multiple-frequency eddy currents to determine multiple properties in parallel strata has been discussed extensively in a previous report.[1] If $N$ different frequencies are used, no more than $2N$ independent pieces of data or readings can be obtained in a single set of measurements. These data are then used to obtain the values of no more than $2N$ independent properties. Sampling pulsed or swept-frequency eddy currents at $2N$ different times or in $2N$ different frequency channels can also provide similar information. We shall consider only the use of multiple sinusoidal frequencies, though the programs and procedures can easily be adapted to equivalent data obtained in other ways.

The complete procedure for optimizing eddy-current tests of coaxial conductors consists of the following steps:

1. The eddy-current readings (of magnitude and phase) are theoretically calculated for a number of different frequencies and all of the anticipated combinations of pertinent sample properties. This step uses the ENCIRM program, which will be described in another report.[2] The calculated magnitudes, phases, and corresponding properties are written on a random access data (RAD) file.

2. The data in the RAD file are read by a second program, LSQENC.[2] It makes a linear least-squares fitting of user-selected polynomials constructed from the magnitude and phase readings to obtain the various desired properties. This program also enables the user to determine what combinations of frequencies and coil designs can best fit those desired properties.
3. Coils of the optimum design are selected or constructed, and the eddy-current instrument is set up to operate at the proper frequencies. The instrument, interfaced with a minicomputer, is then used with a third program, TUBRDG, to make readings of magnitudes and phases on standard tubing samples whose properties are "known" (i.e., determined by some other method). These readings are then written on another RAD file.
4. The data in the latter RAD file are read by a fourth program, TUBFIT, which allows the user to make a linear least-squares fitting[3] of the known standard properties by using various polynomials constructed from the actual experimental readings (of magnitudes and phases at the various frequencies).
5. The polynomial coefficients determined by TUBFIT can be used to calculate the properties of unknown samples. This is done by using the TUBFIT program in the minicomputer or by programming an on-board microcomputer in the eddy-current instrument to use the same least-squares polynomial coefficients with measured magnitudes and phases.

The five steps described above will be discussed in greater detail in later sections of this report, and the TUBRDG and TUBFIT programs will be listed and analyzed in Appendixes A and B respectively.

The instrumentation is shown in the block diagram in Fig. 1 and is described in earlier reports.[1,4] It consists of a multiple-frequency phase-sensitive eddy-current instrument connected to a test coil and a

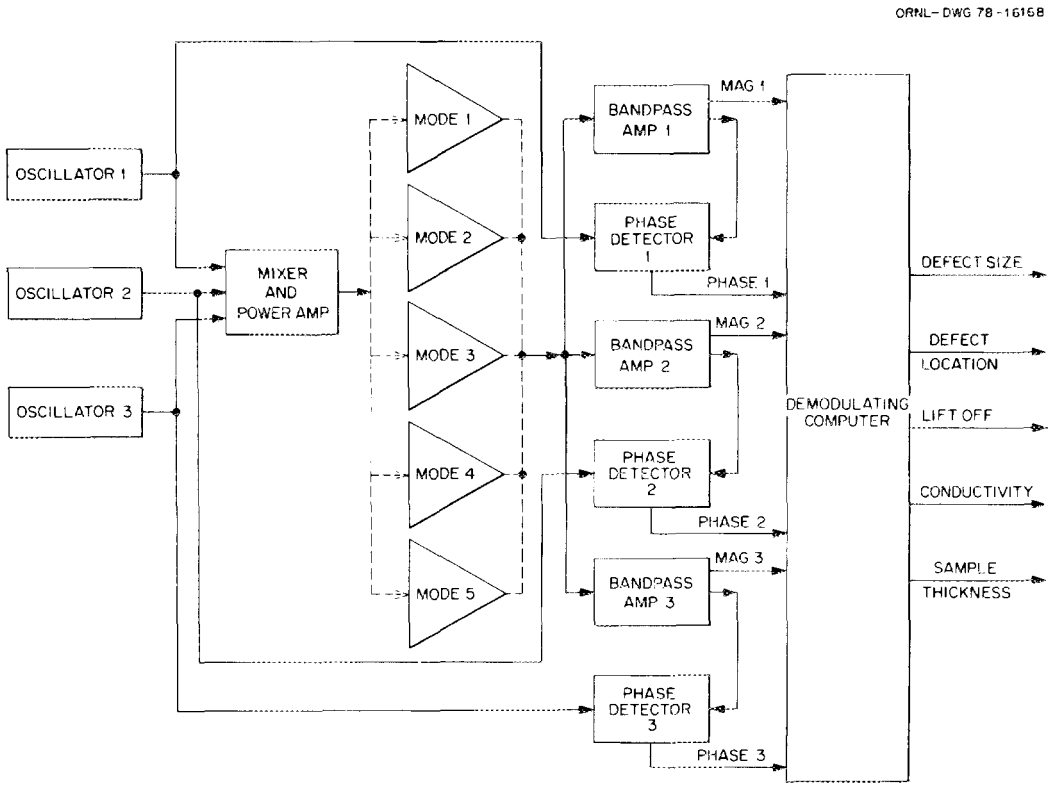


**Fig. 1. Block diagram of eddy-current instrument.**

digital computer capable of measuring and recording the magnitude and phase (for example) of the currents in the test coil at several frequencies.

The programs are to be used with coils that either encircle or are inside coaxial conductors, as shown in Fig. 2. However, the actual configuration is used only for recording the experimental conditions, and data recording and processing are really independent of the particular arrangement of coils and conductors.

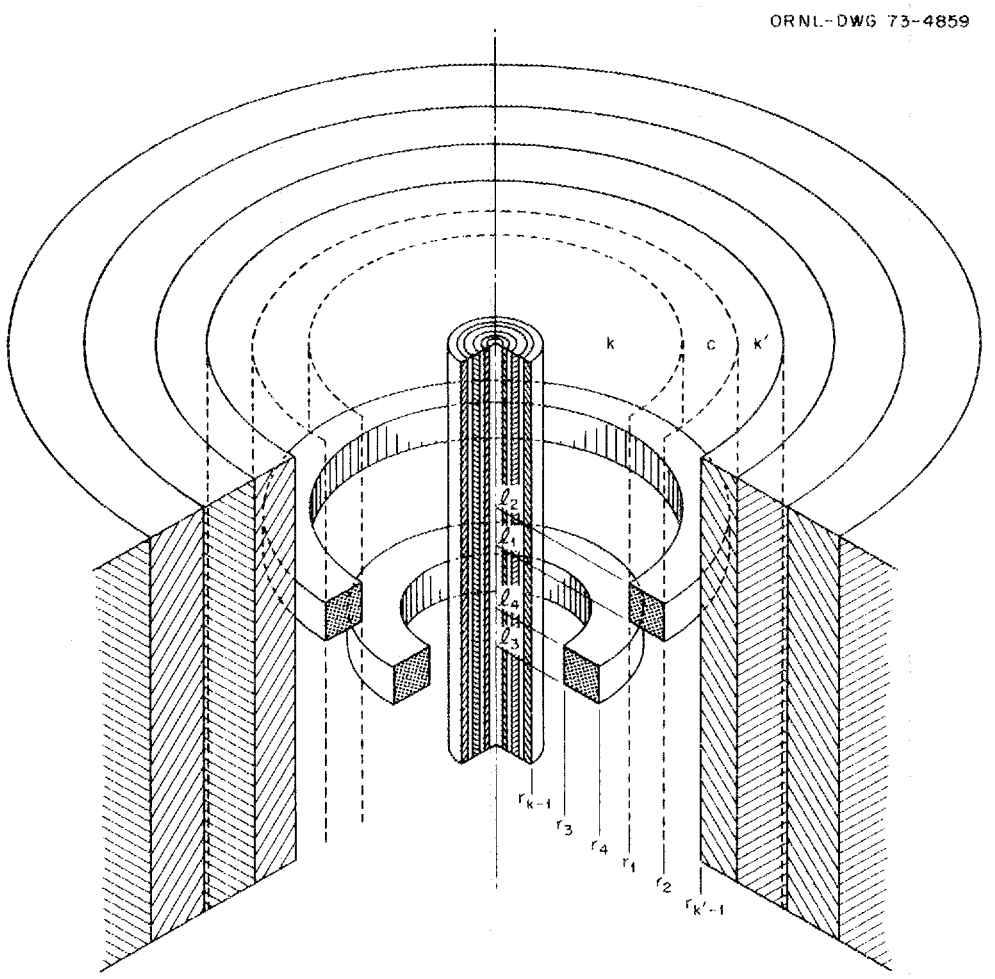


**Fig. 2. Multiple cylindrical conductors encircling and encircled by two disjoint coils.**

## MEASUREMENT OF MULTIPLE PROPERTIES

No more than two pieces of information can be obtained from a given sample with an eddy-current instrument operating at a single frequency — for example, the magnitude and phase of the output voltage. However, many sample properties can affect the readings, and their effects are so interrelated that various combinations of the properties could produce the same pair of instrument readings. Therefore, to distinguish the actual properties from other possible combinations, one must make at least as many independent readings as there are parameters that can both vary and affect the readings. This can be done by using (1) magnitude and phase readings at $N$ different fixed frequencies to obtain $2N$ pieces of information, (2) the

magnitudes of pulsed currents read at $2N$ different times, or (3) the magnitudes and phases of swept-frequency currents read through $N$ different frequency channels or at $N$ different times (or the magnitude read in $2N$ channels or at $2N$ times). Though we shall consider only the first procedure, information obtained by the other methods could also be used.

It is important that the pieces of information be really independent. For example, at very high frequencies the conductor will appear essentially perfect, and the readings will depend mostly upon the lift-off, or coil-to-conductor spacing, and hardly at all upon the other properties. Similarly, at very low frequencies, defects will produce no changes, and even resistivity and thickness are relatively unimportant. Therefore, the range of useful frequencies is generally limited to those with the skin depth of the eddy currents in the material roughly comparable to the thickness of the material.

Assume that there are $N$ parameters that may affect the readings, such as conductivity, permeability, lift-off or fill factor, wall thickness, and defects. Let $(P_{ij})$ be an array of the parameters in which $j = 1, 2, ..., N$ gives the particular parameter or property, and $i = 1, 2, ..., M$ is the index giving the particular sample. We can write the array as

$$(P_{ij}) = \begin{pmatrix} P_{11} & P_{12} & \cdots & P_{1N} \\ P_{21} & P_{22} & \cdots & P_{2N} \\ \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \\ P_{M1} & P_{M2} & \cdots & P_{MN} \end{pmatrix} \tag{1}$$

Let us also suppose that the set of properties $P_{ij}$ $(j = 1, 2, ..., N)$ for the $i$th sample produces a set of $N'$ independent readings $R_{ik}$ $(k = 1, 2, ..., N')$, where $N' \geqslant N$. We want to obtain a set of formulas such that, for the $i$th set of properties,

$$\begin{aligned} P_{i1} &\doteq F_1\,(R_{i1}, R_{i2}, ..., R_{iN'}) \\ P_{i2} &\doteq F_2\,(R_{i1}, R_{i2}, ..., R_{iN'}) \\ &\cdot \\ &\cdot \\ &\cdot \\ P_{iN} &\doteq F_N\,(R_{i1}, R_{i2}, ..., R_{iN'})\,, \end{aligned} \tag{2}$$

where $i = 1, 2, ..., M$, and the symbol $\doteq$ indicates that the functions $F_1, F_2, ..., F_N$ give only approximations to the corresponding properties. To obtain good approximations we may wish to expand the readings into polynomials, such as

$$P_{ij} \doteq F_j = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} \cdots \sum_{n_{N'}=0}^{N_N} C_{jn_1n_2...n_{N'}} R_{i1}^{n_1} R_{i2}^{n_2} \cdots R_{iN'}^{n_{N'}}\,, \tag{3}$$

where $N_1, N_2, ..., N_{N'}$ are the highest powers of the readings $R_{i1}, R_{i2}, ..., R_{iN'}$ in the polynomial. To determine all of the coefficients $C_{jn_1n_2 \cdots n_{N'}}$ (see above) for the $j$th property, one must have at least as

many sets of readings ($i = 1, 2, ..., M$) as there are coefficients to be determined. If $M$ is greater than the number of coefficients to be determined, then a linear least-squares fitting program[4] can be used to determine the set of coefficients to minimize the sum of squares

$$S_j = \sum_{i=1}^{M} (P_{ij} - F_j)^2 \tag{4}$$

for the $j$th property. This is done by a subroutine called ALSQS in the TUBFIT program, which numbers the coefficients used consecutively. The first term in the polynomial expansion $F_j$ is the constant $C_{jo o...o}$, which is called the "offset." Of course, it may be zero.

Once the polynomial coefficients have been determined to calculate a given property with the desired accuracy, they can be used to calculate that property for an unknown sample if the properties of the unknown sample fall within the range of validity of the polynomial expansion. This is assured by fitting the polynomial to a wider range of variations than is expected for the unknown samples.

Thus, dropping the index $i$ from Eq. (3), we can calculate the property $P_j$ of an unknown sample from the readings $R_1, R_2, ..., R_{N'}$ by the formula

$$P_j \doteq F_j = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} ... \sum_{n_{N'}=0}^{N_{N'}} C_{jn_1n_2...n_{N'}} R_1^{n_1} R_2^{n_2} ... R_{N'}^{n_{N'}} . \tag{5}$$

At first glance it seems obvious that any desired precision can be obtained by making the polynomial large enough. This is indeed true. If the number of sets of measured properties, $M$, is equal to the number of sets of linearly independent readings, $N'$, then a unique and exact fit can be obtained for the $M$ values of each property by using only linear terms. However, the function $F_j$ may fluctuate drastically between the fitted points so that intermediate values of the property would not be fitted well.

Furthermore, if each reading $R_j$ is subject to a certain amount of random error, $\Delta R_j$, so that the polynomial becomes

$$F_j = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} ... \sum_{n_{N'}=0}^{N_{N'}} C_{jn_1n_2...n_{N'}} (R_1 \pm \Delta R_1)^{n_1} (R_2 \pm \Delta R_2)^{n_2} ... (R_{N'} \pm \Delta R_{N'})^{n_{N'}} , \tag{6}$$

then we see that the random errors, particularly in the terms involving high powers, may lead to considerable error. The worst possible combination of errors, in which all the errors in the value of the property are in the same direction, will be called the "DRIFT." In the curve-fitting program TUBFIT, the worst possible combination of errors produced by changes of 0.01% in each magnitude and 0.01° in each phase are calculated and printed as the DRIFT, since errors of this amount are characteristic of present eddy-current equipment.

Since most material properties will not oscillate rapidly in a limited range of any given parameter, it is rarely necessary or even useful to include cubic or higher powers of the readings to fit a given property.

Details on the linear least-squares curve-fitting procedure are given in refs. 1 and 3 and will be discussed briefly in a later section. The coefficients $C_{jn_1n_2...n_{N'}}$ in Eq. (3) are determined by the subroutine called ALSQS in the TUBFIT program to minimize the sum of squares in Eq. (4).

## EXPERIMENT DESIGN

The computer program ENCIRM[2] calculates the values of the magnitude and phase of the output voltage at various frequencies and for various combinations of the pertinent properties, including possible defects. The ranges of the various properties should cover all of the expected variations in those properties. The coil and circuit parameters can be varied as the operator chooses. The "readings" – the calculated values of the magnitudes and phases of the output signal for a given coil system at the various frequencies – are written in a RAD file. These data are then read by another program, LSQENC, which allows the operator to try various polynomials (using magnitudes and phases at various combinations of frequencies) to fit the desired properties. In this way the operator can determine the optimum coil design, circuit values, and test frequency combination to measure the desired properties most accurately and to reject the effects of unwanted variables.

This computer-assisted experiment design step will be extensively treated in another report[2] and can be very helpful for optimizing the experimental conditions or for conducting feasibility studies. However, it is not absolutely essential. As a result of previous experience, design data from other similar problems, or limitations of available equipment, one can choose the coil and instrument parameters without using the ENCIRM and LSQENC programs. Our TUBRDG and TUBFIT programs use actual experimental instrument readings, and the coil, circuit, and instrument values are used only for identification purposes.

Although the use of ENCIRM and LSQENC is not essential, we should emphasize that they can help predetermine the accuracy for measuring any particular property and the allowable tolerances in the instrumental parameters.

## INSTRUMENTATION

The required equipment is essentially the same as we have described in other reports.[1,3] It includes a coil system capable of fitting the tubing samples, a multiple-frequency eddy-current instrument to operate this system, and a digital computer with a parallel data acquisition system (called a MODAC for the ModComp IV computer) and a FORTRAN IV compiler compatible with the FR4 compiler of the ModComp MAX IV system processor.[5] The computer programs assume that the tubing will be inspected from one side, either by inner or by encircling coils. However, only the recording of the test conditions would need to be changed for through-transmission inspections. The data processing parts of the programs would need no changes. A block diagram of the eddy-current instrument is shown in Fig. 1. The instrument must also contain a calibration module that will switch in phase-shifting networks on command from the computer to introduce known phase shifts at the various frequencies. Modifications of the programs to allow manual switching of the calibration networks are described in Appendixes A and B.

Ideally, the coil design and the operating frequencies should be optimized by running the ENCIRM and LSQENC programs first.[2] From the results of running those programs or from other considerations, a suitable coil and circuit design is chosen or constructed, and the instrument is set up to operate at the frequencies expected to give the best results.

The actual coil and circuit parameters are then measured so that they can be entered in the data file for the experiment. The values of the coil and circuit parameters are used only for records and not for running TUBRDG and TUBFIT. These two programs use only the actual measured eddy-current readings and the stated material properties (assumed to have been determined previously for the calibration standards).

The computer should have storage and core capacity of about 7K for the operating program itself, disk storage capacity for the random access data files (up to perhaps 50K), and a parallel data acquisition system (such as the MODAC for the ModComp IV).

## ACQUISITION OF DATA FROM STANDARDS WITH TUBRDG

Tubing standards are obtained with properties that span the expected ranges of the various properties to be encountered in the unknown samples, and the optimum coil is selected or constructed. The readings (of the magnitudes and phases of the eddy currents at the selected frequencies) are then made with the actual coil and circuit components and the actual eddy-current instrument by running the TUBRDG program, which will write both the readings measured on the standards and the calibration readings on a RAD file.

The TUBRDG program is written in FORTRAN IV.[5] It asks the operator to type in data concerning the test apparatus, test conditions, and tubing samples. It then takes readings of the magnitude and phase of the output voltage at the various frequencies for the various samples and finally stores the readings in RAD file 37.

To use the program the operator should first call up the TUBRDG program, catalog it under a different name, such as NEWRDG, then exit. The new program (e.g., NEWRDG) can then be edited and run without endangering the original program. Table 1 lists the editorial changes that must be made in the renamed version of TUBRDG before it is run. Table 2 lists the promptings (and suitable responses) displayed as the program is run. Table 3 lists the quantities contained in the COMMON block B3. A detailed description and explanation of the TUBRDG program is given in Appendix A.

The dimensions of the various arrays and RAD files need to be carefully checked. Actually, the dimensions can be larger than needed, as long as the capacity of the computer is not exceeded. However, they must not be smaller than indicated by the guides in lines 37–47 and lines 62–68. RAD file 29 should contain data for various coils, but if a coil with different parameters is to be used, the operator can type in the new values when the program is run.

## FITTING OF PROPERTIES WITH DATA FROM STANDARDS BY USING TUBFIT

The TUBFIT program not only determines least-squares coefficients of polynomials to fit the properties of standard samples, but also stores the coefficients, takes readings (of magnitudes and phases) of unknown samples, and uses the stored coefficients to calculate the properties of the unknown samples. It is an interactive program written in FORTRAN IV[5] and requires about 7K of memory, four RAD files for storing up to 50K of data, and a parallel data acquisition system (called MODAC on the ModComp IV system).

**Table 1. Summary of editorial changes preparatory to running TUBRDG[a]**

| Lines | Action |
|---|---|
| 49–59 | Adjust array dimensions; follow key in lines 37–47. |
| 70–71 | Make sure FILES 29 and 37 have been allocated with correct specifications; follow guide in lines 62–68. |
| 73 | Select operator input terminal (LI), disk input terminal (LID), operator output terminal (LOT), line printer (LPT), number of data channels (NCHS), and RAD file limit (NFILIM), which should be the same as the first dimension of FILE 37. The LID may be the same as LI. |
| 75 | Set MAG = 1 if there is a saturating magnet used with ferromagnetic tubing, otherwise MAG = 0. |
| Line 15 of the READAT subroutine | The real number 819.2 in the definition of CONFAC is the number of A/D converter counts equal to 1.0 V for the instrument used to make the readings. As long as the same value is used in READAT when called by the TUBFIT program, there will be no difficulty. However, the values of magnitude and phase printed out will be incorrect unless CONFAC is the correct conversion factor. |

[a]The "PROTECT" switch for the computer must be in the OFF position to allow data to be read by the MODAC.

Table 2. Interactive program queries and operator responses during execution of the TUBRDG program

| Line | | Query and response |
|---|---|---|
| 99 | | TYPE IN THE FOLLOWING DATA AS REQUESTED |
| 105 | | PROBE #: |
| 106 | * | Enter six-character alphanumeric; may contain leading or trailing blanks (spaces). If response is "OLD," data from previous running of TUBRDG will be used, and the next response required will be at line 271. If response is a zero (or blank space), the user will be asked to type in coil name and data. This will also occur if the user types a coil name not found in FILE 29. If the coil name *is* found in the file, no other response will be needed for this query. |
| 115 | | SERIAL #: |
| 116 | * | Enter free format integer. |
| 118 | | DRIVER SERIES RESISTANCE: |
| 119 | * | Enter free format real number of ohms. |
| 121 | | DRIVER SHUNT CAP: |
| 122 | * | Enter free format real number of farads. |
| 124 | | PICK-UP SHUNT RESISTANCE: |
| 125 | * | Enter free format real number of ohms. |
| 127 | | PICK-UP SHUNT CAP: |
| 128 | * | Enter free format real number of farads. |
| 130 | | CABLE I.D.#: |
| 131 | * | Enter six-character alphanumeric; may contain spaces. |
| 133 | | LENGTH OF CABLE: |
| 134 | * | Enter free format length (real number). |
| 136 | | CAPACITANCE OF CABLE: |
| 137 | * | Enter free format real number of farads. |
| 139 | | EDDY CURRENT INSTRUMENT #: |
| 140 | * | Enter six-character alphanumeric; may contain spaces. |
| 142 | | POWER OSC I.D.: |
| 143 | * | Enter six-character alphanumeric; may contain spaces. |
| 148 | | NO. OF FREQUENCIES: |
| 149 | * | Enter free format integer. |
| 151 | | INPUT THE VALUE OF EACH FREQ SEPARATED BY A SPACE: |
| 152 | * | Enter a free format real number for each of the frequencies, with a space or comma between the numbers. |
| 154 | | INPUT THE FOLLOWING DATA, 6 CHAR WITH A SPACE BETWEEN |
| 155 | | FREQUENCY: (first freq) (second freq) ... |
| 157 | | PICK-UP AMP I.D.: |
| 158 | * | Enter a six-character alphanumeric for each frequency in order, with a space or comma between the I.D. numbers. |
| 161 | | PHASE DETECTOR I.D.: |
| 162 | * | Enter a six-character alphanumeric for each frequency in order, with a space or comma between the I.D. numbers. |
| 164 | | 180-DEG SW(OFF/ON): |

**Table 2 continued.**

| Line | | Query and response |
|---|---|---|
| 165 | * | Enter a three-character alphanumeric [either OFF or ON(space)] for each frequency in order, with a space or comma between the words. |
| 171 | | NUMBER OF TUBING SAMPLES: |
| 172 | * | Enter a free format integer. |
| 174 | | NUMBER OF COAXIAL REGIONS: |
| 175 | * | Enter a free format integer. |
| 180 | | TYPE (No.) PERMEABILITIES FOR REGIONS OF SAMPLE (No.): |
| 181 | * | Enter free format real numbers for the relative permeabilities of the various regions of the designated sample, starting with the region farthest from the coil and putting a space or comma between the numbers. |
| 183 | | TYPE (No.) RESISTIVITIES FOR REGIONS OF SAMPLE (No.): |
| 184 | * | Enter free format real numbers for the resistivities in microhm-cm for each of the regions of the designated sample, putting a space or comma between the values and starting with the region farthest from the coil. |
| 186 | | TYPE (No.) VALUES OF REGION RADII FOR SAMPLE (No.): |
| 187 | * | Enter free format real numbers for the radii in inches for the interfaces between the regions of the designated sample, starting with the boundary between the two regions farthest from the coil and leaving a space or comma between the numbers. |
| 201 | | ERROR: COIL WILL NOT FIT. |
| | * | This message will appear if the limiting radius of the coil will not accommodate the tubing sample. If this occurs the program will stop, and the user should start over after checking coil and tubing dimensions. |
| 203 | | NO. OF DEFECTS IN EACH OF (No.) REGIONS OF SAMPLE (No.): |
| 204 | * | Enter an integer for the number of defects in each region of the designated sample, starting with the region farthest from the coil and *not* including the coil region. Leave a space or comma between the integers. |
| 210 | | TYPE SIZE, AXIAL DIST & RADIAL LOCATION OF DEFECT # (No.) |
| 211 | | FOR SAMPLE # (No.) IN REGION (No.) |
| 212 | * | Enter three free format real numbers for the properties of the designated defect in the designated region. The dimensions should be in cubic inches for the size and inches for the axial displacement from the center of the coil and the radial distance from the region surface nearer the coil. Leave a space or comma between the numbers. |
| 230–270 | * | If the user typed "OLD" in response to the query PROBE #: in line 105, previous data in FILE 37 will be read and displayed on the LOT terminal, ending with the data for the 180 PHASE SWITCH. |
| 271 | | The display on the LOT terminal will pause for inspection until the operator types any character (or a space) followed by a carriage return. |
| 285–286 | | SAMP PROP DIAM THICK RHO PERM DEFSIZE RADLOC AXILOC |
| | | No response is necessary; sample and defect properties will be displayed. |
| 312 | | CALIBRATION READINGS: |
| 313 | | The calibration readings will be taken automatically and will be continually updated until the foot pedal is pressed to stop the calibration. |
| 342 | | SWITCH TO OPERATE AND ANSWER Y IF YOU ARE READY TO TAKE THE FIRST SET OF READINGS, N OTHERWISE: |
| 345 | | Type Y or N. If N is typed go to statement 1140 (line 441). |

Table 2 continued.

| Line | Query and response |
|---|---|
| 367–368 | SAMP PROP DIAM THICK RHO PERM DEFSIZ RADLOC AXILOC |
| | Place the probe at the indicated position on the indicated sample and press the foot switch to take the readings. |
| 412 | (No.) SETS OF READINGS HAVE BEEN MADE. |
| 413–414 | DO YOU WANT TO DELETE THE PREVIOUS SET OF READINGS? TYPE Y OR N: |
| 416 | Type Y or N. If N is typed the calibration and sample readings will be added to previous sets. Otherwise the latest readings will be ignored, and control will jump to statement 1140 (line 441). |
| 443 | DO YOU WANT TO (1) RE-DO THE CALIBRATIONS, (2) MAKE |
| 444 | ANOTHER SET OF READINGS, OR (3) STOP? TYPE 1, 2, OR 3: |
| 445 | Type 1, 2, or 3. If 3 is typed the rest of the procedure will be automatic. |

*Responses preceded by an asterisk will not be necessary if the LID terminal is a disk partition containing prerecorded responses.

Table 3. Quantities in COMMON block B3

| | | |
|---|---|---|
| COIL | = | coil name (containing up to six characters) |
| RBAR | = | mean coil radius (in inches) |
| R1 | = | normalized inner coil radius |
| R2 | = | normalized outer coil radius |
| XL | = | normalized length of pickup and driver coil |
| ZL | = | normalized distance between pickup and driver coils |
| RLIM | = | normalized limiting radius of tube to fit the coil assembly |
| R6 | = | RDCDR = driver coil resistance (ohms) |
| R7 | = | RDCPU = pickup coil resistance (ohms) |
| TNDR | = | number of turns on driver coil |
| TNPU | = | number of turns on pickup coil |
| ZLDR | = | air inductance of driver coil (henries) |
| ZLPU | = | air inductance of pickup coil (henries) |
| X | = | unused parameter in the RAD file |

The operator should first call up the TUBFIT program in the edit mode, rename it (e.g., NEWFIT), and exit. Then the newly named program (e.g., NEWFIT) can be called up, edited, and run without endangering the original program.

The editorial changes that must be made in the renamed version of TUBFIT before it is run are summarized in Table 4. When the input data and dimensions have all been checked, the program can be run. Table 5 shows both the queries that are used to prompt the operator as the program is run and the types of responses needed.

As TUBFIT is run it will read data stored in FILE 37 during the execution of the TUBRDG program. If the coil named NPROBE in the latter program is not found in the coil data file, FILE 29, then the operator will be asked to type in the coil data, as defined in Table 3.

**Table 4. Summary of editorial changes preparatory to running TUBFIT[a]**

| Line | Action |
|---|---|
| 65–79 | Adjust array dimensions; follow key in lines 44–60 |
| 87–97 | Make sure files 29 and 37 are provided with the necessary data; file 37 is written when the TUBRDG program is run and must have the same specifications in both programs. Make sure files 21 and 31 have been allocated with correct specifications. |
| 107 | The NPT value should agree with the first dimension of the TMAGM array, for example. The line printer output device is LOU; LITEK is the operator input terminal; LOTEK is the operator output terminal. The NFILIM value should agree with the first dimension of FILE 37. |
| 108 | Set NUNIT=1 for output of lengths in inches; NUNIT=2 for output lengths in millimeters. |
| 118–119 | Insert NRT*NPT initial values of RHO, U, TH, RDCL, DSIZ, RLOC, and ZLOC. Different values for certain regions will be read from FILE 37 when necessary. |
| 120 | Insert NFT values of GAIN; this is used only for information printout. |

[a]The "PROTECT" switch for the computer should be in the OFF position to allow data to be read by the MODAC.

**Table 5. Interactive program queries and operator responses during execution of TUBFIT**

| Line | Query and response |
|---|---|
| 166 | If the number of lines read from FILE 37 exceeds the limit, NFILIM, an error message will be printed, and the program will be terminated. The operator should check the dimensions and data in the file. |
| 172 | If the coil named NPROBE in FILE 37 is not found in FILE 29, the terminal will display |
| | COIL (Name) NOT FOUND. TYPE FOLLOWING COIL PARAMETERS: RBAR,R1,R2,XL,ZL,RLIM, RDCDR,RDCPU,TNDR,TNPU,ZLDR,ZLPU,X |
| | Type in the data requested (see definitions in Table 3). |
| 225 | If the number of properties (NP) read from FILE 37 does not agree with the number (NPT) used in the array dimensions of TUBFIT, the following error message will be printed: |
| 227 | THERE IS AN ERROR IN THE NUMBER OF PROPERTIES. |
| 228 | NP=(No.) NPT=(No.) NPTT=(No.) |
| | Execution will be halted. Check the input data and dimension statements. |
| 307 | SELECT NUMBER OF PROPERTY TO BE FITTED. |
| 308 | 1=RHO 2=MU 3=THICKNESS 4=RADCLR 5=DEFECT RAD 6=DEFECT SIZE |
| 309 | SELECT REGION TO BE FITTED. LOWEST IS NUMBER 1. |
| 310 | Enter two integers with a comma or space between them: from 1 to 6 for the property and from 1 to NRT for the region where the property is to be fitted, with region 1 farthest from the coil. |
| 322 | TYPE 1 IF THERE IS OFFSET: 0 IF NO OFFSET: |
| 323 | Enter 0 or 1, usually 1. |
| 327–328 | SELECT THE NUMBER OF THE FUNCTION TYPE, POLYNOMIAL DEGREE &# OF |
| 329 | CROSS TERMS FOR EACH MAGNITUDE AND PHASE |
| 331 | FUNCTION TYPE: 1=LINEAR 2=LOG 3=EXP 4=INV |
| 334 | FCTN POL #CROSS<br>TYPE DEG TERMS |
| 342 | MAG AT (Freq) HZ |
| | Enter three integers with commas or spaces between them. The first should be 1, 2, 3, or 4; the second at least 0 but probably not greater than 3; the third integer should be zero. |

**Table 5 continued.**

| Line | Query and response |
|---|---|
| 343 | PHA AT (Freq.) HZ |
| | Enter three more integers: 1, 3, or 4; 0 up; 0 up (not necessarily zero). |
| | Repeat responses to lines 342 and 343 for each of the frequencies. If the sum of the offset number and the second and third integers in these responses exceeds 15, the following error message will be displayed: |
| 357 | ERROR: # OF TERMS IN POLARY EXCEEDS DIMENSION |
| 462–463 | 1 FIT PROP 2 PRT ENTIRE FIT 3 PRT/SV COEF 4 CHG FCTN/POL TYP 5 RUN TEST |
| | Enter 4 and include fewer terms in the polynomial. |
| 423–424 | RMS DIFF IN (Property) IN REG (No.) = (No.)(Units) DRIFT= (No.)(Units) |
| | Information on quality of fit; no response needed. |
| 462-463 | 1 FIT PROP 2 PRT ENTIRE FIT 3 PRT/SV COEF 4 CHG FCTN/POL TYPE 5 RUN TEST |
| | Enter integer from 1 through 5; 1 returns control to statement 300, permitting selection of another property to be fitted; 2 returns control to statement 500 for a printout of the actual and calculated property for each sample, with the difference and "drift"; 3 returns control to statement 590 and saves the coefficients of the fit by writing them on FILES 31 and 21; and 4 returns control to statement 350 and allows a change in the polynomial used to fit the same property just fitted. If 5 is typed the following message will be displayed: |
| 538–539 | PRINT: 1. PARAMETERS 2. MEAS VOLT 3. CAL & DISPLAY PROPS 4. RAW RDG 5. RE-CALIB 6. STOP |
| | Enter integer from 1 through 6; 1 produces a complete printout of input data; 2 lists the measured voltages and phases; 3 is the choice for making measurements and calculating properties from the readings; 4 displays the magnitudes and phases; 5 causes a recalibration; and 6 terminates execution. If 3 is chosen the foot pedal is pressed to take a set of readings on the selected sample, and the program will then return to the above set of choices. |

The operator will then be asked to select a property to be fitted and the region where the property is located. Next the type of polynomial for approximating the property is specified: if there is to be a constant term (called "offset"), the type of functions of the various magnitudes and phases, the powers, and the cross-terms (if any) that will be included. Each magnitude and phase can be converted to a variable that is the same as the original reading (LIN function), the logarithm of the reading (LOG function), the exponential of the reading (EXP function), or the reciprocal of the reading (INV function). The operator can then choose the highest power to which that variable is to be raised as well as the number of cross products between the variables at a given frequency.

When the type of polynomial has been chosen, the computer will do a linear least-squares fitting[3] of the polynomial to the actual properties, as written in FILE 37 by TUBRDG. The differences between the actual properties and those calculated by using the least-squares polynomial are called DIFF and can be printed out as an option. In any case, the root mean square (rms) value of the differences will be printed for a given polynomial fitting a given property. In addition, the worst possible error in the property caused by the worst possible combination of errors of 0.01% in the magnitudes or 0.01° in the phases is called DRIFT. It can be printed out, and the rms value of the drifts is always printed. If the differences and drifts are larger than desired, different polynomials can be tried. If the fits are satisfactory, a complete printout of the actual and calculated property values, differences, and drifts can be obtained, and the coefficients of the fits can be saved and written on FILE 31 (and in hexadecimal binary form in FILE 21). Then another

property and/or region can be fitted in a similar fashion. This can be continued until all desired properties and regions have been covered if the capacities of FILES 21 and 31 are not exceeded.

Appendix B contains a detailed analysis and explanation of the TUBFIT program and also a printout of a sample run.

## FITTING PROPERTIES OF UNKNOWN SAMPLES WITH TUBFIT

The TUBFIT program not only determines least-squares coefficients of polynomials to fit the properties of standard samples but also stores the coefficients, takes readings (of magnitudes and phases) of unknown samples, and uses the stored coefficients to calculate the properties of the unknown samples.

The least-squares coefficients can be stored in the memory of a microcomputer built into a multifrequency eddy-current instrument so that the properties of the unknown samples can be calculated and displayed directly by the instrument.

The TUBFIT program is written in FORTRAN IV[5] and requires a computer with a parallel data acquisition system (called MODAC for the ModComp IV). The TUBRDG program must be run first, to write data from known standards on RAD file 37.

If the third option in lines 538–539 of TUBFIT is chosen (CAL & DISPLAY PROPS) by typing 3 when prompted, the LOTEK terminal will display the name of the properties being measured and the calculated values of the properties for whatever sample the coil is on at the time. These values will also be printed on the LOU output device when the foot pedal is pressed to record the values and to keep a hard copy of the measured sample properties.

If the fourth option is selected (RAW RDGS) by typing 4 when prompted, the instrument readings of magnitudes and phases at the various frequencies will be displayed on the LOTEK terminal and the LOU output device. Although this option will seldom be chosen, it can be used to keep a record of the instrument readings as a check on the operation of the equipment.

## ACKNOWLEDGMENT

The authors wish to acknowledge the assistance of L. D. Chitwood for constructing the instrumentation, G. W. Scott and W. A. Simpson for reviewing the manuscript, and B. G. Ashdown for editing it.

## REFERENCES

1. W. E. Deeds, C. V. Dodd, and G. W. Scott, *Computer-Aided Design of Multifrequency Eddy-Current Tests for Layered Conductors with Multiple Property Variations,* ORNL/TM-6858 (in preparation).

2. C. V. Dodd, D. H. McCollough, and C. W. Nestor, *The ENCIRM Program* (report in preparation).

3. G. Golub, "Numerical Methods for Solving Linear Least Squares Problems," *Numer. Math.* 7, 202–16 (1965).

4. C. V. Dodd and L. D. Chitwood, *Three-Frequency Eddy-Current Instrument for Multiple Property Problems,* ORNL-5495 (March 1979).

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

## Appendix A
## THE TUBRDG PROGRAM

The TUBRDG program is written in FORTRAN IV, which is compatible with the FR4 compiler in a ModComp MAX IV system processor.[1] It prompts the user to type in data on the experimental conditions and equipment, takes calibration and data readings from an eddy-current instrument through a parallel data acquisition system, and then writes all the data on a random access data file (FILE 37).

Some special variables are declared in lines 8 and 9, and some dimensions that do not need to be changed are stated in line 10.

```
 1          PROGRAM TUBRDG
 2   C      (9 APRIL 1979)
 3   C
 4   C THIS PROGRAM READS MAGNITUDE AND PHASE DATA AT DIFFERENT
 5   C      FREQUENCIES FOR ENCIRCLING TYPE COILS, USING THE PHASE
 6   C      SENSITIVE INSTRUMENT.
 7   C
 8          REAL*8 TITLE(3)
 9          REAL*6 NPROBE,NCABLE,INSTNO,POWOSC,PICKAM,PHADET,COIL
10          DIMENSION IDAY(3),ITIM(3),IDY(3),ITM(3)
```

Various integer variables are defined in lines 12–35.

```
11   C
12   C      LI=LOGICAL INPUT TERMINAL(FROM OPERATOR)
13   C      LID=LOGICAL INPUT TERMINAL(MAY BE ASSIGNED TO A DISK FILE)
14   C      LOT=LOGICAL OUTPUT TERMINAL(TO OPERATOR)
15   C      LPT==LINE PRINTER(OUTPUT)TERMINAL
16   C      NCHS=NUMBER OF CHANNELS
17   C      NFT=NUMBER OF FREQUENCIES
18   C      LNF=2*NFT
19   C      NF=FREQUENCY INDEX
20   C      NS=SAMPLE INDEX
21   C      NST=NUMBER OF TUBING SAMPLES
22   C      NSTOP=INDEX TO STOP THE INSTRUMENT READINGS
23   C      MAG=INDEX TO TURN ON THE SATURATING MAGNET(=0 IF NONE USED)
24   C      NR=REGION INDEX(1=FARTHEST FROM COIL;NRT=COIL REGION)
25   C      NRT=NUMBER OF COAXIAL REGIONS
26   C      NRT1=NRT-1
27   C      ND=DEFECT INDEX
28   C      MNDEF=MAXIMUM NUMBER OF DEFECTS FOR ANY REGION
29   C      NDF=NUMBER OF DEFECTS IN A GIVEN SAMPLE
30   C      NFILIM=FILE LIMIT ON LINES IN FILE 37, THE DATA FILE
31   C      NP=PROPERTY INDEX
32   C      NPT=NUMBER OF PROPERTIES=TOTAL NUMBER OF SAMPLES+DEFECTS
33   C      NPHCAL=NUMBER OF PHASE CALIBRATIONS
34   C      NMGCAL=NUMBER OF MAGNITUDE CALIBRATIONS
35   C      NCAL=NPHCAL*NMGCAL=TOTAL NUMBER OF CALIBRATION READINGS
```

In lines 37–47 the dimensions of arrays are stated symbolically in terms of the variables defined above.

```
36   C
37   C      DIMENSION FREQ(NFT),PICKAM(NFT),PHADET(NFT),PHASW(NFT)
38   C      DIMENSION PERM(NST,NRT1),RHO(NST,NRT1),RAD(NST,NRT1),NDF(NST)
39   C      DIMENSION NDEF(NST,NRT1),DFSIZE(NST,NRT1,MNDEF),RADCLR(NST)
40   C      DIMENSION RADLOC(NST,NRT1,MNDEF),AXILOC(NST,NRT1,MNDEF)
41   C      DIMENSION TMAG(NPT,NFT),PHASE(NPT,NFT),TH(NST,NRT1)
42   C      DIMENSION SUMMAG(NPT,NFT),SUMPHA(NPT,NFT),D(NST,NRT1)
43   C      DIMENSION SSMAG(NPT,NFT),SSPHA(NPT,NFT)
44   C      DIMENSION SDVMAG(NPT,NFT),SDVPHA(NPT,NFT)
45   C      DIMENSION SUMCAL(2*NFT,NCAL),SSCAL(2*NFT,NCAL),SDVCAL(2*NFT,NCAL)
46   C      DIMENSION VOLTS(2+2*NFT),RDGC(2+2*NFT,NCAL),OLRDGC(2+2*NFT,NCAL)
47   C      DIMENSION TOFSET(2+2*NFT),TSLOPE(2+2*NFT)
```

The actual dimensions of the various arrays must be inserted in lines 49–59 by using the guides given above. The dimensions may be larger than actually necessary if the storage capacity of the computer is not exceeded.

```
48  C
49          DIMENSION FREQ(4),PICKAM(4),PHADET(4),PHASW(4)
50          DIMENSION PERM(56,7),RHO(56,7),RAD(56,7),NDF(56)
51          DIMENSION NDEF(56,7),DFSIZE(56,7,4),RADCLR(56)
52          DIMENSION RADLOC(56,7,4),AXILOC(56,7,4)
53          DIMENSION TMAG(168,4),PHASE(168,4),TH(56,7)
54          DIMENSION SUMMAG(168,4),SUMPHA(168,4),D(56,7)
55          DIMENSION SSMAG(168,4),SSPHA(168,4)
56          DIMENSION SDVMAG(168,4),SDVPHA(168,4)
57          DIMENSION SUMCAL(6,4),SSCAL(6,4),SDVCAL(6,4)
58          DIMENSION VOLTS(8),RDGC(8,4),OLRDGC(8,4)
59          DIMENSION TOFSET(8),TSLOPE(8)
```

The COMMON block B3 in line 60 is shared with the subroutine CIRCOL and serves to return coil data to the main program when the subroutine reads it from RAD file 29. The quantities in COMMON block B3 are defined in Table 3.

```
60          COMMON/B3/COIL,RBAR,R1,R2,XL,ZL,RLIM,R6,R7,TNDR,TNPU,ZLDR,ZLPU,X
```

The RAD files are defined in lines 70 and 71. File 29 contains the data for various encircling coils; file 37 will contain the data written on it when the TUBRDG program is run. The comments in lines 62–68 describe the minimum acceptable dimensions for RAD file 37.

```
61  C
62  C***CAUTION***BE SURE ENOUGH SPACE HAS BEEN ALLOCATED ON RAD FILE 37
63  C      FOR WRITING ALL THE RECORDS OUTPUT BY THIS PROGRAM. NUMBER OF
64  C      RECORDS IN THIS DEFINE STATEMENT(NFILIM=FIRST # IN PARENTHESES)
65  C      SHOULD EQUAL AT LEAST 4+NPT+2*NFT+NST*(NRT+1).
66  C      THE SECOND SHOULD BE AT LEAST 2 MORE THAN THE GREATEST OF THE
67  C      FOLLOWING: 26, 8*NFT, 4*NCAL, 6*MNDEF, OR 4*NRT-2.
68  C      FOR EXAMPLE: DEFINE FILE 37(NFILIM,8*NFT+2,U,IREC).
69  C
70          DEFINE FILE 37(400,40,U,IREC)
71          DEFINE FILE 29(80,32,U,NCOIL)
```

The data designating the logical input and output units are inserted in line 73. The input terminal at which the operator types input data is LI; LID is the logical input for data (it may be the same as LI or assigned to a disk partition containing prerecorded data); LOT is the CSI terminal where the operator is prompted; LPT is the hard-copy line printer. The data in line 74 will not need to be changed. In line 75, ITIMS is the number of times the data acquisition system is read and will not need to be changed unless the system is altered. To turn on the saturating magnet for ferromagnetic tubing, MAG is used; it should be 1 for inspecting ferromagnetic tubing and 0 otherwise. The number of phase and magnitude calibrations at each frequency is NPHCAL and NMGCAL respectively; they will normally be changed only if the calibrator module and subroutines are changed.

```
72  C
73          DATA LI/1/,LID/9/,LOT/5/,LPT/6/,NCHS/8/,NFILIM/400/
74          DATA TITLE/'TUBING CALIBRATION DATA '/,BLANKS/'     '/
75          DATA ITIMS/1024/,MAG/1/,NPHCAL/2/,NMGCAL/2/
```

The screen of the operating terminal is cleared by jumping to statement 1800 in line 593 and then back to statement 10 in line 87 because line 604 returns control to ISTATE, which was set equal to 10 in line 81.

```
 76          DATA TOFSET/2.2532,-1.4992,.99074,-1.3361,.47935,-.72012,2*0./
 77          DATA TSLOPE/-.1366,.0906,-.0597,.08037,.0286,.04375,2*0./
 78   C
 79   C      GO CLEAR TERMINAL SCREEN & RETURN HERE
 80   C
 81          ASSIGN 10 TO ISTATE
 82          GO TO 1800
590   C THIS SECTION OF ASSEMBLER CODE IS NECESSARY IN ORDER TO CLEAR
591   C      THE SCREEN OF THE TERMINAL.
592   C
593   1800   CONTINUE
594          INLINE
595          LDI,4       #001A       LOAD SUB(ERASE SCREEN) IN REG 4
596          LDI,1       #4002       LOAD 4000 + CHANNEL OFFSET INTO REG 1.
597          OCB,1,8                   SELECT CH 02
598          LDI,3       #801A         LOAD REG 3 WITH NO PARITY,8DATA BITS 1 STOP
599          OCB,3,9                   OUTPUT REG 3
600   READY  ISB,2,9                   INPUT STATUS IN REG 2
601          TBRB,2,8    READY         TEST BIT 8 IN REG 2.STAY IN LOOP UNTIL RDY
602          ODB,4,9                   OUTPUT BITS 8-15 OF REG 4 TO CH A02
603          FINI
604          GO TO ISTATE
```

Since ISTATE = 10, control returns to statement 10, which writes the name of the program on the LOT terminal.

```
 83   C
 84   C FORM FEED ON LPT FOR SUMMARY REPORT.
 85   C PRINT TITLE AND DATE
 86   C
 87   10     WRITE (LOT,20) TITLE
 88   20     FORMAT(1X,3A8)
 89   30     FORMAT(1H )
```

The date and time are obtained and written on the LOT terminal.

```
 90          CALL DATE(IDAY)
 91          CALL TIME(ITIM)
 92          IT=IDAY(1)-1900
 93          IDAY(1)=IDAY(2)
 94          IDAY(2)=IDAY(3)
 95          IDAY(3)=IT
 96          WRITE(LOT,40) IDAY,ITIM
 97   40     FORMAT(' DATE  ',2(I2,1H/),I2,'     TIME  ',2(I2,1H:),I2)
```

The operator is prompted to type in data for the experimental apparatus and conditions. The data are read from the LID terminal, which might be the same as the LI terminal, or it might have been assigned to a disk partition containing prerecorded data.

```
 98          WRITE(LOT,50)
 99   50     FORMAT(' TYPE IN THE FOLLOWING DATA AS REQUESTED.')
100          ANSWER=BLANKS
101   C
102   C INPUT DESCRIPTION OF EXPERIMENTAL APPARATUS
```

```
103  C
104          WRITE(LOT,60)
105  60      FORMAT(' PROBE #:  ')
106          READ(LID,70)NPROBE
107  70      FORMAT(A6)
```

If the operator typed OLD for the probe name, NPROBE, the program jumps to statement 450. Experimental data entered previously in FILE 37 will be read and printed. Otherwise, the subroutine CIRCOL is called, and the properties of the NPROBE coil are read from FILE 29 and returned to the main program in COMMON block B3. If NPROBE is not found in FILE 29, the user is prompted to type in the necessary coil data.

```
108          IF (NPROBE.EQ.'OLD   ') GO TO 450
109  C
110  C       THE COIL PROPERTIES ARE CALLED FROM FILE 29.
111  C
112          CALL CIRCOL(LI,LOT,NPROBE)
```

Next the operator is prompted to type in the coil serial number and data on the circuit components used with the eddy-current instrument. If LID has been assigned to a prerecorded disk partition, the data will be read from there. The operator does not need to make any response until line 271 has been reached.

```
113  C
114          WRITE(LOT,80)
115  80      FORMAT(' SERIAL #:  ')
116          READ(LID,*)NSER
117          WRITE(LOT,90)
118  90      FORMAT(' DRIVER SERIES RESISTANCE:  ')
119          READ(LID,*)RO
120          WRITE(LOT,100)
121  100     FORMAT(' DRIVER SHUNT CAP:  ')
122          READ(LID,*)CAPDR
123          WRITE(LOT,110)
124  110     FORMAT(' PICK-UP SHUNT RESISTANCE:  ')
125          READ(LID,*)R9
126          WRITE(LOT,120)
127  120     FORMAT(' PICK-UP SHUNT CAP:  ')
128          READ(LID,*)CAPPU
129          WRITE(LOT,130)
130  130     FORMAT(' CABLE I.D. #:  ')
131          READ(LID,70)NCABLE
132          WRITE(LOT,140)
133  140     FORMAT(' LENGTH OF CABLE:  ')
134          READ(LID,*)CABLEL
135          WRITE(LOT,150)
136  150     FORMAT(' CAPACITANCE OF CABLE:  ')
137          READ(LID,*)CCABLE
```

The instrument identification and operating conditions are read in.

```
138          WRITE(LOT,160)
139  160     FORMAT(' EDDY CURRENT INSTRUMENT #:  ')
140          READ(LID,70) INSTNO
141          WRITE(LOT,170)
142  170     FORMAT(' POWER OSC I.D.:  ')
143          READ(LID,70)POWOSC
144  C
145  C INPUT FREQUENCY VALUES
146  C
147          WRITE(LOT,190)
148  190     FORMAT(' NO. OF FREQUENCIES:')
149          READ(LID,*)NFT
```

```
      WRITE(LOT,200)
200   FORMAT(' INPUT THE VALUE OF EACH FREQ SEPARATED BY A SPACE:',/)
      READ(LID,*)(FREQ(NF),NF=1,NFT)
      WRITE(LOT,210)(FREQ(NF),NF=1,NFT)
210   FORMAT(' INPUT THE FOLLOWING DATA,6 CHAR WITH A SPACE BETWEEN',/,
     * ' FREQUENCY:',10X,10(1PE9.2))
      WRITE(LOT,220)
220   FORMAT(' PICK-UP AMP I.D.:     ')
      READ(LID,230)(PICKAM(NF),NF=1,NFT)
230   FORMAT(10(A6,1X))
      WRITE(LOT,240)
240   FORMAT(' PHASE DETECTOR I.D.: ')
      READ(LID,230)(PHADET(NF),NF=1,NFT)
      WRITE(LOT,250)
250   FORMAT(' 180-DEG SW(OFF/ON):    ')
      READ(LID,260)(PHASW(NF),NF=1,NFT)
260   FORMAT(10(A3,1X))
```

The tubing properties are read in.

```
C
C INPUT PROPERTY VALUES
C
270   WRITE(LOT,280)
280   FORMAT(' NUMBER OF TUBING SAMPLES: ')
      READ(LID,*) NST
      WRITE(LOT,290)
290   FORMAT(' NUMBER OF COAXIAL REGIONS:  ')
      READ(LID,*) NRT
      NRT1=NRT-1
      NRT3=NRT-3
      DO 400 NS=1,NST
      WRITE(LOT,300) NRT1,NS
300   FORMAT(' TYPE',I3,' PERMEABILITIES FOR REGIONS OF SAMPLE ',I3,/)
      READ(LID,*) (PERM(NS,NR),NR=1,NRT1)
      WRITE(LOT,310) NRT1,NS
310   FORMAT(' TYPE',I3,' RESISTIVITIES FOR REGIONS OF SAMPLE',I3,/)
      READ(LID,*) (RHO(NS,NR),NR=1,NRT1)
      WRITE(LOT,320) NRT1,NS
320   FORMAT(' TYPE',I3,' VALUES OF REGION RADII FOR SAMP # ',I3,/)
      READ(LID,*) (RAD(NS,NR),NR=1,NRT1)
```

The tubing radius next to the coil is called X and is compared with the mean coil radius RBAR to see whether the coil is encircling or inside the tube; it is then compared with the limiting radius RLIM of the coil to see whether the coil will fit the tubing (with radial clearance RADCLR). If not, the error message in FORMAT 360 will be printed.

```
      DO 330 NR=1,NRT1
      D(NS,NR)=2.*RAD(NS,NR)
      IF(NR.NE.1) TH(NS,NR)=RAD(NS,NR-1)-RAD(NS,NR)
      TH(NS,1)=99.999
  330 CONTINUE
      X=RAD(NS,NRT1)
      IF(X.LT.RBAR) GO TO 340
C     COIL INSIDE TUBE
      RADCLR(NS)=X-RLIM*RBAR
      GO TO 350
C     COIL ENCIRCLING TUBE
340   RADCLR(NS)=RLIM*RBAR-X
350   IF(RADCLR(NS).LT.0.) WRITE(LOT,360)
360   FORMAT(' ERROR: COIL WILL NOT FIT.')
```

The defect sizes and locations are read from the LID terminal.

```
202          WRITE(LOT,370) NRT1,NS
203   370    FORMAT(' NO.OF DEFCTS IN EACH OF',I3,' REGIONS OF SAMPLE',I3,3X)
204          READ(LID,*) (NDEF(NS,NR),NR=1,NRT1)
205          DO 400 NR=1,NRT1
206          NDSR=NDEF(NS,NR)
207          IF(NDSR.EQ.0) GO TO 400
208          DO 390 ND=1,NDSR
209          WRITE(LOT,380) ND,NS,NR
210   380    FORMAT(' TYPE SIZE, AXIAL DIST & RADIAL LOCATION OF DEFECT #'
211         *,I2,/,' FOR SAMPLE # ',I3, ' IN REGION ',I3,/)
212          READ(LID,*) DFSIZE(NS,NR,ND),AXILOC(NS,NR,ND),RADLOC(NS,NR,ND)
213   390    CONTINUE
214   400    CONTINUE
```

The total number of properties (including defects) is accumulated as NPT.

```
215          DO 420 NS=1,NST
216          NX=0
217          DO 410 NR=1,NRT1
218          NX=NX+NDEF(NS,NR)
219   410    CONTINUE
220          NDF(NS)=NX
221   420    CONTINUE
222          NPT=0
223          DO 440 NS=1,NST
224          NPT=NPT+1+NDF(NS)
225   440    CONTINUE
```

The program now skips to statement 690, omitting the lines from statement 450 to 690, which are used only if the operator typed OLD for the coil name NPROBE in line 106. In the latter case, data written on FILE 37 in the previous running of TUBRDG are read and printed on the LOT terminal.

```
226          GO TO 690
227   C
228   C      READ AND PRINT PREVIOUS DATA
229   C
230   450    IREC=1
231          READ(37'IREC) TITLE,IDY,ITM
232          READ(37'IREC) NPROBE,NSER,RO,CAPDR,R9,CAPPU,NCABLE,
233         *              CABLEL,CCABLE,INSTNO,POWOSC
234          READ(37'IREC) NFT,NST,NRT,NPT,NPHCAL,NMGCAL
235          NRT1=NRT-1
236          NRT3=NRT-3
237          DO 460 NS=1,NST
238          READ(37'IREC) ((PERM(NS,NR),RHO(NS,NR)),NR=1,NRT1)
239          READ(37'IREC) ((RAD(NS,NR),NDEF(NS,NR)),NR=1,NRT1),RADCLR(NS)
240          DO 460 NR=1,NRT1
241          NDSR=NDEF(NS,NR)
242          IF(NDSR.EQ.0) GO TO 460
243          READ(37'IREC)((RADLOC(NS,NR,ND),AXILOC(NS,NR,ND),
244         *DFSIZE(NS,NR,ND)),ND=1,NDSR)
245   460    CONTINUE
246          READ(37'IREC)((FREQ(NF),PICKAM(NF),PHADET(NF),PHASW(NF)),
247         *NF=1,NFT)
248          WRITE(LOT,470) IDY,ITM
249   470    FORMAT(' TUBRDG: DATE ',2(I2,1H/),I2,', TIME ',2(I2,1H:),I2)
250          WRITE(LOT,480) NPROBE,NSER
251   480    FORMAT(' PROBE NO: ',A6,5X,' SERIAL NO.:',I5)
252          WRITE(LOT,490) RO,CAPDR
253   490    FORMAT(' DRIVER SERIES RESISTANCE:',F10.1,5X,'DRIVER SHUNT',
254         *' CAP.:',E12.4)
255          WRITE(LOT,500) R9,CAPPU
256   500    FORMAT(' PICK-UP SHUNT RESISTANCE:',F10.1,5X,'PICK-UP SHUNT'
257         *,' CAP.:',E12.4)
258          WRITE(LOT,510) NCABLE,CABLEL,CCABLE
```

```
259  510   FORMAT(' CABLE I.D. NO.:',A6,5X,'LENGTH:',F10.1,5X,'CAP.:',
260       *E12.4)
261         WRITE(LOT,520) INSTNO,POWOSC
262  520   FORMAT(' EDDY CURRENT INST.NO.:',A6,' POWER OSC.I.D.:',A6)
263         WRITE(LOT,530) (FREQ(NF),NF=1,NFT)
264  530   FORMAT(1H0,10X,'FREQUENCY:',6(1PE12.4,5X))
265         WRITE(LOT,540) (PICKAM(NF),NF=1,NFT)
266  540   FORMAT(' PICK-UP AMP I.D.:',7X,10(A6,9X))
267         WRITE(LOT,550) (PHADET(NF),NF=1,NFT)
268  550   FORMAT(' PHASE DETECTOR I.D.:',4X,10(A6,9X))
269         WRITE(LOT,560) (PHASW(NF),NF=1,NFT)
270  560   FORMAT(' 180 PHASE SWITCH:',8X,10(A3,12X))
```

To limit the data displayed on the LOT terminal, the READ statement in line 271 has been inserted. After the foregoing data have been inspected, the execution of the program can be continued by typing any character and a carriage return. Then the sample properties will be read and displayed.

```
271         READ(LI,*) XYZ
272         DO 580 NS=1,NST
273         NX=0
274         DO 570 NR=1,NRT1
275         NX=NX+NDEF(NS,NR)
276  570   CONTINUE
277         NDF(NS)=NX
278  580   CONTINUE
279         NP=0
280         DO 680 NS=1,NST
281         NODF=1+NDF(NS)
282         DO 680 KD=1,NODF
283         NP=NP+1
284         WRITE(LOT,590)
285  590   FORMAT(' SAMP PROP  DIAM  THICK     RHO      PERM',
286       *'   DEFSIZ  RADLOC  AXILOC')
287         DO 680 NR=1,NRT1
288         D(NS,NR)=2.*RAD(NS,NR)
289         IF(NR.NE.1) TH(NS,NR)=RAD(NS,NR-1)-RAD(NS,NR)
290         TH(NS,1)=99.999
291         IF(KD.EQ.1) GO TO 620
292         NDSR=NDEF(NS,NR)
293         IF(NDSR.EQ.0) GO TO 620
294         DO 610 ND=1,NDSR
295         IF(ND.NE.(KD-1))GO TO 610
296         WRITE(LOT,600) NS,NP,D(NS,NR),TH(NS,NR),RHO(NS,NR),PERM(NS,NR),
297       *DFSIZE(NS,NR,ND),RADLOC(NS,NR,ND),AXILOC(NS,NR,ND)
298  600   FORMAT(1X,2(I3,2X),2(F6.3,1X),2(F8.1),3(F8.3))
299  610   CONTINUE
300         GO TO 680
301  620   WRITE(LOT,630) NS,NP,D(NS,NR),TH(NS,NR),RHO(NS,NR),PERM(NS,NR)
302  630   FORMAT(1X,2(I3,2X),2(F6.3,1X),2(F8.1))
303  680   CONTINUE
```

To ensure that the program control will return to 790 later, NSTATE is assigned the value of 790.

```
304  690   ASSIGN 790 TO NSTATE
```

The calibration readings are made by calling the CALIB subroutine, which returns the readings in the RDGC array. This array is then transferred to the OLDRDGC array, which stores the old readings for later use or updating. The term LNF is twice the number of frequencies, and NCAL is the number of phase and magnitude calibrations at each frequency.

```
305  C
306  C THIS SECTION TAKES THE CALIBRATION READINGS. CONTROL IS RETURNED
```

```
C       TO SOME SECTION OF THE PROGRAM ACCORDING TO THE CURRENT
C       VALUE OF THE VARIABLE, NSTATE (STATEMENT NO. VALUE).
C
700     ANSWER=BLANKS
        WRITE(LOT,710)
710     FORMAT(' CALIBRATION READINGS:  ')
750     CALL CALIB(RDGC,NCHS,NSTOP)
        IF(NSTOP.NE.0) GO TO 750
        LNF=2*NFT
        NCAL=NPHCAL*NMGCAL
        DO 770 NF=1,LNF
        DO 770 NC=1,NCAL
        OLRDGC(NF,NC)=RDGC(NF,NC)
770     CONTINUE
        GO TO NSTATE
C
C END OF SECTION WHICH TAKES CALIBRATION READINGS
```

Various arrays are initialized.

```
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
790     MSET=0
        DO 800 NF=1,NFT
        DO 800 NP=1,NPT
        SUMMAG(NP,NF)=0.
        SUMPHA(NP,NF)=0.
        SSMAG(NP,NF)=0.
        SSPHA(NP,NF)=0.
800     CONTINUE
        DO 810 NF=1,LNF
        DO 810 NC=1,NCAL
        SUMCAL(NF,NC)=0.
        SSCAL(NF,NC)=0.
810     CONTINUE
```

The operator is asked whether he or she is ready to take readings. If N is typed, control will switch to statement 1140, which will set NPRINT = 0 and then offer various options.

```
        WRITE(LOT,820)
820     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
```

If the answer was not N, the program will proceed to statement 830, where the data-taking loops are started. Then the operator will be prompted to set the coil on the various samples, and the data will be read into the computer via the MODAC when the subroutine READAT is called.

```
C
C THIS SECTION TAKES THE ACTUAL PHASE & MAGNITUDE DATA READINGS
C
830     ANSWER=BLANKS
        NCOUNT=0
        NRET=0
        NP=0
        DO 980 NS=1,NST
        NODF=1+NDF(NS)
        DO 980 KD=1,NODF
        NP=NP+1
        IF(NCOUNT.NE.0) GO TO 850
```

```
359  C
360  C GO CLEAR TERMINAL SCREEN & RETURN HERE
361  C
362          NCOUNT=1
363          ASSIGN 850 TO ISTATE
364          GO TO 1800
365  C
366  850     WRITE(LOT,860)
367  860     FORMAT(' SAMP PROP  DIAM  THICK     RHO      PERM',
368         *'   DEFSIZ  RADLOC  AXILOC')
369          DO 910 NR=1,NRT1
370          IF(KD.EQ.1) GO TO 890
371          NDSR=NDEF(NS,NR)
372          IF(NDSR.EQ.0) GO TO 890
373          DO 880 ND=1,NDSR
374          IF(ND.NE.(KD-1))GO TO 880
375          WRITE(LOT,870) NS,NP,D(NS,NR),TH(NS,NR),RHO(NS,NR),PERM(NS,NR),
376         *DFSIZE(NS,NR,ND),RADLOC(NS,NR,ND),AXILOC(NS,NR,ND)
377  870     FORMAT(1X,I3,2X,I3,2(1X,F6.3),1PE11.3,0PF8.1,3(F8.3))
378  880     CONTINUE
379          GO TO 910
380  890     WRITE(LOT,900) NS,NP,D(NS,NR),TH(NS,NR),RHO(NS,NR),PERM(NS,NR)
381  900     FORMAT(1X,I3,2X,I3,1X,2(F6.3),1PE11.3,0PF8.1)
382  910     CONTINUE
383          MAXCH=2*(NFT-1)+1
384          WRITE(LOT,920)(BLANKS,II,II, II=1,NFT)
385  920     FORMAT(3X,10(A1,'MAG(',I1,')',4X,' PH(',I1,')',5X))
386          WRITE(LOT,30)
```

Because of line 390, READAT is called repeatedly, as long as the foot switch is not pressed (i.e., NSTOP = 0). Each time the last index in the call of READAT is zero; therefore, this is the value of MAG in the subroutine and means that the saturating magnet will be turned off.

```
387  930     CALL READAT(VOLTS,NCHS,ITIMS,NRET,NSTOP,0)
388          WRITE(LOT,950)((VOLTS(JJ),VOLTS(JJ+1)),JJ=1,MAXCH,2)
389  950     FORMAT(1H+,10(F9.4,2X))
390          IF(NSTOP.GT.0) GO TO 930
```

When the foot switch is pressed, NSTOP is greater than zero, and READAT is again called. If the value of MAG was set at 1 in the initial DATA statement of the main program, the readings are first made 1024 times, giving a delay of 0.210 sec to allow the readings to settle down. The desired readings are made when READAT is called again in line 392. The saturating magnet is then turned off by calling the SATMAG subroutine, with the value of MAG set equal to 0.

```
391          IF(MAG.EQ.1)CALL READAT(VOLTS,NCHS,1024,NRET,NSTOP,MAG)
392          CALL READAT(VOLTS,NCHS,ITIMS,NRET,NSTOP,MAG)
393          CALL SATMAG(0)
```

The array of readings called VOLTS is returned from the READAT subroutine and then separated into the phase and magnitude arrays called PHASE and TMAG respectively.

```
394          VOLTS(8)=25.*VOLTS(7)/VOLTS(8)
395          NCH2=NCHS-2
396          DO 960 NC=1,NCH2
397          VOLTS(NC)=VOLTS(NC)+TOFSET(NC)+TSLOPE(NC)*VOLTS(8)
398  960     CONTINUE
399          WRITE(LOT,950)((VOLTS(JJ),VOLTS(JJ+1)),JJ=1,MAXCH,2)
400          NRET=0
401          DO 980 NF=1,NFT
402          TMAG(NP,NF)=VOLTS(2*NF-1)
403          PHASE(NP,NF)=VOLTS(2*NF)
404  980     CONTINUE
```

The number of sets of readings, MSET, is incremented, NSTATE is changed to 1000, and control is shifted back to statement 700 to make another set of calibration readings, after which control will return to statement 1000.

```
405          MSET=MSET+1
406          ASSIGN 1000 TO NSTATE
407          GO TO 700
408    C
409    C END OF A SET OF READINGS. ASK USER IF HE WANTS TO DELETE THIS SET.
```

After return from the calibration procedure, the set of options in FORMAT 1010 is offered.

```
410    C
411    1000  WRITE(LOT,1010)MSET
412    1010  FORMAT(1H ,I5,' SET(S) OF READINGS HAVE BEEN MADE.'/
413         *' DO YOU WANT TO DELETE THE PREVIOUS SET OF READINGS? ',
414         *'TYPE Y OR N:  ')
415          ANSWER=BLANKS
416          READ(LI,1020)ANSWER
417    1020  FORMAT(A1)
```

If the answer was N (not to delete the new set of readings), the new sets of readings will be added to the previous ones – OLRDGC, SUMPHA, and SUMMAG for the calibration, phase, and magnitude readings respectively. If the answer was anything else, the value of MSET will be decremented, and the options in FORMAT 1150 will be offered.

```
418          IF(ANSWER.EQ.'N   ') GO TO 1030
419          MSET=MSET-1
420          GO TO 1140
421    C
422    C ADD NEW CURRENT SET OF READINGS TO SUM OF PREVIOUS SETS
423    C
424    1030  ANSWER=BLANKS
425    1050  DO 1080 NF=1,LNF
426          DO 1080 NC=1,NCAL
427          SUMCAL(NF,NC)=SUMCAL(NF,NC)+OLRDGC(NF,NC)
428          SSCAL(NF,NC)=SSCAL(NF,NC)+OLRDGC(NF,NC)*OLRDGC(NF,NC)
429    1080  CONTINUE
430          ANSWER=BLANKS
431    1100  DO 1130 NF=1,NFT
432          DO 1130 NP=1,NPT
433          SUMMAG(NP,NF)=SUMMAG(NP,NF)+TMAG(NP,NF)
434          SUMPHA(NP,NF)=SUMPHA(NP,NF)+PHASE(NP,NF)
435          SSMAG(NP,NF)=SSMAG(NP,NF)+TMAG(NP,NF)*TMAG(NP,NF)
436          SSPHA(NP,NF)=SSPHA(NP,NF)+PHASE(NP,NF)*PHASE(NP,NF)
437    1130  CONTINUE
438    C
439    C PROMPT USER ABOUT WHAT HE WANTS TO DO NEXT
440    C
441    1140  ANSWER=BLANKS
442          WRITE(LOT,1150)
443    1150  FORMAT(' DO YOU WANT TO (1) RE-DO THE CALIBRATIONS, (2) MAKE',/,
444         *' ANOTHER SET OF READINGS, OR (3) STOP? TYPE 1,2,OR 3:  ')
445          READ(LI,*)NEXT
```

The three options return control to statements 700, 830, and 1170 respectively. If any number other than 1, 2, or 3 was entered, the error message in FORMAT 1160 will be printed, and the options will be offered again.

```
446          GO TO (700,830,1170),NEXT
447          WRITE(LOT,1160)
```

```
1160  FORMAT(' ERROR ON INPUT. ENTER AGAIN.')
      GO TO 1140
```

If the third option, STOP, was chosen, the averages and standard deviations of the calibration, phase, and magnitude readings will be calculated as SUMCAL, SDVCAL, SUMPHA, SDVPHA, SUMMAG, and SDVMAG respectively.

```
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 '/
     *' STOP WITHOUT SAVING OR PRINTING ANYTHING.')
      GO TO 2000
1190  DO 1210 NF=1,LNF
      DO 1210 NC=1,NCAL
      SUMCAL(NF,NC)=SUMCAL(NF,NC)+RDGC(NF,NC)
      SSCAL(NF,NC)=SSCAL(NF,NC)+RDGC(NF,NC)*RDGC(NF,NC)
1210  CONTINUE
      DO 1250 NF=1,LNF
      DO 1250 NC=1,NCAL
      SUMCAL(NF,NC)=SUMCAL(NF,NC)/(FLOAT(MSET+1))
      SDVCAL(NF,NC)=SQRT(ABS((SSCAL(NF,NC)-SUMCAL(NF,NC)*SUMCAL(NF,NC)
     *                  *FLOAT(MSET+1))/FLOAT(MSET)))
1250  CONTINUE
C
C CALCULATE AVERAGES & STANDARD DEVIATIONS OF READINGS
C
      DO 1290 NF=1,NFT
      DO 1290 NP=1,NPT
      SUMMAG(NP,NF)=SUMMAG(NP,NF)/(FLOAT(MSET))
      SUMPHA(NP,NF)=SUMPHA(NP,NF)/(FLOAT(MSET))
      SDVMAG(NP,NF)=SQRT(ABS((SSMAG(NP,NF)-SUMMAG(NP,NF)
     *        *SUMMAG(NP,NF)*FLOAT(MSET))/FLOAT(MSET-1)))
      SDVPHA(NP,NF)=SQRT(ABS((SSPHA(NP,NF)-SUMPHA(NP,NF)
     *        *SUMPHA(NP,NF)*FLOAT(MSET))/FLOAT(MSET-1)))
1290  CONTINUE
```

The data are now written on RAD file 37.

```
C
C STORE ALL INFORMATION IN DIRECT ACCESS FILE #37 ON DISK
C
      IREC=1
      WRITE(37'IREC) TITLE,IDAY,ITIM
      WRITE(37'IREC) NPROBE,NSER,RO,CAPDR,R9,CAPPU,NCABLE,
     *               CABLEL,CCABLE,INSTNO,POWOSC
      WRITE(37'IREC) NFT,NST,NRT,NPT,NPHCAL,NMGCAL
      DO 1300 NS=1,NST
      WRITE(37'IREC) ((PERM(NS,NR),RHO(NS,NR)),NR=1,NRT1)
      WRITE(37'IREC) ((RAD(NS,NR),NDEF(NS,NR)),NR=1,NRT1),RADCLR(NS)
      DO 1300 NR=1,NRT1
      NDSR=NDEF(NS,NR)
      IF(NDSR.EQ.0) GO TO 1300
      WRITE(37'IREC)((RADLOC(NS,NR,ND),AXILOC(NS,NR,ND),
     *DFSIZE(NS,NR,ND)),ND=1,NDSR)
1300  CONTINUE
      WRITE(37'IREC)((FREQ(NF),PICKAM(NF),PHADET(NF),PHASW(NF)),
     *NF=1,NFT)
      DO 1310 NF=1,LNF
      WRITE(37'IREC)((SUMCAL(NF,J),SDVCAL(NF,J)),J=1,NCAL)
1310  CONTINUE
      DO 1320 NP=1,NPT
      WRITE(37'IREC)((SUMMAG(NP,NF),SUMPHA(NP,NF),SDVMAG(NP,NF),
```

```
      *SDVPHA(NP,NF)),NF=1,NFT)
 1320  CONTINUE
       IF(IREC.GE.NFILIM) WRITE(LOT,1330)
 1330  FORMAT(' LIMIT OF FILE 37 IS EXCEEDED.')
 C
 C END OF SECTION WHICH WRITES DIRECT ACCESS FILE
```

The data are also written on the LPT line printer.

```
 C PRINT SUMMARY OF JOB STATISTICS ON LPT
 C
       WRITE(LPT,1335)
 1335  FORMAT(1H1,20X)
       WRITE(LPT,1340) IDAY,ITIM
 1340  FORMAT(' TUBRDG: DATE ',2(I2,1H/),I2,', TIME ',2(I2,1H:),I2)
       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.:',
      *        E12.4)
       WRITE(LPT,1370) R9,CAPPU
 1370  FORMAT(' PICK-UP SHUNT RESISTANCE:',F10.1,5X,'PICK-UP SHUNT CAP.:'
      *         ,E12.4)
       WRITE(LPT,1380) NCABLE,CABLEL,CCABLE
 1380  FORMAT(' CABLE I.D. NO.:',A6,5X,'LENGTH:',F10.1,5X,'CAP.:',
      *         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(NF),NF=1,NFT)
 1410  FORMAT(/,10X,'FREQUENCY:',10(1PE12.4,5X))
       WRITE(LPT,1420) (PICKAM(NF),NF=1,NFT)
 1420  FORMAT(' PICK-UP AMP I.D.:',7X,10(A6,9X))
       WRITE(LPT,1430) (PHADET(NF),NF=1,NFT)
 1430  FORMAT(' PHASE DETECTOR I.D.:',4X,10(A6,9X))
       WRITE(LPT,1440) (PHASW(NF),NF=1,NFT)
 1440  FORMAT(' 180  PHASE SWITCH:',8X,10(A3,12X))
       NP=0
       DO 1600 NS=1,NST
       NODF=1+NDF(NS)
       DO 1550 KD=1,NODF
       NP=NP+1
       WRITE(LPT,1450)
 1450  FORMAT(1X,/,' SAMP PROF REG  DIAM  THICK    RHO       PERM',
      *'   DEFSIZ  RADLOC  AXILOC')
       DO 1500 NR=1,NRT1
       IF(KD.EQ.1) GO TO 1490
       NDSR=NDEF(NS,NR)
       IF(NDSR.EQ.0) GO TO 1490
       DO 1480 ND=1,NDSR
       IF(ND.NE.(KD-1))GO TO 1480
       WRITE(LPT,1470) NS,NP,NR,D(NS,NR),TH(NS,NR),RHO(NS,NR),
      *PERM(NS,NR),DFSIZE(NS,NR,ND),RADLOC(NS,NR,ND),AXILOC(NS,NR,ND)
 1470  FORMAT(1X,I3,2X,I3,2X,I2,2(1X,F6.3),1PE11.3,0PF8.1,3(F8.3))
 1480  CONTINUE
       GO TO 1500
 1490  WRITE(LPT,1495) NS,NP,NR,D(NS,NR),TH(NS,NR),RHO(NS,NR),
      *PERM(NS,NR)
 1495  FORMAT(1X,I3,2X,I3,2X,I2,1X,2(F6.3),1PE11.3,0PF8.1)
 1500  CONTINUE
       WRITE(LPT,1510) ((BLANKS),NF=1,NFT)
 1510  FORMAT(' AV.READINGS',4X,4(A1,'MAG',7X,'PHA',6X))
       WRITE(LPT,1520) ((SUMMAG(NP,NF),SUMPHA(NP,NF)),NF=1,NFT)
 1520  FORMAT(12X,4(F10.5,F10.5))
       WRITE(LPT,1530) ((SDVMAG(NP,NF),SDVPHA(NP,NF)),NF=1,NFT)
 1530  FORMAT(2X,'STD.DEV.',2X,4(F10.5,F10.5))
 1550  CONTINUE
```

```
1600  CONTINUE
      WRITE(LPT,1650) NPHCAL,NMGCAL
1650  FORMAT(1H0,' AVERAGES & STANDARD DEVIATIONS OF CALIBRATION ',
     *      'READINGS:',I3,' MAG',I3,' PHA'/)
      WRITE(LPT,1660) (FREQ(NF),NF=1,NFT)
1660  FORMAT(3(10X,1PE12.4))
      WRITE(LPT,1670)((BLANKS),NF=1,NFT)
1670  FORMAT(8X,3(A1,'MAG',8X,'PHA',7X))
      DO 1700 NC=1,NCAL
      WRITE(LPT,30)
      WRITE(LPT,1680)((BLANKS,SUMCAL(NF,NC)),NF=1,LNF)
1680  FORMAT(5X,6(A1,F10.5))
      WRITE(LPT,1690)((BLANKS,SDVCAL(NF,NC)),NF=1,LNF)
1690  FORMAT(' S.D.',6(A1,F10.5))
1700  CONTINUE
      GO TO 2000
C
C END OF SECTION WHICH WRITES SUMMARY REPORT ON LPT
C
```

The section starting with statement 1800 contains the assembler code to clear the screen of the LOT terminal.

```
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
```

The program is terminated.

```
2000  STOP JOB
      END
```

A sample printout is given below.

```
TUDRDG: DATE  5/ 9/79, TIME 11:15:29
PROBE NO.:2V169S       SERIAL NO.:     1
DRIVER SERIES RESISTANCE: 1000000.0      DRIVER SHUNT CAP.:  0.0000E+00
PICK-UP SHUNT RESISTANCE:  333333.0      PICK-UP SHUNT CAP.:  0.0000E+00
CABLE I.D. NO.:1            LENGTH:       72.0      CAP.:  0.1800E-09
EDDY CURRENT INSTRUMENT NO.:1503
POWER OSC I.D.151-86

          FREQUENCY:  1.0000E+03        5.0000E+03        2.0000E+04
PICK-UP AMP I.D.:        15A-83          15A-84            15A-85
PHASE DETECTOR I.D.:     15A-79          15A-77            15A-78
180  PHASE SWITCH:         ON                                N

SAMP PROP REG  DIAM  THICK    RHO        PERM   DEFSIZ  RADLOC  AXILOC
  1    1   1  2.00099.999  1.000E+20     1.0
  1    1   2  0.626 0.687  1.000E+20     1.0
  1    1   3  0.405 0.110  3.500E+01    10.0
AV.READINGS      MAG        PHA        MAG        PHA        MAG        PHA
               3.31137    2.52167    3.04149    3.44233    3.38340    3.14837
```

```
STD.DEV.        0.00276   0.00338   0.00391   0.00000   0.00478   0.00000

SAMP PROP REG  DIAM  THICK    RHO        PERM   DEFSIZ  RADLOC  AXILOC
 1    2    1  2.00099.999  1.000E+20     1.0
 1    2    2  0.626 0.687  1.000E+20     1.0
 1    2    3  0.405  0.110  3.500E+01    10.0    0.065   0.232   0.000
AV.READINGS      MAG       PHA       MAG       PHA       MAG       PHA
               3.30634   2.52827   3.04149   3.45555   3.39298   3.16259
STD.DEV.       0.00478   0.00756   0.00873   0.00000   0.00677   0.00731

                                    .
                                    .
                                    .

SAMP PROP REG  DIAM  THICK    RHO        PERM   DEFSIZ  RADLOC  AXILOC
 56  168   1  0.72799.999  3.500E+01   200.0
 56  168   2  0.626 0.051  1.000E+20     1.0
 56  168   3  0.443  0.091  3.500E+01     5.0    0.046   0.290   0.000
AV.READINGS      MAG       PHA       MAG       PHA       MAG       PHA
               3.23099   2.27908   2.76510   3.33882   3.11263   3.18757
STD.DEV.       0.00276   0.00338   0.00338   0.00000   0.00276   0.00276

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

          1.0000E+03          5.0000E+03          2.0000E+04
          MAG       PHA       MAG       PHA       MAG       PHA

          3.90004   3.00526   3.97349   2.99976   3.92541   3.00801
S.D.      0.00391   0.00338   0.00391   0.00391   0.00000   0.00338

          1.98966   3.00308   2.02621   2.99654   1.99131   3.01149
S.D.      0.00138   0.00338   0.00138   0.00276   0.00138   0.00195

          3.88414   1.03392   3.96428   1.02506   3.93591   1.05350
S.D.      0.00000   0.00098   0.00276   0.00000   0.00391   0.00138

          1.98234   1.03878   2.02274   1.02811   1.99684   1.05597
S.D.      0.00138   0.00138   0.00195   0.00120   0.00195   0.00169
```

## Reference

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

## Appendix B
## THE TUBFIT PROGRAM

The TUBFIT program is written in FORTRAN IV, which is compatible with the FR4 compiler in a ModComp MAX IV system processor.[1] It reads data from RAD file 37, which was recorded when the TUBRDG program was run. Before running TUBFIT, three preparatory steps are required: the file should be allocated, the TUBRDG program should be run, and the program should be called up in the edit mode and cataloged under another name, such as NEWFIT, so that the newly named program can then be edited and run without endangering the original TUBFIT program.

It is most important that the dimensions of the arrays be correctly specified in lines 65–79 and that the arrays shared with TUBRDG should have identical dimensions. The commented dimension statements in lines 44–60 indicate the necessary dimensions in terms of the quantities defined in lines 7–36. Actually, the use of dimensions larger than necessary would not cause trouble if the computer capacity is not exceeded, *except for the first dimensions of READNG,* TMAGM, and PHASEM; these must be exactly as specified in lines 48 and 53. The term NPT is equal to the total number of properties (including defects). The user must also ensure that RAD files 21 and 31 have enough space to store the data to be recorded when the program is run. RAD file 29 must also be set up. However, if it does not contain the necessary coil data, the user will be prompted to type in whatever coil properties are needed; these data are used only for reference purposes and are not used in any of the calculations in this program.

The data in lines 106–110 may need to be changed. The terms LITEK and LOTEK are the logical input and output units, respectively, for the terminal at which the operator works; LI is the logical input unit for typing in responses to the computer questions, and LOU is the hard-copy output line printer. The other parameter that may need changing is NUNIT, which selects the units to be used for the computer printout. If NUNIT = 1 the output lengths will be in inches, and the defect volume will be in cubic inches. If NUNIT = 2 the lengths will be printed out in millimeters and the volumes in cubic millimeters. In either case, the resistivities will be in microhm-cm, and the permeabilities will be relative. The NFILIM value should be equal to the first number in the dimension statement of FILE 37.

A listing and step-by-step analysis of the program follows. Various variables are defined in lines 7–36.

```
 1          PROGRAM TUBFIT
 2   C      (1 MAY 1979)
 3   C      PROGRAM TO PERFORM A LEAST SQUARES FIT TO DATA READ INTO A DISK
 4   C      FILE BY TUBRDG PROGRAM AND THEN PERFORM CONTINOUS BACK CALCULATIONS
 5   C      USING INSTRUMENT READINGS THAT ARE MADE BETWEEN EACH DISPLAYED SET.
 6   C
 7   C      AXILOC=DISTANCE OF DEFECT ALONG AXIS
 8   C      DFSIZE=VOLUME OF DEFECT
 9   C      IRDFRM=MAXIMUM NUMBER OF COEFFICIENTS IN EXPANSION
10   C      LITEK=LOGICAL INPUT UNIT (1=TTY)
11   C      LOTEK=LSI OUTPUT UNIT FOR PROMPTING AND DISPLAY
12   C      LOU=LOGICAL OUTPUT UNIT FOR PERMANENT RECORD
13   C      MNDF=MAXIMUM NUMBER OF DEFECTS IN ANY REGION
14   C      NCHS=NUMBER OF DATA CHANNELS
15   C      NCOED=FILE VARIABLE FOR MIC-COMP DATA=1ST CALO DATA BYTE HEX OR DEC
16   C      NFILIM=FILE LIMIT ON LINES IN FILE 37, THE DATA FILE
17   C      NF=FREQUENCY INDEX
18   C      NFT=NUMBER OF FREQUENCIES
19   C      LNF=2*NFT
20   C      NP=PROPERTY INDEX
21   C      NPROPM=MAXIMUM NUMBER OF PROPERTIES CALCULATED(=6 NOW)
22   C      NPT=TOTAL NUMBER OF PROPERTIES + NUMBER OF DEFECTS
23   C      NPTT=VALUE OF NPT READ FROM FILE 30
24   C      NPHCAL=NUMBER OF PHASE CALIBRATIONS
25   C      NMGCAL=NUMBER OF MAGNITUDE CALIBRATIONS
26   C      NCAL=NPHCAL*NMGCAL=TOTAL NUMBER OF CALIBRATION READINGS
```

```
27  C       NPRINT=PRINT AND TRANSFER INDEX.
28  C       NR=REGION INDEX(1=FARTHEST FROM COIL;NRT=COIL REGION)
29  C       NRT=NUMBER OF COAXIAL REGIONS,INCLUDING AIR REGION NEXT TO COIL.
30  C       NRT1=NRT-1,NUMBER OF REGIONS EXCLUDING AIR REGION NEXT TO COIL.
31  C       NS=SAMPLE INDEX
32  C       NST=NUMBER OF TUBING SAMPLES
33  C       NSTOP=INDEX TO STOP THE INSTRUMENT READINGS
34  C       MAG=INDEX TO TURN ON THE SATURATING MAGNET(=0 IF NONE USED)
35  C       NUNIT=INDEX OF LENGTH UNITS(1=INCHES;2=MILLIMETERS)
36  C       RADLOC=RADIAL DISTANCE OF DEFECT FROM SURFACE OF CONDUCTOR
```

The dimensions of various arrays are given in lines 39–41 for arrays of fixed dimensions and symbolically in lines 44–60 for arrays whose dimensions may need to be changed. The actual dimensions for the latter arrays must be entered in lines 65–79.

```
37  C
38  C   DIMENSIONS THAT ARE NOT CHANGED:
39          DIMENSION ITIM(3),IDAY(3),PROPTY(6),PRONAM(6)
40          DIMENSION VOLTS(8),OFSET(6),CGAIN(6),STDV(6)
41          DIMENSION CONVRT(2),NCONV(4),UNITS(6,2),UN(6)
42  C
43  C   DIMENSIONS THAT ARE CHANGED:
44  C       DIMENSION RDGC(NCHS,NCAL),RDGO(NCHS,NCAL)
45  C       DIMENSION POLARY(IRDPRM+1,5),PRO(NPT),PROP(NPT,NPROPM)
46  C       DIMENSION TMDFT(NFT),PHDFT(NFT),JPOL(6,NFT,NPROPM)
47  C       DIMENSION COE(IRDPRM),COEF(IRDPRM,NPROPM)
48  C *** DIMENSION READNG(NPT+1,IRDPRM+1),NPOL(6,NFT)
49  C       DIMENSION JOFSET(NPROPM),JRDPR(NPROPM)
50  C       DIMENSION RDCL(NRT,NPT),RHO(NRT,NPT),U(NRT,NPT),TH(NRT,NPT)
51  C       DIMENSION RLOC(NRT,NPT),ZLOC(NRT,NPT),DSIZ(NRT,NPT)
52  C       DIMENSION NDF(NST),FREQ(NFT),GAIN(NFT)
53  C *** DIMENSION TMAGM(NPT,NFT),PHASEM(NPT,NFT)
54  C       DIMENSION SDVMAG(NPT,NFT),SDVPHA(NPT,NFT)
55  C       DIMENSION PICKAM(NFT),PHADET(NFT),PHASW(NFT)
56  C       DIMENSION RAD(NST,NRT1),RADCLR(NST),NDEF(NST,NRT1)
57  C       DIMENSION XRHO(NST,NRT1),PERM(NST,NRT1),DFSIZE(NST,NRT1,MNDF)
58  C       DIMENSION RADLOC(NST,NRT1,MNDF),AXILOC(NST,NRT1,MNDF)
59  C       DIMENSION SUMCAL(NCHS,NCAL),SDVCAL(NCHS,NCAL)
60  C       DIMENSION TOFSET(NCHS),TSLOPE(NCHS)
61  C
62  C   THE APPROPRIATE NUMBERS SHOULD BE INSERTED IN THE FOLLOWING
63  C   DIMENSION STATEMENTS;  COMMENTED DIMENSIONS MARKED *** MANDATORY:
64  C
65          DIMENSION RDGC(6,4),RDGO(6,4)
66          DIMENSION POLARY(16,5),PRO(168),PROP(168,6)
67          DIMENSION TMDFT(3),PHDFT(3),JPOL(6,3,6),COE(15),COEF(15,6)
68          DIMENSION READNG(169,16),NPOL(6,3),JOFSET(6),JRDPR(6)
69          DIMENSION RDCL(4,168),RHO(4,168),U(4,168),TH(4,168)
70          DIMENSION RLOC(4,168),ZLOC(4,168),DSIZ(4,168)
71          DIMENSION NDF(168),FREQ(4),GAIN(4)
72          DIMENSION TMAGM(168,3),PHASEM(168,3)
73          DIMENSION SDVMAG(168,3),SDVPHA(168,3)
74          DIMENSION PICKAM(4),PHADET(4),PHASW(4)
75          DIMENSION RAD(56,7),RADCLR(56),NDEF(56,7)
76          DIMENSION XRHO(56,7),PERM(56,7),DFSIZE(56,7,4)
77          DIMENSION RADLOC(56,7,4),AXILOC(56,7,4)
78          DIMENSION SUMCAL(6,4),SDVCAL(6,4)
79          DIMENSION TOFSET(6),TSLOPE(6)
```

Various special variables and the common blocks are declared in lines 81–85.

```
80  C
81          REAL L2,L4,L3,L5,L6
82          REAL*6 NPROBE,NCABLE,INSTNO,POWOSC,PICKAM,PHADET
83          REAL*8 TITLE(3)
84          COMMON/B3/RBAR,R1,R2,XL,ZL,RLIM,R6,R7,TNDR,TNPU,ZLDR,ZLPU,X
85          COMMON/B6/ L2,L5,L6
```

The RAD files are defined in lines 87–97.

```
 86  C
 87  C      FILE 29 CONTAINS THE COIL DATA.
 88         DEFINE FILE 29(80,32,U,NCOIL)
 89  C
 90  C      FILE 37 CONTAINS THE EXPERIMENTAL MEASUREMENTS MADE USING TUBRDG
 91         DEFINE FILE 37(400,40,U,IREC)
 92  C
 93  C      FILE 21 WILL CONTAIN THE COEFFICIENTS OF THE FITS.
 94         DEFINE FILE 21(30,100,U,ICOEF)
 95  C
 96  C      FILE 31 WILL STORE THE COEFFICIENT DATA FOR THE MICROCOMPUTER.
 97         DEFINE FILE 31(8192,4,U,NCOED)
```

Initial DATA values are entered in lines 100–121. Line 100 contains conversion constants, and lines 101–103 contain printout words; these will not ordinarily need to be changed. Lines 106–108 contain constants used in the computer program and defined in lines 7–36.

```
 98  C
 99  C  DATA THAT DO NOT NEED TO BE CHANGED:
100         DATA TWOPI/6.28318531/,RADN/57.2957795/,CONVRT/1.,25.4/,NLABL/1/
101         DATA PROPTY/4HREST,4HPERM,4HTHIK,4HRDCL,4HDFLC,4HDFSI/
102         DATA UNITS/4HMOCM,4HREL.,4HINCH,4HINCH,4HINCH,4HCUIN,
103        *            4HMOCM,4HREL.,4H MM ,4H MM ,4H MM ,4HCUMM/
104  C
105  C  DATA THAT MAY NEED TO BE CHANGED:
106         DATA IRDPRM/15/,IR/1/,NCHS/6/,ITIMS/1024/,NPROPM/6/,NPROPT/1/
107         DATA NPT/168/,NPRINT/0/,LOU/6/,LITEK/1/,LOTEK/5/,NFILIM/400/
108         DATA STOP/'STOP'/,BLANKS/'    '/,NUNIT/1/,NCOED/2000/,ICOEF/1/
109         DATA TOFSET/2.27,-1.4992,1.01,-1.32,.47935,-.70/
110         DATA TSLOPE/-.1575,.0906,-.0646,.1034,-.0285,.0596/
```

Lines 118 and 119 contain NRT*NPT initial material properties, some of which will be changed when the program is run. The NPT values of GAIN should be entered in line 120; NCHS initial values of calibration gains and offsets should be entered in line 121; they will also be modified as the program is run. The MAG value should be 0 if a saturating magnet is not used; 1 if it is.

```
111  C
112  C      THE MATERIAL PROPERTIES NOW FOLLOW,STARTING WITH THE FIRST
113  C      PROPERTY SET FOR THE LOWERMOST REGION. (RHO=RESISTIVITY IN
114  C      MICROHM-CM;U=RELATIVE PERMEABILITY;TH=THICKNESS IN INCHES)
115  C      SOME OF THESE VALUES ARE CHANGED TO THE VALUES IN THE DATA SET,
116  C      READ FROM FILE 30.
117  C
118         DATA RHO/672*1.E12/,U/672*1./,TH/672*1.E10/
119         DATA RDCL/672*.002/,DSIZ/672*0./,RLOC/672*5.E9/,ZLOC/672*0./
120         DATA GAIN/5.8025E3,3.1094E2,2*3.E3/,MAG/1/
121         DATA OFSET/6*0.0/,CGAIN/6*1./
```

The date and time are read and printed at the LOU terminal.

```
122         CALL TIME(ITIM)
123         CALL DATE(IDAY)
124         IT=IDAY(1)-1900
125         IDAY(1)=IDAY(2)
126         IDAY(2)=IDAY(3)
127         IDAY(3)=IT
128         WRITE (LOU,10)
129      10 FORMAT (1H1,20X)
130         FIND (37'IREC)
131         WRITE (LOU,20) IDAY, ITIM
132   20    FORMAT (10X,'PROGRAM TUBFIT DATE  ', 2(I2,1H/), I2,
133        1  ' TIME  ', 2(I2,1H:), I2)
```

The coefficient data file index, NCOED, is incremented, and the data from FILE 37 are read. If the number of lines read exceeds the preset limit, NFILIM, which should be the same as the first dimension of FILE 37, subsequent calculations will probably contain erroneous data. Therefore, control jumps to statement 855 to print an error message and to produce a choice of options, allowing the operator to stop the program and check the input data and dimensions.

```
      NCOED=NCOED+1
C
C     NCOED INCREMENTED(FROM ADDRESSES START AT 0,TURBFIT ADDRS AT 1)
C  READ INFORMATION IN DIRECT ACCESS FILE #37 ON DTSK
C
      IREC=1
      READ(37'IREC) TITLE,IDAY,ITIM
      READ(37'IREC) NPROBE,NSER,RO,CAPDR,R9,CAPPU,NCABLE,
     *              CABLEL,CCABLE,INSTNO,POWOSC
      READ(37'IREC) NFT,NST,NRT,NFTT,NPHCAL,NMGCAL
      NRT1=NRT-1
      NRT3=NRT-3
      DO 30 NS=1,NST
      READ(37'IREC) ((PERM(NS,NR),XRHO(NS,NR)),NR=1,NRT1)
      READ(37'IREC) ((RAD(NS,NR),NDEF(NS,NR)),NR=1,NRT1),RADCLR(NS)
      DO 30 NR=1,NRT1
      NDSR=NDEF(NS,NR)
      IF(NDSR.EQ.0) GO TO 30
      READ(37'IREC)((RADLOC(NS,NR,ND),AXILOC(NS,NR,ND),
     *DFSIZE(NS,NR,ND)),ND=1,NDSR)
   30 CONTINUE
      READ(37'IREC) ((FREQ(NF),PICKAM(NF),PHADET(NF),PHASW(NF)),
     *NF=1,NFT)
      LNF=2*NFT
      NCAL=NPHCAL*NMGCAL
      DO 40 NF=1,LNF
      READ(37'IREC) ((RDGO(NF,NC),SDVCAL(NF,NC)),NC=1,NCAL)
   40 CONTINUE
      DO 50 NP=1,NFTT
      READ(37'IREC) ((TMAGM(NP,NF),PHASEM(NP,NF),SDVMAG(NP,NF),
     *SDVPHA(NP,NF)),NF=1,NFT)
   50 CONTINUE
      IF(IREC.GE.NFILIM) GO TO 855
```

The CIRCOL subroutine is called to read the data from FILE 29 for the coil named NPROBE or to prompt the user to type in the coil data (see Appendix C for CIRCOL listing).

```
C
C        THE INPUT DATA FOR THE COIL PARAMETERS ARE READ FROM FILE 29
C        USING THE COIL NAME READ FROM FILE 30.SEE FIG.2,P.4
C        AND FIG.4, P.7, ORNL-TM-4107, FOR DEFINITIONS.
C
      CALL CIRCOL(LITEK,LOTEK,NPROBE)
```

The total number of defects, NDF(NS), in the NRT1 regions of the NS sample is found.

```
C
C     RESISTIVITIES, THICKNESSES AND DEFECTS ARE SET FROM THE
C     DATA READ FROM FILE #37. CONVERSION FACTOR SET.
C
      DO 70 NS=1,NST
      NX=0
      DO 60 NR=1,NRT1
      NX=NX+NDEF(NS,NR)
   60 CONTINUE
      NDF(NS)=NX
   70 CONTINUE
```

The conversion factor, CVT, for the units is set, and the property index, NP, is initialized.

```
184          CVT=CONVRT(NUNIT)
185          NP=0
```

The properties in the NR regions of the NP set of properties are set with data read from FILE 37. All other properties not specifically overwritten at this time will remain as initialized in the DATA statements in lines 118 and 119.

```
186   C
187   C       LOOP OVER ALL NST TUBING SAMPLES
188   C
189          DO 100 NS=1,NST
190          NODF=1+NDF(NS)
191          DO 100 KD=1,NODF
192          NP=NP+1
193   C
194   C      LOOP OVER THE NRT1 COAXIAL REGIONS.
195   C      SET THE VALUE OF RHO AND THICKNESS IN THE REGION OF STUDY,NRREP
196   C
197          DO 90 NR=1,NRT1
198          RHO(NR,NP)=XRHO(NS,NR)
199          U(NR,NP)=PERM(NS,NR)
200          IF(NR.NE.1) TH(NR,NP)=(RAD(NS,NR-1)-RAD(NS,NR))*CVT
201          TH(1,NP)=1.E10
202          RDCL(NR,NP)=RADCLR(NS)*CVT
203          IF(NR.NE.1)RLOC(NR,NP)=.5*CVT*(RAD(NS,NR-1)+RAD(NS,NR))
204          ZLOC(NR,NP)=0.
205          DSIZ(NR,NP)=0.
206          IF(KD.EQ.1) GO TO 90
207          NDSR=NDEF(NS,NR)
208          IF(NDSR.EQ.0) GO TO 90
209   C
210   C      LOOP OVER THE NDSR DEFECTS IN REGION NR
211   C
212          DO 80 ND=1,NDSR
213          RHO(NR,NP)=XRHO(NS,NR)
214          U(NR,NP)=PERM(NS,NR)
215          IF(NR.NE.1) TH(NR,NP)=(RAD(NS,NR-1)-RAD(NS,NR))*CVT
216          TH(1,NP)=1.E10
217          RDCL(NR,NP)=RADCLR(NS)*CVT
218          IF(ND.NE.(KD-1)) GO TO 80
219          RLOC(NR,NP)=RADLOC(NS,NR,ND)*CVT
220          ZLOC(NR,NP)=AXILOC(NS,NR,ND)*CVT
221          DSIZ(NR,NP)=DFSIZE(NS,NR,ND)*(CVT**3)
222       80 CONTINUE
223       90 CONTINUE
224      100 CONTINUE
225          IF(NP.EQ.NPT) GO TO 120
226          WRITE (LOTEK,110) NP,NPT,NPTT
227      110 FORMAT(' THERE IS AN ERROR IN THE NUMBER OF PROPERTIES.',/,
228         *' NP=',I3,' NPT=',I3,' NPTT=',I3)
229          GO TO 900
230      120 DO 130 NP=1,NPT
231          DO 130 NP=1,NPT
232      130 CONTINUE
```

The date and time when the data in FILE 37 were taken are printed on the LOU output terminal.

```
233   C
234   C       THE DATE AND TIME ARE PRINTED
235   C
236          WRITE (LOU,140) IDAY, ITIM
237      140 FORMAT (' CALIBRATION DATA TAKEN     ', 2(I2,1H/), I2,
238         1  ' TIME  ', 2(I2,1H:), I2,/)
```

The total shunt capacitances in the driver and pickup circuits are calculated.

```
C
C       THE CIRCUIT PARAMETERS ARE NOW GIVEN.
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
        CAPDR=CAPDR+CCABLE
        CAPPU=CAPPU+CCABLE
```

Statement 150 is used as a return point for a later option. If the variable NPRINT = 1, then the properties in the various regions of each sample are printed on the LOU output terminal.

```
C
  150 CONTINUE
C
C       EACH SET OF PROPERTIES FOR EACH REGION IS NOW PRINTED OUT
C
        IF (NPRINT.NE.1) GO TO 240
        NP=0
        DO 240 NS=1,NST
        NODF=1+NDF(NS)
        DO 230 KD=1,NODF
        NP=NP+1
        WRITE(LOU,160)
  160 FORMAT(1X)
        WRITE(LOU,170)UNITS(3,NUNIT),UNITS(1,NUNIT),UNITS(2,NUNIT),
       *UNITS(4,NUNIT),UNITS(5,NUNIT),UNITS(5,NUNIT),UNITS(6,NUNIT)
  170 FORMAT(' PR.SET  REG   THICK        RESTVY       PERM   RAD CLR
       *RAD LOC  AXIAL LOC  DEF SIZ',/,16X,A4,8X,A4,7X,A4,5X,A4,8X,
       *A4,2(6X,A4))
        DO 220 NR=1,NRT1
        IF(KD.EQ.1) GO TO 200
        NDSR=NDEF(NS,NR)
        IF(NDSR.EQ.0) GO TO 200
        DO 190 ND=1,NDSR
        IF(ND.NE.(KD-1)) GO TO 190
        WRITE(LOU,180) NP,NR,TH(NR,NP),RHO(NR,NP),U(NR,NP),
       *RDCL(NR,NP),RLOC(NR,NP),ZLOC(NR,NP),DSIZ(NR,NP)
  180   FORMAT(1X,2(I5),2(1PE12.4),0PF9.3,1PE12.4,0PF9.4,F9.4,1PE12.4)
  190   CONTINUE
        GO TO 220
  200   WRITE(LOU,210)NP,NR,TH(NR,NP),RHO(NR,NP),U(NR,NP),
       *RDCL(NR,NP)
  210   FORMAT(1X,2(I5),2(1PE12.4),0PF9.3,1PE12.4)
  220   CONTINUE
  230   CONTINUE
  240   CONTINUE
        WRITE(LOU,160)
```

If NPRINT = 2 the phase and magnitude of the output voltage at each frequency are printed on the LOU output terminal.

```
C
        IF(NPRINT.NE.2) GO TO 280
        WRITE(LOU,250) ((BLANKS),NF=1,NFT)
  250 FORMAT(' AV.RDGS.SAMP.',2X,4(A1,'MAG',7X,'PHA',6X))
        DO 270 NP=1,NPT
        WRITE(LOU,260) NP,((TMAGM(NP,NF),PHASEM(NP,NF)),NF=1,NFT)
  260 FORMAT(9X,I3,2X,4(F10.5,F10.5))
  270 CONTINUE
  280 CONTINUE
```

An index used to prevent repetition of certain printouts, such as property headings, is NLABL. If NPRINT = 0, control shifts to statement 630, where various printout options are offered; if NPRINT ≠ 0, control shifts to statement 880, where options are offered to print out data and measure unknown samples.

```
C
      NLABL=1
      IF(NPRINT.NE.0)GO TO 880
      GO TO 630
```

The property-fitting part of the program starts now.

```
C
C     LEAST SQUARES DESIGN SECTION.
C
C     SELECT PROPERTY TO BE FITTED AND SET UP PROPERTY ARRAY.
C
```

The first dimension of the PROP array, MSET1, is set, and the second index, NPROPT, is checked to see that it does not exceed the maximum allowed value, NPROPM.

```
  300 MSET=NPT
      MSET1=MSET+1
      IF(NPROPT.GT.NPROPM) GO TO 860
```

The choices of property and region to be studied are offered and read; the property name, XP, and the units, UT, are selected. Then the NPROPT column of the PROP array is set up.

```
  310 WRITE(LOTEK,320)
      WRITE(LOTEK,330)
  320 FORMAT(' SELECT NUMBER OF THE PROPERTY TO BE FITTED:'/
     *' 1=RHO 2=MU 3=THICKNESS 4=RADCLR 5=DEFECT RAD 6=DEFECT SIZE')
  330 FORMAT(' SELECT REGION TO BE FITTED. LOWEST IS NUMBER 1.',/)
      READ(LITEK,*)NPROP,NREG
      XP=PROPTY(NPROP)
      UT=UNITS(NPROP,NUNIT)
      DO 340 NP=1,NPT
      IF(NPROP.EQ.1)PROP(NP,NPROPT)=RHO(NREG,NP)
      IF(NPROP.EQ.2)PROP(NP,NPROPT)=U(NREG,NP)
      IF(NPROP.EQ.3)PROP(NP,NPROPT)=TH(NREG,NP)
      IF(NPROP.EQ.4)PROP(NP,NPROPT)=RDCL(NREG,NP)
      IF(NPROP.EQ.5)PROP(NP,NPROPT)=RLOC(NREG,NP)
      IF(NPROP.EQ.6)PROP(NP,NPROPT)=DSIZ(NREG,NP)
  340 CONTINUE
```

The type of polynomial is selected to fit the chosen property. The number of terms in the polynomial is IRDPR; the array specifying the terms is JPOL; the first index indicates whether the term is a magnitude or phase and specifies the type of function, its highest power, and the number of cross products; the second index specifies the frequency; the third index gives the property. The subset of JPOL for the particular property being fitted in the current operation is NPOL. The subroutine POLTYP actually sets up the polynomial in the array called POLARY.

```
  350 WRITE(LOTEK,360)
  360 FORMAT(' TYPE 1 IF THERE IS OFFSET; 0 IF NO OFFSET:',/)
      READ(LITEK,*) JOFSET(NPROPT)
      IOFSET=JOFSET(NPROPT)
      IRDPR=IOFSET
  370 WRITE(LOTEK,380)
  380 FORMAT(' SELECT THE NUMBER OF THE FUNCTION TYPE, POLYNOMIAL',
```

```
     *' DEGREE, & # OF '/,
     *' CROSS TERMS FOR EACH MAGNITUDE & PHASE')
      WRITE(LOTEK,390)
  390 FORMAT(' FUNCTION TYPE:1=LINEAR 2=LOG 3=EXP 4=INV ')
  400 WRITE(LOTEK,160)
      WRITE(LOTEK,410)
  410 FORMAT(25X,'FCTN POL # CROSS'/,25X'TYPE DEG TERMS')
      DO 450 NF=1,NFT
      DO 440 NC=1,2
      NCC=NC*3
      NCP=NCC-1
      NCF=NCP-1
      IF(NC.EQ.1) WRITE(LOTEK,420) FREQ(NF)
      IF(NC.EQ.2) WRITE(LOTEK,430) FREQ(NF)
  420 FORMAT(' MAG AT ',1PE12.6,' HZ    ')
  430 FORMAT(' PHA AT ',1PE12.6,' HZ    ')
      READ(LITEK,*)
     *JPOL(NCF,NF,NPROPT),JPOL(NCP,NF,NPROPT),JPOL(NCC,NF,NPROPT)
C     JPOL(3,NF,NPROPT)=0
      IRDPR=IRDPR+JPOL(NCP,NF,NPROPT)+JPOL(NCC,NF,NPROPT)
      JRDPR(NPROPT)=IRDPR
      NPOL(NCF,NF)=JPOL(NCF,NF,NPROPT)
      NPOL(NCP,NF)=JPOL(NCP,NF,NPROPT)
      NPOL(NCC,NF)=JPOL(NCC,NF,NPROPT)
  440 CONTINUE
  450 CONTINUE
      IRDPR1=IRDPR+1
      IF(IRDPRM.LT.JRDPR(NPROPT))WRITE(LOTEK,460)
      IF (IRDPRM.LT.JRDPR(NPROPT))GO TO 630
  460 FORMAT(' ERROR: # OF TERMS IN POLARY EXCEEDS DIMENSION')
      JROW=IRDPRM+1
      CALL POLTYP(POLARY,JROW,IRDPR,NPOL,6,NFT,2,IOFSET,LOTEK)
```

By using the selected polynomial type, the instrument readings are expanded into the READNG array by calling the RDGEXP subroutine. The working set of properties is PRO.

```
C
C       EXPAND THE RAW READINGS INTO IRDPR READINGS.
C
  470 DO 480 NF=1,NFT
      TMDFT(NF)=0.
      PHDFT(NF)=0.
  480 CONTINUE
      DO 490 NP=1,NPT
      CALL RDGEXP(READNG,TMAGM,PHASEM,NPOL,IOFSET,TMDFT,PHDFT,
     *MSET1,IRDPR1,NP,NFT,1,1,NP,NPT)
      PRO(NP)=PROP(NP,NPROPT)
  490 CONTINUE
```

The least-squares fitting of the READNG polynomial expansions to the PRO properties is done by the ALSQS subroutine.

```
C
C       DO THE LEAST SQUARES FIT OF THE READINGS TO THE PROPERTIES.
C
      CALL ALSQS(READNG,PRO,COE,RSQS,MSET,IRDPR,MSET1)
```

The sum of squares of the differences between the actual and calculated values of the property, SSDIFF, and the sum of squares of the drifts, SSDRIF, are initialized, and printout headings are written if NPRINT = 2.

```
C
C       CALCULATE THE DIFFERENCES IN THE FIT AND THE MAXIMUM DRIFTS.
C
```

```
  500 SSDRIF=0.
      SSDIFF=0.
      IF(NPRINT.EQ.2)WRITE(LOU,510)XP,NREG,XP,UT,UT,UT,UT
      IF(NPRINT.EQ.2)WRITE(LOTEK,510)XP,NREG,XP,UT,UT,UT,UT
  510 FORMAT(1X,/,'  NP ',A4,' IN REG',I2,'  CAL ',A4,6X,'DIFF',8X,
     *'DRIFT',/,10X,A4,3(8X,A4))
```

The "drift" is initialized for each property value. Loops are started to introduce a 0.01% error in a magnitude or a 0.01° error in a phase shift, the polynomial expansion with one erroneous reading is set up by calling the RDGEXP subroutine, and the "drifted" property value is accumulated as SUM.

```
      DO 570 NP=1,NPT
      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*TMAGM(NP,NF)
      IF(NC.EQ.2)PHDFT(NF)=.01
      CALL RDGEXP(READNG,TMAGM,PHASEM,NPOL,IOFSET,TMDFT,PHDFT
     *,MSET1,IRDPR1,NP,NFT,1,1,NP,NPT)
C
C     THE POLYNOMIAL IS CALCULATED
C
      SUM=0.
      DO 520 IR=1,IRDPR
      SUM=SUM+COE(IR)*READNG(NP,IR)
  520 CONTINUE
```

The value of READNG (NP, IRDPR1) is the value of the NP property obtained by the least-squares fitting of the polynomial by the subroutine ALSQS. Its difference from SUM is accumulated without regard to sign as DRIFT, and its difference from the actual value of the property, PRO (NP), is called DIFF. The squares of the differences and drifts are accumulated as SSDIFF and SSDRIF.

```
      DRIFT=DRIFT+ABS(READNG(NP,IRDPR1)-SUM)
      TMDFT(NF)=0.
      PHDFT(NF)=0.
  530 CONTINUE
  540 CONTINUE
      DIFF=PRO(NP)-READNG(NP,IRDPR1)
      SSDIFF=SSDIFF+DIFF*DIFF
      SSDRIF=SSDRIF+DRIFT*DRIFT
```

If NPRINT = 2 the actual and calculated values of the property and the corresponding difference and drift are printed on the LOTEK and LOU output terminals.

```
      IF(NPRINT.NE.2)GO TO 570
C
C      THE ENTIRE FIT IS PRINTED OUT
C
      WRITE(LOU,560)NP,PRO(NP),READNG(NP,IRDPR1),DIFF,DRIFT
      WRITE(LOTEK,560)NP,PRO(NP),READNG(NP,IRDPR1),DIFF,DRIFT
  560 FORMAT(1X,I3,4F12.5)
  570 CONTINUE
```

The rms values of the drifts and differences are calculated as SDRIF and SDIFF; they would be the standard deviations of the drifts and differences if the distributions were normal. In any case, the name, region, and rms values of the selected property are printed on both output terminals, with the selected units, UT.

```
      SDRIF=SQRT(SSDRIF/FLOAT(MSET))
      SDIFF=SQRT(SSDIFF/FLOAT(MSET))
      WRITE(LOU,580)XP,NREG,SDIFF,UT,SDRIF,UT
      WRITE(LOTEK,580)XP,NREG,SDIFF,UT,SDRIF,UT
  580 FORMAT(1X,/,' RMS DIFF IN ',A4,' IN REG #',I2,'=  F10.5,1X,A4,
```

If NPRINT = 3 the least-squares coefficients of the fit, COE, are converted to binary hexadecimal by the subroutine CONVR8 or CONVR9 (depending upon which microcomputer is being used) and then written on FILE 31. The file index, NCOED, is incremented, and the property name is written on the PRONAM array.

```
     *', DRIFT=',F10.5,1X,A4)
  590 IF (NPRINT.NE.3)GO TO 630
      IF(NPROPT.NE.1)GO TO 597
      DO 595 NC=1,NCAL
      DO 595 NF=1,LNF
      CALL CONVR9(RDGO(NF,NC),1,NCONV)
      DO 595 II=1,4
      WRITE(31'NCOED)NCONV(II)
  595 CONTINUE
  597 WRITE(LOU,160)
      WRITE(LOTEK,160)
      NCOED1=NCOED+4*IRDPRM+1
      WRITE(31'NCOED)IRDPR
      DO 610 I=1,IRDPR
      CALL CONVR9(COE(I),1,NCONV)
      WRITE(LOU,600)I,COE(I),(NCONV(II),II=1,4),(POLARY(I,J),J=1,5)
      WRITE(LOTEK,600)I,COE(I),(NCONV(II),II=1,4),(POLARY(I,J),J=1,5)
  600 FORMAT(' COEF(',I2,')=',1PE15.7,4X,4(Z2,1X),1X,5A4)
      COEF(I,NPROPT)=COE(I)
      DO 610 NCO=1,4
      WRITE(31'NCOED)NCONV(NCO)
  610 CONTINUE
      NCOED=NCOED1
      PRONAM(NPROPT)=XP
```

The unconverted coefficients of the fit are also written on FILE 21. Then the file index ICOEF is set back so that the last written line of FILE 21 (containing STOP) will be overwritten the next time data are entered in FILE 21. Finally, the property index is incremented to be ready for data for the next property.

```
C
C     THE COEFFICIENT,OFFSET,NPOL AND IRDPR ARE WRITTEN ON THE RAD DISC
C     FILE #21.
C
      WRITE(21'ICOEF)XP,IRDPR,JOFSET(NPROPT)
      WRITE(21'ICOEF)(COEF(IR,NPROPT),IR=1,IRDPR)
      DO 620 NF=1,NFT
      WRITE(21'ICOEF)(NPOL(I,NF),I=1,6)
  620 CONTINUE
      WRITE(21'ICOEF)STOP,NFT,NFT
      ICOEF=ICOEF-1
      NPROPT=NPROPT+1
```

The operation to be performed next is selected.

```
C
  630 WRITE(LOTEK,640)
  640 FORMAT(' 1 FIT PROP 2 PRT ENTIRE FIT 3 PRT/SV COEF 4 CHG FCTN/POL'
     *,'TYP 5 RUN TEST',/)
      READ(LITEK,*)NPRINT
      GO TO(300,500,590,350,650),NPRINT
```

If the last option was selected, the property number is set back to the previous value, and control jumps to statement 880, where another choice of options is offered.

```
466     650 NPROPT=NPROPT-1
467         WRITE(LOU,660)
468     660 FORMAT(1X)
469         GO TO 880
```

If the value of NPRINT was typed in as 5, control is transferred to statement 900, which terminates the program. If NPRINT is typed in as 3 or 4, control jumps to statement 700 to take new readings from the eddy-current instrument.

```
470  C      CALCULATES PROPERTIES FROM MAGNITUDES AND PHASES AND CONTINOUSLY
471  C      DISPLAYS THE VALUES ON THE LSI TERMINAL.
472  C
473     700 IF(NPRINT.EQ.3)WRITE(LOTEK,710)(PRONAM(NPRO),NPRO=1,NPROPT)
474     710 FORMAT(1X,6(6X,A4,3X))
475         IF(NPRINT.EQ.4)WRITE(LOTEK,715)((VOLTS(2*NF-1),VOLTS(2*NF))
476        *,NF=1,NFT)
477     715 FORMAT(1X,3('      MAG(',I1,')      PHA(',I1,')'))
478         WRITE(LOTEK,660)
```

A loop is now started to take magnitude and phase readings (TMAGM and PHASEM) to expand them in the chosen polynomials (JPOL) by calling the polynomial expansion subroutine (RDGEXP), and to use the stored coefficients (COEF) with the polynomial terms of powers of the measurements (READNG) to calculate the property values (PRO). The instrument readings are updated by calling the READAT subroutine. The loop is continued by the test in line 521 until the foot pedal is depressed, which will set NSTOP to 0 and terminate the loop with the last calculated values of the properties (PRO), magnitudes (TMAGM), and phases (PHASEM).

```
479  C
480  C      EXPANSION OF TMAGM(NF,NF) AND PHASEM(NF,NF) INTO READNG(1,IRDPRM)
481  C
482     720  DO 800 NPRO=1,NPROPT
483          DO 790 NF=1,NFT
484          DO 780 I=1,6
485          NPOL(I,NF)=JPOL(I,NF,NPRO)
486     780 CONTINUE
487          TMDFT(NF)=0.
488          PHDFT(NF)=0.
489     790 CONTINUE
490          IOFSET=JOFSET(NPRO)
491          IRDPR=JRDPR(NPRO)
492          IRDPR1=IRDPR+1
493          CALL RDGEXP(READNG,TMAGM,PHASEM,NPOL,IOFSET,TMDFT,PHDFT,
494         *MSET1,IRDPR1,1,NFT,1,1,1,NPT)
495          PRO(NPRO)=0.
496          DO 800 IR=1,IRDPR
497          PRO(NPRO)=PRO(NPRO)+COEF(IR,NPRO)*READNG(1,IR)
498     800 CONTINUE
499          NSTART=1
500          IF(NPRINT.EQ.3)WRITE(LOTEK,805)(PRO(NPRO),NPRO=1,NPROPT)
501     805 FORMAT(1H+,6(F13.5))
502  C
503  C      NEW READINGS ARE MADE FROM THE EDDY CURRENT INSTRUMENT.
504  C
505          NCH2=NCHS+2
506     825 IF(MAG.EQ.1) CALL READAT(VOLTS,NCH2,1024,1,NSTOP,MAG)
507          CALL READAT(VOLTS,NCH2,ITIMS,NRET,NSTOP,MAG)
508          IF(MAG.EQ.1) CALL SATMAG(0)
```

```
509          VOLTS(8)=25.*VOLTS(7)/VOLTS(8)
510          DO 830 NC=1,NCHS
511          VOLTS(NC)=VOLTS(NC)+TOFSET(NC)+TSLOPE(NC)*VOLTS(8)
512      830 CONTINUE
513          CALL CORRDG(VOLTS,OFSET,CGAIN,NCHS)
514      840 DO 845 NF=1,NFT
515          TMAGM(1,NF)=VOLTS(2*NF-1)
516          PHASEM(1,NF)=VOLTS(2*NF)
517      845 CONTINUE
518          IF(NPRINT.EQ.4)WRITE(LOTEK,820)((VOLTS(2*NF-1),VOLTS(2*NF))
519         *,NF=1,NFT)
520      820 FORMAT(1H+,6(F13.4))
521          IF(NSTOP.GT.0)GO TO 720
522    C
523    C     PROGRAM WILL STAY IN THIS LOOP UNTIL THE FOOT PEDAL IS PRESSED.
524    C
```

After the foot pedal is depressed and computer control escapes from the loop, the properties calculated last (PRO) or magnitudes and phases (TMAGM and PHASEM) are printed on the LOU terminal, depending upon whether NPRINT was chosen as 3 or 4 respectively. Then control jumps to statement 880, where the various options for the next job are offered.

```
525          IF(NLABL+NPRINT.EQ.4)WRITE(LOU,710)(PRONAM(NPRO),NPRO=1,NPROPT)
526          NLABL=NPRINT
527          IF(NPRINT.EQ.3)WRITE(LOU,850)(PRO(NPRO),NPRO=1,NPROPT)
528      850 FORMAT(1H ,6(F13.5))
529          IF(NPRINT.EQ.4)WRITE(LOU,850)((VOLTS(2*NF-1),VOLTS(2*NF))
530         *,NF=1,NFT)
531          GO TO 880
```

If the number of lines read from FILE 37 exceeded the limit set by NFILIM in line 107, the error message in FORMAT statement 857 will be displayed on the LOTEK terminal.

```
532      855 WRITE(LOTEK,857)
533      857 FORMAT(' LIMIT OF FILE 37 IS EXCEEDED.')
534          GO TO 880
```

If the number of properties being fit with polynomial expansions exceeds the limit set by NPROPM in line 304, the error message in FORMAT statement 870 will be displayed on the LOTEK terminal.

```
535      860 WRITE(LOTEK,870)
536      870 FORMAT(' PROP ARRAY IS FILLED.')
```

The choices for the next procedure are offered, and the last option terminates the program.

```
537      880 WRITE(LOTEK,890)
538      890 FORMAT(' PRINT :1.PARAMETERS 2.MEAS VOLT 3.CAL&DISPLAY PROPS'
539         *,'4.RAW RDGS.5.RE-CALIB 6.STOP',/)
540          READ(LITEK,*)NPRINT
541          GO TO (150,200,700,700,895,900),NPRINT
542      895 CALL CALIB(RDGC,NCHS,NSTOP)
543          CALL RESET(RDGC,RDGO,OFSET,GAIN,STDV,NCHS)
544          GO TO 880
545      900 STOP JOB
546          END
```

A sample printout is given below.

```
        PROGRAM TUBFIT DATE  11/ 9/78 TIME  15:35:42
CALIBRATION DATA TAKEN   11/ 6/78 TIME  18:20:54


RMS DIFF IN THIK IN REG # 3=   0.00055 INCH, DRIFT=   0.00030 INCH

 NP THIK IN REG 3  CAL THIK      DIFF        DRIFT
        INCH        INCH        INCH        INCH
  1     0.05110     0.05094     0.00016     0.00048
  2     0.05110     0.04995     0.00115     0.00030
  3     0.05110     0.05043     0.00067     0.00030
  4     0.05110     0.05061     0.00049     0.00030
  5     0.05110     0.05038     0.00072     0.00030
  6     0.05110     0.05100     0.00010     0.00030
  7     0.05110     0.05056     0.00054     0.00030
  8     0.05110     0.05086     0.00024     0.00030
  9     0.05110     0.05089     0.00021     0.00030
 10     0.05110     0.05070     0.00040     0.00030
 11     0.05110     0.05185    -0.00075     0.00030
 12     0.05110     0.05159    -0.00049     0.00030
 13     0.05110     0.05178    -0.00068     0.00030
 14     0.05110     0.05177    -0.00067     0.00030
 15     0.05110     0.05148    -0.00038     0.00030
 16     0.04225     0.04249    -0.00024     0.00030
 17     0.04225     0.04126     0.00099     0.00030
 18     0.04225     0.04114     0.00111     0.00030
 19     0.04225     0.04130     0.00095     0.00030
 20     0.04225     0.04156     0.00069     0.00030
 21     0.04225     0.04257    -0.00032     0.00030
                              .
                              .
                              .
 68     0.03824     0.03827    -0.00003     0.00030
 69     0.03824     0.03822     0.00002     0.00030
 70     0.03824     0.03826    -0.00002     0.00030
 71     0.03824     0.03806     0.00018     0.00030
 72     0.03824     0.03788     0.00036     0.00030
 73     0.03824     0.03767     0.00057     0.00030
 74     0.03824     0.03757     0.00067     0.00030
 75     0.03824     0.03721     0.00103     0.00030

RMS DIFF IN THIK IN REG # 3=   0.00055 INCH, DRIFT=   0.00030 INCH

RMS DIFF IN DFSI IN REG # 3=   0.00563 CUIN, DRIFT=   0.00557 CUIN

 NP DFSI IN REG 3  CAL DFSI      DIFF        DRIFT
        CUIN        CUIN        CUIN        CUIN
  1     0.00000     0.00754     0.00754     0.03186
  2     0.02000     0.01399     0.00601     0.00345
                              .
                              .
                              .
```

## Reference

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

## Appendix C
## THE SUBROUTINES

### CIRCOL

The subroutine CIRCOL retrieves data for the coil named COLNAM from FILE 29 and transfers the data back to the main program in COMMON block B3. If COLNAM = 0 the operator will be asked to type in a six-character coil name. In any case, a search will be made through FILE 29 for a coil named COLNAM. If no such coil is found, the message in FORMAT statement 60 will be displayed on the LOTEK terminal, and the operator's responses will be read at the LITEK terminal.

```
  1  C
  2        SUBROUTINE CIRCOL(LI,LOU,COLNAM)
  3  C *** SUBROUTINE TO OBTAIN COIL DATA *** (20 APRIL 1978)
  4        REAL*6 COIL,COLNAM
  5        COMMON /B3/COLNAM,RBAR,R1,R2,XL,ZL,RLIM,R6,R7,TNDR,TN,X,ZLDR,ZLPU
  6        DEFINE FILE 26(80,30,U,NCOL)
  7        NCOL=1
  8        IF(COLNAM.NE.0.)GO TO 40
  9     10 WRITE (LOU,20)
 10     20 FORMAT (' TYPE COIL NAME (UP TO 6 LETTERS):')
 11        READ (LI,30) COLNAM
 12     30 FORMAT (A4)
 13     40 READ(26'NCOL)COIL,RBAR,R1,R2,XL,ZL,RLIM,R6,R7,TNDR,TN,X,ZLDR,ZLPU
 14        IF (COIL.EQ.COLNAM) GO TO 70
 15        IF (COIL.EQ.'END   ') GO TO 50
 16        GO TO 40
 17     50 WRITE (LOU,60) CNAM
 18     60 FORMAT(' COIL ',A4,' NOT FOUND. TYPE FOLLOWING COIL PARAMETERS:',
 19       */,' RBAR,R1,R2,XL,ZL,RLIM,RDCDR,RDCPU,TNDR,TN,X,ZLDR,ZLPU:')
 20        READ(LI,*) RBAR,R1,R2,XL,ZL,RLIM,RDCDR,RDCPU,TNDR,TN,X,ZLDR,ZLPU
 21     70 RETURN
 22        END
 23  C
TOTAL RECORDS WRITTEN =     24
EXIT
$AVR CI 4
$END  LIST
```

### READAT

The READAT subroutine is not included in the listed program but is stored in the user library (UL) in binary object code. It controls the acquisition of data from the eddy-current instrument via the parallel data acquisition system of the computer (called MODAC for the ModComp IV computer). The data are stored in an array called MARRAY and converted to floating-point volts in line 51. As long as the foot pedal is held down, the test in line 53 will keep the control looping back to statement 30. When the pedal is released, the last values calculated for VOLTS are returned to the main program.

For a different computer system this subroutine would probably need to be rewritten, particularly the part in assembly language. The objective is to return the NCHS instrument readings of magnitudes and phases at the various frequencies back to the main program in the array called VOLTS.

```
  1        SUBROUTINE READAT(VOLTS,NCHS,ITIMS,NRET,NSTOP,MAG)
  2  C
  3  C     SUBROUTINE TO READ DATA FROM AN EDDY CURRENT TEST USING THE MODAC.
  4  C     PROGRAM SENDS PROMPT SIGNALS OVER THE AO2 CHANNEL AND DISPLAYS THE
  5  C     CURRENT READING VALUES OF THE DESIGNATED CHANNELS OVER AO2.
  6  C     NOTE: SYSTEM PROTECT SWITCH MUST BE OFF FOR PROGRAM TO RUN.
  7  C     AFTER NSTOP HAS GONE LOW(FOOT PEDAL DEPRESSED) AND ROUTINE HAS
  8  C        RETURNED ONCE TO SIGNAL PROGRAM TO MAKE READING (NRET=1),
```

```
C          THEN HANG IN HERE (LOOP NRET=0) UNTIL NSTOP GOES HIGH (=1),
C          IMPLYING THAT FOOT PEDAL HAS BEEN RELEASED.
C
        INTEGER*2 ITIM,NCH,STP,IMAG
        DIMENSION VOLTS(12),MARRAY(12)
        ITIM=ITIMS
        CONFAC=819.2*FLOAT(ITIMS)
        IMAG=MAG*8192
        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            LOAD THE NO.OF TIMES IN REG.6
        LDM,7          IMAG            LOAD IMAG INTO REG.7
ITLP    LDI,3          1               LOAD 1 IN REG.3
        LDI,2          0               ZERO REG.2,INDEX REG. FOR MARRAY
CHLP    ABR,3,0                        ADD BIT 0 TO REG.3,START ADC,SWITCH
        ADR,3,7                        ADD REG.7 TO REG.3(IMAG ADDED)
        ODD,3,2                        OUTPUT REG.3 TO PORT 32
        SBR,3,0                        SUBTRACT START BIT
        ODD,3,2                        OUTPUT REG.3,SEND HI,THEN LO PULSE
        SUR,3,7                        SUTRACT IMAG BACK OFF
        LAS,3,1                        SHIFT REG.3 LEFT ONE PLACE
BUSY    IDD,5,0                        INPUT PORT 30 INTO REG.5
        TBRB,5,14      BUSY            TRANSFER BACK TO 'BUSY' IF BIT 14 HI
        ODD,7,2                        ZERO SENT OUT ON 32(TURN ALL OFF)
        RAS,5,2                        SHIFT REG.5(HAS ADC #) TWO POS.RIGHT
        ESS,4,5                        CONVRT ADC #  TO DOUB REG WORD IN 4,5
        ADMD,4,2       MARRAY          MARRAY(REG.2)ADDED TO REG.4,5
        STMD,4,2       MARRAY          RESULT STORED BACK IN MARRAY(REG.2)
        ABR,2,14                       REG.2 INCREMENTED BY 2(MARRAY IS I*2)
        CRMB,2         NCH,CHLP,CHLP   LOOP BACK TO CHLP IF(REG.2.LE.NCH)
        SBRB,6,15      ITLP            DECREMENT # OF TIMES IN REG.6,
*                                      TRANSFER BACK IF NOT 0
        IDD,5,0                        READ PORT AGAIN,LOOK AT FOOTSWITCH
        ETI,5          #FFFE           EXTRACT BIT 15 (& WITH COMPLEMENT)
        STM,5          STP             STORE IN STP
        FINI
        DO 20 N=1,NCHS
        VOLTS(N)=FLOAT(MARRAY(N))/CONFAC
   20 CONTINUE
        IF(STP.LT.1.AND.NRET.EQ.0) GO TO 30
        NSTOP=STP
        NRET=1
        RETURN
        END
TOTAL RECORDS WRITTEN =     58
```

## CORRDG

This subroutine takes the values of VOLTS for each channel from the main program, multiplies by the GAIN for that channel, adds the corresponding OFSET (offset), and returns the corrected value of VOLTS to the main program.

```
        SUBROUTINE CORRDG(VOLTS,OFSET,GAIN,NCHS)
        DIMENSION VOLTS(8),OFSET(6),GAIN(6)
        DO 100 NCH=1,NCHS
        VOLTS(NCH)=OFSET(NCH)+GAIN(NCH)*VOLTS(NCH)
  100 CONTINUE
        RETURN
        END
TOTAL RECORDS WRITTEN =      8
EXIT
```

## CALIB

This subroutine automatically takes the calibration readings by calling the READAT subroutine with appropriate calling parameters to set the relay switches in the calibrator module. Actually, READAT is called twice for each channel, with a delay after the first switching to allow the readings to settle down (primarily to allow the low-frequency filters to have sufficient discrimination time). The delay is produced by taking readings 1024 times in line 391 before calling READAT again in line 392. The values of VOLTS returned from READAT are called RDGC (calibration readings) before being returned to the main program.

For a different computer or a different instrument, this subroutine would probably need to be rewritten. This version is written for an automatic calibrator module, which switches known phase shift and attenuator networks into the circuit at the various frequencies by means of relays activated by the program. If no automatic calibrator is available, this subroutine can be eliminated, the switching can be done by hand, and the data can be taken by the READAT subroutine when the foot pedal is pressed.

```
  1          SUBROUTINE CALIB(RDGC,NCHS,NSTOP)
  2   C
  3   C      PROGRAM TO DRIVE AUTOMATIC CALIBRATOR MODULE
  4   C
  5          INTEGER*2 NRDG
  6          DIMENSION RDGC(NCHS,4),VOLTS(8)
  7          DO 100 NRDG=1,4
  8          INLINE
  9          LDM,2          NRDG            LOAD VALUE OF CHANNEL SELECT IN REG2
 10          SBR,2,15                       SUBTRACT OFF 1(GO FROM 0 TO 3)
 11          ODD,2,3                        OUTPUT TO PORT 33,SET CALIBRATOR
 12          FINI
 13          CALL READAT(VOLTS,NCHS,1024,1,NSTOP,0)
 14   C
 15   C      DELAY FOR 6 MS TO ALLOW RELAY TO SWITCH,READINGS TO SETTLE
 16   C
 17          CALL READAT(VOLTS,NCHS,8192,1,NSTOP,0)
 18          DO 50 NCH=1,NCHS
 19          RDGC(NCH,NRDG)=VOLTS(NCH)
 20       50 CONTINUE
 21      100 CONTINUE
 22          INLINE
 23          LDI,2          7               LOAD VALUE FOR OPERATE MODE IN REG2
 24          ODD,2,3                        OUTPUT TO PORT 33,SET CALIBRATOR
 25          FINI
 26          WRITE(5,210)(I,I,I=1,3)
 27          DO 200 NRDG=1,4
 28          WRITE(5,220)NRDG,(RDGC(NCH,NRDG),NCH=1,NCHS)
 29      200 CONTINUE
 30      210 FORMAT(3X,3('      MAG(',I1,')      PHA(',I1,')'))
 31      220 FORMAT(1X,I2,6(F11.5))
 32          RETURN
 33          END
TOTAL RECORDS WRITTEN =     34
EXIT
```

## RESET

The RESET subroutine does a least-squares fitting the NRD=NCAL=NPHCAL*NMGCAL calibration readings for each of the NCHS channels. It finds the best-fitting values of offset and gain to correct the latest calibration readings (RDGC) back to the values (RDG0) when the data in FILE 37 were taken. It uses the formula

RDG0 = OFSET + GAIN*RDGC .

The best (least-squares) values of OFSET and GAIN to make this correction for each of the channels are then returned to the main program with the standard deviations of the fits.

```
      SUBROUTINE RESET(RDGC,RDGO,OFSET,GAIN,STDV,NCHS)
C
C     SUBROUTINE DOES NCHS LEAST SQUARES FITS OF RDGO=OFSET+GAIN*RDGC
C     LEAST SQUARES SUBROUTINE IS STANDARD FOR Y(J)=OFSET+GAIN*X(J)
C
      DIMENSION RDGC(NCHS,4),RDGO(NCHS,4),OFSET(NCHS),GAIN(NCHS)
      DIMENSION STDV(NCHS)
      DATA NRD/4/
      DO 100 NCH=1,NCHS
      SUMX=0.0
      SUMY=0.0
      SUMXY=0.0
      SUMXX=0.0
      SUMYY=0.0
C
C     PERFORM LEAST SQUARES FIT OVER NRD READINGS
C
      DO 50 NR=1,NRD
      SUMX=SUMX+RDGC(NCH,NR)
      SUMY=SUMY+RDGO(NCH,NR)
      SUMXY=SUMXY+RDGC(NCH,NR)*RDGO(NCH,NR)
      SUMXX=SUMXX+RDGC(NCH,NR)*RDGC(NCH,NR)
      SUMYY=SUMYY+RDGO(NCH,NR)*RDGO(NCH,NR)
   50 CONTINUE
      XBAR=SUMX/NRD
      YBAR=SUMY/NRD
      GAIN(NCH)=(SUMXY-SUMX*YBAR)/(SUMXX-SUMX*XBAR)
      OFSET(NCH)=YBAR-GAIN(NCH)*XBAR
      STDV(NCH)=SQRT(ABS((SUMYY-SUMY*YBAR-GAIN(NCH)*(SUMXY-SUMX*YBAR))/
     *FLOAT(NRD-2)))
  100 CONTINUE
      WRITE(5,210)(I,I,I=1,3)
      WRITE(5,220)(OFSET(NCH),NCH=1,NCHS)
      WRITE(5,230)(GAIN(NCH),NCH=1,NCHS)
      WRITE(5,240)(STDV(NCH),NCH=1,NCHS)
  210 FORMAT(6X,3('     MAG(',I1,')      PHA(',I1,')'))
  220 FORMAT(' OFSET',6(F11.5))
  230 FORMAT(' GAIN ',6(F11.5))
  240 FORMAT(' STDV ',6(F11.5))
      RETURN
      END
TOTAL RECORDS WRITTEN =     42
REM RESET
EXIT
```

## SATMAG

If ferromagnetic samples are being tested, a saturating electromagnet is usually needed. It can be turned on and off by the computer by using the SATMAG subroutine. If the value of MAG is 0 in the call, the MODAC port 32 sends out an instruction to switch off the magnet; if MAG = 1 the magnet is switched on.

```
C
      SUBROUTINE SATMAG(MAG)
C
C     IF MAG = 1,SATURATING COIL IS SWITCHED ON ,IF MAG = 0,FIELD IS
C     TURNED OFF.IF MAG.GT.1,ROUTINE RETURNS WITH NO ACTION.
C
      INTEGER*2 IBIT
      IF(MAG.GT.1)GO TO 100
      IBIT=MAG*8192
      INLINE
      LDM,2          IBIT       THE VALUE OF MAG(0 OR 1) IS PUT IN BIT 2
```

```
      ODD,2,2                    THE BIT IS SENT OUT ON MODAC PORT #32
      FINI
  100 RETURN
      END
C
TOTAL RECORDS WRITTEN =     17
EXIT
```

## 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. The A, Y, B, and R2 terms correspond to READNG, PRO, COEF, and RSOS respectively.

```
      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
```

## 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.[1]

```
        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
```

## CONVR9

Floating-point numbers from the ModComp minicomputer are converted to the hexadecimal representation of the floating-point binary used by the NDT-COMP9 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.[1]

```
        SUBROUTINE CONVR9(XNUM,NTYPE,NCONV)
        INTEGER*2 NEXP1,NHI1,NMD1,NLO1
        DIMENSION NCONV(4)
C
C       SUBROUTINE TO CONVERT MODCOMP FLOATING POINT NUMBERS TO THE
C       HEX REPRESENTATION OF THE FLOATING POINT BINARY THAT THE NDT-COMP9
C       USES.THE FORM IS EXPONENT,FRACTION WITH THE FIRST TWO NUMBERS
C       THE EXPONENT AND THE LAST SIX THE FRACTION.  VERSION 13 9 78
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
        ADI,6           #8000         ADD EXPONENT OFFSET TO HI BYTE
        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           #0000         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 MANTISSA IS XFERED TO 2
        STM,2           NHI1          AND STORED.
        MBR,2,7                       THE MID ORDER MANTISSA IS MOVED TO 2
        STM,2           NMD1          AND STORED.
        MLR,2,7                       THE LOW ORDER BYTE IS MOVED TO 2
        STM,2           NLO1          AND STORED.
```

```
 31          FINI
 32          NCONV(1)=NEXP1
 33          NCONV(2)=NHI1
 34          NCONV(3)=NMD1
 35          NCONV(4)=NLO1
 36          RETURN
 37          END
TOTAL RECORDS WRITTEN =     38
EXIT
```

## POLTYP

Subroutine POLTYP constructs the POLARY array for printing a representation of the polynomial expansion. The ENCODE instruction 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."

```
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
639          POLARY(NROW,1)=POLARY(NROW-1,1)
640          POLARY(NROW,2)=POLARY(NROW-1,2)
641          ENCODE(4,40,POLARY(NROW,3)) INUM(I),ROLD
642   40     FORMAT(A1,1X,A2)
643          ENCODE(4,50,POLARY(NROW,4)) IROLD(2),IROLD(3)
644   50     FORMAT(2A2)
645          ENCODE(4,60,POLARY(NROW,5))IROLD(4),INUM(J)
646   60     FORMAT(A2,A1,1X)
647          J=J-1
648          NROW=NROW+1
649   500    CONTINUE
650   300    CONTINUE
651   200    CONTINUE
652   C
653   C  PRINT RESULTS
654   C
655          WRITE(IDEV,21)((POLARY(I,J),J=1,5),I=1,MROW)
656   21     FORMAT(' POLARY='/(1X,5A4))
657          GO TO 1000
658   C
659   C  PRINT ERROR MESSAGES
660   C
661   99     WRITE(IDEV,31)
662   31     FORMAT(' ERROR: CANNOT HAVE A CROSS TERM ON 1ST ITERATION')
663   C
664   1000   RETURN
665          END
```

## 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 C.1). Subroutine RDGEXP constructs the $M$th 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.

**Table C.1. 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.

```
      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
```

## Reference

1. Modular Computer Systems, Inc., Ft. Lauderdale, Fla., *Reference Manual, MAX II/III/IV System Processors, Assemblers,* Publication 210-600500-008H00/B00, 1976.

## Appendix D
## THE COILDATA PROGRAM

A separate program called COILDATA is used to list, change, or add coil data on FILES 28 and 29. FILE 28 contains the data for reflection and through-transmission coils; FILE 29 contains data for encircling or inner-diameter coils.

Before COILDATA is run, FILES 28 and 29 should be allocated as specified in lines 7 and 8. Also, the input terminal (LI), output terminal (LOU), and the line printer (LPT) should be specified in line 9.

```
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/
```

The user is asked to choose between reflection coils (FILE 28) and encircling coils (FILE 29) by typing REF or CIR respectively. If neither is typed, NFILE remains zero, and control returns to statement 5.

```
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
```

The procedural options are offered in statement 17, and several input and output formats are listed in lines 18–21. The options are read in line 22 and used in line 23.

```
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
```

The LIST option starts by initializing the file index (NCOIL) in statement 100. The proper data file is selected, read, and printed line by line in lines 28–45. The process is cycled through the complete file by line 46 until the end of the file is found in line 34, which causes control to jump out of the loop and back to the list of options in statement 16.

```
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
```

The second (CHANGE) option allows the user to change the data in a specified line of the file, the one with index NCOIL, read from the terminal in statement 220. The new coil name is read as COILN; the proper file is selected in lines 55 and 56. The user is prompted to type in the data for the coil, and the data are read into the appropriate line of the file. The definitions of the encircling coil data are listed earlier in Table 3; corresponding definitions for reflection coils are given in another report.[1]

```
47  C
48  C       SECTION TO CHANGE THE COIL DATA
49      200 WRITE(LOU,210)
50      210 FORMAT(' CHANGE COIL IN LINE NO:(TYPE LINE NO IN DATA FILE) ')
51      220 READ(LI,*)NCOIL
52          WRITE(LOU,225)
53      225 FORMAT(' NEW COIL NAME (UP TO 6 CHARACTERS) ')
54          READ(LI,30)COILN
55          IF(NFILE.EQ.28)GO TO 227
56          IF(NFILE.EQ.29)GO TO 250
57          GO TO 5
58      227 WRITE(LOU,230)
59      230 FORMAT(' TYPE:RBAR,R1,R2,L3,R3,R4,L4,L5,L6,RDCDR,RDCPU'
60         *,',TNDR,TNPU',/)
61          READ(LI,*)RBAR,R1,R2,XL3,R3,R4,XL4,XL5,XL6,RDCDR,RDCPU,TNDR,TNPU
62          WRITE(LOU,40)NCOIL,COILN,RBAR,R1,R2,XL3,R3,R4,XL4,XL5,XL6
63         *,RDCDR,RDCPU,TNDR,TNPU
64          WRITE(28'NCOIL)COILN,RBAR,R1,R2,XL3,R3,R4,XL4,XL5,XL6
65         *,RDCDR,RDCPU,TNDR,TNPU
66          GO TO 16
67      250 WRITE (LOU,260)
68      260 FORMAT(' TYPE:RBAR,R1,R2,XL,ZL,RLIM,RDCDR,RDCPU,TNDR,TNPU,',
69         *'ZLDR,ZLPU,X',/)
70          READ(LI,*) RBAR,R1,R2,XL,ZL,RLIM,RDCDR,RDCPU,TNDR,
71         *TNPU,ZLDR,ZLPU,X
72          WRITE(LOU,40)NCOIL,COILN,RBAR,R1,R2,XL,ZL,RLIM,
73         *RDCDR,RDCPU,TNDR,TNPU,ZLDR,ZLPU,X
74          WRITE(29'NCOIL)COILN,RBAR,R1,R2,XL,ZL,RLIM,
75         *RDCDR,RDCPU,TNDR,TNPU,ZLDR,ZLPU,X
76          GO TO 16
```

The third option, to add data to one of the files, sends control to statement 300, where the file index (NCOIL) is initialized. The NFILE file is then read through in the loop in lines 81 and 82 until the coil END is found. The coil index is decremented by 1 in line 83 to overwrite the data listed for the coil previously called END.

```
77  C
78  C       SECTION TO ADD TO COIL DATA FILE
79  C
80      300 NCOIL =1
81      310 READ(NFILE'NCOIL)COILN
82          IF(COILN.NE.'END    ')GO TO 310
83          NCOIL=NCOIL-1
```

Statement 315 prompts the user to type in the new coil name; statement 317 tells how to stop the process. The value of NFILE selects the data requested, which is then read from the terminal and written on the appropriate file. The process is looped back in line 106 until the coil name END is typed in, at which time control jumps to statement 390, and the final line of the file is written with coil name END followed by a number of zeros. The execution is then stopped.

The fourth option (STOP) also causes control to shift to statement 400 and stop execution.

```
 84     315 WRITE(LOU,225)
 85         WRITE(LOU,317)
 86     317 FORMAT('  TYPE END TO STOP ')
 87         READ(LI,30)COILN
 88         IF(COILN.EQ.'END    ') GO TO 390
 89         IF(NFILE.EQ.28)GO TO 320
 90         IF(NFILE.EQ.29)GO TO 350
 91         GO TO 5
 92     320 WRITE(LOU,230)
 93         READ(LI,*)RBAR,R1,R2,XL3,R3,R4,XL4,XL5,XL6,RDCDR,RDCPU,TNDR,TNPU
 94         WRITE(28'NCOIL)COILN,RBAR,R1,R2,XL3,R3,R4,XL4,XL5,XL6
 95        *,RDCDR,RDCPU,TNDR,TNPU
 96         WRITE(LOU,40)NCOIL,COILN,RBAR,R1,R2,XL3,R3,R4,XL4,XL5,XL6
 97        *,RDCDR,RDCPU,TNDR,TNPU
 98         GO TO 315
 99     350 WRITE (LOU,260)
100         READ(LI,*) RBAR,R1,R2,XL,ZL,RLIM,RDCDR,RDCPU,TNDR,TNPU,
101        *ZLDR,ZLPU,X
102         WRITE(LOU,40)NCOIL,COILN,RBAR,R1,R2,XL,ZL,RLIM,RDCDR,RDCPU,
103        *TNDR,TNPU,ZLDR,ZLPU,X
104         WRITE(29'NCOIL)COILN,RBAR,R1,R2,XL,ZL,RLIM,RDCDR,RDCPU,TNDR,TNPU,
105        *ZLDR,ZLPU,X
106         GO TO 315
107     390 ZZ=0.
108         WRITE (NFILE'NCOIL)COILN,ZZ,ZZ,ZZ,ZZ,ZZ,ZZ,ZZ,ZZ,ZZ,ZZ,ZZ,ZZ
109         GO TO 16
110     400 STOP JOB
111         END
```

A printout of the data in FILE 29, the encircling coil file, is given below. In running other programs the data for the chosen coil are read from the first line of the file, which had previously been changed to contain the data for that coil. Hence, the data for coil I300-A appear twice. The initial "I" in the coil name indicates an inner coil; an "E" would indicate an encircling coil.

```
$$
```

| N | COIL | RBAR | R1 | R2 | XL | ZL | RLIM | RDCDR | RDCPU | TNDR | TNPU | ZLDR | ZLPU | X |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | I300-A | 0.3000 | 0.9167 | 1.0830 | 0.1667 | 0.1667 | 1.1000 | 12.2000 | 12.2000 | 120.0 | 120.0 | 0.300E-03 | 0.300E-03 | 0.0 |
| 2 | I169SB | 0.1690 | 0.8876 | 1.1124 | 0.5920 | 0.5920 | 1.1213 | 380.0000 | 380.0000 | 1000.0 | 1000.0 | 0.978E-02 | 0.978E-02 | 0.0 |
| 3 | I291-A | 0.2910 | 0.9140 | 1.0820 | 0.1718 | 0.1718 | 1.1065 | 37.9700 | 37.8000 | 222.0 | 222.0 | 0.120E-02 | 0.196E-02 | 0.0 |
| 4 | I300-A | 0.3000 | 0.9167 | 1.0830 | 0.1667 | 0.1667 | 1.1000 | 12.2000 | 12.2000 | 120.0 | 120.0 | 0.300E-03 | 0.300E-03 | 0.0 |
| 5 | I300-B | 0.3000 | 0.8867 | 1.0500 | 0.1667 | 0.1667 | 1.1000 | 39.9300 | 39.3600 | 222.0 | 222.0 | 0.124E-02 | 0.125E-02 | 0.0 |
| 6 | I352-A | 0.3520 | 0.9496 | 1.0500 | 0.0710 | 0.0710 | 1.0920 | 43.0000 | 47.6000 | 222.0 | 222.0 | 0.140E-02 | 0.155E-02 | 0.0 |
| 7 | I356-A | 0.3565 | 0.9300 | 1.0700 | 0.1400 | 0.1400 | 1.0850 | 12.8000 | 12.8000 | 100.0 | 100.0 | 0.300E-03 | 0.300E-03 | 0.0 |
| 8 | 2V159S | 0.1590 | 0.8962 | 1.1164 | 0.6289 | 0.6289 | 2.2642 | 7.5000 | 7.5000 | 114.0 | 0.0 | 0.600E-03 | 0.000E+00 | 0.0 |
| 9 | 2V169S | 0.1690 | 0.9083 | 1.0947 | 0.5917 | 0.5917 | 2.2426 | 7.8000 | 7.8000 | 116.0 | 0.0 | 0.600E-03 | 0.000E+00 | 0.0 |
| 10 | 2V173S | 0.1730 | 0.9017 | 1.1127 | 0.5780 | 0.5780 | 2.3064 | 7.3000 | 7.3000 | 110.0 | 0.0 | 0.600E-03 | 0.000E+00 | 0.0 |

## Reference

1. W. E. Deeds, C. V. Dodd, and G. W. Scott, *Computer-Aided Design of Multifrequency Eddy-Current Tests for Layered Conductors with Multiple Property Variations,* ORNL/TM-6858 (in preparation).

NUREG/CR-0967
ORNL/NUREG/TM-335
Distribution
Category R5

*INTERNAL DISTRIBUTION*

1–2. Central Research Library
3. Document Reference Section
4–5. Laboratory Records Department
6. Laboratory Records, ORNL R.C.
7. ORNL Patent Section
8. Nuclear Safety Information Center
9. R. H. Bryan
10–14. W. E. Deeds
15–28. C. V. Dodd
29. W. P. Eatherly
30. T. G. Godfrey, Jr.
31. T. P. Hamrick
32–34. M. R. Hill
35. R. W. McClung
36. F. R. Mynatt
37. G. W. Scott
38. R. L. Shepherd
39. G. M. Slaughter
40. J. H. Smith
41. R. W. Balluffi (Consultant)
42. A. L. Bement, Jr. (Consultant)
43. W. R. Hibbard, Jr. (Consultant)
44. E. H. Kottcamp, Jr. (Consultant)
45. M. J. Mayfield (Consultant)
46. J. T. Stringer (Consultant)

*EXTERNAL DISTRIBUTION*

47. NRC, OFFICE OF NUCLEAR REGULATORY RESEARCH, Washington, DC 20555
Program Sponsor
48. DOE, OAK RIDGE OPERATIONS OFFICE, P.O. Box E, Oak Ridge, TN 37830
Assistant Manager, Energy Research and Development
49–50. DOE, TECHNICAL INFORMATION CENTER, P.O. Box 62, Oak Ridge, TN 37830
51–390. For Distribution Category R5 (10 — NTIS)

☆ U. S. GOVERNMENT PRINTING OFFICE: 79-640-245-295