2

NUREG/CR-5553
ORNL/TM-11505

Computer Programs for Eddy-Current Defect Studies

Prepared by J. R. Pate, C. V. Dodd

Oak Ridge National Laboratory

Prepared for
U.S. Nuclear Regulatory Commission

DISCLAIMER

AVAILABILITY NOTICE

Availability of Reference Materials Cited in NRC Publications

Most documents cited in NRC publications will be available from one of the following sources:

1. The NRC Public Document Room, 2120 L Street, NW, Lower Level, Washington, DC 20555

2. The Superintendent of Documents, U.S. Government Printing Office, P.O. Box 37082, Washington, DC 20013-7082

3. The National Technical Information Service, Springfield, VA 22161

Although the listing that follows represents the majority of documents cited in NRC publications, it is not intended to be exhaustive.

Referenced documents available for inspection and copying for a fee from the NRC Public Document Room include NRC correspondence and internal NRC memoranda; NRC Office of Inspection and Enforcement bulletins, circulars, information notices, inspection and investigation notices; Licensee Event Reports; vendor reports and correspondence; Commission papers; and applicant and licensee documents and correspondence.

The following documents in the NUREG series are available for purchase from the GPO Sales Program: formal NRC staff and contractor reports, NRC-sponsored conference proceedings, and NRC booklets and brochures. Also available are Regulatory Guides, NRC regulations in the *Code of Federal Regulations*, and *Nuclear Regulatory Commission Issuances*.

Documents available from the National Technical Information Service include NUREG series reports and technical reports prepared by other federal agencies and reports prepared by the Atomic Energy Commission, forerunner agency to the Nuclear Regulatory Commission.

Documents available from public and special technical libraries include all open literature items, such as books, journal and periodical articles, and transactions. *Federal Register* notices, federal and state legislation, and congressional reports can usually be obtained from these libraries.

Documents such as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings are available for purchase from the organization sponsoring the publication cited.

Single copies of NRC draft reports are available free, to the extent of supply, upon written request to the Office of Information Resources Management, Distribution Section, U.S. Nuclear Regulatory Commission, Washington, DC 20555.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at the NRC Library, 7920 Norfolk Avenue, Bethesda, Maryland, and are available there for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from the American National Standards Institute, 1430 Broadway, New York, NY 10018.

NUREG/CR--5553
TI90 013306

Computer Programs for Eddy-Current Defect Studies

Manuscript Completed: March 1990
Date Published: June 1990

Prepared by
J. R. Pate, C. V. Dodd

Oak Ridge National Laboratory
Operated by Martin Marietta Energy Systems, Inc.

Oak Ridge National Laboratory
Oak Ridge, TN 37831

Prepared for
Division of Engineering
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555
NRC FIN B0417

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

CONTENTS

COMPUTER PROGRAMS FOR EDDY-CURRENT DEFECT STUDIES*

J. R. Pate and C. V. Dodd

ABSTRACT

Several computer programs to aid in the design of eddy-current tests and probes have been written. The programs, written in Fortran, deal in various ways with the response to defects exhibited by four types of probes: the pancake probe, the reflection probe, the circumferential boreside probe, and the circumferential encircling probe. Programs are included which calculate the impedance or voltage change in a coil due to a defect, which calculate and plot the defect sensitivity factor of a coil, and which invert calculated or experimental readings to obtain the size of a defect. The theory upon which the programs are based is the Burrows point defect theory, and thus the calculations of the programs will be more accurate for small defects.

INTRODUCTION

This report contains computer programs for a number of eddy-current probes applied to various test situations. The probes analyzed and presented here are the pancake probe, the reflection probe, the circumferential boreside probe, and the circumferential encircling probe. Both absolute and differential probes are used for the last two cases. The programs are written to run in Ryan-McFarland Fortran, although some of them have been run using Microsoft Fortran and NDP Fortran with very little change. The programs have been run on PC-AT clones, using either an Intel 80286 or 80386 microprocessor. Grafmatic plotting software from Microcompatibles is used for the contour plots and also must be installed on the machines.

The purpose of these programs is to analyze the effects of defects in conductors for the design of eddy-current tests, probes, and instrumentation. We can use them first to calculate the changes in eddy-

*Research Sponsored by the Office of Nuclear Regulatory Research, Division of Engineering Safety, U. S. Nuclear Regulatory Commission under Interagency Agreement DOE 1886-8010-9B with the U.S. Department of Energy under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

current readings that various types of defects produce. The defects can be treated as point defects, as point defects averaged over the depth of the actual defects, or as point defects averaged over the volume of the defects. The effect of a defect can also be inverted so that the volume and depth of the defect can be calculated from the instrument readings. In addition, we can plot the "defect sensitivity factor" (DSF) for the various probes and designs. The DSF at a point in a conductor is proportional to the coil voltage produced by a point defect at that location.

The programs were written and used for the analysis of eddy-current steam generator problems of interest to the Nuclear Regulatory Commission (NRC). Experimental measurements have been made that verify the accuracy of these programs for the NRC problems. However, they are quite general and may be used for a wide number of different eddy-current problems that have similar geometries. The programs are relatively fast running and sufficiently accurate so that eddy-current design studies of specific problems can be run with a minimum investment of time and equipment.

The accuracy of the impedance or voltage calculations for the cases without the defects is on the order of 0.005%. This is more accurate than the coils and standards can be constructed. The accuracy of the impedance change produced by the defect depends in general on the size of the defect. In addition, several different methods for calculating the defect are given. The accuracy of the theory increases as the defect size decreases, but the experimental error increases. For the optimum defect size, the accuracy of the impedance change due to the defect is on the order of 20%. However, most of the problem of eddy-current steam generator inspection is the elimination of the effect of unwanted property variations on the defect measurements, and the theory and programs listed here are well suited to study this problem.

This report is divided into five sections, one for each of the four coil types used and a fifth for common subroutines. We have attempted to make the description of each program sufficient (except for the subroutines), but it will probably be necessary to refer to an earlier program discussion in some cases. We also have given the equations that the programs are evaluating, along with the program variables used, so that the reader can more easily modify the programs for other uses.

The defect theory is based on the model of M. L. Burrows, with a correction of the sign of the term. Burrows gave an expression for both a "current defect" and a "magnetic defect," which only occurs in ferromagnetic materials. While the theory and discussion do carry the terms for ferromagnetic materials, the programs are written for non-ferromagnetic materials, with a relative permeability of unity. None of the equations for the "magnetic defect" are given. The equation[1] for the voltage V_{2d} induced in coil 2 by the perturbation of the eddy-current flow from coil 1 due to the presence of a defect is:

$$\frac{V_{2d}}{I_1} = - \left[\frac{3}{2} \sigma\omega^2 \left(\frac{A_2}{I_2} \right) \left(\frac{A_1}{I_1} \right) \right] \times \left[Vol \; \alpha_{22} \right]$$

where I_1 and I_2 are the currents flowing in coil 1 and coil 2, respectively, and A_1 and A_2 are the vector potentials at the defect due to the currents flowing in the coils. The term σ is the conductivity of the material, ω is the angular frequency, *Vol* is the volume of the defect and α_{22} is a shape and orientation factor for the defect. It is equal to unity for a spherical defect and assumed to be unity for all the other cases in this report.

The theory and equations are presented in this report only to aid in the explanation of the computer programs. A complete and rigorous derivation is left for other work.[2-4]

PANCAKE COIL PROGRAMS

The programs in this section perform various functions relating to the effect of a defect in a single conducting plate on the impedance of a pancake coil. Figure 1 shows a cutaway view of a pancake coil. In general, the shorter the coil, the greater its sensitivity to the property changes in the material. Thus, short flat coils shaped like a pancake, with a small relative inner diameter, are more sensitive to defects in the conductor. This is reasonable since the turns of wire are closer to the conductor than in a long narrow solenoid. However, the theory and programs work for any shape of coil and conductor with this geometry.

ORNL-DWG 90-8085

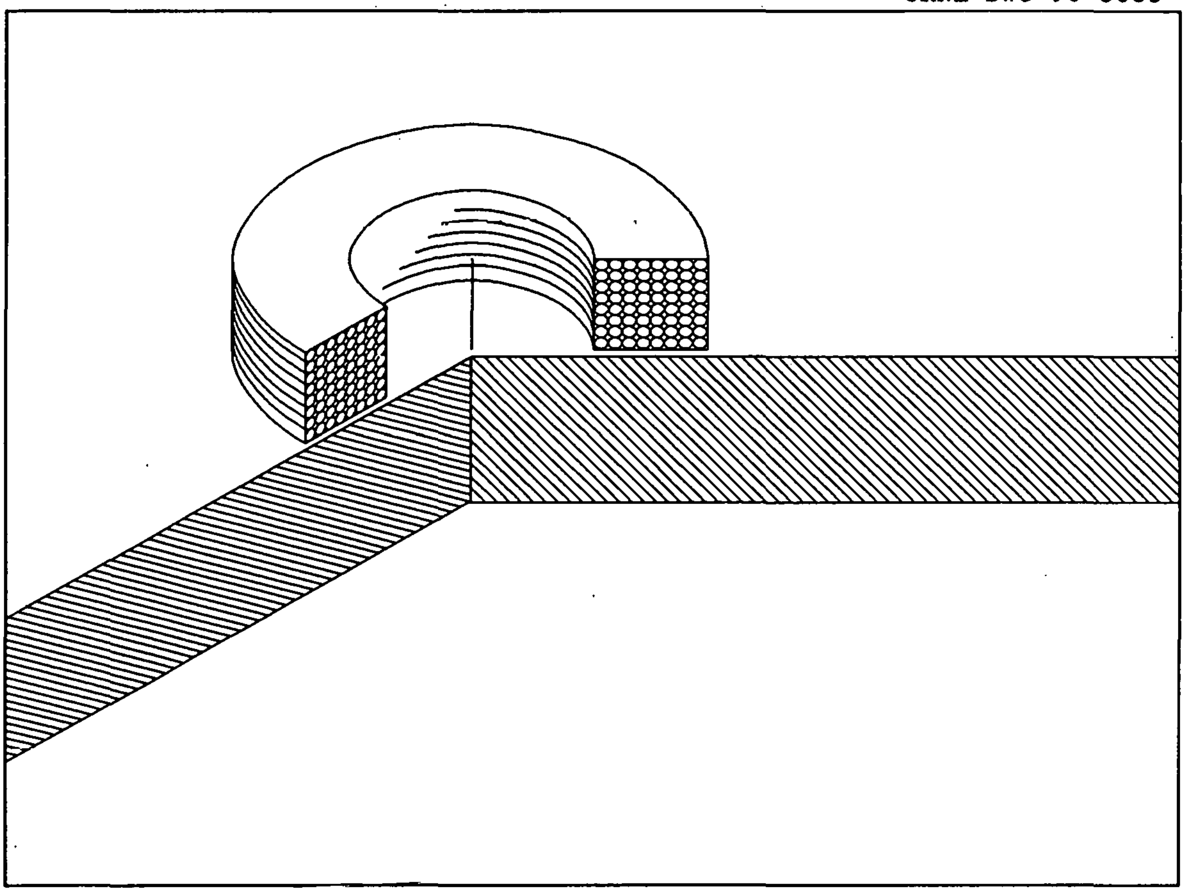

Fig. 1. Cutaway view of a pancake coil above a conducting plate.

Figure 2 shows a cross section of a pancake coil above a plate. The coil has been labeled with variables relating to its geometry which are common to all of the programs in this section. The coil is above a spherical defect, located at r and z with respect to the coil and the top surface of the plate, respectively.

The basic equations for the programs are presented below, with a detailed

ORNL-DWG 90-8086

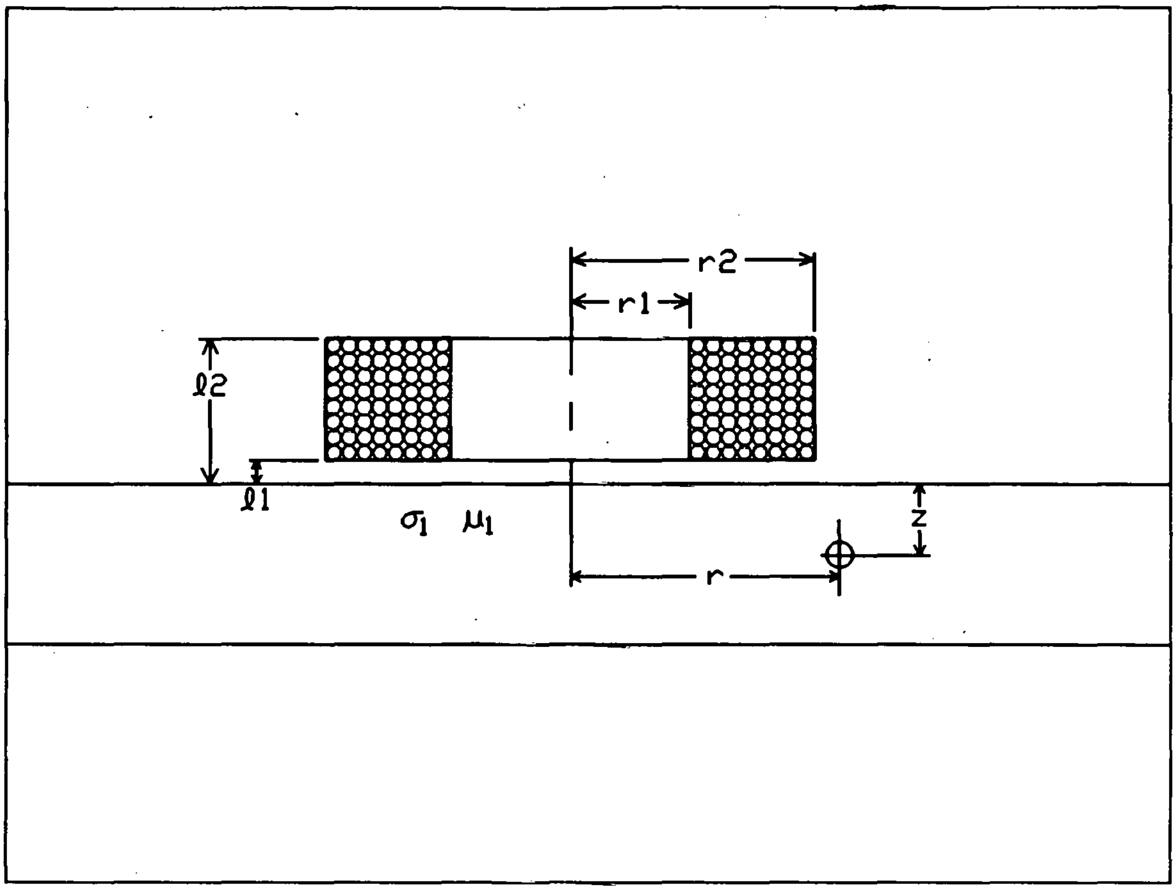

Fig. 2. Cross section of a pancake coil above a conducting plate.

derivation given elsewhere. The impedance change due to a small spherical defect at r,z is:

$$Z_{nd}(r,z) = \frac{-3(\omega\mu\sigma_1\bar{r}^2)}{2\pi I_{air}} Vol_n \left[\int_0^\infty \frac{J(r_2,r_1)\ J_1(\alpha r)(e^{-\alpha\ell_1} - e^{-\alpha\ell_2})\ F(\alpha,\alpha_1,z)}{\alpha^3}\, d\alpha \right]^2 \quad (1)$$

$$\text{where} \quad J(r_2,r_1) = \int_{\alpha r_1}^{\alpha r_2} xJ_1(x)\,dx \quad (2)$$

$$\text{and} \quad \alpha_1 = (\alpha^2 + j\omega\mu\sigma_1\bar{r}^2)^{1/2} \quad (3)$$

The term I_{air} is related to the air inductance of the coil and is:

$$I_{air} = \int_0^\infty \frac{1}{\alpha^6}\ [J(r_2,r_1)]^2\ 2[\alpha(\ell_2-\ell_1) + \exp(-\alpha\ell_2+\alpha\ell_1) - 1]\ d\alpha \quad (4)$$

The term $F(\alpha,\alpha_1,z)$ depends on the number of planar layers of conductors. For the simple case of a semi-infinite plane beneath the coil we have:

$$F(\alpha,\alpha_1,z) = \frac{\alpha e^{\alpha_1 z}}{\alpha + \alpha_1} \tag{5}$$

We can take the square root of both sides of the equation, multiply by a weighting factor $rJ_1(ar)$, and integrate over the signal produced as we scan across the defect:

$$\int_0^\infty rJ_1(ar)\sqrt{-Z_{nd}(r,z)}\,dr =$$

$$\left[\frac{3\omega\mu\sigma_1\bar{r}^2}{2\pi I_{air}} Vol_n\right]^{1/2} \int_0^\infty rJ_1(ar)dr \int_0^\infty \frac{J(r_2,r_1)J_1(\alpha r)(e^{-\alpha\ell_1} - e^{-\alpha\ell_2})F(\alpha,\alpha_1,z)}{\alpha^3}\, d\alpha \tag{6}$$

We shall now use the Fourier-Bessel Integral, which is:

$$f(a) = \int_0^\infty rJ_1(ar)\int_0^\infty \alpha J_1(\alpha r)f(\alpha)\,d\alpha\,dr \tag{7}$$

to simplify the above equation. The result is:

$$\int_0^\infty rJ_1(ar)\sqrt{-Z_{nd}(r,z)}\,dr = \left[\frac{3\omega\mu\sigma_1\bar{r}^2}{2\pi I_{air}}\mathrm{Vol}_n\right]^{1/2}\frac{J(r_2,r_1)(e^{-a\ell_1} - e^{-a\ell_2})F(a,a_1,z)}{a^4} \tag{8}$$

We now transpose the equation and simplify the terms using some definitions:

$$\sqrt{\mathrm{Vol}_n}\, e^{a_1 z} = CM_0 e^{i\theta} \tag{9}$$

where

$$C = \sqrt{\frac{2\pi I_{air}}{3\omega\mu\sigma_1\bar{r}^2}}\ \frac{a^3}{J(r_2,r_1)[\exp(-a\ell_1) - \exp(-a\ell_2)]} \tag{10}$$

$$M_0 = \mathrm{Mag}\left[(a + a_1)\int_0^\infty \sqrt{-Z_{nd}(r,z)}\, rJ_1(ar)\,dr\right] \tag{11}$$

and

$$\theta = \text{Pha}\left[(a + a_1) \int_0^{\infty} \sqrt{-Z_{nd}(r,z)}\; rJ_1(ar)\; dr \right] \tag{12}$$

Then the defect depth and volume can be calculated from the magnitude and phase shift of the expression and the real and imaginary parts of a_1, which we will call x and y.

Thus $$z = \theta / y \tag{13}$$

and $$Vol_n = [CM_0 \exp(-x\theta/y)]^2 \tag{14}$$

For the simple case of the pancake coil above the semi-infinite conducting plate, we are able to directly invert the defect signal for a spherical defect and obtain the volume and depth of the defect. However, the equations become more messy for the case of a plate of thickness c. We generate functions that cannot be solved directly so we must use a lookup table. The program PCBLDF is used for this purpose and is discussed below.

PCBLDF builds a magnitude and phase lookup file

Program PCBLDF builds a lookup file containing the magnitude and phase of the following integral

$$\int_0^{\infty} \frac{1}{\alpha^3} J(r_2,r_1) \left[\frac{J_0(\alpha r_1)-J_0(\alpha r_2)}{\alpha} \right] [\exp(-\alpha \ell_1)-\exp(-\alpha \ell_2)]$$

$$\left[\frac{\alpha(\alpha_1+\alpha)\exp(\alpha_1(2c+z))+\alpha(\alpha_1-\alpha)\exp(-\alpha_1 z)}{-(\alpha-\alpha_1)^2+(\alpha_1+\alpha)^2\exp(2\alpha_1 c)} \right] d\alpha \qquad (15)$$

at different depths z in a conducting plate. This is similar to the equation that we had for the semi-infinite plate except the term for $F(\alpha,\alpha_1,z)$ has been replaced by the term in the final set of large brackets. We have replaced the semi-infinite plate with a plate of thickness c. Also, we have used a weighting term so that the data used for the inversion are concentrated near the coil windings. This allows us to use data that have a higher signal-to-noise ratio. The weighting function is unity over the dimensions of the coil, from r_1 to r_2, and zero elsewhere. The result of this particular choice is that the function $J_0(\alpha r_1)-J_0(\alpha r_2)$ is produced.

This lookup file can be used by programs PCINV and PCRTSCAN to calculate the depth and volume of defects in the plate.

<u>Summary</u>

1. Dimension arrays and declare variable types.
2. Initialize variables.
3. Select a value for the depth in the plate at which to calculate the integral.
4. Calculate the integral.
5. Store the depth along with the magnitude and phase of the integral at this depth.
6. Loop to 3 until the calculations have been done at points all the way through the plate.

<u>Variables</u>

Starred variables must be assigned by the user.

AIRIND* The inductance in henries of the coil in air.

FREQ*	The operating frequency in hertz.
L*	The length of the coil. The value is input in inches and normalized by the program.
L1*	The lift-off of the coil. The value is input in inches and normalized by the program.
L2	The normalized distance between the top of the coil and the plate. This value is computed by the program.
LOD*	The number of the I/O unit connected to the output data file.
LOU	The number of the I/O unit connected to the printer.
MZT*	The number of depths throughout the plate at which the program does the calculations.
NS	The side of the plate which is nearer to the point at which the integral is being calculated. If NS = 1, the point is closer to the near side; if NS = 2, the point is closer to the far side.
NZT*	The number of parts into which each defect is divided to perform the calculations.
R1*	The inner radius of the coil. The value is input in inches and normalized by the program.
R2*	The outer radius of the coil. The value is input in inches and normalized by the program.
R3	The mean radius of the coil in inches.
RHSMAG	The magnitude of the integral at a certain depth in the plate.
RHSPHA	The phase in degrees of the integral at a certain depth in the plate.
RHO1*	The electrical resistivity of the plate in $\mu\Omega$-cm.
T1*	The thickness of the plate. The value is input in inches and normalized by the program.
TRN*	The number of turns in the coil.
U1*	The relative magnetic permeability of the plate.
WUSRR	The product of the angular frequency, the magnetic permeability, the electrical conductivity, and the square of the mean coil radius.
ZD	The normalized distance from the side of the plate where the defect is located to the bottom of the defect. A negative number.
ZD2	The normalized distance from the near side of the plate to the center of the defect. A negative number.
ZMSTEP	The normalized axial distance between the depths at which the program does the calculations.

Notes

1. Program PCBLDF assists program PCINV in the inversion process. The programs use the equation

$$\int_{r_1}^{r_2} [-Z_{nd}(r,z)]^{1/2}\, dr = \tag{16}$$

$$\left[\frac{3\omega\mu\sigma_1 \bar{r}^2 Vol_n \alpha_{22}}{2\pi I_{air}} \right]^{1/2} \left[\int_0^{\infty} \frac{1}{\alpha^3} J(r_2,r_1) \left[\frac{J_0(\alpha r_1)-J_0(\alpha r_2)}{\alpha} \right] \right.$$

$$\left. [\exp(-\alpha \ell_1)-\exp(-\alpha \ell_2)] \left[\frac{\alpha(\alpha_1+\alpha)\exp(\alpha_1(2c+z))+\alpha(\alpha_1-\alpha)\exp(-\alpha_1 z)}{-(\alpha-\alpha_1)^2+(\alpha_1+\alpha)^2\exp(2\alpha_1 c)} \right] d\alpha \right]$$

Program PCBLDF calculates the integral on the righthand side of this equation. It is clear that since Vol_n, the defect volume, can be factored out of the right side of the equation, it has no effect on the phase of the right side of the equation. Therefore, the phase depends only upon z, the depth of the defect. So when program PCINV calculates the integral on the left-hand side, it obtains a value for the phase of this integral, and it can search the table built by program PCBLDF until it finds this phase. The depth in the table corresponding to this phase is the depth of the defect. Then, knowing the magnitude of both integrals (the magnitude of the integral on the right was calculated and stored in the file by program PCBLDF, and the magnitude of the integral was previously calculated by program PCINV), program PCINV can solve for the volume of the defect.

Integration Section of Program PCBLDF

Symbol definitions

The following are definitions of the symbols used to describe the program variables appearing in the integration section of this program. All lengths are normalized unless otherwise noted.

α	Integration variable
α_1	$(\alpha^2+j\omega\mu\sigma_1\bar{r}^2)^{1/2}$
c	Plate thickness
$J(x_2,x_1)$	Integral of $xJ_1(x)$ with respect to x from αx_1 to αx_2
$J_0(x)$	Bessel function of the first kind of order 0
$J_1(x)$	Bessel function of the first kind of order 1
ℓ	Length of coil
ℓ_1	Lift-off of coil
ℓ_2	Distance from top of coil to plate
μ	Relative magnetic permeability of plate
$\underline{r}$	Coil-to-defect radial distance
$\bar{r}$	Mean radius of coil in inches
r_1	Inner radius of coil
r_2	Outer radius of coil
σ_1	Conductivity of plate
ω	Angular frequency at which circuit is driven
z	Depth to center of defect

Variables appearing in the integration section

Program variable	Symbolic equivalent
AN	α
AN2	α^2
AN4	α^4
ANJR1	$J(r_1,0)/\alpha^3$
ANJR2	$J(r_2,0)/\alpha^3$
ANR1	αr_1
ANR2	αr_2
ARHSI	$\mathrm{Im}\left[\int_0^\infty \frac{1}{\alpha^3} J(r_2,r_1)\left[\frac{J_0(\alpha r_1)-J_0(\alpha r_2)}{\alpha}\right]\right.$

$$\left.[\exp(-\alpha\ell_1)-\exp(-\alpha\ell_2)]\left[\frac{\alpha(\alpha_1+\alpha)\exp(\alpha_1(2c+z))+\alpha(\alpha_1-\alpha)\exp(-\alpha_1 z)}{-(\alpha-\alpha_1)^2+(\alpha_1+\alpha)^2\exp(2\alpha_1 c)}\right] d\alpha\right]$$

ARHSR	$\mathrm{Re}\left[\int_0^\infty \frac{1}{\alpha^3} J(r_2,r_1)\left[\frac{J_0(\alpha r_1)-J_0(\alpha r_2)}{\alpha}\right]\left[\exp(-\alpha\ell_1)-\exp(-\alpha\ell_2)\right]\left[\frac{\alpha(\alpha_1+\alpha)\exp(\alpha_1(2c+z))+\alpha(\alpha_1-\alpha)\exp(-\alpha_1 z)}{-(\alpha-\alpha_1)^2+(\alpha_1+\alpha)^2\exp(2\alpha_1 c)}\right]d\alpha\right]$
ASTP	$d\alpha$
DCOYNT1	$\mathrm{Re}[\exp(\mathrm{Im}(2\alpha_1 c))]$
DCOYNZ	$\mathrm{Re}[\exp(\mathrm{Im}(\alpha_1 z))]$
DEXNT1	$\exp[\mathrm{Re}(2\alpha_1 c)]$
DEXNZ	$\exp[\mathrm{Re}(\alpha_1 z)]$
DNI	$\mathrm{Im}[-(\alpha-\alpha_1)^2+(\alpha_1+\alpha)^2\exp(2\alpha_1 c)]$
DNR	$\mathrm{Re}[-(\alpha-\alpha_1)^2+(\alpha_1+\alpha)^2\exp(2\alpha_1 c)]$
DSIYNT1	$\mathrm{Im}[\exp(\mathrm{Im}(2\alpha_1 c))]$
DSIYNZ	$\mathrm{Im}[\exp(\mathrm{Im}(\alpha_1 z))]$
FI	$\mathrm{Im}\left[\frac{\alpha(\alpha_1+\alpha)\exp(2\alpha_1 c)\exp(\alpha_1 z)+\alpha(\alpha_1-\alpha)\exp(-\alpha_1 z)}{-(\alpha-\alpha_1)^2+(\alpha_1+\alpha)^2\exp(2\alpha_1 c)}\right]$
FR	$\mathrm{Re}\left[\frac{\alpha(\alpha_1+\alpha)\exp(2\alpha_1 c)\exp(\alpha_1 z)+\alpha(\alpha_1-\alpha)\exp(-\alpha_1 z)}{-(\alpha-\alpha_1)^2+(\alpha_1+\alpha)^2\exp(2\alpha_1 c)}\right]$
J01MJ02	$J_0(\alpha r_1)-J_0(\alpha r_2)$
J0R1	$J_0(\alpha r_1)$
J0R2	$J_0(\alpha r_2)$
JANR21	$J(r_2,r_1)/\alpha^3$
NMI	$\mathrm{Im}[\alpha(\alpha_1+\alpha)\exp(2\alpha_1 c)\exp(\alpha_1 z)+\alpha(\alpha_1-\alpha)\exp(-\alpha_1 z)]$
NMI1	$\mathrm{Im}[\alpha(\alpha_1+\alpha)\exp(2\alpha_1 c)\exp(\alpha_1 z)]$
NMI2	$\mathrm{Im}[\alpha(\alpha_1-\alpha)\exp(-\alpha_1 z)]$
NMR	$\mathrm{Re}[\alpha(\alpha_1+\alpha)\exp(2\alpha_1 c)\exp(\alpha_1 z)+\alpha(\alpha_1-\alpha)\exp(-\alpha_1 z)]$
NMR1	$\mathrm{Re}[\alpha(\alpha_1+\alpha)\exp(2\alpha_1 c)\exp(\alpha_1 z)]$
NMR2	$\mathrm{Re}[\alpha(\alpha_1-\alpha)\exp(-\alpha_1 z)]$
RDEXNZ	$\exp[\mathrm{Re}(-\alpha_1 z)]$
RFAC	$\frac{1}{\alpha^3} J(r_2,r_1)\left[\frac{J_0(\alpha r_1)-J_0(\alpha r_2)}{\alpha}\right][\exp(-\alpha\ell_1)-\exp(-\alpha\ell_2)]\ d\alpha$

RHSI $\quad \mathrm{Im}\left[\frac{1}{\alpha^3} J(r_2,r_1)\left[\frac{J_0(\alpha r_1)-J_0(\alpha r_2)}{\alpha}\right]\right.$

$$\left.[\exp(-\alpha\ell_1)-\exp(-\alpha\ell_2)]\left[\frac{\alpha(\alpha_1+\alpha)\exp(\alpha_1(2c+z))+\alpha(\alpha_1-\alpha)\exp(-\alpha_1 z)}{-(\alpha-\alpha_1)^2+(\alpha_1+\alpha)^2\exp(2\alpha_1 c)}\right] d\alpha\right]$$

RHSR $\quad \mathrm{Re}\left[\frac{1}{\alpha^3} J(r_2,r_1)\left[\frac{J_0(\alpha r_1)-J_0(\alpha r_2)}{\alpha}\right]\right.$

$$\left.[\exp(-\alpha\ell_1)-\exp(-\alpha\ell_2)]\left[\frac{\alpha(\alpha_1+\alpha)\exp(\alpha_1(2c+z))+\alpha(\alpha_1-\alpha)\exp(-\alpha_1 z)}{-(\alpha-\alpha_1)^2+(\alpha_1+\alpha)^2\exp(2\alpha_1 c)}\right] d\alpha\right]$$

SMAIR $\quad \displaystyle\int_0^\infty \frac{2(\alpha\ell+\exp(-\alpha\ell)-1)[J(r_2,r_1)]^2\, d\alpha}{\alpha^6}$

WUSRR2 $\quad (\omega\mu\sigma_1\bar{r}^2)^2$

XN $\quad \mathrm{Re}[\alpha_1]$

YN $\quad \mathrm{Im}[\alpha_1]$

<u>Notes for the integration section</u>

I1. The program has been tested and found to be accurate enough with a step size of 0.01 and with the upper limit of the integration equal to 50.

Listing

```
      PROGRAM PCBLDF
C     November 14, 1988
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 L,L1,L2,JANR21,LHSPHA
      REAL*8 NMR1A,NMR1B,NMR1,NMR2,NMR
      REAL*8 NMI1A,NMI1B,NMI1,NMI2,NMI
      REAL*8 JOR1,JOR2,J01MJ02
      DATA LOU/8/,LOD/40/,PI/3.141592653/
      DATA FREQ/500./,RHO1/4.054/,U1/1.0/
      DATA TRN/800./,T1/0.25/,ASTP/0.01/
      DATA AIRIND/6.252919E-03/,MZT/25/,NZT/1/
C     TIME AND DATE ARE PRINTED
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT(' PCBLDF    TIME ',I2,':',I2,':',I2
     *,'  DATE ',I2,'/',I2,'/',I2)
      WRITE(LOU,5)
    5 FORMAT(6X,'IN RAD',4X,'OT RAD',4X,'LENGTH',4X,'LIF OFF'
     *,3X,'CLADTH')
      R1=0.1000
      R2=0.4100
      L=0.1000
      L1=0.01
      R3=0.5*(R1+R2)
      WRITE(LOU,10)R1,R2,L,L1,T1
      R1=R1/R3
      R2=2.0-R1
      L=L/R3
      L1=L1/R3
      T1=T1/R3
      L2=L+L1
      WUSRR=0.5093979*U1*R3*R3*FREQ/RHO1
      WRITE(LOU,15)R1,R2,L,L1,T1
   10 FORMAT('ACT    ',5(F7.4,3X))
   15 FORMAT('NOR    ',5(F7.4,3X))
      WRITE(LOU,20)R3,FREQ,RHO1,U1,WUSRR
   20 FORMAT(' RBAR',F7.4,' FREQ= ',1PE13.6,' RHO=',0PF9.4,
     *' PERM=',F7.3,' WUSRR=',F9.4)
      SMAIR=AIRIND*(L*(R2-R1))**2/(0.0254*4.E-07*TRN*TRN*R3*PI*PI)
C
      WUSRR2=WUSRR*WUSRR
      ZMSTEP=T1/MZT
      DO 1200 MZ=0,MZT
      IF(MZ.GT.(0.5*MZT)) THEN
C    Far side defect
      NS=2
      ZM=(MZT-MZ)*ZMSTEP
```

```
      ZD=-2.*ZM
      ZD2=-MZ*ZMSTEP
      ELSE
C   Near side defect
      NS=1
      ZM=MZ*ZMSTEP
      ZD=-2.*ZM
      ZD2=-ZM
      END IF
C      NZT=20
      ARHSR=0.
      ARHSI=0.
      DO 1180 NZ=1,NZT
      Z=(REAL(NZ)-0.5)*ZD/NZT
      IF(NS.EQ.2)Z=-T1-Z
C      Z=-T1-(FLOAT(NZ)-0.5)*ZD/FLOAT(NZT)
      SRHSR=0.
      SRHSI=0.
      AN=0.
      DO 1150 K=1,5000
      AN=AN+ASTP
      AN2=AN*AN
      AN4=AN2*AN2
C      WRITE(0,37)K,AN
   37 FORMAT(I3,F8.2,F8.2,F8.2)
      XN=DSQRT(0.5*(AN2+DSQRT(AN4+WUSRR2)))
      YN=WUSRR/(2.*XN)
C   Definitions of often-used quantities
      DEXNT1=DEXP(2*XN*T1)
      DEXNZ=DEXP(XN*Z)
      RDEXNZ=1/DEXNZ
      DSIYNT1=DSIN(2*YN*T1)
      DSIYNZ=DSIN(YN*Z)
      DCOYNT1=DCOS(2*YN*T1)
      DCOYNZ=DCOS(YN*Z)
C   Real part of the first term of the numerator.
      NMR1A=AN*(XN+AN)*DEXNT1*DCOYNT1
      NMR1A=NMR1A-AN*YN*DEXNT1*DSIYNT1
      NMR1A=NMR1A*DEXNZ*DCOYNZ
      NMR1B=AN*YN*DEXNT1*DCOYNT1
      NMR1B=NMR1B+AN*(XN+AN)*DEXNT1*DSIYNT1
      NMR1B=NMR1B*DEXNZ*DSIYNZ
      NMR1=NMR1A-NMR1B
C   Imaginary part of the first term of the numerator.
      NMI1A=AN*YN*DEXNT1*DCOYNT1
      NMI1A=NMI1A+AN*(XN+AN)*DEXNT1*DSIYNT1
      NMI1A=NMI1A*DEXNZ*DCOYNZ
      NMI1B=AN*(XN+AN)*DEXNT1*DCOYNT1
      NMI1B=NMI1B-AN*YN*DEXNT1*DSIYNT1
      NMI1B=NMI1B*DEXNZ*DSIYNZ
      NMI1=NMI1A+NMI1B
```

```
C    Real part of the second term of the numerator.
      NMR2=AN*(XN-AN)*RDEXNZ*DCOYNZ
      NMR2=NMR2+AN*YN*RDEXNZ*DSIYNZ
C    Imaginary part of the second term of the numerator.
      NMI2=AN*YN*RDEXNZ*DCOYNZ
      NMI2=NMI2-AN*(XN-AN)*RDEXNZ*DSIYNZ
C    Real part of the denominator.
      DNR=-(AN-XN)*(AN-XN)+YN*YN
      DNR=DNR+DEXNT1*((XN+AN)*(XN+AN)-(YN*YN))*DCOYNT1
      DNR=DNR-DEXNT1*(2*YN*(XN+AN))*DSIYNT1
C    Imaginary part of the denominator.
      DNI=2*YN*(AN-XN)
      DNI=DNI+DEXNT1*(2*(XN+AN)*YN)*DCOYNT1
      DNI=DNI+DEXNT1*((XN+AN)*(XN+AN)-YN*YN)*DSIYNT1
C
      NMR=NMR1+NMR2
      NMI=NMI1+NMI2
      FR=(NMR*DNR+NMI*DNI)/(DNR*DNR+DNI*DNI)
      FI=(NMI*DNR-NMR*DNI)/(DNR*DNR+DNI*DNI)
      CALL BESSEL(ANJR2,AN,R2)
      CALL BESSEL(ANJR1,AN,R1)
      JANR21=ANJR2-ANJR1
      ANR1=AN*R1
      CALL BESO(JOR1,ANR1)
      ANR2=AN*R2
      CALL BESO(JOR2,ANR2)
      J01MJO2=JOR1-JOR2
      RFAC=JANR21*J01MJO2*(DEXP(-AN*L1)-DEXP(-AN*L2))*ASTP/AN
      RHSR=RFAC*FR
      RHSI=RFAC*FI
C       WRITE(0,1149)AN,RHSR,RHSI
 1149 FORMAT(F9.3,3X,D11.3,3X,D11.3)
      SRHSR=SRHSR+RHSR
      SRHSI=SRHSI+RHSI
 1150 CONTINUE
      ARHSR=ARHSR+SRHSR
      ARHSI=ARHSI+SRHSI
 1180 CONTINUE
      ARHSR=ARHSR/NZT
      ARHSI=ARHSI/NZT
      RHSMAG=DSQRT(ARHSR*ARHSR+ARHSI*ARHSI)
      RHSPHA=ATAN2(ARHSI,ARHSR)*180./PI
      WRITE(LOD,1205)ZD2,RHSMAG,RHSPHA
      WRITE(LOU,1205)ZD2,RHSMAG,RHSPHA
      WRITE(0,1205)ZD2,RHSMAG,RHSPHA
 1200 CONTINUE
 1205 FORMAT(F6.3,1X,D11.3,1X,F7.2)
      STOP
      END
```

PCDSF calculates mag. and phase of DSF for lattice of points

Program PCDSF calculates the magnitude and phase of the defect sensitivity factor of a pancake coil at a lattice of points throughout a plate. It stores the calculations so that they can be plotted by program PCDSFPLT. In Fig. 3 we show a pancake coil above a conducting plate, with the plate divided into a lattice of points.

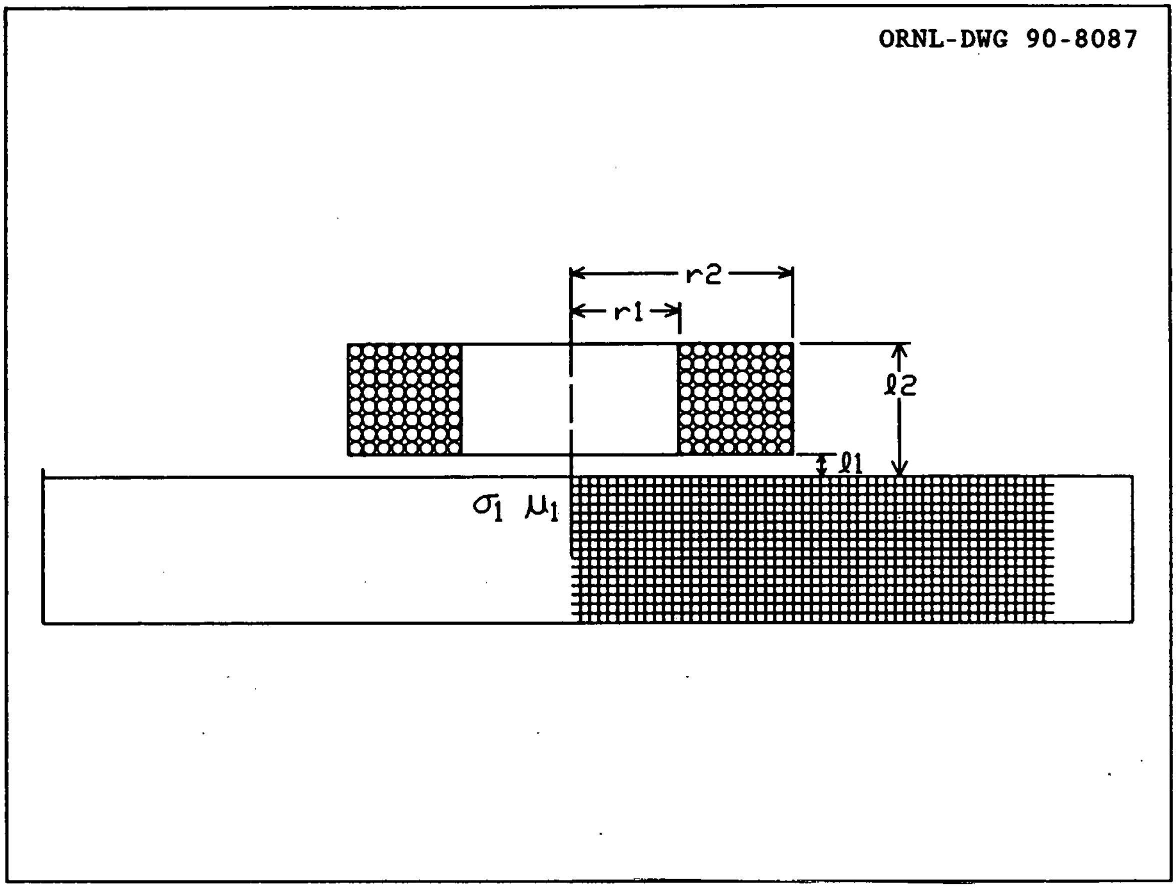

Fig. 3. Pancake coil with the conductor divided into a lattice of points.

The defect sensitivity factor for the pancake coil above a conducting plate at a point *r,z* in the plate is:

$$DSF(r,z) = \frac{-3(\omega\mu\sigma_1\bar{r}^2)}{2\pi I_{air}} \left[\int_0^\infty \frac{J(r_2,r_1)\; J_1(\alpha r)(e^{-\alpha\ell_1} - e^{-\alpha\ell_2})\; F(\alpha,\alpha_1,z)}{\alpha^3} d\alpha \right]^2 \qquad (17)$$

The various terms in the equation are explained in Eqs. (2), (3), (4), and (15). Since the points to be calculated are in a regular lattice, and the terms in the numerical integration are factorable, a considerable amount of computation time can be saved by performing the computations that vary over the r dimension and the z dimension separately for each value of α. The values of the different factors that depend only on r or z are computed once and stored in an array. Then for the different locations the array values are multiplied and then summed to compute the integral. The output from this program is stored in the data file PCDSF.DAT.

Summary

1. Dimension arrays and declare variable types.
2. Initialize variables.
3. Write the information about the coil and plate geometry to a file.
4. Do the parts of the integration which are independent of the position of the point.
5. Choose a value for the axial coordinate of the position of the point. Do the parts of the integration that depend only upon this coordinate.
6. Choose a value for the radial coordinate of the position of the point. Complete the integration for this point.
7. Loop to 6 until done.
8. Loop to 5 until done.
9. Write the results to a data file.

Variables

DELTAR*	The normalized radial distance between adjacent points at which the program calculates the defect sensitivity factor.
DELTAZ	The normalized axial distance between adjacent points at which the program calculates the defect sensitivity factor.
DSFI	The imaginary part of the defect sensitivity factor.
DSFM	The magnitude of the defect sensitivity factor.
DSFP	The phase in radians of the defect sensitivity factor.
DSFR	The real part of the defect sensitivity factor.
FREQ*	The operating frequency in hertz.
L*	The length of the coil. The value is input in inches and normalized by the program
L1*	The lift-off of the coil. The value is input in inches and normalized by the program.
LOD*	The number of the I/O unit connected to the output data file.
LOU	The number of the I/O unit connected to the printer.
NRT*	The total number of points in the radial direction at which the defect sensitivity factor is to be calculated.
NZT*	The total number of points in the axial direction at which the defect sensitivity factor is to be calculated.
R1*	The inner radius of the coil. The value is input

	in inches and normalized by the program.
R2*	The outer radius of the coil. The value is input in inches and normalized by the program.
R3	The mean radius of the coil in inches.
RD	The normalized radial distance between the coil axis and the point at which the program is calculating the defect sensitivity factor.
RHO1*	The resistivity of the plate in μΩ-cm.
T1*	The thickness of the plate. The value is input in inches and normalized by the program.
TRN*	The number of turns in the coil.
U1*	The relative magnetic permeability of the plate.
WUSRR	The product of the angular operating frequency, the magnetic permeability of the plate, the electrical conductivity of the plate, and the square of the mean radius of the coil.

Notes

1. The integration in this program is very similar to the integration in program PCAVZSCN. Program PCAVZSCN calculates the impedance change in a pancake coil due to a defect, but in order to do this, it must calculate the defect sensitivity factor. The main difference between the programs is that program PCDSF calculates the defect sensitivity factor at different depths in the plate while program PCAVZSCN calculates the average defect sensitivity factor over a range of depths.

Sample output

Output sent to printer:

```
PCDSF      TIME  7:44:26  DATE  8/ 9/89
     IN RAD    OT RAD    LENGTH    LIFTOFF   CLADTH
ACT  0.1000    0.4250    0.0500    0.0100    0.2500
NOR  0.3810    1.6190    0.1905    0.0381    0.9524
 RBAR 0.2625 FREQ=6.000000E+02  RHO=4.0540  PERM=1.000  WUSRR=5.1950
 NORM IMPD:RL  0.190682 IM  0.777969 AIR IND 1.099840E-02
```

Partial listing of output stored on PCDSF.DAT:

```
      40         40
     0.05000        0.02442
     0.38095        1.61905
     0.19048        0.03810
     0.95238
    1       1  0.17513D-03  0.24106D+01
    1       2  0.16525D-03  0.23596D+01
    1       3  0.15496D-03  0.23083D+01
    1       4  0.14450D-03  0.22563D+01
    1       5  0.13407D-03  0.22037D+01
    1       6  0.12385D-03  0.21504D+01
```

1	7	0.11397D-03	0.20963D+01
1	8	0.10454D-03	0.20413D+01
1	9	0.95618D-04	0.19856D+01
1	10	0.87259D-04	0.19292D+01

Listing

```
      PROGRAM PCDSF
C     VERSION November 16, 1988
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 L,L1
      DIMENSION S1(6),S2(6),ERR(6),SMZDFR(100,40),SMZDFI(100,40),RJ(100)
      DATA LOU/8/,PI/3.141592653/,LOD/39/
      DATA S1/0.005,0.02,0.05,0.1,0.5,2./
      DATA S2/1.0,2.0,5.0,10.0,50.0,200.0/
      DATA ERR/0.1,0.01,0.001,1.E-4,1.E-5,1.E-6/
      DATA FREQ/600./,RHO1/4.054/,U1/1.0/,NRT/40/,NZT/40/
      DATA TRN/1000./,T1/0.250/,DELTAR/0.05/
C     TIME AND DATE ARE PRINTED
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT('PCDSF     TIME ',I2,':',I2,':',I2,
     *'  DATE ',I2,'/',I2,'/',I2)
      WRITE(LOU,5)
    5 FORMAT(5X,'IN RAD',4X,'OT RAD',4X,'LENGTH',4X,'LIFTOFF'
     *,3X,'CLADTH')
C     COIL P250A
      R1=0.100
      R2=0.425
      L=0.050
      L1=0.01
      R3=0.5*(R1+R2)
      WRITE(LOU,10)R1,R2,L,L1,T1
      R1=R1/R3
      R2=2.0-R1
      L=L/R3
      L1=L1/R3
      T1=T1/R3
      DELTAZ=T1/(NZT-1)
      WUSRR=0.5093979*U1*R3*R3*FREQ/RHO1
C   Open the file for input data for PCDSFPLT.
      OPEN(LOD,FILE='PCDSF.DAT',STATUS='NEW')
      WRITE(LOU,15)R1,R2,L,L1,T1
      WRITE(LOD,16)NRT,NZT
      WRITE(LOD,17)DELTAR,DELTAZ
      WRITE(LOD,17)R1,R2
      WRITE(LOD,17)L,L1
      WRITE(LOD,18)T1
   10 FORMAT('ACT ',5(F7.4,3X))
   15 FORMAT('NOR ',5(F7.4,3X))
   16 FORMAT(I8,1X,I8)
   17 FORMAT(F12.5,1X,F12.5)
   18 FORMAT(F12.5)
      WRITE(LOU,20)R3,FREQ,RHO1,U1,WUSRR
```

```
   20 FORMAT(' RBAR',F7.4,' FREQ= ',1PE13.6,' RHO=',0PF9.4,
     *' PERM=',F7.3,' WUSRR=',F9.4)
      SMAIR=0.0
      SMIMPR=0.0
      SMIMPI=0.0
      DO 25 NR=1,NRT
      DO 25 NZ=1,NZT
      SMZDFR(NR,NZ)=0.0
   25 SMZDFI(NR,NZ)=0.0
      B1=0.0
      B2=S2(1)
      DO 100 JKL=1,6
   30 RI9=SMAIR
      X=B1-0.5*S1(JKL)
C     DETERMINE NUMBER OF STEPS,ISTEPS,BASED ON STEP SIZE&INTERVAL
      ISTEPS=DNINT((B2-B1)/S1(JKL))
      DO 95 I=1,ISTEPS
      X=X+S1(JKL)
      CALL BESSEL(XJR2,X,R2)
      CALL BESSEL(XJR1,X,R1)
      XL=X*L
      IF(XL.GT.5.0E-3) GO TO 60
      A1=XL*XL*(0.5-XL/6.0)
      GO TO 80
   60 IF(XL.GT.75.0) GO TO 70
      A1=XL+DEXP(-XL)-1.0
      GO TO 80
   70 A1=XL-1.0
   80 CFAC=S1(JKL)*(XJR2-XJR1)*(XJR2-XJR1)
      SMAIR=SMAIR+CFAC*2.*A1
      IF(X*L1.GT.75.)GO TO 95
      XX=X*X
      X1=DSQRT(0.5*(XX+DSQRT(XX*XX+WUSRR*WUSRR)))/U1
      Y1=WUSRR/(2.*X1*U1*U1)
      A2=XL-A1
      A3=DEXP(-X*L1)
      APBR=(X+X1)*(X+X1)-Y1*Y1
      APBI=2.*Y1*(X+X1)
      AMBR=(X-X1)*(X-X1)-Y1*Y1
      AMBI=-2.*Y1*(X-X1)
      A2BR=0.0
      A2BI=-2.*X1*Y1
      ZNUR=A2BR
      ZNUI=A2BI
      DENR=APBR
      DENI=APBI
      DNCJ=DENR*DENR+DENI*DENI
C     SECTION THAT MULTIPLIES BY DEXP(ALPHA1*ZDEFECT)
      DO 91 NZ=1,NZT
      FZD=-(NZ-1)*T1/(NZT-1)
      ZDR=X1*U1*FZD
```

```
      IF(ZDR.LT.-60.0)GO TO 93
      ZDI=Y1*U1*FZD
      XPDR=DEXP(ZDR)
      CSDI=DCOS(ZDI)*XPDR
      SNDI=DSIN(ZDI)*XPDR
      XX1=X*X1+XX
      XY1=X*Y1
      X1X=X*X1-XX
      ZNDR=XX1*CSDI-XY1*SNDI
      ZNDI=XX1*SNDI+XY1*CSDI
C     SECTION THAT MULTIPLIES BY DEXP(ALPHA1*(2*TH+ZDEFECT))
      TZR=X1*U1*(2.*T1+FZD)
      IF(TZR.GT.60.)GO TO 87
      TZI=Y1*U1*(2.*T1+FZD)
      XPZR=DEXP(-TZR)
      CSZI=DCOS(TZI)*XPZR
      SNZI=DSIN(TZI)*XPZR
      ZNDR=XX1*CSDI-XY1*SNDI+X1X*CSZI+XY1*SNZI
      ZNDI=XX1*SNDI+XY1*CSDI+XY1*CSZI-X1X*SNZI
C     SECTION THAT MULTIPLIES BY DEXP(-ALPHA1*2*CLADTH)
      TR=2.*X1*U1*T1
      IF(TR.GT.60.)GO TO 87
      TI=2.*Y1*U1*T1
      XPTR=DEXP(-TR)
      CSTI=DCOS(TI)*XPTR
      SNTI=DSIN(TI)*XPTR
      DENR=APBR-AMBR*CSTI-AMBI*SNTI
      DENI=APBI-AMBI*CSTI+AMBR*SNTI
      ZNUR=A2BR-A2BR*CSTI-A2BI*SNTI
      ZNUI=A2BI-A2BI*CSTI+A2BR*SNTI
      DNCJ=DENR*DENR+DENI*DENI
   87 ZDRL=(ZNDR*DENR+ZNDI*DENI)/DNCJ
      ZDIM=(DENR*ZNDI-ZNDR*DENI)/DNCJ
      DFAC=A2*A3*S1(JKL)*(XJR2-XJR1)
C     LOOP OVER THE R VARIATION FOR THE DEFECT
      DO 90 NR=1,NRT
      IF(NZ.GT.1)GO TO 89
      RD=(FLOAT(NR)-.5)*DELTAR
      XRD=X*RD
      CALL BESEL1(XRD,RJ1)
      RJ(NR)=RJ1
   89 SMZDFR(NR,NZ)=SMZDFR(NR,NZ)+RJ(NR)*ZDRL*DFAC
   90 SMZDFI(NR,NZ)=SMZDFI(NR,NZ)+RJ(NR)*ZDIM*DFAC
   91 CONTINUE
   93 ZRL=(ZNUR*DENR+ZNUI*DENI)/DNCJ
      ZIM=(DENR*ZNUI-ZNUR*DENI)/DNCJ
      SMIMPR=SMIMPR+A2*A2*A3*A3*ZRL*CFAC
      SMIMPI=SMIMPI+A2*A2*A3*A3*ZIM*CFAC
   95 CONTINUE
      B1=B2
      B2=B2+S2(JKL)
```

```
      CHECK=(SMAIR-RI9)/SMAIR
      IF(ABS(CHECK).GT.ERR(JKL)) GO TO 30
  100 CONTINUE
  135 Q6=0.0254*4.0E-7*TRN*TRN*R3*PI*PI*SMAIR/(L*(R2-R1))**2
      ZNIM=(SMIMPR+SMAIR)/SMAIR
      ZNRL=-SMIMPI/SMAIR
      WRITE(LOU,140)ZNRL,ZNIM,Q6
  140 FORMAT(' NORM IMPD:RL',F10.6,' IM',F10.6,
     *' AIR IND',1PE13.6)
      DO 200 NR=1,NRT
      RD=(FLOAT(NR)-.5)*DELTAR
      DO 220 NZ=1,NZT
      AVZDFR=SMZDFR(NR,NZ)
      AVZDFI=SMZDFI(NR,NZ)
      DSFR=-1.5*WUSRR*(AVZDFR*AVZDFR-AVZDFI*AVZDFI)/(SMAIR*PI)
      DSFI=-1.5*WUSRR*2.0*AVZDFR*AVZDFI/(SMAIR*PI)
      DSFM=DSQRT(DSFR*DSFR+DSFI*DSFI)
      DSFP=DATAN2(DSFI,DSFR)
      WRITE(LOD,221)NR,NZ,DSFM,DSFP
  220 CONTINUE
  221 FORMAT(I5,2X,I5,2X,D11.5,2X,D11.5)
  200 CONTINUE
 1001 STOP 'JOB '
      END
```

PCDSFPLT generates a contour plot of magnitude of DSF

Program PCDSFPLT generates a contour plot of the magnitude of the defect sensitivity factor for a pancake coil using calculations performed and stored by program PCDSF. The cross section of the coil and the coil axis are also drawn on the plot with the same scale as the contours, to show their relative positions. Only a small change in the program is necessary to compute and plot contours of the phase of the defect sensitivity factor. There are two lines with the label 140, and one is commented out. With the first line in, the magnitude contours are plotted, and with the second line the phase contours are plotted. Both use the same data file from PCDSF.

Summary

1. Dimension arrays and declare variable types.
2. Initialize variables.
3. Open the file created by program PCDSF.
4. Read in the information about the coil and the plate from the file.
5. Calculate the position of the data points in the normalized coordinate system.
6. Set the label flags for the contours.
7. Read the data stored by program PCDSF into array DSFMA.
8. Specify the values of the magnitude of the defect sensitivity factor where the contours are to be drawn.
9. Call the PRINTMATIC contour initialization routines.
10. Draw the contours.
11. Draw the coil and the plate.

Variables

Starred variables must be set by the user.

CNM*	Array giving the values of the magnitude of the defect sensitivity factor at which contours are to be drawn.
DELTAR	The normalized distance in the radial direction between adjacent data points.
DELTAZ	The normalized distance in the axial direction between adjacent data points.
DSFMA	Array containing the values of the magnitude of the defect sensitivity factor which were read in from the data file.
DSFMMAX	The maximum value of the magnitude of the defect sensitivity factor.
DSFMMIN	The minimum value of the magnitude of the defect sensitivity factor.
L	The normalized length of the coil.
L1	The normalized lift-off of the coil.
LBM*	Array which tells the program which of the

	contours are to be labeled with their values. If all elements of LBM are zero, none of the contours will be labeled.
LOE*	The number of the I/O unit connected to the input data file.
NAME*	Character variable which contains the name of the file which this program uses for output.
NC*	The number of contours to be drawn. This number must be less than or equal to 10.
NRT	The total number of points in the radial direction at which calculations were performed.
NZT	The total number of points in the axial direction at which calculations were performed.
R1	The normalized inner coil radius.
R2	The normalized outer coil radius.
T1	The normalized thickness of the plate.
XX	Array describing the radial location of the data points in the normalized coordinate system.
YY	Array describing the axial location of the data points in the normalized coordinate system.

<u>Notes</u>

1. The coordinate system set up and used by this program has its origin at the intersection of the coil axis and the near side of the plate. One unit of distance in the coordinate system is equal to one mean radius of the driver coil.

2. The array DSFMA must be dimensioned to exactly NRT by NZT. Each time the value of NRT or NZT is changed in program PCDSF, the statement dimensioning the array in program PCDSFPLT must be changed also.

3. The statements in this program which seem to do nothing but write variables to the screen actually have a more important function. Due to a bug in either the PRINTMATIC routines or in RM/FORTRAN, the PRINTMATIC routine DLINE, which is supposed to draw a straight line, refuses to work. It was discovered by accident that putting a WRITE statement near the call to the routine corrects the problem.

4. Program PCDSFPLT does not actually send anything to the printer; it merely creates a file whose name is given by the program variable NAME. If the value of NAME is 'filename.ext', then to print the file created by program PCDSFPLT, enter

```
DPRINT filename.ext
```

DPRINT.EXE is a program supplied by PRINTMATIC. For this particular program the variable NAME is set to PCDSF.FIL, so to make a plot one would type:

```
DPRINT PCDSF.FIL
```

Sample output

An example of the contour plot of the magnitude of the defect sensitivity factor is shown in Fig. 4.

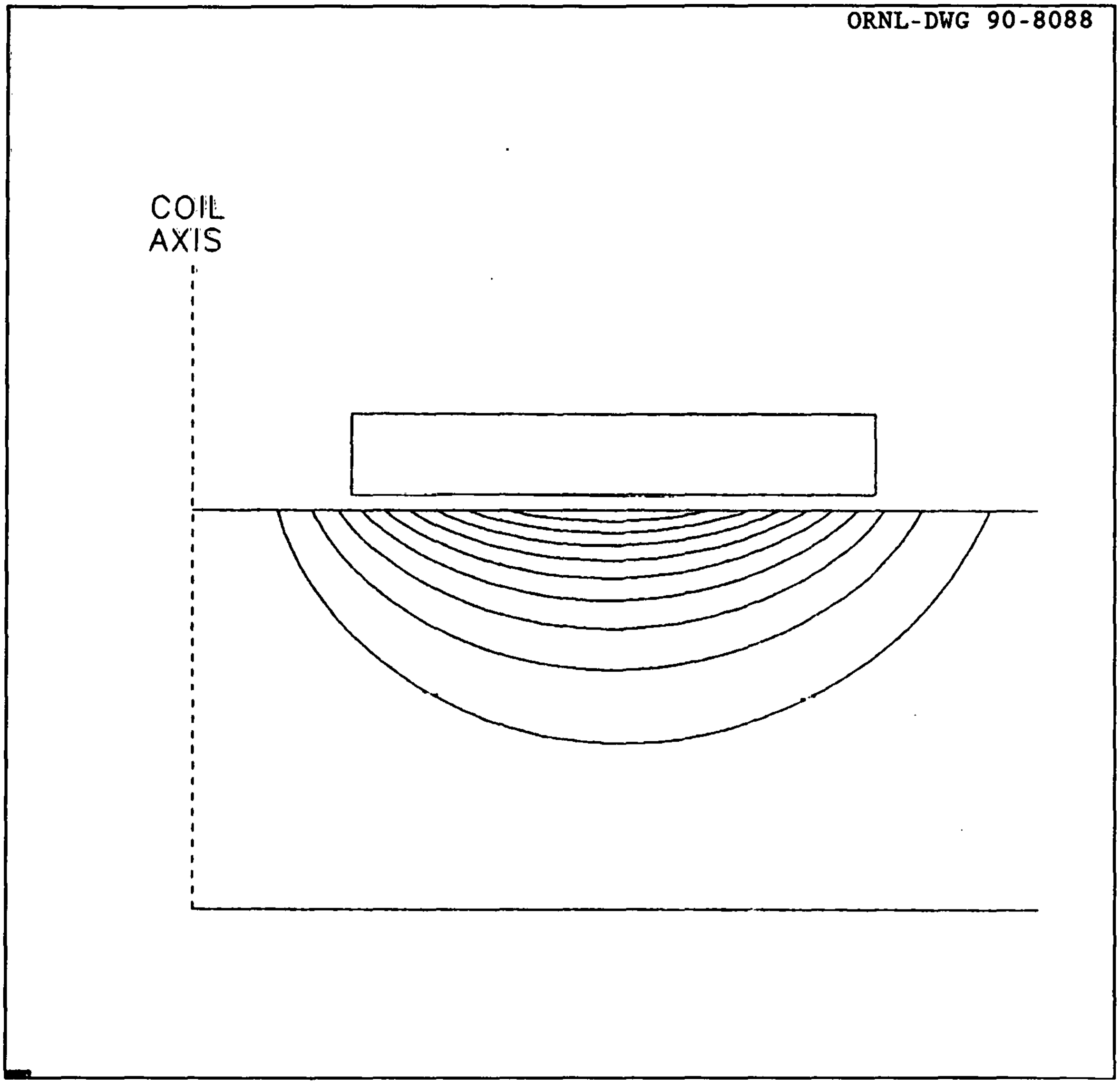

Fig. 4. Plot of the magnitude of the defect sensitivity factor.

Listing

```
      PROGRAM PCDSFPLT
C     VERSION November 16, 1988
C     Program to generate a contour plot of the magnitude of the
C     defect sensitivity factor of a pancake coil.
      CHARACTER*80 NAME
      IMPLICIT REAL*4 (A-H,O-Z)
      REAL*4 DSFMA(40,40)
      REAL*4 XX(40),YY(40)
      REAL*4 CNM(10),CNP(10)
      REAL*4 L,L1
      INTEGER*2 LBM(10)
      INTEGER*2 I1,J1,I2,J2
      DATA XSCALE/1.0/,NC/9/
      DATA IDEF/2/,LOE/40/
C   Open the file created by program PCDSF.
      OPEN(LOE,FILE='PCDSF.DAT',STATUS='OLD')
C   Read in the coil information.
      READ(LOE,*)NRT,NZT
      READ(LOE,*)DELTAR,DELTAZ
      READ(LOE,*)R1,R2
      READ(LOE,*)L,L1
      READ(LOE,*)T1
C   Calculate the position of the data points in the
C    normalized coordinate system.
      DO 110 I=0,(NRT-1)
      XX(I+1)=REAL(I)*DELTAR
  110 CONTINUE
      DO 120 I=0,(NZT-1)
      YY(I+1)=-((NZT-I-1)*DELTAZ)
  120 CONTINUE
C   Set the label flags for the contours.
      DO 130 I=1,10
      LBM(I)=0
  130 CONTINUE
C   Read in the data stored by program PCDSF.
  140 READ(LOE,*,END=150)NR,NZ,DSFM
c 140 READ(LOE,*,END=150)NR,NZ,DUM,DSFM
      NZ=NZT-NZ+1
      IF(DSFM.GT.DSFMMAX)DSFMMAX=DSFM
      IF(DSFMMIN.EQ.0.)DSFMMIN=DSFM
      IF(DSFM.LT.DSFMMIN)DSFMMIN=DSFM
      DSFMA(NR,NZ)=DSFM
      GO TO 140
C   Specify the values at which the contours are to be drawn.
  150 VARMAG=DSFMMAX-DSFMMIN
      CNTDIF=VARMAG/(NC+1)
      DO 160 I=1,NC
      CNM(I)=DSFMMAX-I*CNTDIF
  160 CONTINUE
```

```
C    Call the necessary initialization routines.
       NAME='PCDSF.FIL'
       CALL DINIT(NAME)
       CALL DPLOT(1.,1.,6.,6.,-0.1,1.9,-1.,1.,0.,0.)
       CALL DCTRDEF(1,1,1,1,1)
C    Draw the contours.
       CALL DCNTOUR(XSCALE,XX,YY,DSFMA,CNM,LBM,NRT,NZT,NC,IDEF)
C    Draw the plate.
       write(0,*)j2
       X1=0.
       Y1=0.
       CALL DRTOI(X1,Y1,I1,J1)
       X2=2.
       Y2=-T1
       CALL DRTOI(X2,Y2,I2,J2)
       write(0,*)j2
       CALL DLINE(I1,J1,I2,J1)
       write(0,*)j2
       CALL DLINE(I1,J2,I2,J2)
       write(0,*)j2
C    Draw the coil.
       X1=R1
       Y1=L1
       CALL DRTOI(X1,Y1,I1,J1)
       X2=R2
       Y2=L+L1
       CALL DRTOI(X2,Y2,I2,J2)
       write(0,*)j2
       CALL DLINE(I1,J1,I2,J1)
       CALL DLINE(I1,J2,I2,J2)
       CALL DLINE(I1,J1,I1,J2)
       WRITE(0,*)I2,J1,I2,J2
       CALL DLINE(I2,J1,I2,J2)
       WRITE(0,*)I2,J1,I2,J2
C    Draw the coil axis.
       WRITE(0,*)I1,J1
       X1=0.
       Y1=-T1
       CALL DRTOI(X1,Y1,I1,J1)
       X2=0.
       Y2=0.6
       CALL DRTOI(X2,Y2,I2,J2)
       WRITE(0,*)I1,J1,I2,J2
       CALL DDASH(I1,J1,I2,J2,1,10,10)
       CALL DRTOI(-0.1,0.7,I1,J1)
       CALL DFONT(4,'COIL',I1,J1,1)
       CALL DRTOI(-0.1,0.62,I1,J1)
       CALL DFONT(4,'AXIS',I1,J1,1)
```

```
CALL DFINIS
write(0,*)j2
stop
END
```

PCFIX converts raw data to normalized impedance change

Program PCFIX converts raw pancake coil experimental data into the normalized impedance change in the pancake coil due to a defect. The raw data are read directly from voltmeters as the coil is moved over the surface of a plate by a program such as MIZSCN2. The voltmeters are connected to the vertical and horizontal channels of the Zetec MIZ-17. The MIZ-17 makes relative readings of the *x* and *y* components of the coil impedance. By comparing the measured change for a known amount of lift-off to the calculated change, both the phase (rotation) and gain of the signal are corrected. The values of the voltage readings with and without the lift-off must be measured and typed into the program.

Summary

1. Dimension arrays and declare variable types.
2. Initialize variables.
3. Calculate the constant by which the raw readings must be multiplied to convert them to the values of the normalized impedance change in the coil.
4. Read in a data point.
5. Subtract the reading taken on a part of a plate with no defects from the raw reading.
6. Multiply the reading by the constant obtained in 3.
7. Loop to 4 until done.
8. Average the readings taken on opposite sides of the defect.
9. Write out the results.

Variables

CLOM*	The calculated value of the change in the magnitude of the normalized impedance due to the amount of additional lift-off we are using.
CLOP*	The calculated value of the change in the phase of the normalized impedance due to the amount of additional lift-off we are using.
LOD*	The number of the I/O unit connected to the file which is used for output.
LOE*	The number of the I/O unit connected to the file containing the raw data.
LOI	The change in the imaginary part of the raw reading due to the additional lift-off.
LOR	The change in the real part of the raw reading due to the additional lift-off.
RLOI*	The imaginary part of the raw reading made with additional lift-off.
RLOR*	The real part of the raw reading made with additional lift-off.
SF	The ratio of the magnitude of the calculated change in the normalized impedance to the change

	in the raw reading due to additional lift-off.
PD	The difference between the phase of the calculated change in the normalized impedance and the phase of the change in the raw reading due to additional lift-off.
ZEROI*	The imaginary part of the raw reading taken on a part of the plate with no defects.
ZEROR*	The real part of the raw reading taken on a part of the plate with no defects.
ZLOI*	The imaginary part of the raw reading made on the same part of the plate where the readings for RLOR and RLOI were taken, but ZLOI is the reading without additional lift-off.
ZLOR*	The real part of the raw reading made on the same part of the plate where the readings for RLOR and RLOI were taken, but ZLOR is the reading without additional lift-off.
ZMAG	The magnitude of the change in the normalized impedance of the coil due to the defect.
ZPHA	The phase of the change in the normalized impedance of the coil due to the defect.

<u>Notes</u>

1. The raw readings taken directly from the voltmeters by a program such as MIZSCN2 differ from the normalized impedance change in the coil by both an additive and a multiplicative constant. To determine the additive constant, we need only take readings on a part of the plate with no defects. This reading is subtracted from the raw readings. To determine the multiplicative constant, we find both the calculated and the experimental changes in readings due to a certain amount of additional lift-off. All subsequent raw readings are then multiplied by the ratio of the calculated value to the measured value. This directly normalizes the readings, and includes any amplification factors in the instrument.

2. This program averages the readings which precede the defect with those which follow the defect. The defect signal is supposed to be symmetric about the center of the defect. Any asymmetry should be due to random noise or to changes in the plate which we are not interested in. Averaging the signal will reduce the effect that such changes have on our results.

Listing

```
      PROGRAM PCFIX
      VERSION November 21, 1988
C     PROGRAM TO CONVERT RAW EXPERIMENTAL DATA TAKEN BY
C     PANCAKE COILS INTO THE NORMALIZED IMPEDANCE CHANGE IN
C     THE PANCAKE COIL DUE TO A DEFECT.
      DIMENSION VRA(250),VIA(250),ZMAG(250),ZPHA(250)
      REAL OFFSET
      REAL LOM,LOP,LOR,LOI
      DATA LOD/39/,LOE/38/
      DATA PI/3.141592653/
      DATA FREQ/500/
      DATA ZEROR/-7.6/,ZEROI/-2.2/
      DATA RLOR/-9.5/,RLOI/-0.5/,ZLOR/-7.77/,ZLOI/-2.14/
      DATA CLOM/0.020030/,CLOP/2.44/
      OPEN(LOE,FILE='AARAWN2.DAT',STATUS='OLD')
      OPEN(LOD,FILE='AAEXPN2.DAT',STATUS='NEW')
      LOR=RLOR-ZLOR
      LOI=RLOI-ZLOI
      I=1
   11 READ(LOE,*,END=30)X,VRA(I),VIA(I)
C      WRITE(0,*)VRA(I),VIA(I)
      I=I+1
      GOTO 11
   30 I=I-1
      IMAX=I
      J=1
      LOM=SQRT(LOR*LOR+LOI*LOI)
      LOP=ATAN2(LOI,LOR)
      SF=CLOM/LOM
      PD=CLOP-LOP
   35 IF(I.LT.J) GO TO 200
      VR=0.5*(VRA(I)+VRA(J))
      VI=0.5*(VIA(I)+VIA(J))
      VR=VR-ZEROR
      VI=VI-ZEROI
      ZMAG(I)=SQRT(VR*VR+VI*VI)*SF
      ZPHA(I)=(ATAN2(VI,VR)+PD)*180./PI
   75 FORMAT(E11.4,1X,F9.3)
      I=I-1
      J=J+1
      GO TO 35
  200 I=I+1
      DO 300 J=I,IMAX
      WRITE(LOD,75)ZMAG(J),ZPHA(J)
  300 CONTINUE
      STOP
      END
```

PCAVZSCN calculates defect impedance change, average over depth

Program PCAVZSCN calculates the change in the impedance of a pancake coil due to the presence of a defect in a conducting plate. It does the calculations for a number of different radial distances between the coil axis and the center of the defect, and it has the ability to divide the defect into smaller parts centered on the axis of the defect and to perform the calculations for each part separately, a method of treating the defect that usually gives more accurate results. The assumption is made that the defect has a constant cross section as the depth is varied, such as in a flat bottomed hole. If this is not the case, a weighted average with depth should be used. The output from PCAVZSCN is stored in the file PCAVZSCN.DAT.

<u>Summary</u>

1. Dimension arrays and declare variable types.
2. Initialize variables.
3. Begin the integration loops, calculating the expressions that are independent of the position of the defect.
4. Assign a value to FZD, the center of the part of the defect we wish to work with.
5. Do the calculations that depend only on the axial position of the defect.
6. Select a value for RD, the radial distance between the coil and the part of the defect we are working with.
7. Complete the integration.
8. Loop to 6 until done.
9. Loop to 4 until done.
10. Write the results of the calculations to a file.

<u>Variables</u>

DELTAR*	The normalized radial distance between adjacent data points.
DFDEP*	The distance from the side of the plate where the defect is located to the bottom of the defect in inches.
DFDIAM*	The diameter of the defect in inches.
DFM	The magnitude of the change in the normalized impedance of the coil due to the defect.
DFP	The phase of the change in the normalized impedance of the coil due to the defect.
DSFI	The imaginary part of the defect sensitivity factor of the coil.
DSFR	The real part of the defect sensitivity factor of the coil.
FREQ*	The operating frequency in hertz.
L*	The length of the coil. The value is input in inches and normalized by the program.
L1*	The lift-off of the coil. The value is input in

	inches and normalized by the program.
LOD*	The number of the I/O unit connected to the output data file.
LOU	The number of the I/O unit connected to the printer.
NRT*	The number of different radial distances between the coil axis and the center of the defect at which the calculations are performed.
NS*	The side of the plate where the defect is located. If NS=1, the defect is on the near side; if NS=2, the defect is on the far side.
NZT*	The total number of parts into which the defect is divided along its axis when the calculations are performed.
Q6	The inductance of the coil in air.
R1*	The inner radius of the coil. The value is input in inches and normalized by the program.
R2*	The outer radius of the coil. The value is input in inches and normalized by the program.
R3	The mean radius of the coil in inches.
RD	The normalized radial distance from the axis of the coil to the center of the defect.
RHO1*	The electrical resistivity of the plate in $\mu\Omega$-cm.
T1*	The thickness of the plate. The value is input in inches and normalized by the program.
TRN*	The number of turns in the coil.
U1*	The relative magnetic permeability of the plate.
VOLN	The normalized volume of the defect.
WUSRR	The product of the angular frequency, the relative magnetic permeability, the electrical conductivity, and the square of the mean coil radius.
ZD	The normalized distance from the side of the plate where the defect is located to the bottom of the defect. A negative number.
ZNDFI	The imaginary part of the change in the normalized impedance of the coil due to the defect.
ZNDFR	The real part of the change in the normalized impedance of the coil due to the defect.
ZNIM	The imaginary part of the normalized coil impedance with no defects present.
ZNRL	The real part of the normalized coil impedance with no defects present.

Notes

1. Variable NZT controls the number of parts into which the defect is divided to perform the calculations. Since the theory upon which this program is based is more accurate for small defects, it is desirable to work with the defect in parts rather than as a whole.

Integration Section of Program PCAVZSCN

Symbol definitions

The following are definitions of the symbols used to describe the program variables appearing in the integration section of this program. All lengths are normalized unless otherwise noted.

α	Integration variable
α_1	$(\alpha^2+j\omega\mu\sigma_1\bar{r}^2)^{1/2}$
α_{22}	Defect shape and orientation factor
β_1	$(\alpha^2+j\omega\mu\sigma_1\bar{r}^2)^{1/2}/\mu$
c	Plate thickness
$J(x_2,x_1)$	Integral of $xJ_1(x)$ with respect to x from αx_1 to αx_2
$J_1(x)$	Bessel function of the first kind of order 1
ℓ	Length of coil
ℓ_1	Lift-off of coil
μ	Relative magnetic permeability of plate
$\underline{r}$	Coil-to-defect radial distance
$\bar{r}$	Mean radius of coil in inches
r_1	Inner radius of coil
r_2	Outer radius of coil
σ_1	Conductivity of plate
Vol_n	Normalized volume of defect
ω	Angular frequency at which circuit is driven
z	Depth to center of defect

Variables appearing in the integration section

Program variable	Symbolic equivalent
A1	$\alpha\ell+\exp(-\alpha\ell)-1$
A2	$1-\exp(-\alpha\ell)$
A2BI	$\mathrm{Im}[\alpha^2-\beta_1^2]$
A2BR	$\mathrm{Re}[\alpha^2-\beta_1^2]$
A3	$\exp(-\alpha\ell_1)$
AMBI	$\mathrm{Im}[(\alpha-\beta_1)^2]$
AMBR	$\mathrm{Re}[(\alpha-\beta_1)^2]$
APBI	$\mathrm{Im}[(\alpha+\beta_1)^2]$
APBR	$\mathrm{Re}[(\alpha+\beta_1)^2]$
AVZDFI	(See note I3.)
AVZDFR	(See note I3.)
B1	(See note I2.)
B2	(See note I2.)

CFAC	$[J(r_2, r_1)]^2$
CHECK	{See note I2.}
CSDI	$\mathrm{Re}[\exp(\alpha_1 z)]$
CSTI	$\mathrm{Re}[\exp(-2\alpha_1 c)]$
CSZI	$\mathrm{Re}[\exp(-\alpha_1(2c+z))]$
DENI	$\mathrm{Im}[(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)]$
DENR	$\mathrm{Re}[(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)]$
DFAC	$\frac{1}{\alpha^3}(1-\exp(-\alpha\ell))\exp(-\alpha\ell_1)J(r_2, r_1)\ d\alpha$
ERR	{See note I2.}
FZD	z {See note I4.}
ISTEPS	{See note I2.}
RI9	{See note I2.}
RJ1	$J_1(\alpha r)$
S1	$d\alpha$
S2	{See note I2.}
SMAIR	$\int_0^\infty \frac{2(\alpha\ell+\exp(-\alpha\ell)-1)[J(r_2, r_1)]^2}{\alpha^6}\ d\alpha$
SMIMPI	$\mathrm{Im}\left[\int_0^\infty \frac{(1-\exp(-\alpha\ell))^2\exp(-2\alpha\ell_1)[J(r_2, r_1)]^2}{\alpha^6}\left[\frac{(\alpha^2-\beta_1^2)-(\alpha^2-\beta_1^2)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]d\alpha\right]$
SMIMPR	$\mathrm{Re}\left[\int_0^\infty \frac{(1-\exp(-\alpha\ell))^2\exp(-2\alpha\ell_1)[J(r_2, r_1)]^2}{\alpha^6}\left[\frac{(\alpha^2-\beta_1^2)-(\alpha^2-\beta_1^2)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]d\alpha\right]$

SMZDFI $\text{Im}\left[\int_0^\infty \frac{1}{\alpha^3}(1-\exp(-\alpha\ell))\exp(-\alpha\ell_1)J(r_2,r_1)J_1(\alpha r)\right.$

$\left.\alpha\left[\frac{(\alpha+\beta_1)\exp(\alpha_1 z)-(\alpha-\beta_1)\exp(-\alpha_1(2c+z))}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]d\alpha\right]$

SMZDFR $\text{Re}\left[\int_0^\infty \frac{1}{\alpha^3}(1-\exp(-\alpha\ell))\exp(-\alpha\ell_1)J(r_2,r_1)J_1(\alpha r)\right.$

$\left.\alpha\left[\frac{(\alpha+\beta_1)\exp(\alpha_1 z)-(\alpha-\beta_1)\exp(-\alpha_1(2c+z))}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]d\alpha\right]$

SNDI $\text{Im}[\exp(\alpha_1 z)]$

SNTI $-\text{Im}[\exp(-2\alpha_1 c)]$

SNZI $-\text{Im}[\exp(-\alpha_1(2c+z))]$

TI $\text{Im}[2\alpha_1 c]$

TR $\text{Re}[2\alpha_1 c]$

TZI $\text{Im}[\alpha_1(2c+z)]$

TZR $\text{Re}[\alpha_1(2c+z)]$

X α

X1 $\text{Re}[\beta_1]$

X1X $\text{Re}[\alpha(\beta_1-\alpha)]$

XJR1 $J(r_1,0)$

XJR2 $J(r_2,0)$

XL $\alpha\ell$

XPDR $\exp[\text{Re}(\alpha_1 z)]$

XPTR $\exp[\text{Re}(-2\alpha_1 c)]$

XPZR $\exp[\text{Re}(-\alpha_1(2c+z))]$

XRD αr

XX α^2

XX1 $\text{Re}[\alpha(\alpha+\beta_1)]$

XY1 $\text{Im}[\alpha(\alpha+\beta_1)]=\text{Im}[\alpha(\beta_1-\alpha)]$

Y1 $\text{Im}[\beta_1]$

ZDI $\text{Im}[\alpha_1 z]$

ZDIM $\mathrm{Im}\left[\dfrac{(\alpha+\beta_1)\exp(\alpha_1 z)-(\alpha-\beta_1)\exp(-\alpha_1(2c+z))}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]$

ZDR $\mathrm{Re}[\alpha_1 z]$

ZDRL $\mathrm{Re}\left[\alpha\dfrac{(\alpha+\beta_1)\exp(\alpha_1 z)-(\alpha-\beta_1)\exp(-\alpha_1(2c+z))}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]$

ZIM $\mathrm{Im}\left[\alpha\dfrac{(\alpha^2-\beta_1^2)-(\alpha^2-\beta_1^2)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]$

ZNDI $\mathrm{Im}[\alpha(\alpha+\beta_1)\exp(\alpha_1 z)]$

ZNDR $\mathrm{Re}[\alpha(\alpha+\beta_1)\exp(\alpha_1 z)]$

ZNUI $\mathrm{Im}[(\alpha^2-\beta_1^2)-(\alpha^2-\beta_1^2)\exp(-2\alpha_1 c)]$

ZNUR $\mathrm{Re}[(\alpha^2-\beta_1^2)-(\alpha^2-\beta_1^2)\exp(-2\alpha_1 c)]$

ZRL $\mathrm{Re}\left[\dfrac{(\alpha^2-\beta_1^2)-(\alpha^2-\beta_1^2)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]$

Notes for the integration section

I1. A number of the variables in the integration section are not always assigned their exact values but are approximated in certain cases to save time. For example, rather than calculate the exponential of a very small number that has its argument subtracted from it, the Maclaurin series expansion is sometimes used. Also, the exponential of a very large negative number is usually treated as zero.

I2. Several variables appear in the integration section of the program which play no part in the calculations being done. They are merely there to do such things as to determine the maximum step size which can be used while still accurately calculating the integrals.

I3. Variables AVZDFR and AVZDFI are the averages of the elements in arrays SMZDFR and SMZDFI, respectively. For improved accuracy, the defect is divided into NZT parts to perform the calculations. The NZT elements of each array contain the calculations for these NZT parts. These elements are averaged to give the total effect of all of these parts on the impedances of the coils.

I4. The variable FZD is the normalized depth from the near side of the plate to the center of the part of the defect with which the program is working. It is not the depth of the center of the actual defect.

Sample output

Printer output of program PCAVZSCN:

```
PCAVZSCN    20 POINTS USED   FAR SIDE DEFECT  TIME  8: 9:55  DATE  8/ 9/89
     IN RAD      OT RAD      LENGTH      LIFTOFF     CLADTH      DF DEP
ACT  0.1000      0.4100      0.1000      0.0100      0.2500     -0.0780
NOR  0.3922      1.6078      0.3922      0.0392      0.9804     -0.3059
RBAR 0.2550 FREQ=5.000000E+02 RHO=4.0540 PERM=1.000 WUSRR=4.0853
 NORM IMPD:RL  0.155825 IM  0.832098 AIR IND 6.252919E-03 VOLN 2.1905E-02
```

Partial listing of file PCAVZSCN.DAT:

```
 0.1550D-06    54.632
 0.1390D-05    54.602
 0.3832D-05    54.540
 0.7428D-05    54.448
 0.1210D-04    54.322
 0.1774D-04    54.162
 0.2423D-04    53.966
 0.3143D-04    53.733
 0.3919D-04    53.459
 0.4733D-04    53.144
```

Listing

```
      PROGRAM PCAVZSCN
C     VERSION August 9, 1989
C     PROGRAM TO CALCULATE THE NORMALIZED IMPEDANCE CHANGE
C     FOR A DEFECT IN A SINGLE PLANAR CONDUCTOR.  THE DEFECT IS
C     CALCULATED AT AN ARRAY OF POINTS RATHER THAN A SINGLE POINT
C     AND AVERAGED OVER THE DEPTH.  THE CALCULATIONS ARE STEPPED IN THE
C     R DIRECTION, AND THE MAGNITUDE AND PHASE IS STORED IN A DATA FILE.
      CHARACTER SIDE(2)*4
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 L,L1
      CHARACTER*1 FF
      DIMENSION S1(6),S2(6),ERR(6),SMZDFR(200,40),SMZDFI(200,40),RJ(200)
      DATA LOU/8/,PI/3.141592653/,NZT/20/,NRT/40/
      DATA S1/.005,.02,.05,.1,.5,2./
      DATA S2/1.0,2.0,5.0,10.0,50.0,200.0/
      DATA ERR/.1,.01,.001,1.E-4,1.E-5,1.E-6/
      DATA FREQ/500./,RHO1/4.054/,U1/1.0/,T1/0.250/
      DATA TRN/800./,SIDE/'NEAR',' FAR'/,DELTAR/0.05/
      DATA LOD/39/,DFDIAM/0.0770/,DFDEP/0.0780/,NS/2/
      FF=CHAR(12)
      OPEN(LOD,FILE='PCAVZSCN.DAT',STATUS='NEW')
C     TIME AND DATE ARE PRINTED
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,2)NZT,SIDE(NS),IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT('PCAVZSCN  ',I3,' POINTS USED  ',A4,' SIDE DEFECT',
     A'  TIME ',I2,':',I2,':',I2,'  DATE ',I2,'/',I2,'/',I2)
      WRITE(LOU,5)
    5 FORMAT(5X,'IN RAD',4X,'OT RAD',4X,'LENGTH',4X,'LIFTOFF'
     *,3X,'CLADTH',4X,'DF DEP')
C     COIL P250A
      R1=0.1000
      R2=0.4100
      L=0.1000
      L1=0.01
      R3=.5*(R1+R2)
      ZD=-DFDEP
      WRITE(LOU,10)R1,R2,L,L1,T1,ZD
      R1=R1/R3
      R2=2.0-R1
      L=L/R3
      L1=L1/R3
      ZD=ZD/R3
      T1=T1/R3
C     VOLN=0.1666667*PI*(DFDIAM/R3)*(DFDIAM/R3)*(DFDIAM/R3)
      VOLN=PI*DFDIAM*DFDIAM*DFDEP/(4.*R3*R3*R3)
      WUSRR=0.5093979*U1*R3*R3*FREQ/RHO1
      WRITE(LOU,15)R1,R2,L,L1,T1,ZD
```

```
   10 FORMAT('ACT ',7(F7.4,3X))
   15 FORMAT('NOR ',7(F7.4,3X))
  18  FORMAT(' CYL FLAW: DIAM=',F7.4,', DEPTH=',F7.4)
      WRITE(LOU,20)R3,FREQ,RHO1,U1,WUSRR
   20 FORMAT(' RBAR',F7.4,' FREQ= ',1PE13.6,' RHO=',0PF9.4,
     *' PERM=',F7.3,' WUSRR=',F9.4)
      SMAIR=0.0
      SMIMPR=0.0
      SMIMPI=0.0
      DO 25 NR=1,NRT
      DO 25 NZ=1,NZT
      SMZDFR(NR,NZ)=0.0
   25 SMZDFI(NR,NZ)=0.0
      B1=0.0
      B2=S2(1)
      DO 100 JKL=1,6
   30 RI9=SMAIR
      X=B1-0.5*S1(JKL)
C     DETERMINE NUMBER OF STEPS,ISTEPS,BASED ON STEP SIZE&INTERVAL
      ISTEPS=DNINT((B2-B1)/S1(JKL))
      DO 95 I=1,ISTEPS
      X=X+S1(JKL)
      CALL BESSEL(XJR2,X,R2)
      CALL BESSEL(XJR1,X,R1)
      XL=X*L
      IF(XL.GT.5.0E-3) GO TO 60
      A1=XL*XL*(0.5-XL/6.0)
      GO TO 80
   60 IF(XL.GT.75.0) GO TO 70
      A1=XL+DEXP(-XL)-1.0
      GO TO 80
   70 A1=XL-1.0
   80 CFAC=S1(JKL)*(XJR2-XJR1)*(XJR2-XJR1)
      SMAIR=SMAIR+CFAC*2.*A1
      IF(X*L1.GT.75.)GO TO 95
      XX=X*X
      X1=DSQRT(0.5*(XX+DSQRT(XX*XX+WUSRR*WUSRR)))/U1
      Y1=WUSRR/(2.*X1*U1*U1)
      A2=XL-A1
      A3=DEXP(-X*L1)
      APBR=(X+X1)*(X+X1)-Y1*Y1
      APBI=2.*Y1*(X+X1)
      AMBR=(X-X1)*(X-X1)-Y1*Y1
      AMBI=-2.*Y1*(X-X1)
      A2BR=0.0
      A2BI=-2.*X1*Y1
      ZNUR=A2BR
      ZNUI=A2BI
      DENR=APBR
      DENI=APBI
      DNCJ=DENR*DENR+DENI*DENI
```

```
C     SECTION THAT MULTIPLIES BY DEXP(ALPHA1*ZDEFECT)
      DO 91 NZ=1,NZT
C     NEAR SIDE DEFECT CALCULATION
      FZD=(FLOAT(NZ)-.5)*ZD/FLOAT(NZT)
C     FAR SIDE DEFECT CALCULATION
      IF(NS.EQ.2)FZD=-T1-FZD
      ZDR=X1*U1*FZD
      IF(ZDR.LT.-60.0)GO TO 93
      ZDI=Y1*U1*FZD
      XPDR=DEXP(ZDR)
      CSDI=DCOS(ZDI)*XPDR
      SNDI=DSIN(ZDI)*XPDR
      XX1=X*X1+XX
      XY1=X*Y1
      X1X=X*X1-XX
      ZNDR=XX1*CSDI-XY1*SNDI
      ZNDI=XX1*SNDI+XY1*CSDI
C     SECTION THAT MULTIPLIES BY DEXP(ALPHA1*(2*TH+ZDEFECT))
      TZR=X1*U1*(2.*T1+FZD)
      IF(TZR.GT.60.)GO TO 87
      TZI=Y1*U1*(2.*T1+FZD)
      XPZR=DEXP(-TZR)
      CSZI=DCOS(TZI)*XPZR
      SNZI=DSIN(TZI)*XPZR
      ZNDR=XX1*CSDI-XY1*SNDI+X1X*CSZI+XY1*SNZI
      ZNDI=XX1*SNDI+XY1*CSDI+XY1*CSZI-X1X*SNZI
C     SECTION THAT MULTIPLIES BY DEXP(-ALPHA1*2*CLADTH)
      TR=2.*X1*U1*T1
      IF(TR.GT.60.)GO TO 87
      TI=2.*Y1*U1*T1
      XPTR=DEXP(-TR)
      CSTI=DCOS(TI)*XPTR
      SNTI=DSIN(TI)*XPTR
      DENR=APBR-AMBR*CSTI-AMBI*SNTI
      DENI=APBI-AMBI*CSTI+AMBR*SNTI
      ZNUR=A2BR-A2BR*CSTI-A2BI*SNTI
      ZNUI=A2BI-A2BI*CSTI+A2BR*SNTI
      DNCJ=DENR*DENR+DENI*DENI
   87 ZDRL=(ZNDR*DENR+ZNDI*DENI)/DNCJ
      ZDIM=(DENR*ZNDI-ZNDR*DENI)/DNCJ
      DFAC=A2*A3*S1(JKL)*(XJR2-XJR1)
C     LOOP OVER THE R VARIATION FOR THE DEFECT
      DO 90 NR=1,NRT
      IF(NZ.GT.1)GO TO 89
      RD=(FLOAT(NR)-.5)*DELTAR
      XRD=X*RD
      CALL BESEL1(XRD,RJ1)
      RJ(NR)=RJ1
   89 SMZDFR(NR,NZ)=SMZDFR(NR,NZ)+RJ(NR)*ZDRL*DFAC
   90 SMZDFI(NR,NZ)=SMZDFI(NR,NZ)+RJ(NR)*ZDIM*DFAC
   91 CONTINUE
```

```
   93 ZRL=(ZNUR*DENR+ZNUI*DENI)/DNCJ
      ZIM=(DENR*ZNUI-ZNUR*DENI)/DNCJ
      SMIMPR=SMIMPR+A2*A2*A3*A3*ZRL*CFAC
      SMIMPI=SMIMPI+A2*A2*A3*A3*ZIM*CFAC
   95 CONTINUE
      B1=B2
      B2=B2+S2(JKL)
      CHECK=(SMAIR-RI9)/SMAIR
      IF(ABS(CHECK).GT.ERR(JKL)) GO TO 30
  100 CONTINUE
  135 Q6=0.0254*4.0E-7*TRN*TRN*R3*PI*PI*SMAIR/(L*(R2-R1))**2
      ZNIM=(SMIMPR+SMAIR)/SMAIR
      ZNRL=-SMIMPI/SMAIR
      WRITE(LOU,140)ZNRL,ZNIM,Q6,VOLN
  140 FORMAT(' NORM IMPD:RL',F10.6,' IM',F10.6,
     *' AIR IND',1PE13.6,' VOLN',1PE11.4)
  150 FORMAT(' NORM CHG:RL ',0PF10.6,' IM',0PF10.6,
     *' MAG ',0PF10.6,' PHA ',0PF7.2)
  160 FORMAT(' NORM DSF:RL',1PE11.4,' IM',1PE11.4,' VOLN',1PE11.4)
      DO 200 NR=1,NRT
      RD=(FLOAT(NR)-.5)*DELTAR
C     AVERAGE OVER DEFECT POINTS AT DIFFERENT DEPTHS
      AVZDFR=0.0
      AVZDFI=0.0
      DO 220 NZ=1,NZT
      AVZDFR=AVZDFR+SMZDFR(NR,NZ)
  220 AVZDFI=AVZDFI+SMZDFI(NR,NZ)
      AVZDFR=AVZDFR/FLOAT(NZT)
      AVZDFI=AVZDFI/FLOAT(NZT)
      DSFR=-1.5*WUSRR*(AVZDFR*AVZDFR-AVZDFI*AVZDFI)/(SMAIR*PI)
      DSFI=-1.5*WUSRR*2.0*AVZDFR*AVZDFI/(SMAIR*PI)
      ZNDFR=VOLN*DSFR
      ZNDFI=VOLN*DSFI
      DFM=VOLN*SQRT(DSFR*DSFR+DSFI*DSFI)
      DFP=ATAN2(DSFI,DSFR)*180./PI
  200 WRITE(LOD,178)DFM,DFP
  178 FORMAT(D11.4,1X,F8.3)
 1001 STOP
      END
```

PCAVVSCN calculates defect impedance change, average over volume

Program PCAVVSCN calculates the change in the impedance of a pancake coil due to the presence of a flat-bottomed hole in a conducting plate. It first calculates the defect sensitivity factor for a lattice of points that extends from the coil axis to the outer edge of the coil field, as shown in Fig. 3. Although this method of averaging over the dimensions of the defect is not mathematically correct, it usually gives more accurate results than assuming that the defect is a point or only averaging over the defect depth. Once the defect sensitivity factor has been computed, the defect can be considered to be at any radial location with respect to the coil and its average effect calculated by summing over the defect sensitivity factor values at the different radial positions multiplied by the volume of that particular element. In Fig. 5 we show the relationship of a flat-bottomed hole to the coil center, looking down on the plate.

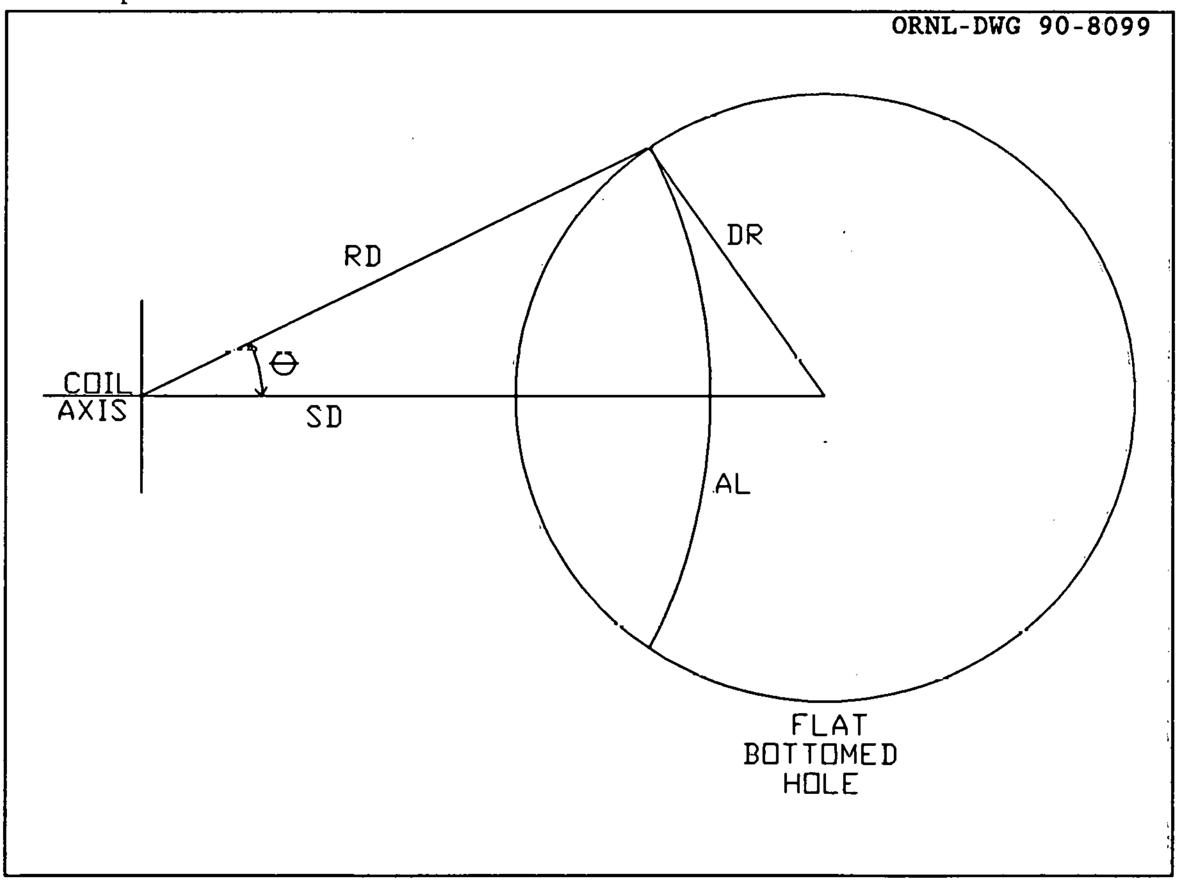

Fig. 5. Top view of flat-bottomed hole referenced to the coil axis.

The defect signal is first averaged over the defect depth in a manner similar to the PCAVZSCN program. Then the arc length AL is multiplied by the real and imaginary value of the average defect signal and this is summed as the distance from the coil axis RD is stepped from one side of the defect to the other. This is repeated for different values of the distance from the coil axis to the defect center, SD. This simulates the coil scanning across the defect. It should be noted that the defect sensitivity factor integration does not have to be repeated as SD increases or if another size defect is used if the lattice of points is fine enough. However, rather than choose the lattice fine enough to cover all defect depths presently stored in PCAVVSCN, we have set the program up so that it will compute a new lattice for each defect depth.

The arc length is calculated by the expression:

$$AL = 2\ RD \left[\cos^{-1}\left(\frac{SD^2 + RD^2 - DR^2}{2\ RD\ SD}\right)\right] \tag{18}$$

unless the defect encloses the coil origin. In this case, the arc length is:

$$AL = 2\ \pi\ RD$$

as long as RD is less than DR − SD. Then the expression in Eq. (18) should be used.

The defect volume can also be calculated by multiplying the arc length AL by the incremental step in RD. A comparison of the volume calculated by this method shows an agreement to within 0.1% if the lattice is fine enough to have 40 points across the defect. A lattice of 20 points will give an error of 0.3%.

The defect signal is averaged over the defect volume using the expression:

$$AVZDR + jAVZDI = \frac{\sum_{NR} AL\ (AVZDFR(NR) + jAVZDFI(NR))}{\sum_{NR} AL} \tag{19}$$

The division by the summation of the arc length in Eq. (19) helps reduce any errors in the computation of the defect volume and furnishes the proper normalization. The limit of the summation is only done for NR values that lie within the defect.

The output for the program is written on the file FORT9.

Summary

1. Dimension arrays and declare variable types.
2. Initialize variables.
3. Begin the integration loops, calculating the expressions that are independent of the position of the defect.
4. Assign a value to FZD, the center of the part of the defect we wish to work with.
5. Do the calculations that depend only upon the axial position of the defect.
6. Select a value for RD, the radial distance between the coil and the part of the defect we are working with.

7. Complete the integration.
8. Loop to 6 until done.
9. Loop to 4 until done.
10. Write the results of the calculations to a file.

Variables

ANG	The angle of half the arc length AL, θ in Figure 5.
DELTAR'	The normalized radial distance between adjacent data points.
DFDEP'	The distance from the side of the plate where the defect is located to the bottom of the defect in inches.
DFDIAM'	The diameter of the defect in inches.
DFM	The magnitude of the change in the normalized impedance of the coil due to the defect.
DFP	The phase of the change in the normalized impedance of the coil due to the defect.
DR	The normalized defect radius.
DRR	The normalized defect radius best described by the NDT points.
DSFI	The imaginary part of the defect sensitivity factor of the coil.
DSFR	The real part of the defect sensitivity factor of the coil.
FREQ*	The operating frequency in hertz.
L'	The length of the coil. The value is input in inches and normalized by the program.
L1'	The lift-off of the coil. The value is input in inches and normalized by the program.
LOD'	The number of the I/O unit connected to the output data file.
LOU	The number of the I/O unit connected to the printer.
NDF	The number denoting the particular defect in the array of defects.
NDT	The number of radial points across the defect.
NRD	The integer that denotes the radial location at which the defect sensitivity is calculated.
NRT'	The total number of radial points at which the defect sensitivity calculations are performed.
NRTT	The total number of points in the radial direction at which the averaged impedance change is calculated. Presently set to 2/3 of NRT.
NS'	The side of the plate where the defect is located. If NS=1, the defect is on the near side; if NS=2, the defect is on the far side.
NZT'	The total number of parts into which the defect is divided along its axis when the calculations are performed.

Q6	The inductance of the coil in air.
R1*	The inner radius of the coil. The value is input in inches and normalized by the program.
R2*	The outer radius of the coil. The value is input in inches and normalized by the program.
R3	The mean radius of the coil in inches.
RD	The normalized radial distance from the axis of the coil to the center of the defect.
RHO1*	The electrical resistivity of the plate in μΩ-cm.
T1*	The thickness of the plate. The value is input in inches and normalized by the program.
TRN*	The number of turns in the coil.
U1*	The relative magnetic permeability of the plate.
VOLN	The normalized volume of the defect.
WUSRR	The product of the angular frequency, the relative magnetic permeability, the electrical conductivity, and the square of the mean coil radius.
ZD	The normalized distance from the side of the plate where the defect is located to the bottom of the defect. A negative number.
ZNDFI	The imaginary part of the change in the normalized impedance of the coil due to the defect.
ZNDFR	The real part of the change in the normalized impedance of the coil due to the defect.
ZNIM	The imaginary part of the normalized coil impedance with no defects present.
ZNRL	The real part of the normalized coil impedance with no defects present.

Notes

1. Variable NZT controls the number of parts into which the defect is divided to perform the calculations. Since the theory upon which this program is based is more accurate for small defects, it is desirable to work with the defect in parts rather than as a whole.

Integration Section of Program PCAVVSCN

Symbol definitions

The following are definitions of the symbols used to describe the program variables appearing in the integration section of this program. All lengths are normalized unless otherwise noted.

α	Integration variable
α_1	$(\alpha^2+j\omega\mu\sigma_1\bar{r}^2)^{1/2}$
α_{22}	Defect shape and orientation factor
β_1	$(\alpha^2+j\omega\mu\sigma_1\bar{r}^2)^{1/2}/\mu$
c	Plate thickness
$J(x_2,x_1)$	Integral of $xJ_1(x)$ with respect to x from αx_1 to αx_2
$J_1(x)$	Bessel function of the first kind of order 1
ℓ	Length of coil
ℓ_1	Lift-off of coil
μ	Relative magnetic permeability of plate
$\underline{r}$	Coil-to-defect radial distance
$\bar{r}$	Mean radius of coil in inches
r_1	Inner radius of coil
r_2	Outer radius of coil
σ_1	Conductivity of plate
Vol_n	Normalized volume of defect
ω	Angular frequency at which circuit is driven
z	Depth to center of defect

Variables appearing in the integration section

Program variable	Symbolic equivalent
A1	$\alpha\ell+\exp(-\alpha\ell)-1$
A2	$1-\exp(-\alpha\ell)$
A2BI	$Im[\alpha^2-\beta_1^2]$
A2BR	$Re[\alpha^2-\beta_1^2]$
A3	$\exp(-\alpha\ell_1)$
AMBI	$Im[(\alpha-\beta_1)^2]$
AMBR	$Re[(\alpha-\beta_1)^2]$
APBI	$Im[(\alpha+\beta_1)^2]$
APBR	$Re[(\alpha+\beta_1)^2]$
AVZDFI(NR)	(See note I3.)
AVZDFR(NR)	(See note I3.)
B1	(See note I2.)
B2	(See note I2.)

CFAC	$[J(r_2,r_1)]^2$
CHECK	{See note I2.}
CSDI	$\mathrm{Re}[\exp(\alpha_1 z)]$
CSTI	$\mathrm{Re}[\exp(-2\alpha_1 c)]$
CSZI	$\mathrm{Re}[\exp(-\alpha_1(2c+z))]$
DENI	$\mathrm{Im}[(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)]$
DENR	$\mathrm{Re}[(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)]$
DFAC	$\frac{1}{\alpha^3}\ (1-\exp(-\alpha\ell))\exp(-\alpha\ell_1)J(r_2,r_1)\ d\alpha$
ERR	{See note I2.}
FZD	z {See note I4.}
ISTEPS	{See note I2.}
RI9	{See note I2.}
RJ1	$J_1(\alpha r)$
S1	$d\alpha$
S2	{See note I2.}
SMAIR	$\int_0^\infty \frac{2(\alpha\ell+\exp(-\alpha\ell)-1)[J(r_2,r_1)]^2\ d\alpha}{\alpha^6}$
SMIMPI	$\mathrm{Im}\left[\int_0^\infty \frac{(1-\exp(-\alpha\ell))^2\exp(-2\alpha\ell_1)[J(r_2,r_1)]^2}{\alpha^6}\left[\frac{(\alpha^2-\beta_1^{\ 2})-(\alpha^2-\beta_1^{\ 2})\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]d\alpha\right]$

SMIMPR	$\mathrm{Re}\left[\int_0^\infty \frac{(1-\exp(-\alpha\ell))^2\exp(-2\alpha\ell_1)[J(r_2,r_1)]^2}{\alpha^6}\left[\frac{(\alpha^2-\beta_1^2)-(\alpha^2-\beta_1^2)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]d\alpha\right]$
SMZDFI	$\mathrm{Im}\left[\int_0^\infty \frac{1}{\alpha^3}(1-\exp(-\alpha\ell))\exp(-\alpha\ell_1)J(r_2,r_1)J_1(\alpha r)\left[\frac{(\alpha+\beta_1)\exp(\alpha_1 z)-(\alpha-\beta_1)\exp(-\alpha_1(2c+z))}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]d\alpha\right]$
SMZDFR	$\mathrm{Re}\left[\int_0^\infty \frac{1}{\alpha^3}(1-\exp(-\alpha\ell))\exp(-\alpha\ell_1)J(r_2,r_1)J_1(\alpha r)\left[\frac{(\alpha+\beta_1)\exp(\alpha_1 z)-(\alpha-\beta_1)\exp(-\alpha_1(2c+z))}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]d\alpha\right]$
SNDI	$\mathrm{Im}[\exp(\alpha_1 z)]$
SNTI	$-\mathrm{Im}[\exp(-2\alpha_1 c)]$
SNZI	$-\mathrm{Im}[\exp(-\alpha_1(2c+z))]$
TI	$\mathrm{Im}[2\alpha_1 c]$
TR	$\mathrm{Re}[2\alpha_1 c]$
TZI	$\mathrm{Im}[\alpha_1(2c+z)]$
TZR	$\mathrm{Re}[\alpha_1(2c+z)]$
X	α
X1	$\mathrm{Re}[\beta_1]$
X1X	$\mathrm{Re}[\alpha(\beta_1-\alpha)]$
XJR1	$J(r_1,0)$
XJR2	$J(r_2,0)$
XL	$\alpha\ell$
XPDR	$\exp[\mathrm{Re}(\alpha_1 z)]$

XPTR	$\exp[\mathrm{Re}(-2\alpha_1 c)]$
XPZR	$\exp[\mathrm{Re}(-\alpha_1(2c+z))]$
XRD	αr
XX	α^2
XX1	$\mathrm{Re}[\alpha(\alpha+\beta_1)]$
XY1	$\mathrm{Im}[\alpha(\alpha+\beta_1)]=\mathrm{Im}[\alpha(\beta_1-\alpha)]$
Y1	$\mathrm{Im}[\beta_1]$
ZDI	$\mathrm{Im}[\alpha_1 z]$
ZDIM	$\mathrm{Im}\left[\dfrac{(\alpha+\beta_1)\exp(\alpha_1 z)-(\alpha-\beta_1)\exp(-\alpha_1(2c+z))}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]$
ZDR	$\mathrm{Re}[\alpha_1 z]$
ZDRL	$\mathrm{Re}\left[\dfrac{(\alpha+\beta_1)\exp(\alpha_1 z)-(\alpha-\beta_1)\exp(-\alpha_1(2c+z))}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]$
ZIM	$\mathrm{Im}\left[\dfrac{(\alpha^2-\beta_1^2)-(\alpha^2-\beta_1^2)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]$
ZNDI	$\mathrm{Im}[\alpha(\alpha+\beta_1)\exp(\alpha_1 z)]$
ZNDR	$\mathrm{Re}[\alpha(\alpha+\beta_1)\exp(\alpha_1 z)]$
ZNUI	$\mathrm{Im}[(\alpha^2-\beta_1^2)-(\alpha^2-\beta_1^2)\exp(-2\alpha_1 c)]$
ZNUR	$\mathrm{Re}[(\alpha^2-\beta_1^2)-(\alpha^2-\beta_1^2)\exp(-2\alpha_1 c)]$
ZRL	$\mathrm{Re}\left[\dfrac{(\alpha^2-\beta_1^2)-(\alpha^2-\beta_1^2)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]$

<u>Notes for the integration section</u>

I1. A number of the variables in the integration section are not always assigned their exact values but are approximated in certain cases to save time. For example, rather than calculate the exponential of a very small number, the Maclaurin series expansion is sometimes used. Also, the exponential of a very large negative number is usually treated as zero.

I2. Several variables appear in the integration section of the program which play no part in the calculations being done. They are merely there to do such things as to determine the maximum step size which can be used while still accurately calculating the integrals.

I3. Variables AVZDFR(NR) and AVZDFI(NR) are the averages of the elements in arrays SMZDFR(NZ,NR) and SMZDFI(NZ,NR), respectively, summed over NZT. For improved accuracy, the defect is divided into NZT parts over the depth of the defect, to perform the calculations. The NZT elements of each array contain the calculations for these NZT parts. In addition, the defect is divided into NRTT parts in the radial direction. The contributions from these elements are averaged to give the total effect of all of these parts on the impedances of the coils.

I4. The variable FZD is the normalized depth from the near side of the plate to the center of the part of the defect with which the program is working. It is not the depth of the center of the actual defect.

Sample output

Partial printer output of program PCAVVSCN:

```
PCAVVSCN 20 POINTS USED   FAR SIDE DEFECT  TIME  9: 1:43  DATE  8/ 9/89
     IN RAD      OT RAD      LENGTH   O-LIFTOFF  L.O. VAR    CLADTH     DF DEP
ACT  0.1000      0.4100      0.1000     0.0100     0.0200     0.2540    -0.2210
NOR  0.3922      1.6078      0.3922     0.0392     0.0784     0.9961    -0.8667
 RBAR 0.2550 FREQ=5.000000E+02 RHO=4.0900 PERM=1.000 WUSRR=4.0493
 NORM IMPD:RL  0.155213 IM  0.833067 AIR IND 6.252919E-03 VOLN 5.1590E-01
 LIFT-OFF PHASE=  140.00
  R(NOR)       MAG   PHASE
  0.010   0.001331  93.93
  0.030   0.001343  93.93
  0.050   0.001358  93.92
  0.070   0.001380  93.91
  0.090   0.001409  93.90
  0.110   0.001446  93.88
  0.130   0.001490  93.86
  0.150   0.001541  93.84
  0.170   0.001601  93.81
  0.190   0.001668  93.78
```

Listing

```
      PROGRAM PCAVVSCN
C     VERSION August 9, 1989
C     PROGRAM TO CALCULATE THE NORMALIZED IMPEDANCE CHANGE FOR A FLAT
C     BOTTOM HOLE IN A SINGLE PLANAR CONDUCTOR.  THE DEFECT IS
C     CALCULATED AT AN ARRAY OF POINTS RATHER THAN A SINGLE POINT
C     THE DEFECT SIGNAL IS AVERAGED OVER BOTH THE R AND Z DIMENSIONS.
C     THE CALCULATIONS ARE STEPPED IN THE R DIRECTION, AS THE DEFECT IS
C     SCANNED BY THE PROBE.
C DIA NS .221 .188 .158 .129 .097 .076 FS .222 .189 .158 .129 .098 .077
C DEP NS .221 .188 .157 .128 .096 .078 FS .221 .188 .156 .128 .096 .078
      CHARACTER SIDE(2)*4
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 L,L1,L2
      DIMENSION DFDIAM(12),DFDEP(12),NSIDE(12)
      DIMENSION S1(6),S2(6),ERR(6),RJ(200)
      DIMENSION SMZDFR(20,200),SMZDFI(20,200),AVZDFR(200),AVZDFI(200)
      DATA LOU/8/,LOD/9/,PI/3.141592653/,NZT/20/,NDF/7/,NRT/150/
      DATA S1/.005,.02,.05,.1,.5,2./
      DATA S2/1.0,2.0,5.0,10.0,50.0,200.0/
      DATA ERR/.1,.01,.001,1.E-4,1.E-5,1.E-6/
      DATA FREQ/500./,RHO1/4.09/,U1/1.0/
      DATA TRN/800./,T1/0.254/,SIDE/'NEAR',' FAR'/,DELTAR/0.02/
      DATA DFDIAM/
     *.221,.188,.158,.129,.097,.076,.222,.189,.158,.129,.098,.077/
      DATA DFDEP/
     *.221,.188,.157,.128,.096,.078,.221,.188,.156,.128,.096,.078/
      DATA NSIDE/1,1,1,1,1,1,2,2,2,2,2,2/
      NS=NSIDE(NDF)
C     TIME AND DATE ARE PRINTED
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,2)NZT,SIDE(NS),IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT('PCAVVSCN',I3,' POINTS USED  ',A4,' SIDE DEFECT',
     *'  TIME ',I2,':',I2,':',I2,'  DATE ',I2,'/',I2,'/',I2)
      WRITE(LOU,5)
    5 FORMAT(5X,'IN RAD',4X,'OT RAD',4X,'LENGTH',2X,'O-LIFTOFF'
     *,2X,'L.O. VAR',3X,'CLADTH',4X,'DF DEP')
C     COIL P255A
      R1=0.100
      R2=0.410
      L=0.100
C     COIL P371A      R1=0.275      R2=0.4665      L=0.265
      L1=0.01
      L2=0.02
      R3=.5*(R1+R2)
      ZD=-DFDEP(NDF)
      WRITE(LOU,10)R1,R2,L,L1,L2,T1,ZD
      R1=R1/R3
```

```
      R2=2.0-R1
      L=L/R3
      L1=L1/R3
      L2=L2/R3
      ZD=ZD/R3
      T1=T1/R3
C    VOLN=0.1666667*PI*(DFDIAM(NDF)/R3)*(DFDIAM(NDF)/R3)*(DFDIAM(NDF)/R3)

      VOLN=PI*DFDIAM(NDF)*DFDIAM(NDF)*DFDEP(NDF)/(4.*R3*R3*R3)
      DR=DFDIAM(NDF)*0.5/R3
      WUSRR=0.5093979*U1*R3*R3*FREQ/RHO1
      WRITE(LOU,15)R1,R2,L,L1,L2,T1,ZD
   10 FORMAT('ACT ',7(F7.4,3X))
   15 FORMAT('NOR ',7(F7.4,3X))
C       WRITE(LOU,18) DFDIAM(NDF),DFDEP(NDF)
  18  FORMAT(' CYL FLAW: DIAM=',F7.4,', DEPTH=',F7.4)
      WRITE(LOU,20)R3,FREQ,RHO1,U1,WUSRR
   20 FORMAT(' RBAR',F7.4,' FREQ= ',1PE13.6,' RHO=',0PF9.4,
     *' PERM=',F7.3,' WUSRR=',F9.4)
      SMAIR=0.0
      SMIMPR=0.0
      SMIMPI=0.0
      SMIMPR1=0.0
      SMIMPI1=0.0
      DO 25 NR=1,NRT
      DO 25 NZ=1,NZT
      SMZDFR(NZ,NR)=0.0
   25 SMZDFI(NZ,NR)=0.0
      B1=0.0
      B2=S2(1)
      DO 100 JKL=1,6
   30 RI9=SMAIR
      X=B1-0.5*S1(JKL)
C     DETERMINE NUMBER OF STEPS,ISTEPS,BASED ON STEP SIZE&INTERVAL
      ISTEPS=DNINT((B2-B1)/S1(JKL))
      DO 95 I=1,ISTEPS
      X=X+S1(JKL)
      CALL BESSEL(XJR2,X,R2)
      CALL BESSEL(XJR1,X,R1)
      XL=X*L
      IF(XL.GT.5.0E-3) GO TO 60
      A1=XL*XL*(0.5-XL/6.0)
      GO TO 80
   60 IF(XL.GT.75.0) GO TO 70
      A1=XL+DEXP(-XL)-1.0
      GO TO 80
   70 A1=XL-1.0
   80 CFAC=S1(JKL)*(XJR2-XJR1)*(XJR2-XJR1)
      SMAIR=SMAIR+CFAC*2.*A1
      IF(X*L1.GT.75.)GO TO 95
      XX=X*X
```

```
      X1=DSQRT(0.5*(XX+DSQRT(XX*XX+WUSRR*WUSRR)))/U1
      Y1=WUSRR/(2.*X1*U1*U1)
      A2=XL-A1
      A3=DEXP(-X*L1)
      A4=DEXP(-X*L2)
      APBR=(X+X1)*(X+X1)-Y1*Y1
      APBI=2.*Y1*(X+X1)
      AMBR=(X-X1)*(X-X1)-Y1*Y1
      AMBI=-2.*Y1*(X-X1)
      A2BR=0.0
      A2BI=-2.*X1*Y1
      ZNUR=A2BR
      ZNUI=A2BI
      DENR=APBR
      DENI=APBI
      DNCJ=DENR*DENR+DENI*DENI
C     SECTION THAT MULTIPLIES BY DEXP(ALPHA1*ZDEFECT)
      DO 91 NZ=1,NZT
C     NEAR SIDE DEFECT CALCULATION
      FZD=(FLOAT(NZ)-.5)*ZD/FLOAT(NZT)
C     FAR SIDE DEFECT CALCULATION
      IF(NS.EQ.2)FZD=-T1-FZD
      ZDR=X1*U1*FZD
      IF(ZDR.LT.-60.0)GO TO 93
      ZDI=Y1*U1*FZD
      XPDR=DEXP(ZDR)
      CSDI=DCOS(ZDI)*XPDR
      SNDI=DSIN(ZDI)*XPDR
      XX1=X*X1+XX
      XY1=X*Y1
      X1X=X*X1-XX
      ZNDR=XX1*CSDI-XY1*SNDI
      ZNDI=XX1*SNDI+XY1*CSDI
C     SECTION THAT MULTIPLIES BY DEXP(ALPHA1*(2*TH+ZDEFECT))
      TZR=X1*U1*(2.*T1+FZD)
      IF(TZR.GT.60.)GO TO 87
      TZI=Y1*U1*(2.*T1+FZD)
      XPZR=DEXP(-TZR)
      CSZI=DCOS(TZI)*XPZR
      SNZI=DSIN(TZI)*XPZR
      ZNDR=XX1*CSDI-XY1*SNDI+X1X*CSZI+XY1*SNZI
      ZNDI=XX1*SNDI+XY1*CSDI+XY1*CSZI-X1X*SNZI
C     SECTION THAT MULTIPLIES BY DEXP(-ALPHA1*2*CLADTH)
      TR=2.*X1*U1*T1
      IF(TR.GT.60.)GO TO 87
      TI=2.*Y1*U1*T1
      XPTR=DEXP(-TR)
      CSTI=DCOS(TI)*XPTR
      SNTI=DSIN(TI)*XPTR
      DENR=APBR-AMBR*CSTI-AMBI*SNTI
      DENI=APBI-AMBI*CSTI+AMBR*SNTI
```

```
      ZNUR=A2BR-A2BR*CSTI-A2BI*SNTI
      ZNUI=A2BI-A2BI*CSTI+A2BR*SNTI
      DNCJ=DENR*DENR+DENI*DENI
   87 ZDRL=(ZNDR*DENR+ZNDI*DENI)/DNCJ
      ZDIM=(DENR*ZNDI-ZNDR*DENI)/DNCJ
      DFAC=A2*A3*S1(JKL)*(XJR2-XJR1)
C     LOOP OVER THE R VARIATION FOR THE DEFECT
      DO 90 NR=1,NRT
      IF(NZ.GT.1)GO TO 89
      RD=(FLOAT(NR)-.5)*DELTAR
      XRD=X*RD
      CALL BESEL1(XRD,RJ1)
      RJ(NR)=RJ1
   89 SMZDFR(NZ,NR)=SMZDFR(NZ,NR)+RJ(NR)*ZDRL*DFAC
   90 SMZDFI(NZ,NR)=SMZDFI(NZ,NR)+RJ(NR)*ZDIM*DFAC
   91 CONTINUE
   93 ZRL=(ZNUR*DENR+ZNUI*DENI)/DNCJ
      ZIM=(DENR*ZNUI-ZNUR*DENI)/DNCJ
      SMIMPR=SMIMPR+A2*A2*A3*A3*ZRL*CFAC
      SMIMPI=SMIMPI+A2*A2*A3*A3*ZIM*CFAC
      SMIMPR1=SMIMPR1+A2*A2*A4*A4*ZRL*CFAC
      SMIMPI1=SMIMPI1+A2*A2*A4*A4*ZIM*CFAC
   95 CONTINUE
      B1=B2
      B2=B2+S2(JKL)
      CHECK=(SMAIR-RI9)/SMAIR
      IF(ABS(CHECK).GT.ERR(JKL)) GO TO 30
  100 CONTINUE
  135 Q6=0.0254*4.0E-7*TRN*TRN*R3*PI*PI*SMAIR/(L*(R2-R1))**2
      ZNIM=(SMIMPR+SMAIR)/SMAIR
      ZNRL=-SMIMPI/SMAIR
      ZNIM1=(SMIMPR1+SMAIR)/SMAIR
      ZNRL1=-SMIMPI1/SMAIR
      ZLOR=ZNRL1-ZNRL
      ZLOI=ZNIM1-ZNIM
      PLO=ATAN2(ZLOI,ZLOR)*180./PI
      WRITE(LOU,140)ZNRL,ZNIM,Q6,VOLN
      WRITE(LOU,145)PLO
  145 FORMAT(' LIFT-OFF PHASE= ',0PF7.2)
C      WRITE(LOU,160)DSFR,DSFI,VOLN
  140 FORMAT(' NORM IMPD:RL',F10.6,' IM',F10.6,
     *' AIR IND',1PE13.6,' VOLN',1PE11.4)
  150 FORMAT(' NORM CHG:RL ',0PF10.6,' IM',0PF10.6,
     *' MAG ',0PF10.6,' PHA ',0PF7.2)
  160 FORMAT(' NORM DSF:RL',1PE11.4,' IM',1PE11.4,' VOLN',1PE11.4)
      DO 200 NR=1,NRT
C     AVERAGE OVER DEFECT POINTS AT DIFFERENT DEPTHS
      AVZDFR(NR)=0.0
      AVZDFI(NR)=0.0
      DO 220 NZ=1,NZT
      AVZDFR(NR)=AVZDFR(NR)+SMZDFR(NZ,NR)
```

```
  220 AVZDFI(NR)=AVZDFI(NR)+SMZDFI(NZ,NR)
      AVZDFR(NR)=AVZDFR(NR)/FLOAT(NZT)
      AVZDFI(NR)=AVZDFI(NR)/FLOAT(NZT)
  200 CONTINUE
C     AVERAGE OVER THE DEFECT AT DIFFERENT R VALUES
C     SD=LOCATION OF DEFECT CENTER
C     RD=FIELD POINT WHERE DEFECT IMPEDANCE CHANGE IS CALCULATED
C     DR=DEFECT RADIUS - ALL DIMENSIONS ARE NORMALIZED
      WRITE(LOU,*)' R(NOR)      MAG   PHASE'
      NDT=2.0*DR/DELTAR
      DRR=FLOAT(NDT)*DELTAR*0.5
C     START SD AT 0 IF NDT EVEN, 1ST HALF STEP IF NDT ODD
      SD=FLOAT(MOD(NDT,2))*.5*DELTAR
      NRTT=2*NRT/3
      DO 400 NR=1,NRTT
      AVZDR=0.0
      AVZDI=0.0
      SUML=0.0
      RD=SD-DRR+DELTAR*.5
      DO 300 ND=1,NDT
      AL=0.0
      IF(RD.LT.0.0)GO TO 290
      NRD=(0.50001+RD/DELTAR)
      IF(RD.LT.DR-SD) AL=2.*3.14159*RD
      IF(RD.LT.DR-SD) GO TO 280
      ANG=(SD*SD+RD*RD-DR*DR)/(2.*RD*SD)
      AL=2.*RD*ACOS(ANG)
  280 CONTINUE
  290 RD=RD+DELTAR
      SUML=SUML+AL
      AVZDR=AVZDR+AVZDFR(NRD)*AL
      AVZDI=AVZDI+AVZDFI(NRD)*AL
  300 CONTINUE
      AVZDR=AVZDR/SUML
      AVZDI=AVZDI/SUML
      DSFR=-1.5*WUSRR*(AVZDR*AVZDR-AVZDI*AVZDI)/(SMAIR*PI)
      DSFI=-1.5*WUSRR*2.0*AVZDR*AVZDI/(SMAIR*PI)
      ZNDFR=VOLN*DSFR
      ZNDFI=VOLN*DSFI
      DFM=VOLN*SQRT(DSFR*DSFR+DSFI*DSFI)
      DFP=ATAN2(DSFI,DSFR)*180./PI
      WRITE(LOU,380)SD,DFM,DFP
      WRITE(LOD,390)DFM,DFP
  380 FORMAT(F7.3,F10.6,F7.2)
  390 FORMAT(E12.5,F7.2)
      SD=SD+DELTAR
  400 CONTINUE
 1000 STOP 'JOB '
      END
```

PCGRAPH plots two sets of data on same graph

Program PCGRAPH plots two sets of data on the same graph and sends the output to the screen and the printer. It is normally used to compare the calculated and the experimental values of the impedance change of a pancake coil due to a defect at different distances from the defect. The initial plot is made on the CRT and then the program sends the data on the screen to the printer.

Summary

1. Dimension arrays and declare variable types.
2. Initialize variables.
3. Open the files containing the data.
4. Read in the data and convert them to a form usable by the graphics subroutines.
5. Graph the data on the screen.
6. Send the contents of the screen to the printer.

Variables

CX	An array containing the real parts of one set of the data to be plotted.
CY	An array containing the imaginary parts of one set of the data to be plotted.
DFDEP*	The depth to the bottom of the defect in inches.
DFDIAM*	The diameter of the defect in inches.
EX	An array containing the real parts of one set of the data to be plotted.
EY	An array containing the imaginary parts of one set of the data to be plotted.
FF	Character variable containing the form-feed character.
FREQ*	The operating frequency in hertz.
GIM	Factor by which the imaginary parts of the data to be graphed are multiplied to make the graphs as large as possible.
GRL	Factor by which the real parts of the data to be graphed are multiplied to make the graphs as large as possible.
L*	The length of the coil. The value is input in inches and normalized by the program.
L1*	The lift-off of the coil. The value is input in inches and normalized by the program.
LOEC*	The number of the I/O unit connected to the file containing the calculated data.
LOEE*	The number of the I/O unit connected to the file containing the experimental data.
LOU	The number of the I/O unit connected to the printer.

MODE	The screen graphics mode to be used. Mode 16 is the EGA high resolution mode.
OIM	A number which must be added to the imaginary parts of the data to be graphed in order to move the origin to the desired place on the screen.
ORL	A number which must be added to the real parts of the data to be graphed in order to move the origin to the desired place on the screen.
R1*	The inner radius of the coil. The value is input in inches and normalized by the program.
R2*	The outer radius of the coil. The value is input in inches and normalized by the program.
R3	The mean radius of the coil in inches.
RHO1*	The electrical resistivity of the plate in $\mu\Omega$-cm.
T1*	The thickness of the plate. The value is input in inches and normalized by the program.
TRN*	The number of turns in the coil.
U1*	The relative magnetic permeability of the plate.
VOLN	The normalized volume of the defect.
WUSRR	The product of the angular operating frequency, the magnetic permeability of the plate, the electrical conductivity of the plate, and the square of the coil mean radius.
ZD	The normalized depth to the bottom of the defect. A negative number.

Listing

```
      PROGRAM PCGRAPH
      CHARACTER*1 FF
      DIMENSION CX(200),CY(200),EX(200),EY(200)
      REAL GRL,GIM,L,L1
      DATA LOU/8/,LOEE/38/,LOEC/39/
      DATA PI/3.141592653/,MODE/16/
      DATA ORL/330/,OIM/50/
      DATA FREQ/500./,RHO1/4.054/,U1/1.0/,TRN/800./
      DATA DFDIAM/0.1881/,DFDEP/0.1881/
      FF=CHAR(12)
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT(' PCGRAPH   TIME ',I2,':',I2,':',I2
     *,'  DATE ',I2,'/',I2,'/',I2,\)
      OPEN(LOEE,FILE='QEXPN2.DAT',STATUS='OLD')
      OPEN(LOEC,FILE='OCALN2.DAT',STATUS='OLD')
      WRITE(LOU,*)'     NEAR SIDE DEFECT'
      WRITE(LOU,5)
    5 FORMAT(5X,'IN RAD',4X,'OT RAD',4X,'LENGTH',4X,'LIFTOFF'
     *,3X,'CLADTH',4X,'DF DEP',4X,'DFDIAM')
      R1=0.1000
      R2=0.4100
      L=0.1000
      L1=0.010
      T1=0.250
      R3=0.5*(R1+R2)
      ZD=-DFDEP
      WRITE(LOU,10)R1,R2,L,L1,T1,ZD,DFDIAM
   10 FORMAT('ACT:',7(F7.4,3X))
      R1=R1/R3
      R2=2.0-R1
      L=L/R3
      L1-L1/R3
      T1=T1/R3
      ZD=ZD/R3
      DFDIAM=DFDIAM/R3
      WRITE(LOU,15)R1,R2,L,L1,T1,ZD,DFDIAM
   15 FORMAT('NOR:',7(F7.4,3X))
      VOLN=PI*DFDIAM*DFDIAM*DFDEP/(4.*R3*R3*R3)
      WUSRR-0.5093979*U1*R3*R3*FREQ/RHO1
      WRITE(LOU,18)R3,RHO1,U1,WUSRR
   18 FORMAT(' RBAR',F7.4,' RHO=',0PF7.4,
     *' PERM=',F7.3,' WUSRR=',F9.4)
      CALL QSMODE(MODE)
      CALL GRID
   11 CXMAX=0.
      CYMAX=0.
```

```
      EXMAX=0.
      EYMAX=0.
      EMMAX=0.
      CMMAX=0.
      I=1
   20 READ(LOEE,*,END=29)EMAG,EPHA
      EPHA=EPHA*PI/180.
      EX(I)=EMAG*COS(EPHA)
      EY(I)=EMAG*SIN(EPHA)
      IF(EX(I).GT.EXMAX)EXMAX=EX(I)
      IF(EY(I).GT.EYMAX)EYMAX=EY(I)
      IF(EMAG.GT.EMMAX) THEN
      EMMAX=EMAG
      EPMMAX=EPHA
      END IF
      I=I+1
      GO TO 20
   29 EX(I)=999.
      I=1
   30 READ(LOEC,*,END=40)CMAG,CPHA
      CPHA=CPHA*PI/180.
      CX(I)=CMAG*COS(CPHA)
      CY(I)=CMAG*SIN(CPHA)
      IF(CX(I).GT.CXMAX)CXMAX=CX(I)
      IF(CY(I).GT.CYMAX)CYMAX=CY(I)
      IF(CMAG.GT.CMMAX) THEN
      CMMAX=CMAG
      CPMMAX=CPHA
      END IF
      I=I+1
      GO TO 30
   40 CX(I)=999.
      EGIM=300./EXMAX
      EGRL=300./EYMAX
      IF(EGIM.GT.EGRL) THEN
      EGIM=EGRL
      ELSE
      EGRL=EGIM
      END IF
      CGIM=300./CXMAX
      CGRL=300./CYMAX
      IF(CGIM.GT.CGRL) THEN
      CGIM=CGRL
      ELSE
      CGRL=CGIM
      END IF
      IF(EGIM.GT.CGIM) THEN
      GIM=CGIM
      GRL=CGRL
      ELSE
      GIM=EGIM
```

```
      GRL=EGRL
      END IF
      EPMMAX=EPMMAX*180./PI
      CPMMAX=CPMMAX*180./PI
      WRITE(LOU,*)
      WRITE(LOU,*)'          MAX MAG       PHA AT MAX MAG'
      WRITE(LOU,52)EMMAX,EPMMAX
      WRITE(LOU,53)CMMAX,CPMMAX
   52 FORMAT(' EXP:   ',D11.4,7X,F7.3)
   53 FORMAT(' CAL:   ',D11.4,7X,F7.3)
      IR1=GRL*EX(1)+ORL
      IM1=GIM*EY(1)+OIM
      I=2
   50 IF(EX(I).GT.900) GO TO 59
      IR2=GRL*EX(I)+ORL
      IM2=GIM*EY(I)+OIM
      CALL QLINE(IR1,IM1,IR2,IM2,15)
      IR1=IR2
      IM1=IM2
      I=I+1
      GO TO 50
   59 IR1=GRL*CX(1)+ORL
      IM1=GIM*CY(1)+OIM
      I=2
   60 IF(CX(I).GT.900) GO TO 70
      IR2=GRL*CX(I)+ORL
      IM2=GIM*CY(I)+OIM
      CALL QLINE(IR1,IM1,IR2,IM2,11)
      IR1=IR2
      IM1=IM2
      I=I+1
      GO TO 60
   70 CONTINUE
      WRITE(LOU,*)
      WRITE(LOU,*)
      WRITE(LOU,*)
      CALL PRTSC
      WRITE(LOU,*)FF
 1000 END
```

PCINV inverts scan of pancake coil data to get depth and volume

Program PCINV calculates the depth and volume of a defect given the change in the impedance of a pancake coil as it scans past the defect. The program works with experimental data stored in a file by program PCFIX or with calculated data stored by program PCAVZSCN, and it uses a lookup file built by program PCBLDF. The program calculates the integral of minus the impedance change in the coil due to the defect with respect to the radial distance between the coil axis and the defect from the inner radius of the coil to the outer radius of the coil. It then compares the phase of this complex integral with the phases stored in a lookup file by program PCBLDF. When the phase of the integral matches the phase in the lookup file, the program reads the corresponding depth and magnitude from the lookup file. The depth is equal to the depth of the defect, and the magnitude can be used to find the volume of the defect. The defect is assumed to have the shape of a flat-bottomed hole.

Summary

1. Declare variable types.
2. Initialize variables.
3. Open the file containing the experimental data, read in the data, and calculate the integral.
4. Find the depth and magnitude in the lookup file corresponding to the phase of the integral.
5. Calculate the inverted defect depth and volume based on the experimental data.
6. Open the file containing the calculated data, read in the data, and calculate the integral.
7. Find the depth and magnitude in the lookup file corresponding to the phase of the integral.
8. Calculate the inverted defect depth and volume based on the calculated data.

Variables

AIRIND*	The inductance in henries of the coil in air.
DELRDC	The normalized distance between adjacent calculated data points.
DELRDE	The normalized distance between adjacent experimental data points.
DEPTH	The inverted depth of the center of the defect. A negative number.
DFDEP*	The actual depth of the defect in inches.
DFDIAM*	The actual diameter of the defect in inches.
DFM	The magnitude of the change in the impedance of the coil due to the defect.
DFP	The phase of the change in the impedance of the coil due to the defect.
FREQ*	The operating frequency in hertz.

L*	The length of the coil. The value is input in inches and normalized by the program.
L1*	The lift-off of the coil. The value is input in inches and normalized by the program.
LHSPHA	The phase of the integral calculated by the program.
LOEC*	The number of the I/O unit connected to the file containing the calculated data.
LOEE*	The number of the I/O unit connected to the file containing the experimental data.
LOU	The number of the I/O unit connected to the printer.
R1*	The inner radius of the coil. The value is input in inches and normalized by the program.
R2*	The outer radius of the coil. The value is input in inches and normalized by the program.
R3	The mean radius of the coil in inches.
RDC	The radial distance between the axis of the coil and the center of the defect in the part of the program which inverts the calculated data.
RDE	The radial distance between the axis of the coil and the center of the defect in the part of the program which inverts the experimental data.
RHO1*	The electrical resistivity of the plate in $\mu\Omega$-cm.
RHSMAG	The magnitude retrieved from the lookup file.
SMAIR	A quantity related to the inductance of the coil in air. It is used to normalize the impedance of the coil.
T1*	The thickness of the plate. The value is input in inches and normalized by the program.
TRN*	The number of turns in the coil.
U1*	The relative magnetic permeability of the plate.
VOL1	The inverted normalized volume of the defect.
VOLN	The actual normalized volume of the defect.
WUSRR	The product of the angular operating frequency, the magnetic permeability of the plate, the electrical conductivity of the plate, and the square of the mean radius of the coil.
XMAG	The magnitude of the integral calculated by the program.
ZD	The normalized depth of the center of the defect. A negative number.

Listing

```
      PROGRAM PCINV
C     VERSION July 27, 1988
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 L,L1,L2,JANR21,LHSPHA
      REAL*8 NMR1A,NMR1B,NMR1,NMR2,NMR
      REAL*8 NMI1A,NMI1B,NMI1,NMI2,NMI
      DATA LOEE/38/,LOEC/39/,LOU/8/,PI/3.141592653/
      DATA FREQ/500./,RHO1/4.054/,U1/1.0/,DFDIAM/0.1881/,DFDEP/0.1881/
      DATA DELRDE/0.0392/,DELRDC/0.05/,TRN/800./,T1/0.25/
      DATA AIRIND/6.252919E-03/
      OPEN(LOEE,FILE='QEXPN2.DAT',STATUS='OLD')
      OPEN(LOEC,FILE='OCALN2.DAT',STATUS='OLD')
C     TIME AND DATE ARE PRINTED
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT(' PCINV    TIME ',I2,':',I2,':',I2
     *,'  DATE ',I2,'/',I2,'/',I2)
      WRITE(LOU,5)
    5 FORMAT(6X,'IN RAD',4X,'OT RAD',4X,'LENGTH',4X,'LIF OFF'
     *,3X,'CLADTH',4X,'DF RAD',4X,'DF DEP')
      R1=0.1000
      R2=0.4100
      L=0.1000
      L1=0.01
      R3=.5*(R1+R2)
      RDE=R3*DELRDE*.5
      RDC=R3*DELRDC*.5
      ZD=-0.5*DFDEP
      WRITE(LOU,10)R1,R2,L,L1,T1,RD,ZD
      R1=R1/R3
      R2=2.0-R1
      L=L/R3
      L1=L1/R3
      RDE=RDE/R3
      RDC=RDC/R3
      ZD=ZD/R3
      T1=T1/R3
      L2=L+L1
      VOLN=PI*DFDIAM*DFDIAM*DFDEP/(4.*R3*R3*R3)
      WUSRR=0.5093979*U1*R3*R3*FREQ/RHO1
      WRITE(LOU,15)R1,R2,L,L1,T1,RD,ZD
   10 FORMAT('ACT   ',7(F7.4,3X))
   15 FORMAT('NOR   ',7(F7.4,3X))
      WRITE(LOU,20)R3,FREQ,RHO1,U1,WUSRR
   20 FORMAT(' RBAR',F7.4,' FREQ= ',1PE13.6,' RHO=',0PF9.4,
     *' PERM=',F7.3,' WUSRR=',F9.4)
   28 CONTINUE
```

```
      WRITE(LOU,*)
      WRITE(LOU,*)
      WRITE(LOU,278)
      WRITE(LOU,279)ZD,VOLN
  278 FORMAT('              DEPTH          VOLUME')
  279 FORMAT('ACTUAL:     ',F12.5,1X,E14.5)
      SMAIR=AIRIND*(L*(R2-R1))**2/(0.0254*4.E-07*TRN*TRN*R3*PI*PI)
      SIVR=0.0
      SIVI=0.0
      M=0
   70 READ(LOEE,*,END=78)DFM,DFP
      IF(RDE.LT.R1)GOTO 76
      IF(RDE.GT.R2)GOTO 78
C      WRITE(0,*)RDE,DFM,DFP
      DFP=DFP*(PI/180.)
      XFACT=DSQRT(DFM)*DELRDE
      SIVI=SIVI-XFACT*DCOS(0.5*DFP)
      SIVR=SIVR+XFACT*DSIN(0.5*DFP)
      LHSPHA=ATAN2(SIVI,SIVR)
      XMAG=SQRT(SIVR*SIVR+SIVI*SIVI)
   76 RDE=RDE+DELRDE
      GO TO 70
   78 CONTINUE
      LHSPHA=LHSPHA*180./PI
C      WRITE(LOU,*)LHSPHA
      CALL NDEP(DEPTH,RHSMAG,LHSPHA)
      VOL1=(2*SMAIR*PI)*(XMAG*XMAG)/((3*WUSRR)*(RHSMAG*RHSMAG))
      WRITE(LOU,379)DEPTH,VOL1
  379 FORMAT('INV EXP:  ',F12.5,1X,E14.5)
   79 SIVR=0.0
      SIVI=0.0
      M=0
   80 READ(LOEC,*,END=88)DFM,DFP
      IF(RDC.LT.R1)GOTO 86
      IF(RDC.GT.R2)GOTO 88
      DFP=DFP*(PI/180.)
      M=M+1
      XFACT=DSQRT(DFM)*DELRDC
      SIVI=SIVI-XFACT*DCOS(0.5*DFP)
      SIVR=SIVR+XFACT*DSIN(0.5*DFP)
      LHSPHA=ATAN2(SIVI,SIVR)
      XMAG=SQRT(SIVR*SIVR+SIVI*SIVI)
   86 RDC=RDC+DELRDC
      GO TO 80
   88 CONTINUE
      LHSPHA=LHSPHA*180./PI
C      WRITE(LOU,*)LHSPHA
      CALL NDEP(DEPTH,RHSMAG,LHSPHA)
```

```
      VOL1=(2*SMAIR*PI)*(XMAG*XMAG)/((3*WUSRR)*(RHSMAG*RHSMAG))
      WRITE(LOU,479)DEPTH,VOL1
  479 FORMAT('INV CAL:  ',F12.5,1X,E14.5)
      END
```

PCRTSCAN converts raw voltage readings to impedance change

Program PCRTSCAN converts raw experimental voltage readings taken by a pancake coil scanning across a plate into the impedance change in the coil due to a defect. The program can then use this array of impedance changes to locate defects in the plate and to calculate the depth and volume of the defects. The program locates the defects by constructing three windows, a "zero" window and two windows directly under the coil windings. These two windows are located from R1 to R2 on either side of the coil axis, and a running sum of the magnitude of the impedance change, referenced to the zero window, is kept for each of these windows. A dot product is performed between the impedance change in these two windows, and the defect center is located at the maximum value of this product. The "zero" window is shifted so that it will be in a clean region of the sample, but on either side of the defect. In Fig. 6 we show the magnitude of the impedance change plotted for a scan of six defects on the near side of the plate.

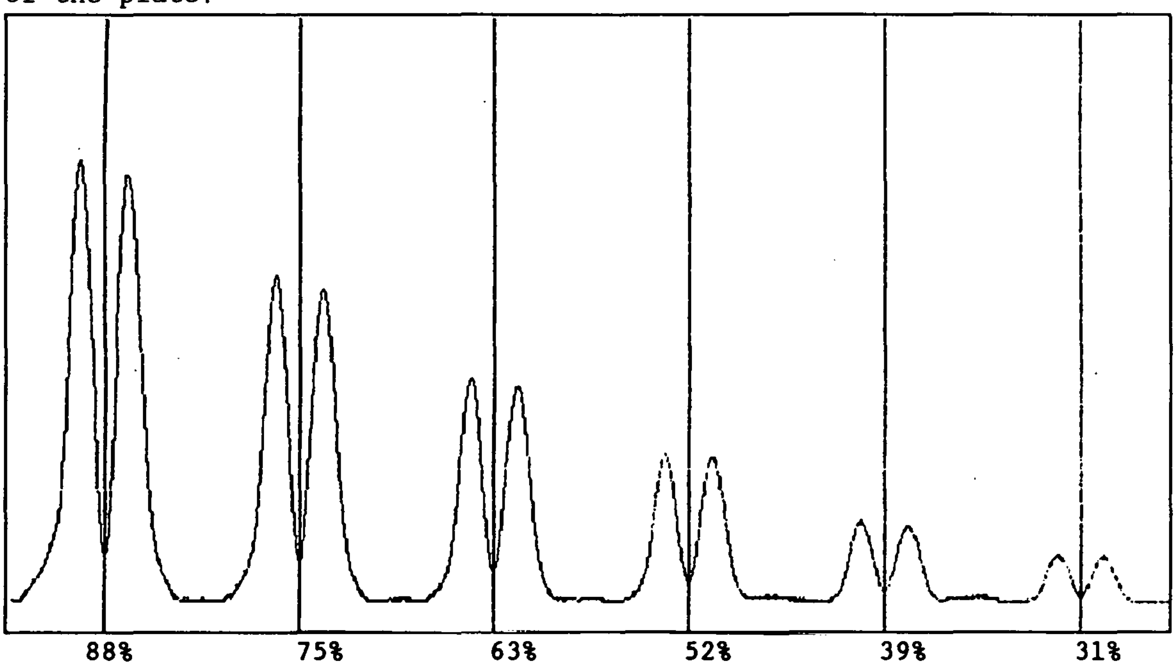

Figure 6 Magnitude of impedance change measured with a pancake coil for near side defects.

The defects have a diameter approximately equal to the depth of the defect, so that the defect volume falls off as the depth cubed. In Fig. 7 we show a similar plot for defects on the far side of the plate. Note that the noise has increased such that the last two defects have not been located using the present noise cut-off level. However, these are very low volume defects compared to those normally detected by eddy-current tests. Relative to the wall thickness, the ASME Section XI 40% standard defect has a volume 94 times greater than the 39% defects in the test plate. The defects were chosen to be relatively small so that better agreement would be obtained with the theory.

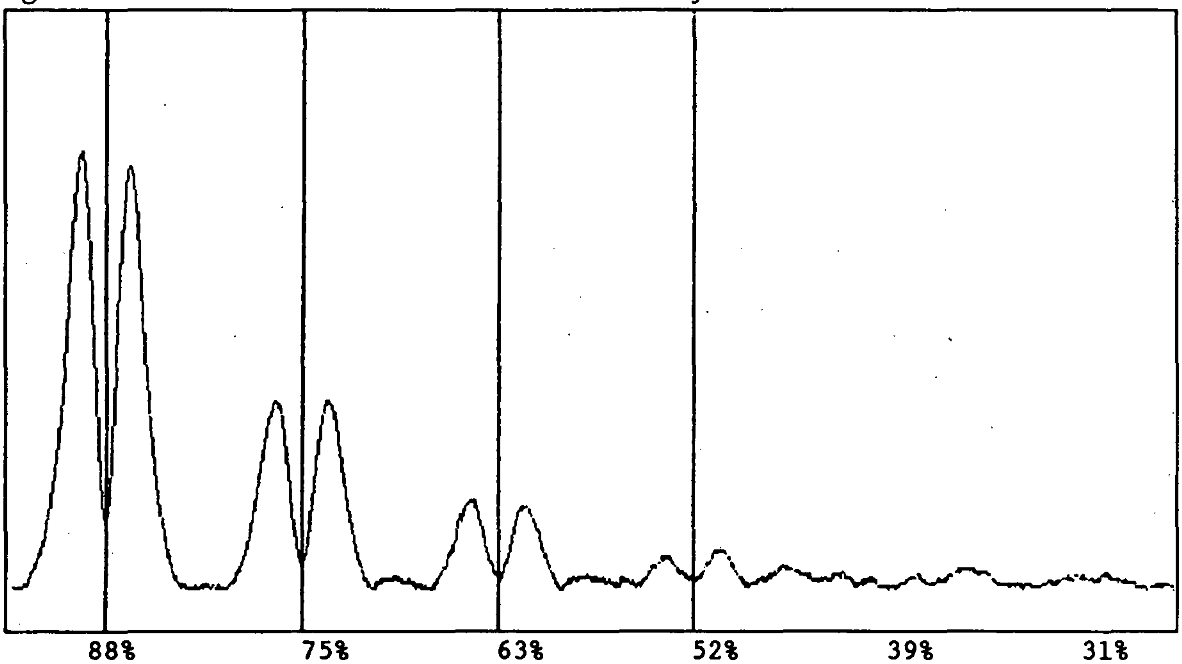

Figure 7 Magnitude of impedance change measured with a pancake coil for far side defects.

The original version of PCRTSCAN read data directly from voltmeters connected to the MIZ17, but later versions, such as the present listing, read the data taken from the MIZ17 and stored in a data file.

<u>Summary</u>

1. Dimension arrays and declare variable types.
2. Initialize variables.
3. Get the voltage readings with additional lift-off and convert these readings to the impedance change caused by the additional lift-off.
4. Take enough readings to fill up both of the active windows and the region between the windows.
5. Average the first NZ readings taken on the plate and use this average value as the voltage in the absence of defects.
6. Calculate the integrals in the active windows and solve for the depth and volume of a possible defect centered between the active windows.
7. Check to see if a defect is located between the active windows. If there is a defect, find and record its depth and volume.
8. Advance the zero window one point. Check to see if the entire zero window is out of the range of defects. If it is, average the values in the zero window to find the new value of the voltage in the absence of defects.
9. Advance the active windows one point.
10. Go to 6 until the entire plate has been scanned.

11. Graph the results.

Variables

AIRIND*	The inductance of the coil in air.
CLOM*	The magnitude of the calculated value of the change in the impedance of the coil due to additional lift-off.
CLOP*	The phase in radians of the calculated value of the change in the impedance of the coil due to additional lift-off.
DEFECT	An array which contains the position of each defect located by the program. It contains the distance in inches from each defect to the point where the scan began.
DELR	The normalized distance between adjacent data points.
DELTAX*	The distance between adjacent data points in inches.
DEPA	An array containing the inverted depths at each point along the plate.
DP	The dot product of the integrals from the two active windows.
FF	A character variable containing the form feed character.
FREQ*	The operating frequency in hertz.
HA1	The number of the first data point in the first active window.
HA2	The number of the first data point in the second active window.
HZ	The number of the first data point in the zero window.
ICP	Flag which is set when a defect is located and is reset when the dot product of the integrals in the two active windows stops decreasing. This flag must be 0 for the program to signal that it has found a defect.
L*	The length of the coil. The value is input in inches and normalized by the program.
L1*	The lift-off of the coil. The value is input in inches and normalized by the program.
L2	The distance from the top of the coil to the plate.
LA1	The number of the last data point in the first active window.
LA2	The number of the last data point in the second active window.
LHSMAG	The magnitude of the average of the integrals from the two active regions.
LHSPHA	The phase in degrees of the average of the

	integrals from the two active regions.
LOD*	The number of the I/O unit connected to the lookup file built by program PCBLDF.
LOE*	The number of the I/O unit connected to the file containing the raw experimental data.
LOU	The number of the I/O unit connected to the printer.
LZ	The number of the last data point in the zero window.
NA	The number of data points in each active window.
ND	The number of defects located by the program.
NE	The number of data points in the region between the active windows.
NZ*	The number of data points in the zero window.
R1*	The inner radius of the coil. The value is input in inches and normalized by the program.
R2*	The outer radius of the coil. The value is input in inches and normalized by the program.
R3	The mean radius of the coil in inches.
RAWI	An array containing the imaginary parts of the raw readings taken at each point across the plate.
RAWR	An array containing the real parts of the raw readings taken at each point across the plate.
RHO1*	The electrical resistivity of the plate in $\mu\Omega$-cm.
SACTI	The imaginary part of the average of the integrals from the two active regions.
SACTR	The real part of the average of the integrals from the two active regions.
SCFAC	The ratio of the magnitude of the calculated change in the normalized impedance to the magnitude of the change in the raw reading due to additional lift-off.
SCPHA	The difference between the phase of the calculated change in the normalized impedance and the phase of the change in the raw reading due to additional lift-off.
TACTI1	The imaginary part of the integral over the region spanned by the first active window.
TACTI2	The imaginary part of the integral over the region spanned by the second active window.
TACTR1	The real part of the integral over the region spanned by the first active window.
TACTR2	The real part of the integral over the region spanned by the second active window.
TRN*	The number of turns in the coil.
U1*	The relative magnetic permeability of the plate.
VLOI*	The imaginary part of the experimental reading taken with additional lift-off.
VLOM	The magnitude of the experimental reading taken with additional lift-off.

VLOP	The phase of the experimental reading taken with additional lift-off.
VLOR	The real part of the experimental reading taken with additional lift-off.
VOLA	An array containing the inverted volume at each point along the plate.
VTOL	The maximum amount of drift in the readings in the zero window that the program will tolerate for it to re-zero the readings.
WUSRR	The product of the angular frequency, the magnetic permeability, the electrical conductivity, and the square of the mean coil radius.
ZEROI	The value of the imaginary part of the voltage on a section of the plate with no defects.
ZEROR	The value of the real part of the voltage on a section of the plate with no defects.
ZEROY	The difference between the number 20 and the number of data points since the program last found a zero point on the plate. If it has been more than 20 points since the last time the program found a zero, ZEROY=0.

Notes

1. For the program to signal that it has found a defect, four conditions must be satisfied:

   (a) The inversion of the preliminary data must result in a defect which is inside the plate.

   (b) The scalar product of the integrals from the two active windows must decrease twice consecutively after having increased.

   (c) The scalar product of the integrals from the two active windows must be greater than 4.0E-04.

   (d) No zero can have been detected within the last 20 readings.

Some of these criteria may be unnecessary.

2. If the program is too sensitive to zeros, that is, if it finds zeros in places it should not, it can be made less sensitive by increasing the value of NZ or by decreasing the value of VTOL. Conversely, if the program fails to find a zero in a clean region of the plate, it may be made more sensitive to such regions by decreasing NZ or by increasing VTOL.

3. When the program locates a defect, it searches until it finds the next zero on the plate, and it averages the zeros on both sides of the defect. While this makes the inversion more accurate, it also makes it possible that the program will overlook a defect. If there are two defects on the plate which are not separated by a region that the program recognizes as clean, the program will not detect the second defect.

4. The program averages the first NZ readings and uses this as the value on the clean part of the plate until it finds a clean section of the plate. If the very first region presented to the program has a defect in it, the results will be unpredictable and very likely undesirable because the program uses the very first reading in its calculation of SCFAC and SCPHA which are used to convert all subsequent readings to the normalized impedance change of the coil.

Listing

```
      PROGRAM PCRTSCAN
C     VERSION September 12, 1988
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER HZ,HA1,HA2
      REAL*8 L,L1,L2
      REAL*8 LHSPHA,LHSMAG
      REAL*8 RAWR(1500),RAWI(1500)
      REAL*8 TMZR(20),TMZI(20)
      REAL*8 ZMAGA(1500),ZPHAA(1500)
      REAL*8 DEPA(1500),VOLA(1500)
      REAL*8 DEFECT(100)
      CHARACTER*1 FF
      DATA LOU/8/,LOD/38/,LOE/40/
      DATA PI/3.141592653/
      DATA R1/0.1000/,R2/0.4100/,L/0.1000/,TRN/800./,L1/0.010/
      DATA FREQ/500./,RHO1/4.054/,U1/1.0/,T1/0.250/
      DATA DELTAX/0.01/,AIRIND/6.252919E-03/,VTOL/0.02/
      DATA CLOM/0.020030/,CLOP/2.44/
      FF=CHAR(12)
      K=0
      OPEN(LOE,FILE='TEMP.DAT',STATUS='OLD')
      R3=0.5*(R1+R2)
      L2=L+L1
      R1=R1/R3
      R2=R2/R3
      L=L/R3
      L1=L1/R3
      L2=L2/R3
      T1=T1/R3
      DELR=DELTAX/R3
      WUSRR=0.5093979*U1*R3*R3*FREQ/RHO1
      SMAIR=AIRIND*(L*(R2-R1))**2/(0.0254*4.E-07*TRN*TRN*R3*PI*PI)
C   NZ = Number of points in the zero window
C   NA = Number of points in each active window
C   NE = Number of points in the eye
      NZ=15
      NA=(R2-R1)/DELR
      NE=(2*R1)/DELR
      LZ=1
      HZ=NZ
      LA1=NZ+1
      HA1=NZ+NA
      LA2=NZ+NA+NE+1
      HA2=NZ+NA+NE+NA
      GO TO 201
C   Initialize scanner
      CALL INITSC
C    Take readings with liftoff
      XX1=1.0
```

```
      YY1=2.0
      DELTAX=0.0
      CALL GDAT(XX1,YY1,DELTAX,VLOR,VLOI)
      PAUSE 'PUT SHIM UNDER COIL; PRESS RETURN'
      XX1=1.0
      YY1=2.0
      DELTAX=0.01
      DO 200 J=1,10
      CALL GDAT(XX1,YY1,DELTAX,VLOR,VLOI)
      TVLOR=TVLOR+VLOR
      TVLOI=TVLOI+VLOI
  200 CONTINUE
  201 CONTINUE
C      VLOR=0.1*TVLOR
C      VLOI=0.1*TVLOI
C      PAUSE 'REMOVE SHIM; PRESS RETURN'
      VLOR=-1.74
      VLOI=-0.31
      XX1=0.0
      YY1=1.0
      DELTAX=0.01
C  Fill the active windows and the region between the active
C   windows with data so that the scan can get started.
      DO 300 J=1,HA2
C      CALL GDAT(XX1,YY1,DELTAX,VR,VI)
      READ(LOE,*)XX1,VR,VI
      RAWR(J)=VR
      RAWI(J)=VI
  300 CONTINUE
      TZEROR=0.
      TZEROI=0.
C   Find the zero to be used until another comes along.
      DO 400 J=LZ,HZ
      TZEROR=TZEROR+RAWR(J)
      TZEROI=TZEROI+RAWI(J)
  400 CONTINUE
      ZEROR=TZEROR/NZ
      ZEROI=TZEROI/NZ
C  Calculate the factors for converting the raw readings
C   to the impedance change in the coil.
      VLOR=VLOR-ZEROR
      VLOI=VLOI-ZEROI
      VLOM=DSQRT(VLOR*VLOR+VLOI*VLOI)
      VLOP=DATAN2(VLOI,VLOR)
      SCFAC=CLOM/VLOM
      SCPHA=CLOP-VLOP
  450 TACTR1=0.
      TACTI1=0.
      TACTR2=0.
      TACTI2=0.
C  Calculate the integral for the first active window.
```

```
      DO 500 J=LA1,HA1
      VR=RAWR(J)-ZEROR
      VI=RAWI(J)-ZEROI
      ZMAG=DSQRT(VR*VR+VI*VI)*SCFAC
      ZPHA=(DATAN2(VI,VR)+SCPHA)
      ZMAGA(J)=ZMAG
      ZPHAA(J)=ZPHA
      XFACT=DSQRT(ZMAG)*DELR
      RZR=XFACT*DSIN(0.5*ZPHA)
      RZI=-XFACT*DCOS(0.5*ZPHA)
      TACTR1=TACTR1+RZR
      TACTI1=TACTI1+RZI
  500 CONTINUE
C   Calculate the integral for the second active window.
      DO 600 J=LA2,HA2
      VR=RAWR(J)-ZEROR
      VI=RAWI(J)-ZEROI
      ZMAG=DSQRT(VR*VR+VI*VI)*SCFAC
      ZPHA=(DATAN2(VI,VR)+SCPHA)
      ZMAGA(J)=ZMAG
      ZPHAA(J)=ZPHA
      XFACT=DSQRT(ZMAG)*DELR
      RZR=XFACT*DSIN(0.5*ZPHA)
      RZI=-XFACT*DCOS(0.5*ZPHA)
      TACTR2=TACTR2+RZR
      TACTI2=TACTI2+RZI
  600 CONTINUE
C   Average the integrals from the two active windows.
      SACTR=0.5*(TACTR1+TACTR2)
      SACTI=0.5*(TACTI1+TACTI2)
      LHSPHA=DATAN2(SACTI,SACTR)*180./PI
      LHSMAG=DSQRT(SACTR*SACTR+SACTI*SACTI)
      K=K+1
C   Invert the integrals.
      CALL NDEP(DEPTH,RHSMAG,LHSPHA,T1,L1,L2,R1,R2,WUSRR,U1)
      VOL1=(2*SMAIR*PI)*(LHSMAG*LHSMAG)/((3*WUSRR)*(RHSMAG*RHSMAG))
      DEPA(K)=DEPTH
      WRITE(0,*)DEPTH,VOL1
      IF(DEPTH.EQ.0) THEN
      VOLA(K)=0
      ELSE
      VOLA(K)=VOL1
      END IF
      DP=TACTR1*TACTR2+TACTI1*TACTI2
      WRITE(0,*)'   ',DP,PRDP
      IF(DEPTH.NE.0.) THEN
      IF((DP.LT.PRDP).AND.(DP.GT.4.E-04).AND.(PRDP.LT.PRPRDP)) THEN
      IF((ZEROY.EQ.0.).AND.(ICP.EQ.0)) THEN
C   A defect has been found.
      XX=XX1-R2*R3-2.*DELTAX
       WRITE(LOU,29)XX,DEPTH,VOL1
```

```
      WRITE(0,*)'D'
      ND=ND+1
      DEFECT(ND)=XX
C  Find the next zero and recalculate.
      ITP=HA2
      DO 610 KN=1,NZ
      ITP=ITP+1
      READ(LOE,*,END=1000)XX1,VR,VI
      IF(VR.EQ.999)GO TO 1000
      RAWR(ITP)=VR
      RAWI(ITP)=VI
      WRITE(0,*)XX1
      TMZR(KN)=VR
      TMZI(KN)=VI
  610 CONTINUE
  615 TZEROR2=0.
      TZEROI2=0.
C  Check to see if the entire zero window is out of the range
C   of defects.
      DO 620 KN=1,NZ
      IF(ABS(ABS(TMZR(KN))-ABS(TMZR(8))).GT.VTOL)GO TO 630
      IF(ABS(ABS(TMZR(KN))-ABS(TMZR(8))).GT.VTOL)GO TO 630
      TZEROR2=TZEROR2+TMZR(KN)
      TZEROI2=TZEROI2+TMZI(KN)
  620 CONTINUE
      ZEROR2=TZEROR2/NZ
      ZEROI2=TZEROI2/NZ
      GO TO 640
C  The zero window is not out of the range of defects.
C  Advance the window one step and retest.
  630 ITP=ITP+1
      READ(LOE,*,END=1000)XX1,VR,VI
      RAWR(ITP)=VR
      RAWI(ITP)=VI
      DO 635 KN=1,NZ-1
      TMZR(KN)=TMZR(KN+1)
      TMZI(KN)=TMZI(KN+1)
  635 CONTINUE
      TMZR(NZ)=VR
      TMZI(NZ)=VI
      GO TO 615
C  The zero window is out of the range of defects.
C  Average the leading and trailing zero.
  640 ZEROR=(ZEROR+ZEROR2)/2.
      ZEROI=(ZEROI+ZEROI2)/2.
      TACTR1=0.
      TACTI1=0.
      TACTR2=0.
      TACTI2=0.
C  Reconvert the raw readings from the first active window to
C   the impedance changes in the coil using the new value for zero.
```

```
      DO 650 KN=LA1,HA1
      VR=RAWR(KN)-ZEROR
      VI=RAWI(KN)-ZEROI
      ZMAG=DSQRT(VR*VR+VI*VI)*SCFAC
      ZPHA=(DATAN2(VI,VR)+SCPHA)
      ZMAGA(KN)=ZMAG
      ZPHAA(KN)=ZPHA
C  Recalculate the integral for the first active window using
C   the new data.
      XFACT=DSQRT(ZMAG)*DELR
      RZR=XFACT*DSIN(0.5*ZPHA)
      RZI=-XFACT*DCOS(0.5*ZPHA)
      TACTR1=TACTR1+RZR
      TACTI1=TACTI1+RZI
  650 CONTINUE
C  Reconvert the raw readings from the second active window to
C   the impedance changes in the coil using the new value for zero.
      DO 660 KN=LA2,HA2
      VR=RAWR(KN)-ZEROR
      VI=RAWI(KN)-ZEROI
      ZMAG=DSQRT(VR*VR+VI*VI)*SCFAC
      ZPHA=(DATAN2(VI,VR)+SCPHA)
      ZMAGA(KN)=ZMAG
      ZPHAA(KN)=ZPHA
C  Recalculate the integral for the second active window using
C   the new data.
      XFACT=DSQRT(ZMAG)*DELR
      RZR=XFACT*DSIN(0.5*ZPHA)
      RZI=-XFACT*DCOS(0.5*ZPHA)
      TACTR2=TACTR2+RZR
      TACTI2=TACTI2+RZI
  660 CONTINUE
C  Average the new integrals.
      SACTR=0.5*(TACTR1+TACTR2)
      SACTI=0.5*(TACTI1+TACTI2)
      LHSPHA=DATAN2(SACTI,SACTR)*180./PI
      LHSMAG=DSQRT(SACTR*SACTR+SACTI*SACTI)
C  Invert once again.
      CALL NDEP(DEPTH,RHSMAG,LHSPHA,T1,L1,L2,R1,R2,WUSRR,U1)
      VOL1=(2*SMAIR*PI)*(LHSMAG*LHSMAG)/((3*WUSRR)*(RHSMAG*RHSMAG))
      DEPA(K)=DEPTH
      IF(DEPTH.EQ.0)THEN
      VOLA(K)=0
      ELSE
      VOLA(K)=VOL1
      END IF
C  Print the results of the inversion.
      WRITE(LOU,29)XX,DEPTH,VOL1
      HA2=ITP
      LA2=ITP-NA+1
      HA1=ITP-NA-NE
```

```
      LA1=HA1-NA+1
      END IF
      ICP=1
      ELSE
C  No defect was found.
      ICP=0
      END IF
      END IF
      PRPRDP=PRDP
      PRDP=DP
C       WRITE(LOD,29)XX1,DEPTH,VOL1
C       WRITE(0,29)XX1,DEPTH,VOL1
   29 FORMAT(F8.4,2X,D11.4,2X,D11.4)
C  Advance all of the windows one step and continue looking
C   for defects.
      LZ=LZ+1
      HZ=HZ+1
      LA1=LA1+1
      HA1=HA1+1
      LA2=LA2+1
      HA2=HA2+1
C       CALL GDAT(XX1,YY1,DELTAX,VR,VI)
C  Get the next raw data reading.
      READ(LOE,*,END=1000)XX1,VR,VI
      IF(VR.EQ.999.)GO TO 1000
      WRITE(0,*)XX1
      RAWR(HA2)=VR
      RAWI(HA2)=VI
C  Shift the readings in the zero window down one step in the array.
      DO 700 KN=1,NZ-1
      TMZR(KN)=TMZR(KN+1)
      TMZI(KN)=TMZI(KN+1)
  700 CONTINUE
C  Assign the new reading to the highest element in the zero array.
      TMZR(NZ)=RAWR(HA2)
      TMZI(NZ)=RAWI(HA2)
      ZEROY=ZEROY-1.
      IF(ZEROY.LT.0.)ZEROY=0.
C  Check to see if the entire zero window is out of the range of defects.
      DO 710 KN=1,NZ
      IF(ABS(ABS(TMZR(KN))-ABS(TMZR(8))).GT.VTOL)GO TO 780
      IF(ABS(ABS(TMZI(KN))-ABS(TMZI(8))).GT.VTOL)GO TO 780
  710 CONTINUE
C  The zero window is out of the range of defects.
      ZEROY=20.
      WRITE(0,*)'Z'
      TZEROR=0.
      TZEROI=0.
C  Average the readings in the zero window to find the new zero.
      DO 720 KN=1,NZ
      TZEROR=TZEROR+TMZR(KN)
```

```
      TZEROI=TZEROI+TMZI(KN)
  720 CONTINUE
      ZEROR=TZEROR/NZ
      ZEROI=TZEROI/NZ
  780 GOTO 450
 1000 CONTINUE
C   Graph the results of the scan.
      DO 1020 K=1,1200
      IF(ABS(ZMAGA(K)).GT.ZMAGMX)ZMAGMX=ABS(ZMAGA(K))
      IF(ABS(ZPHAA(K)).GT.ZPHAMX)ZPHAMX=ABS(ZPHAA(K))
 1020 CONTINUE
C      WRITE(LOU,*)FF
C      WRITE(LOU,*)'MAX MAG ',ZMAGMX
C      WRITE(LOU,*)'MAX PHA ',ZPHAMX
      CALL QSMODE(16)
      DO 1025 K=1,ND
      IY1=1
      IY2=350
      IX1=DEFECT(K)*600./12.
      IX2=IX1
      CALL QLINE(IX1,IY1,IX2,IY2,7)
 1025 CONTINUE
      IX1=0
      IY1=ZMAGA(1)*150./ZMAGMX+150
      DO 1030 K=2,1200
      IX2=K/2
      IY2=ZMAGA(K)*150./ZMAGMX+150
      CALL QLINE(IX1,IY1,IX2,IY2,7)
      IX1=IX2
      IY1=IY2
 1030 CONTINUE
C      CALL PRTSC
C      WRITE(LOU,*)FF
      PAUSE
      CALL QSMODE(16)
      IX1=0
      IY1=ZPHAA(1)*150./ZPHAMX+150
      DO 1040 K=2,1200
      IX2=K/2
      IY2=ZPHAA(K)*150/ZPHAMX+150
      CALL QLINE(IX1,IY1,IX2,IY2,7)
      IX1=IX2
      IY1=IY2
 1040 CONTINUE
C      CALL PRTSC
 1090 END
```

REFLECTION COIL PROGRAMS

The programs in this section perform various functions relating to the effect of a defect in a single conducting plate on the induced voltage in the pickup coils of a reflection probe. A reflection probe above a conducting plate is shown in Fig. 8. The probe consists of a large driver coil with two pick-up coils mounted at either end. The pick-up coils are connected in opposition so that their signal cancels out with the probe in air. When the probe is placed on a conductor, the field can be considered to be equal to the field reflected back from the conductor.

ORNL-DWG 90-8089

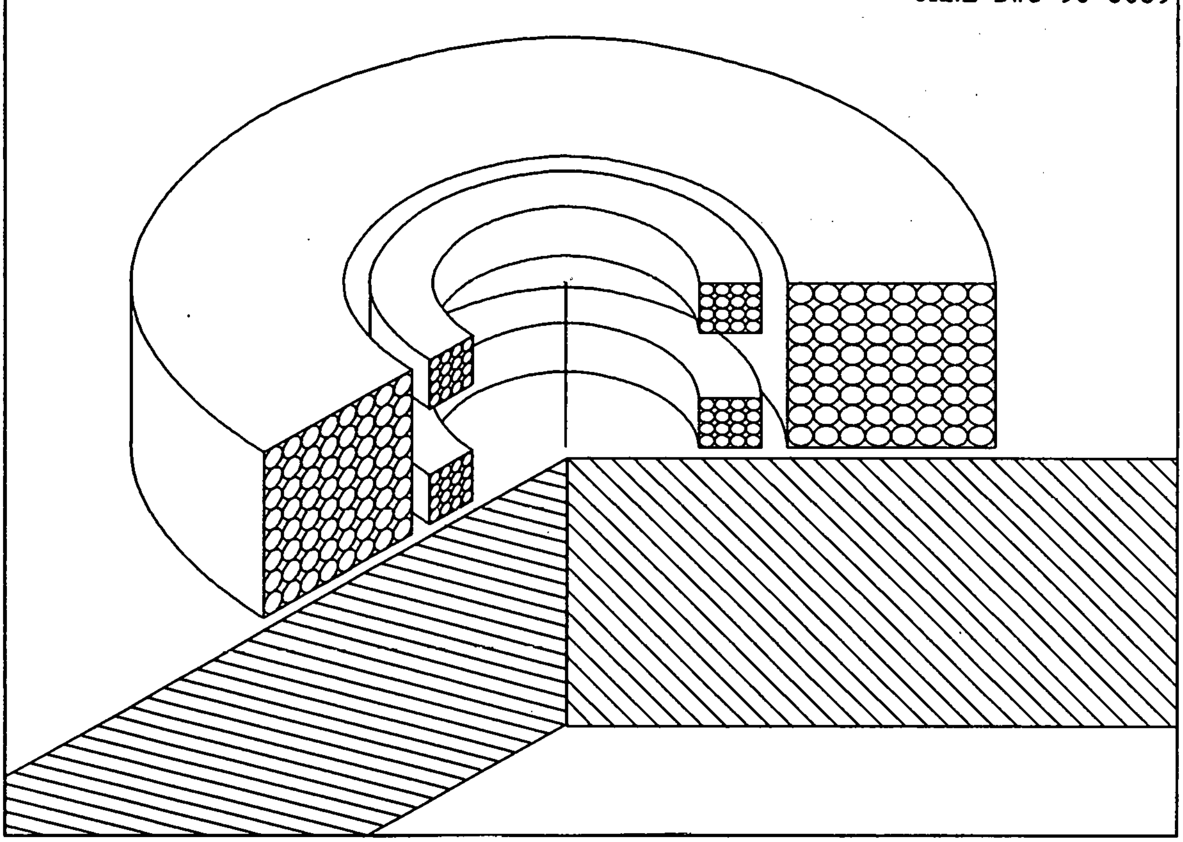

Fig. 8. Cutaway of a reflection probe above a conducting plate.

The electrical connections for the reflection coil circuit are shown in Figure 9. The voltage output from this circuit V_{out} is given by:

$$V_{out} = -\,j\omega M V_0 R_9 A \div \Big\{(\omega C_6 R_0 - j)(\omega C_7 R_9 - j)(\omega M)^2 + \Big[(\omega C_6 R_0 - j)(Z_D + R_6) - jR_0\Big]\Big[(\omega C_7 R_9 - j)(Z_{PU} + R_7) - jR_9\Big]\Big\} \tag{20}$$

where A is the amplifier gain and the rest of the terms are defined in Fig. 9.

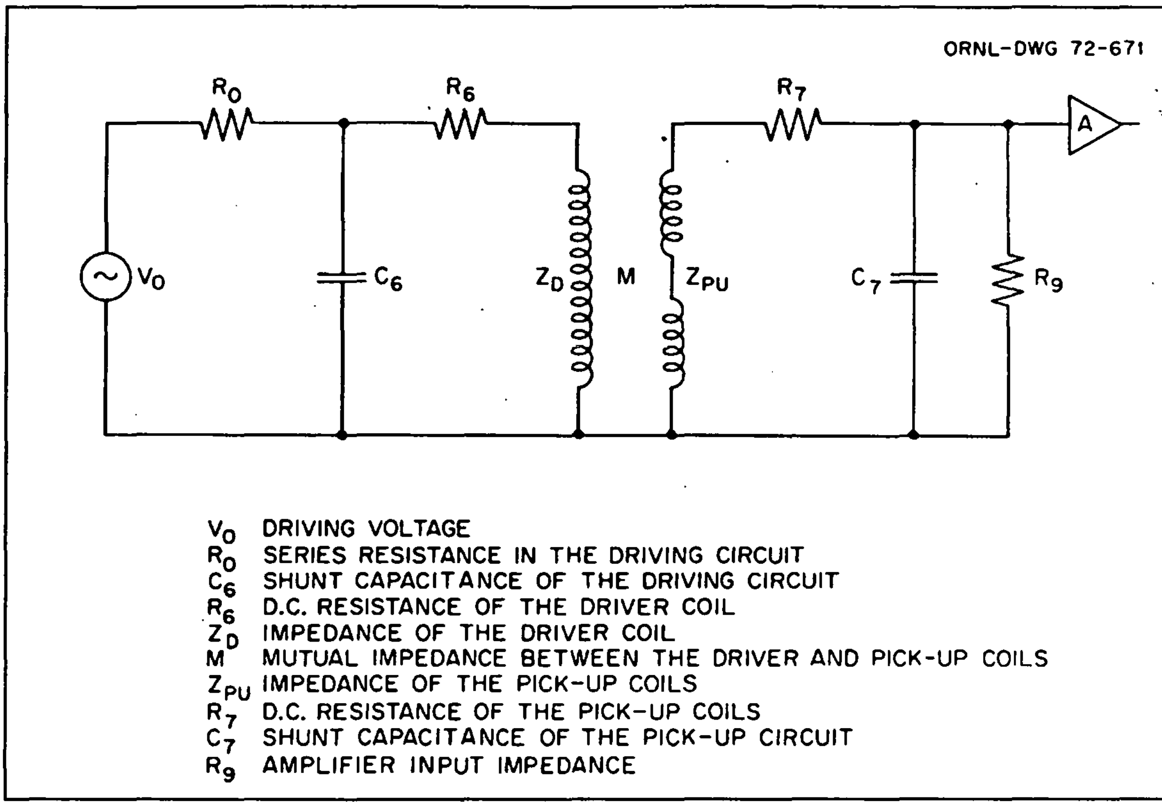

Fig. 9. Simplified circuit diagram for an eddy-current reflection type probe.

Fig. 10 shows a cross section of the reflection coil above the metal plate with the dimensions labeled as they are used in the equations and the programs. The programs are based on the signal from a small spherical defect, as shown in the figure.

As we can see from Eq. (20), we must calculate the impedance of both the driver and pick-up coils, as well as the mutual coupling between the two. The equations for the driver coil impedance, including the change due to the defect, is:

$$Z_D = \frac{\omega\pi\mu_0 N_3^2 R_5}{(r_2-r_1)^2 \ell_3^2} \left\{ j\int_0^\infty \frac{1}{\alpha^6} J^2(r_2,r_1)\left\{2(\alpha\ell_3+e^{-\alpha\ell_3}-1)+(1-e^{-\alpha\ell_3})^2 e^{-2\alpha\ell_6} F(\alpha,\alpha_1,c)\right\}d\alpha \right.$$

$$\left. -\frac{Vo1_n\alpha_{22}3\omega\mu\sigma\bar{r}^2}{8\pi}\left[\int_0^\infty \frac{e^{-\alpha\ell_6}}{\alpha^3} J(r_2,r_1)(1-e^{-\alpha\ell_3})J_1(\alpha r)\ F(\alpha,\alpha_1,z)\,d\alpha\right]^2 \right\} \qquad (21)$$

where $$F(\alpha,\alpha_1,z) = 2\alpha\left[\frac{(\alpha+\beta_1)\exp(\alpha_1 z)-(\alpha-\beta_1)\exp(-\alpha_1(2c+z))}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right] \qquad (22)$$

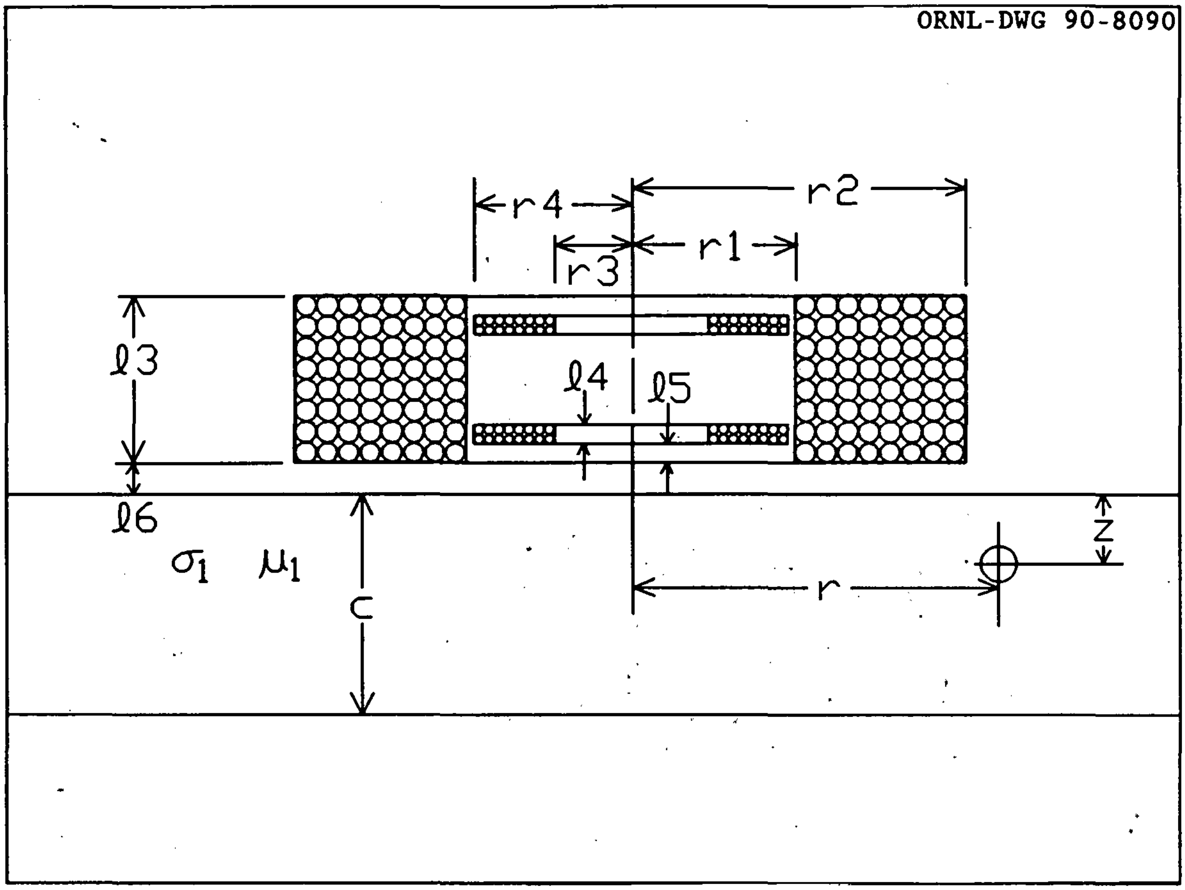

Fig. 10. Cross section of a reflection coil above a conducting plate with a spherical defect.

and $$F(\alpha,\alpha_1,c) = \left[\frac{(\alpha-\beta_1)(\alpha+\beta_1)-(\alpha-\beta_1)(\alpha+\beta_1)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)} \right] \tag{23}$$

and $$\beta_1 = (\alpha^2+j\omega\mu\sigma_1\bar{r}^2)^{1/2}/\mu \tag{24}$$

The dimensions for the coil are shown in Fig. 10. For the pick-up coil impedance we have:

$$Z_p = \frac{\omega\pi\mu_0 N_4^2 R_5}{(r_4-r_3)^2\ell_4^2} \times \left\{ j\int_0^\infty \frac{J^2(r_4,r_3)}{\alpha^6} \right. \tag{25}$$

$$\left\{-4(\alpha\ell_4+e^{-\alpha\ell_4}-1)+(1-e^{-\alpha(\ell_3-\ell_4-2\ell_5)})^2(1-e^{-\alpha\ell_4})^2 e^{-2\alpha(\ell_6+\ell_5)} F(\alpha,\alpha_1,c)e^{-\alpha(\ell_3-2\ell_4-2\ell_5)}\right\}d\alpha$$

$$\left. - \frac{Vol_n\alpha_{22}3\omega\mu\sigma\bar{r}^2}{8\pi}\left[\int_0^\infty \frac{e^{-\alpha(\ell_6+\ell_5)}}{\alpha^3} J(r_4,r_3)(1-e^{-\alpha(\ell_3-\ell_4-2\ell_5)})(1-e^{-\alpha\ell_4})J_1(\alpha r)F(\alpha,\alpha_1,z)d\alpha\right]^2 \right\}$$

and for the mutual impedance between the driver and pickup coils we have:

$$j\omega M = \frac{\omega\pi\mu_0 N_3 N_4 R_5}{(r_2-r_1)\ell_3(r_4-r_3)\ell_4} \times \tag{26}$$

$$\left\{ j\int_0^\infty \frac{J(r_2,r_1)J(r_4,r_3)}{\alpha^6}(1-e^{-\alpha\ell_3})(1-e^{-\alpha(\ell_3-\ell_4-2\ell_5)})(1-e^{-\alpha\ell_4})e^{-\alpha(2\ell_6+\ell_5)}F(\alpha,\alpha_1,c)d\alpha\right. .$$

$$\frac{-Vol_n\alpha_{22}3\omega\mu\sigma\bar{r}^2}{8\pi}\left[\int_0^\infty \frac{e^{-\alpha\ell_6}}{\alpha^3} J(r_2,r_1)(1-e^{-\alpha\ell_3})J_1(\alpha r)\ F(\alpha,\alpha_1,z)d\alpha\right] \times$$

$$\left.\left[\int_0^\infty \frac{e^{-\alpha(\ell_6+\ell_5)}}{\alpha^3} J(r_4,r_3)(1-e^{-\alpha(\ell_3-\ell_4-2\ell_5)})(1-e^{-\alpha\ell_4})J_1(\alpha r)F(\alpha,\alpha_1,z)d\alpha\right]\right\}$$

These are the basic programs used for the calculation of the change in magnitude and phase of the eddy-current signal due to a defect for a reflection probe. In Eqs. (21), (25), and (26) the terms multiplied by j are the normal values without the defect, while the terms multiplied by Vol_n are the changes due to the defect. If the values of driving and input impedance, R_0 and R_9, are large and the circuit is operated well below resonance, the major change due to a defect will be in the term for M. Most of the tests run are usually designed for these conditions.

The program RFDSF calculates the defect sensitivity factor of a reflection coil at points throughout a conducting plate, and program RFDSFPLT produces a contour plot of these calculations. Program RFAVZSCN does the theoretical calculations for the change in the induced voltage in a pickup coil due to a defect in a conducting plate, and program RFFIX converts raw experimental data to the change in induced voltage in the coil. Program RFGRAPH can plot the data from these two programs side-by-side so the theoretical predictions for a defect can be compared to the actual data. Finally, programs RFBLDF and RFINV can take data stored by either RFAVZSCN or RFFIX and solve for the depth and volume of the defect that produced that data. These programs and their discussion closely parallel the programs for the pancake coils in the previous section.

RFBLDF builds a lookup file of magnitude and phase of DSF

Program RFBLDF builds a lookup file containing the magnitude and phase of the integral of the defect sensitivity factor of a reflection coil with respect to *r*, the radial distance from the coil axis to the point where the calculations are performed, from the inner radius of the pickup coil to the outer radius of the driver coil at different depths in a plate. This lookup file can be used by program RFINV to calculate the depth and volume of defects. (See note #2.)

Summary

1. Dimension arrays and declare variable types.
2. Initialize variables.
3. Select a value for the depth in the plate at which the calculations are to be performed.
4. Select a value for the radial distance between the coil axis and the point at which the calculations are to be performed.
5. Find the defect sensitivity factor at this point.
6. Loop to 4 until done.
7. Calculate the integral of the defect sensitivity factor over the range of *r* and write the result to a data file.
8. Loop to 3 until done.

Variables

NOTE: Most of the variables which occur inside the integration loops are uninteresting because they do not correspond to anything physical and because they should never require user intervention. Therefore, these variables will not be discussed here. For a detailed description of these variables, see the documentation of program RFAVZSCN. For a discussion of the major differences between the integration section of program RFBLDF and the integration section of program RFAVZSCN, see note #1.

Starred variables must be set by the user.

C6*	The total shunt capacitance in farads of the driving circuit.
C7*	The total shunt capacitance in farads of the pickup circuit.
DELTAR	The normalized radial distance between successive data points.
FREQ*	The operating frequency in hertz.
GAIN*	Gain of pickup amplifier.
L3	The normalized length of the driver coil.
L4	The normalized length of the pickup coil.
L5	The normalized distance of recess of the pickup coil.
L6	The normalized lift-off of the driver coil.
LOD*	The number of the I/O unit connected to the output

	data file.
LOU	The number of the I/O unit connected to the printer.
MZT*	The number of depths throughout the plate at which the program does calculations. Increasing this number usually improves the accuracy of the inversion somewhat because it brings the interpolated points closer together, so the variation of the magnitude and phase of the integral between the points is more nearly linear. (See note #4.)
NPROBE*	Character variable which contains the name of the reflection probe to be used in the calculations.
NRT	The total number of points in the radial direction at which calculations are performed. (See note #4.)
NS	The side where the defect is located. If NS = 1, the defect is on the near side; if NS = 2, the defect is on the far side. The value of NS is assigned according to the location of the center of the defect. If the center of the defect is nearer the near side of the plate, NS is set equal to 1. If the center of the defect is nearer the far side of the plate, NS is set equal to 2.
NZT	The number of parts each defect is divided into along its axis to perform the calculations. (See note #4.)
RO*	Output series resistance of driving amplifier in ohms.
R1	The normalized inner radius of the driver coil.
R2	The normalized outer radius of the driver coil.
R3	The normalized inner radius of the pickup coil.
R4	The normalized outer radius of the pickup coil.
R5	The mean radius of the driver coil in inches.
R6	DC resistance of the driver coil in ohms.
R7	DC resistance of both pickup coils in ohms.
R9*	Input shunt resistance of pickup amplifier in ohms.
RD	The radial distance from the axis of the coil to the point where the calculations are being done.
RHO1*	The resistivity of the plate in $\mu\Omega$-cm.
RHSI	The imaginary part of the integral of the defect sensitivity factor with respect to RD from R3 to R2.
RHSM	The magnitude of the integral of the defect sensitivity factor with respect to RD from R3 to R2.
RHSP	The phase in radians of the integral of the defect sensitivity factor with respect to RD from R3 to R2.
RHSR	The real part of the integral of the defect

	sensitivity factor with respect to RD from R3 to R2.
T1*	The thickness of the plate. It is input in inches and normalized by the program.
TD	The density of turns in the driver coil.
TNDR	The number of turns in the driver coil.
TNPU	The number of turns in each pickup coil.
TP	The density of turns in the pickup coil.
U1*	The relative magnetic permeability of the plate.
VIN*	Output voltage of driving amplifier in volts.
W	The angular operating frequency.
ZD	The normalized distance from the side of the plate where the defect is located to the bottom of the defect. A negative number.
ZD2	The normalized distance from the near side of the plate to the center of the defect. A negative number.
ZDTIO	The imaginary part of the self impedance of the driver coil with no defects present.
ZDTRO	The real part of the self impedance of the driver coil with no defects present.
ZMSTEP	The normalized axial distance between the depths at which the program does the calculations.
ZMTDI	The imaginary part of the change in the mutual impedance between the driver coil and the pickup coil due to a defect.
ZMTDR	The real part of the change in the mutual impedance between the driver coil and the pickup coil due to a defect.
ZMTIO	The imaginary part of the mutual impedance between the driver coil and the pickup coil with no defects present.
ZMTRO	The real part of the mutual impedance between the driver and pickup coils with no defects present.
ZPTIO	The imaginary part of the self impedance of the pickup coil with no defects present.
ZPTRO	The real part of the self impedance of the pickup coil with no defects present.

Notes

1. The integration in this program is very similar to the integration in program RFAVZSCN, but some very minor differences do exist. Most of these come from the fact that this program calculates only the defect sensitivity factor, which does not depend on the volume of the defect, and program RFAVZSCN calculates the voltage change due to the defect, which does depend on the volume of the defect. While it is necessary to calculate the defect sensitivity factor to calculate the voltage change due to the defect, program RFAVZSCN never calculates the defect

sensitivity factor as a separate quantity. Thus, some of the variables in RFAVZSCN which have the same name as variables in program RFBLDF have the defect volume as an extra factor.

2. The way that program RFINV uses the output of this program to calculate the depth and volume of a defect is very simple. Program RFINV takes either experimental data stored by program RFFIX or calculated data stored by program RFAVZSCN to find the integral of the voltage change due to the defect with respect to r from the inner radius of the pickup coil to the outer radius of the driver coil. The phase of this integral depends only upon the depth of the defect, and it is a single valued function of the depth of the defect. (See note #3.) The phase of the integral of the defect sensitivity factor will be the same as the phase of the voltage change due to the defect, because the defect sensitivity factor differs from the voltage change only by a constant real factor. Therefore, program RFINV can search through the file created by program RFBLDF until it finds the depth corresponding to the phase it obtained when it calculated the integral. This will be the depth of the defect. Then program RFINV can divide the integral it calculated by the integral calculated by program RFBLDF. Since the phases of the integrals are equal, the factors which contain the phase will cancel, and the program needs only to divide the magnitude of one integral by the other. The result of this division is the constant factor by which the two integrals differ, which is equal to the defect volume multiplied by α_{22}, the defect shape and orientation factor. We normally assume that α_{22} is equal to 1, so the program is left with the defect volume.

3. One of the necessary assumptions for the reflection coil inversion programs to work is that the phase of the integral calculated by this program depends only upon the depth of the defect and that it is therefore independent of the defect volume. This is not exactly true. The expression for the defect sensitivity factor contains terms in the denominator which do depend slightly on the defect volume, and since this volume dependence cannot be factored out, it has a small effect on the phase of the integral. However, the error introduced by this slight volume dependence is not significant, as shown by the accuracy of the inversion of data calculated by program RFAVZSCN, which considers the dependence of the phase on the volume. The error for these inversions is typically less than one percent.

4. The selection of the point in the plate about which the calculations are done is complex in the program. The outside position determining loop runs from 0 to MZT. This loop sets the value of the depth of the center of the defect by assigning values to variables ZD and ZD2. It also determines whether a defect is on the near side of the plate or on the far side of the plate. The next position determining loop, which occurs inside this one, runs from 1 to NRT. It sets the value for the radial distance from the coil axis to the center of the defect by assigning a value to variable RD. The final position determining loop lies inside both of these, and it runs from 1 to NZT. This loop is present for the purpose of dividing the large defect located at cylindrical coordinates

RD and ZD2 into a number of smaller defects so that more accurate calculations can be obtained for the entire defect. In the case of a near side defect, the large defect is divided into NZT smaller defects centered at even intervals along the axis of the large defect between the bottom of the defect and the near side of the plate. In the case of a far side defect, the large defect is divided into NZT smaller defects centered at even intervals along the axis of the large defect between the bottom of the defect and the far side of the plate.

Listing

```
      PROGRAM RFBLDF
C     VERSION November 7, 1988
      CHARACTER NPROBE*6,COIL*6
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 ZMT,ZDT,ZPT,Z0,Z6,Z7,Z9
      COMPLEX*16 ZMT0,ZPT0,ZDT0,RHSC,EBW
      REAL*8 L3,L4,L5,L6
      DIMENSION S1(6),S2(6),ERR(6)
      DIMENSION SDDR(120),SDDI(120),SDPR(120),SDPI(120)
      DATA LOU/8/,PI/3.141592653/,LOD/39/
      DATA S1/0.005,0.02,0.05,0.1,0.5,2./
      DATA S2/1.0,2.0,5.0,10.0,50.0,200.0/
      DATA ERR/0.1,0.01,0.001,1.E-4,1.E-5,1.E-6/
      DATA RHO1/4.054/,U1/1.0/,MZT/10/,NRT/100/,NZT/1/
      DATA T1/0.250/,NPROBE/'250A  '/
      DATA R0/3050./,R9/1.0D6/,C6/15.E-11/,C7/15.E-11/
      DATA FREQ/500./,GAIN/1./,VIN/1.100/
C     TIME AND DATE ARE PRINTED
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT('RFBLDF      TIME ',I2,':',I2,':',I2
     *,'  DATE ',I2,'/',I2,'/',I2)
      W=2.0*PI*FREQ
      FNZT=FLOAT(NZT)
      OPEN(28,FILE='REF.DAT',STATUS='OLD')
   10 READ(28,11)COIL,R5,R1,R2,L3,R3,R4,L4,L5,L6
     *,R6,R7,TNDR,TNPU
   11 FORMAT(A6,9F8.4,F10.4,F11.4,2F8.1)
      IF(COIL.EQ.'END   ')WRITE(0,*)' COIL NOT FOUND'
      IF(COIL.EQ.'END   ')GO TO 1040
      IF(COIL.NE.NPROBE)GO TO 10
      L6=L6+0.010/R5
      WRITE(LOU,3)NPROBE,T1
    3 FORMAT('PROBE ',A6,'  PLATE THICKNESS',F7.4)
      WRITE(LOU,5)
    5 FORMAT('COIL    IN RAD  OT RAD  LENGTH OLO/REC   TURNS',
     *'  COIL RES  CKT: RES       CAP')
      WRITE(LOU,14)R1,R2,L3,L6,TNDR,R6,R0,C6
      WRITE(LOU,15)R3,R4,L4,L5,TNPU,R7,R9,C7
   14 FORMAT('DRIVER ',4(F7.4,1X),F8.1,3(1PE10.3))
   15 FORMAT('PICKUP ',4(F7.4,1X),F8.1,3(1PE10.3))
      WUSRR=0.5093979*U1*R5*R5*FREQ/RHO1
      WRITE(LOU,20)R5,FREQ,RHO1,U1,WUSRR
   20 FORMAT('RBAR',F7.4,' FREQ= ',1PE13.6,' RHO=',0PF9.4,
     *' PERM=',F7.3,' WUSRR=',F9.4)
      WRITE(0,23)NPROBE,T1,FREQ
   23 FORMAT('PROBE ',A6,' PLATE THK',F7.4,' FREQ=',F8.1)
```

```
      WRITE(LOU,*)
      WRITE(LOU,24)
   24 FORMAT('DEPTH      MAGNITUDE     PHASE')
      T1=T1/R5
      ZMSTEP=T1/MZT
      DELTAR=(R2-R3)/NRT
      TD=TNDR/((R2-R1)*L3)
      TP=TNPU/((R4-R3)*L4)
      ZDF=6.300475E-7*FREQ*TD*TD*R5
      ZPF=6.300475E-7*FREQ*TP*TP*R5
      ZMF=6.300475E-7*FREQ*TD*TP*R5
      DMF=0.1193662*WUSRR
      DO 1020 MZ=0,MZT
      WRITE(0,*)'MZ ',MZ
      IF(MZ.GT.(0.5*MZT)) THEN
C   Far side defect
      NS=2
      ZM=(MZT-MZ)*ZMSTEP
      ZD=-2.*ZM
      ZM=MZ*ZMSTEP
      ZD2=-ZM
      ELSE
C   Near side defect
      NS=1
      ZM=MZ*ZMSTEP
      ZD=-2.*ZM
      ZD2=-ZM
      END IF
      RD-R3 (0.5*DELTAR)
      SRHSR=0.
      SRHSI=0.
      DO 1010 NR=1,NRT
      WRITE(0,*)MZ,NR
      RD=RD+DELTAR
      SAIR1=0.0
      SAIR2=0.0
      SZDR=0.0
      SZDI=0.0
      SZPR=0.0
      SZPI=0.0
      SZMR=0.0
      SZMI=0.0
      DO 25 NZ=1,NZT
      SDDR(NZ)=0.0
      SDDI(NZ)=0.0
      SDPR(NZ)=0.0
      SDPI(NZ)-0.0
   25 CONTINUE
      B1=0.0
      B2=S2(1)
      DO 100 JKL=1,6
```

```
   30 RI9=SAIR1
      X=B1-0.5*S1(JKL)
C     DETERMINE NUMBER OF STEPS,ISTEPS,BASED ON STEP SIZE&INTERVAL
      ISTEPS=DNINT((B2-B1)/S1(JKL))
      DO 90 I=1,ISTEPS
      X=X+S1(JKL)
      CALL BESSEL(XJR2,X,R2)
      CALL BESSEL(XJR1,X,R1)
      CALL BESSEL(XJR4,X,R4)
      CALL BESSEL(XJR3,X,R3)
      R21=XJR2-XJR1
      R43=XJR4-XJR3
      XL3=X*L3
      IF(XL3.GT.5.0E-3) GO TO 60
      A1=XL3*XL3*(0.5-XL3/6.0)
      GO TO 80
   60 IF(XL3.GT.75.0) GO TO 70
      A1=XL3+DEXP(-XL3)-1.0
      GO TO 80
   70 A1=XL3-1.0
   80 A3=XL3-A1
      SFD=S1(JKL)*R21*R21
      SFP=S1(JKL)*R43*R43
      SFM=S1(JKL)*R21*R43
      SAIR1=SAIR1+SFD*2.0*A1
      XL4=X*L4
      IF(XL4.GT.5.0E-3) GO TO 81
      A2=XL4*XL4*(0.5-XL4/6.0)
      GO TO 83
   81 IF(XL4.GT.75.0) GO TO 82
      A2=XL4+DEXP(-XL4)-1.0
      GO TO 83
   82 A2=XL4-1.0
   83 A4=XL4-A2
      A13=1.0-A3
      A14=1.0-A4
      A5=DEXP(-X*L5)
      XL8=X*(L3-2.*L4-2.*L5)
      IF(XL8.GT.75.)A8=0.
      IF(XL8.LE.75.)A8=DEXP(-XL8)
      SAIR2=SAIR2+SFP*(4.0*A2-2.0*A4*A4*A8)
      IF(X*L6.GT.75.)GO TO 90
C        CALCULATION OF GAMMA FACTORS
      XX=X*X
      X1=DSQRT(0.5*(XX+DSQRT(XX*XX+WUSRR*WUSRR)))/U1
      Y1=WUSRR/(2.*X1*U1*U1)
      A6=DEXP(-X*L6)
      XL7=X*(L3-L4-2.*L5)
      IF(XL7.GT.75.)A7=1.0
      IF(XL7.LE.75.)A7=1.0-DEXP(-XL7)
      A9=A4*A5*A6*A7
```

```
      APBR=(X+X1)*(X+X1)-Y1*Y1
      APBI=2.*Y1*(X+X1)
      AMBR=(X-X1)*(X-X1)-Y1*Y1
      AMBI=-2.*Y1*(X-X1)
      A2BR=0.0
      A2BI=-2.*X1*Y1
      ZNUR=A2BR
      ZNUI=A2BI
      DENR=APBR
      DENI=APBI
      DNCJ=DENR*DENR+DENI*DENI
C     SECTION THAT MULTIPLIES BY DEXP(ALPHA1*ZDEFECT)
      DO 88 NZ=1,NZT
C     NEAR SIDE DEFECT CALCULATION
      FZD=(FLOAT(NZ)-.5)*ZD/FNZT
C     FAR SIDE DEFECT CALCULATION
      IF(NS.EQ.2)FZD=-T1-FZD
      ZDR=X1*U1*FZD
      IF(ZDR.LT.-60.0)GO TO 89
      ZDI=Y1*U1*FZD
      XPDR=DEXP(ZDR)
      CSDI=DCOS(ZDI)*XPDR
      SNDI=DSIN(ZDI)*XPDR
      XRD=X*RD
      CALL BESEL1(XRD,RJ1)
      XX1=X*X1+XX
      XY1=X*Y1
      X1X=X*X1-XX
      ZNDR=XX1*CSDI-XY1*SNDI
      ZNDI=XX1*SNDI+XY1*CSDI
C     SECTION THAT MULTIPLIES BY DEXP(ALPHA1*(2*TH+ZDEFECT))
      TZR=X1*U1*(2.*T1+FZD)
      IF(TZR.GT.60.)GO TO 87
      TZI=Y1*U1*(2.*T1+FZD)
      XPZR=DEXP(-TZR)
      CSZI=DCOS(TZI)*XPZR
      SNZI=DSIN(TZI)*XPZR
      ZNDR=XX1*CSDI-XY1*SNDI+X1X*CSZI+XY1*SNZI
      ZNDI=XX1*SNDI+XY1*CSDI+XY1*CSZI-X1X*SNZI
C     SECTION THAT MULTIPLIES BY DEXP(-ALPHA1*2*CLADTH)
      TR=2.*X1*U1*T1
      IF(TR.GT.60.)GO TO 87
      TI=2.*Y1*U1*T1
      XPTR=DEXP(-TR)
      CSTI=DCOS(TI)*XPTR
      SNTI=DSIN(TI)*XPTR
      DENR=APBR-AMBR*CSTI-AMBI*SNTI
      DENI=APBI-AMBI*CSTI+AMBR*SNTI
      ZNUR=A2BR-A2BR*CSTI-A2BI*SNTI
      ZNUI=A2BI-A2BI*CSTI+A2BR*SNTI
      DNCJ=DENR*DENR+DENI*DENI
```

```
   87 ZDRL=(ZNDR*DENR+ZNDI*DENI)/DNCJ
      ZDIM=(DENR*ZNDI-ZNDR*DENI)/DNCJ
C     SUM DEFECT SENSITIVITY FACTORS FOR THE DRIVER & PICKUP COILS
      SDDR(NZ)=SDDR(NZ)+A3*A6*RJ1*2*ZDRL*R21*S1(JKL)
      SDDI(NZ)=SDDI(NZ)+A3*A6*RJ1*2*ZDIM*R21*S1(JKL)
      SDPR(NZ)=SDPR(NZ)+A9*RJ1*2*ZDRL*R43*S1(JKL)
      SDPI(NZ)=SDPI(NZ)+A9*RJ1*2*ZDIM*R43*S1(JKL)
   88 CONTINUE
   89 ZRL=(ZNUR*DENR+ZNUI*DENI)/DNCJ
      ZIM=(DENR*ZNUI-ZNUR*DENI)/DNCJ
      SZDI=SZDI+A3*A3*A6*A6*ZRL*SFD
      SZDR=SZDR-A3*A3*A6*A6*ZIM*SFD
      SZPI=SZPI+A9*A9*ZRL*SFP
      SZPR=SZPR-A9*A9*ZIM*SFP
      SZMI=SZMI+A3*A6*A9*ZRL*SFM
      SZMR=SZMR-A3*A6*A9*ZIM*SFM
   90 CONTINUE
      B1=B2
      B2=B2+S2(JKL)
      CHECK=(SAIR1-RI9)/SAIR1
      IF(ABS(CHECK).GT.ERR(JKL)) GO TO 30
  100 CONTINUE
C     COMPUTATION OF DRIVER INDUCTANCE
      Q6=ZDF*SAIR1/W
C     DEFINE COMPLEX QUANTITIES THAT ARE CONSTANT
      Z0=DCMPLX(0.0D0,-R0)
      Z6=DCMPLX(W*C6*R0,-1.0D0)
      Z7=DCMPLX(W*C7*R9,-1.0D0)
      Z9=DCMPLX(0.0D0,-R9)
C     AVERAGE DEFECT VALUES OVER DEPTH
      ADDR=0.0
      ADDI=0.0
      ADPR=0.0
      ADPI=0.0
      DO 125 NZ=1,NZT
      ADDR=ADDR+SDDR(NZ)/FNZT
      ADDI=ADDI+SDDI(NZ)/FNZT
      ADPR=ADPR+SDPR(NZ)/FNZT
      ADPI=ADPI+SDPI(NZ)/FNZT
  125 CONTINUE
  135 ZDTR0=ZDF*SZDR
      ZDTI0=ZDF*(SAIR1+SZDI)
      ZPTR0=ZPF*SZPR
      ZPTI0=ZPF*(SAIR2+SZPI)
      ZMTR0=ZMF*SZMR
      ZMTI0=ZMF*SZMI
      ZMTDR=-ZMF*DMF*(ADDR*ADPR-ADDI*ADPI)
      ZMTDI=-ZMF*DMF*(ADDI*ADPR+ADDR*ADPI)
C     DEFINE COMPLEX QUANTITIES, DO COMPLEX CIRCUIT CALCULATIONS
      ZDT=DCMPLX(ZDTR,ZDTI)
      ZPT=DCMPLX(ZPTR,ZPTI)
```

```
      ZMT=DCMPLX(ZMTR,ZMTI)
 1000 CONTINUE
      SRHSR=ZMTDR*DELTAR+SRHSR
      SRHSI=ZMTDI*DELTAR+SRHSI
  180 FORMAT(F6.3,1X,D11.3,1X,F7.2)
C  181 FORMAT(F6.3,2X,D11.3,3X,F7.2)
 1010 CONTINUE
      EBW=Z6*Z7*ZMT0*ZMT0+(Z6*(ZDT0+R6)+Z0)*(Z7*(ZPT0+R7)+Z9)
      RHSC=VIN*R9*GAIN*DCMPLX(SRHSR,SRHSI)/EBW
      RHSR=REAL(RHSC)
      RHSI=DIMAG(RHSC)
      RHSM=DSQRT(RHSR*RHSR+RHSI*RHSI)
      RHSP=DATAN2(RHSI,RHSR)*180./PI
      WRITE(LOD,180)ZD2,RHSM,RHSP
      WRITE(LOU,180)ZD2,RHSM,RHSP
      WRITE(0,180)ZD2,RHSM,RHSP
 1020 CONTINUE
 1040 END
```

RFDSF calculates mag. and phase of DSF for a lattice of points

Program RFDSF calculates the magnitude and phase of the defect sensitivity factor of a reflection coil at a lattice of points throughout a conducting plate, as shown in Fig. 11.

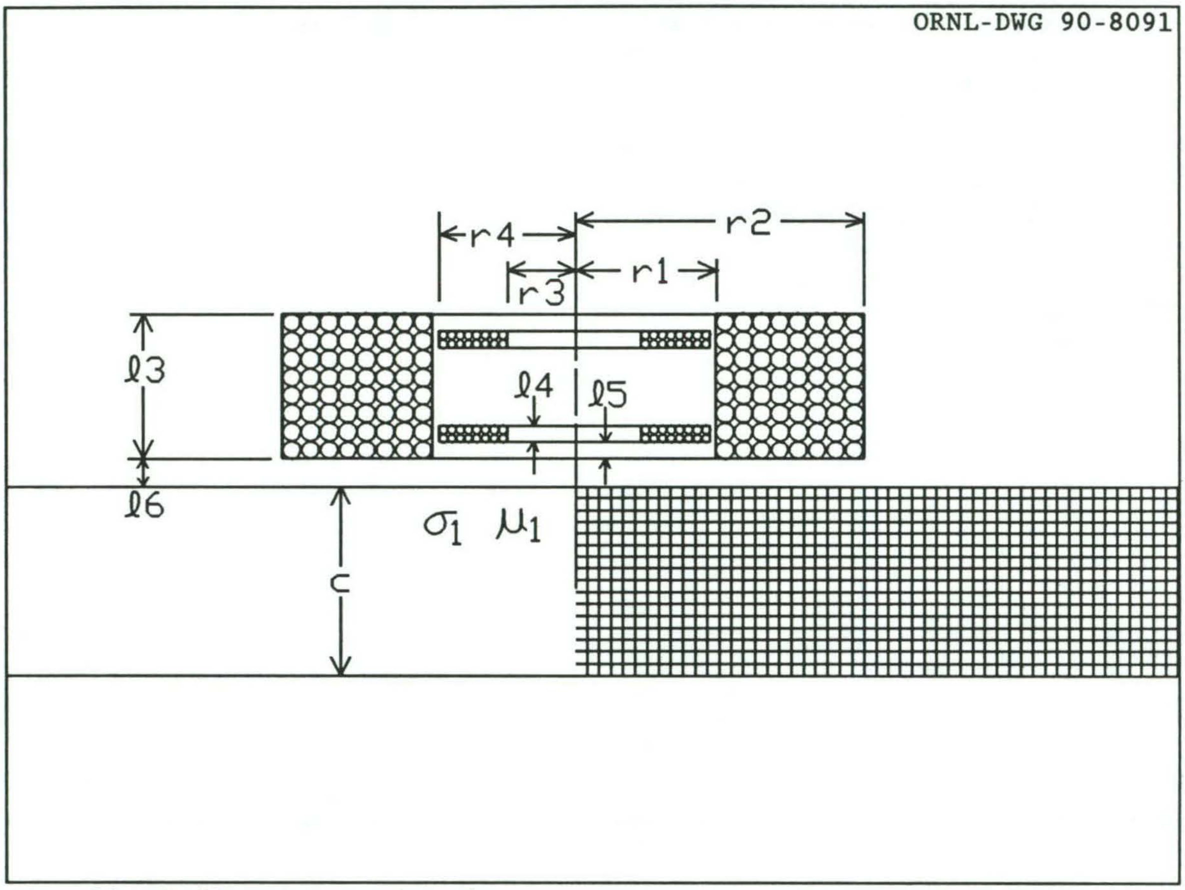

Fig. 11. Reflection probe above a conducting plane with a lattice of points.

The defect sensitivity is the mutual coupling term between the driver coil and pickup coils due to the defect in Eq. (26), and is given by:

$$DSF(r,z) = \frac{3\omega\mu\sigma\bar{r}^2}{8\pi}\left[\int_0^\infty \frac{e^{-\alpha\ell_6}}{\alpha^3} J(r_2,r_1)(1-e^{-\alpha\ell_3})J_1(\alpha r)\ F(\alpha,\alpha_1,z)d\alpha\right] \times$$

$$\left[\int_0^\infty \frac{e^{-\alpha(\ell_6+\ell_5)}}{\alpha^3} J(r_4,r_3)(1-e^{-\alpha(\ell_3-\ell_4-2\ell_5)})(1-e^{-\alpha\ell_4})J_1(\alpha r)F(\alpha,\alpha_1,z)d\alpha\right]\Bigg\} \quad (27)$$

As we can see from the circuit equation, (20), the mutual impedance term is not exactly equal to the voltage change due to the defect, but it is the dominant term. The defect sensitivity magnitudes and phases are stored in a file named FORT40 so they can be plotted by program RFDSFPLT.

Summary

1. Dimension arrays and declare variable types. Initialize variables.
2. Write the coil and plate information to a data file.
3. Select a point at which to calculate the defect sensitivity factor by choosing a value for RD, the radial distance from the axis of the coil, and for ZD, the distance of the point from the near side of the plate.
4. Perform the integration necessary to calculate the defect sensitivity factor at this point.
5. Store the calculations in a file.
6. Loop to 3 until done.

Variables

NOTE: Most of the variables which occur inside the integration loops are uninteresting because they do not correspond to anything physical and because they should never require user intervention. Therefore, these variables will not be discussed here. For a detailed description of these variables, see the documentation of program RFAVZSCN. For a discussion of the major differences between the integration section of program RFDSF and the integration section of program RFAVZSCN, see note #1.

Starred variables must be set by the user.

C6*	The total shunt capacitance in farads of the driving circuit.
C7*	The total shunt capacitance in farads of the pickup circuit.
DELTAR	The normalized distance in the radial direction between adjacent data points.
DELTAZ	The normalized distance in the axial direction between adjacent data points.
FREQ*	The operating frequency in hertz.
L3	The normalized length of the driver coil.
L4	The normalized length of the pickup coil.

L5	The normalized distance of recess of the pickup coil.
L6	The normalized lift-off of the driver coil.
LOD*	The channel on which the output data file is opened.
NPROBE*	Character variable which contains the name of the reflection probe which is to be used in the calculations.
NRT*	The total number of points in the radial direction at which the defect sensitivity factor is calculated.
NZT*	The total number of points in the axial direction at which the defect sensitivity factor is calculated.
R0*	Output series resistance of driving amplifier in ohms.
R1	The normalized inner radius of the driver coil.
R2	The normalized outer radius of the driver coil.
R3	The normalized inner radius of the pickup coil.
R4	The normalized outer radius of the pickup coil.
R5	The mean radius of the driver coil in inches.
R6	DC resistance of the driver coil in ohms.
R7	DC resistance of the pickup coil in ohms.
R9*	The amplifier input impedance.
RHO1*	The resistivity in $\mu\Omega$-cm of the plate.
T1*	The thickness of the plate. When it first occurs, it is in inches, but it is normalized by the program.
TD	The density of turns in the driver coil.
TP	The density of turns in the pickup coil.
TNDR	The number of turns in the driver coil.
TNPU	The number of turns in the pickup coil.
U1*	The relative magnetic permeability of the plate.

Notes

1. The integration in this program is very similar to the integration in program RFAVZSCN, but some very minor differences do exist. Most of these come from the fact that this program calculates only the defect sensitivity factor, which does not depend on the volume of the defect, and program RFAVZSCN calculates the voltage change due to the defect, which does depend on the volume of the defect. While it is necessary to calculate the defect sensitivity factor to calculate the voltage change due to the defect, program RFAVZSCN never calculates the defect sensitivity factor as a separate quantity. Thus, some of the variables in RFAVZSCN which have the same name as variables in program RFDSF have an extra factor of the defect volume. This program does carry and calculate some quantities that are not directly used by this program but are used by similar programs. The calculation of these variables does not add any significant running time to the program.

Sample output

Printer output of program RFDSF:

```
RFDSF      TIME  9:24:38  DATE  8/ 9/89
PROBE 250A   PLATE THIK 0.2500
COIL   IN RAD  OT RAD  LENGTH OLO/REC  TURNS COIL RES  CKT: RES   CAP
DRIVER 0.7500  1.2500  0.6000  0.0600 2350.0 5.310E+02 3.050E+03 8.470E-11
PICKUP 0.3500  0.7000  0.2000  0.0000 3450.0 5.147E+03 1.000E+06 8.450E-11
RBAR 0.2500 FREQ=  5.000000E+02 RHO=  4.0540 PERM=  1.000 WUSRR=   3.9267
```

Partial listing of file FORT40:

```
      25        20
     0.08000       0.05263
     0.75000       1.25000
     0.35000       0.70000
     0.60000       0.20000
     0.06000       0.00000
     1.00000
   1     1     0.11353D-04 -0.46336D+00
   1     2     0.97720D-05 -0.53972D+00
   1     3     0.81151D-05 -0.61645D+00
   1     4     0.65622D-05 -0.69443D+00
   1     5     0.52051D-05 -0.77420D+00
   1     6     0.40726D-05 -0.85603D+00
   1     7     0.31569D-05 -0.94004D+00
   1     8     0.24324D-05 -0.10262D+01
   1     9     0.18679D-05 -0.11143D+01
   1    10     0.14326D-05 -0.12040D+01
```

Listing

```
      PROGRAM RFDSF
C     VERSION August 7, 1989
C    PROGRAM CALCULATES THE CHANGE IN MAGNITUDE AND PHASE DUE TO A DEFECT
C     AT A LATTICE OF LOCATIONS IN THE R AND Z DIMENSIONS IN A PLATE
      CHARACTER NPROBE*6,COIL*6
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 L3,L4,L5,L6
      COMPLEX DEN,Z0,Z6,Z7,Z9,ZDT,ZPT,ZMT,DSFC
      DIMENSION S1(6),S2(6),ERR(6),RJ(120)
      DIMENSION SDDR(120,40),SDDI(120,40),SDPR(120,40),SDPI(120,40)
      DATA LOU/8/,PI/3.141592653/,LOD/40/
      DATA S1/.005,.02,.05,.1,.5,2./
      DATA S2/1.0,2.0,5.0,10.0,50.0,200.0/
      DATA ERR/.1,.01,.001,1.E-4,1.E-5,1.E-6/
      DATA RHO1/4.054/,U1/1.0/,NRT/25/,NZT/20/
      DATA T1/0.250/,NPROBE/'250A  '/
      DATA R0/3050./,R9/1.0D6/,C6/8.47E-11/,C7/8.45E-11/
      DATA FREQ/500./,VIN/1.1/,GAIN/1./,DELTAR/0.08/
C     TIME AND DATE ARE PRINTED
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,2) IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT('RFDSF      TIME ',I2,':',I2,':',I2
     *,'  DATE ',I2,'/',I2,'/',I2)
      W=2.0*PI*FREQ
      OPEN(28,FILE='REF.DAT',STATUS='OLD')
   10 READ(28,11)COIL,R5,R1,R2,L3,R3,R4,L4,L5,L6
     *,R6,R7,TNDR,TNPU
   11 FORMAT(A6,9F8.4,F10.4,F11.4,2F8.1)
      IF(COIL.EQ.'END   ')WRITE(0,*)' COIL NOT FOUND'
      IF(COIL.EQ.'END   ')GO TO 1020
      IF(COIL.NE.NPROBE)GO TO 10
      L6=L6+0.010/R5
      WRITE(LOU,3)NPROBE,T1
    3 FORMAT('PROBE ',A6,' PLATE THIK',F7.4)
      WRITE(LOU,5)
    5 FORMAT('COIL   IN RAD  OT RAD  LENGTH OLO/REC   TURNS',
     *'  COIL RES  CKT: RES      CAP')
      WRITE(LOU,14)R1,R2,L3,L6,TNDR,R6,R0,C6
      WRITE(LOU,15)R3,R4,L4,L5,TNPU,R7,R9,C7
   14 FORMAT('DRIVER ',4(F7.4,1X),F8.1,3(1PE10.3))
   15 FORMAT('PICKUP ',4(F7.4,1X),F8.1,3(1PE10.3))
      WUSRR=0.5093979*U1*R5*R5*FREQ/RHO1
      WRITE(LOU,20)R5,FREQ,RHO1,U1,WUSRR
   20 FORMAT('RBAR',F7.4,' FREQ= ',1PE13.6,' RHO=',0PF9.4,
     *' PERM=',F7.3,' WUSRR=',F9.4)
      WRITE(0,23)NPROBE,T1,FREQ
   23 FORMAT('PROBE ',A6,' PLATE THK',F7.4,' FREQ=',E12.4)
```

```
      TD=TNDR/((R2-R1)*L3)
      TP=TNPU/((R4-R3)*L4)
      ZDF=6.300475E-7*FREQ*TD*TD*R5
      ZPF=6.300475E-7*FREQ*TP*TP*R5
      ZMF=6.300475E-7*FREQ*TD*TP*R5
      DDF=0.1193662*WUSRR
      DPF=0.1193662*WUSRR
      DMF=0.1193662*WUSRR
      T1=T1/R5
      ZFT=-T1
      DELTAZ=T1/(NZT-1)
      WRITE(LOD,7)NRT,NZT
      WRITE(LOD,8)DELTAR,DELTAZ
      WRITE(LOD,8)R1,R2
      WRITE(LOD,8)R3,R4
      WRITE(LOD,8)L3,L4
      WRITE(LOD,8)L6,L5
      WRITE(LOD,9)T1
    7 FORMAT(I8,1X,I8)
    8 FORMAT(F12.5,1X,F12.5)
    9 FORMAT(F12.5)
      SAIR1=0.0
      SAIR2=0.0
      SZDR=0.0
      SZDI=0.0
      SZPR=0.0
      SZPI=0.0
      SZMR=0.0
      SZMI=0.0
      DO 27 NR=1,NRT
      DO 25 NZ=1,NZT
      SDDR(NR,NZ)=0.0
      SDDI(NR,NZ)=0.0
      SDPR(NR,NZ)=0.0
   25 SDPI(NR,NZ)=0.0
   27 CONTINUE
      B1=0.0
      B2=S2(1)
      DO 100 JKL=1,6
   30 RI9=SAIR1
      X=B1-0.5*S1(JKL)
C     DETERMINE NUMBER OF STEPS,ISTEPS,BASED ON STEP SIZE&INTERVAL
      ISTEPS=DNINT((B2-B1)/S1(JKL))
      DO 95 I=1,ISTEPS
      X=X+S1(JKL)
      CALL BESSEL(XJR2,X,R2)
      CALL BESSEL(XJR1,X,R1)
      CALL BESSEL(XJR4,X,R4)
      CALL BESSEL(XJR3,X,R3)
      R21=XJR2-XJR1
      R43=XJR4-XJR3
```

```
      XL3=X*L3
      IF(XL3.GT.5.0E-3) GO TO 60
      A1=XL3*XL3*(0.5-XL3/6.0)
      GO TO 80
   60 IF(XL3.GT.75.0) GO TO 70
      A1=XL3+DEXP(-XL3)-1.0
      GO TO 80
   70 A1=XL3-1.0
   80 A3=XL3-A1
      SFD=S1(JKL)*R21*R21
      SFP=S1(JKL)*R43*R43
      SFM=S1(JKL)*R21*R43
      SAIR1=SAIR1+SFD*2.0*A1
      XL4=X*L4
      IF(XL4.GT.5.0E-3) GO TO 81
      A2=XL4*XL4*(0.5-XL4/6.0)
      GO TO 83
   81 IF(XL4.GT.75.0) GO TO 82
      A2=XL4+DEXP(-XL4)-1.0
      GO TO 83
   82 A2=XL4-1.0
   83 A4=XL4-A2
      A13=1.0-A3
      A14=1.0-A4
      A5=DEXP(-X*L5)
      XL8=X*(L3-2.*L4-2.*L5)
      IF(XL8.GT.75.)A8=0.
      IF(XL8.LE.75.)A8=DEXP(-XL8)
      SAIR2=SAIR2+SFP*(4.0*A2-2.0*A4*A4*A8)
      IF(X*L6.GT.75.)GO TO 95
C        CALCULATION OF GAMMA FACTORS
      XX=X*X
      X1=DSQRT(0.5*(XX+DSQRT(XX*XX+WUSRR*WUSRR)))/U1
      Y1=WUSRR/(2.*X1*U1*U1)
      A6=DEXP(-X*L6)
      XL7=X*(L3-L4-2.*L5)
      IF(XL7.GT.75.)A7=1.0
      IF(XL7.LE.75.)A7=1.0-DEXP(-XL7)
      A9=A4*A5*A6*A7
      APBR=(X+X1)*(X+X1)-Y1*Y1
      APBI=2.*Y1*(X+X1)
      AMBR=(X-X1)*(X-X1)-Y1*Y1
      AMBI=-2.*Y1*(X-X1)
      A2BR=0.0
      A2BI=-2.*X1*Y1
      ZNUR=A2BR
      ZNUI=A2BI
      DENR=APBR
      DENI=APBI
      DNCJ=DENR*DENR+DENI*DENI
C     SECTION THAT MULTIPLIES BY DEXP(ALPHA1*ZDEFECT)
```

```
      DO 91 NZ=1,NZT
      FZD=(NZ-1)*ZFT/(NZT-1)
      ZDR=X1*U1*FZD
      IF(ZDR.LT.-60.0)GO TO 93
      ZDI=Y1*U1*FZD
      XPDR=DEXP(ZDR)
      CSDI=DCOS(ZDI)*XPDR
      SNDI=DSIN(ZDI)*XPDR
      XX1=X*X1+XX
      XY1=X*Y1
      X1X=X*X1-XX
      ZNDR=XX1*CSDI-XY1*SNDI
      ZNDI=XX1*SNDI+XY1*CSDI
C     SECTION THAT MULTIPLIES BY DEXP(ALPHA1*(2*TH+ZDEFECT))
      TZR=X1*U1*(2.*T1+FZD)
      IF(TZR.GT.60.)GO TO 87
      TZI=Y1*U1*(2.*T1+FZD)
      XPZR=DEXP(-TZR)
      CSZI=DCOS(TZI)*XPZR
      SNZI=DSIN(TZI)*XPZR
      ZNDR=XX1*CSDI-XY1*SNDI+X1X*CSZI+XY1*SNZI
      ZNDI=XX1*SNDI+XY1*CSDI+XY1*CSZI-X1X*SNZI
C     SECTION THAT MULTIPLIES BY DEXP(-ALPHA1*2*CLADTH)
      TR=2.*X1*U1*T1
      IF(TR.GT.60.)GO TO 87
      TI=2.*Y1*U1*T1
      XPTR=DEXP(-TR)
      CSTI=DCOS(TI)*XPTR
      SNTI=DSIN(TI)*XPTR
      DENR=APBR-AMBR*CSTI-AMBI*SNTI
      DENI=APBI-AMBI*CSTI+AMBR*SNTI
      ZNUR=A2BR-A2BR*CSTI-A2BI*SNTI
      ZNUI=A2BI-A2BI*CSTI+A2BR*SNTI
      DNCJ=DENR*DENR+DENI*DENI
   87 ZDRL=(ZNDR*DENR+ZNDI*DENI)/DNCJ
      ZDIM=(DENR*ZNDI-ZNDR*DENI)/DNCJ
C     LOOP OVER THE R VARIATION FOR THE DEFECT
      DO 90 NR=1,NRT
      IF(NZ.GT.1) GO TO 89
      RD=FLOAT(NR)*DELTAR
      XRD=X*RD
      CALL BESEL1(XRD,RJ1)
      RJ(NR)=RJ1
C     SUM DEFECT SENSITIVITY FACTORS FOR THE DRIVER & PICKUP COILS
   89 SDDR(NR,NZ)=SDDR(NR,NZ)+A3*A6*RJ(NR)*2*ZDRL*R21*S1(JKL)
      SDDI(NR,NZ)=SDDI(NR,NZ)+A3*A6*RJ(NR)*2*ZDIM*R21*S1(JKL)
      SDPR(NR,NZ)=SDPR(NR,NZ)+A9*RJ(NR)*2*ZDRL*R43*S1(JKL)
   90 SDPI(NR,NZ)=SDPI(NR,NZ)+A9*RJ(NR)*2*ZDIM*R43*S1(JKL)
   91 CONTINUE
   93 ZRL=(ZNUR*DENR+ZNUI*DENI)/DNCJ
      ZIM=(DENR*ZNUI-ZNUR*DENI)/DNCJ
```

```
      SZDI=SZDI+A3*A3*A6*A6*ZRL*SFD
      SZDR=SZDR-A3*A3*A6*A6*ZIM*SFD
      SZPI=SZPI+A9*A9*ZRL*SFP
      SZPR=SZPR-A9*A9*ZIM*SFP
      SZMI=SZMI+A3*A6*A9*ZRL*SFM
      SZMR=SZMR-A3*A6*A9*ZIM*SFM
   95 CONTINUE
      B1=B2
      B2=B2+S2(JKL)
      CHECK=(SAIR1-RI9)/SAIR1
      IF(ABS(CHECK).GT.ERR(JKL)) GO TO 30
  100 CONTINUE
      DO 130 NR=1,NRT
      DO 125 NZ=1,NZT
      DSFR=DMF*(SDDR(NR,NZ)*SDPR(NR,NZ)-SDDI(NR,NZ)*SDPI(NR,NZ))
      DSFI=DMF*(SDDI(NR,NZ)*SDPR(NR,NZ)+SDDR(NR,NZ)*SDPI(NR,NZ))
      DSFM=SQRT(DSFR*DSFR+DSFI*DSFI)
      DSFP=DATAN2(DSFI,DSFR)
      WRITE(LOD,126)NR,NZ,DSFM,DSFP
  125 CONTINUE
  130 CONTINUE
  126 FORMAT(I4,1X,I4,1X,D14.5,1X,D12.5)
 1010 CONTINUE
 1020 STOP
      END
```

RFDSFPLT generates a contour plot of magnitude of DSF

Program RFDSFPLT generates a contour plot of the magnitude of the defect sensitivity factor for a reflection coil using calculations performed and stored by program RFDSF.

Summary

1. Declare arrays and variable types. Initialize variables.
2. Open the file created by program RFDSF.
3. Read in the information about the coil and the plate from the data file.
4. Calculate the position in the normalized coordinate system (see note #1) of the data points which are about to be read in.
5. Set the label flags for the contours.
6. Read the data stored by program RFDSF into array DSFMA.
7. Specify the values of the magnitude of the defect sensitivity factor where the contours are to be drawn.
8. Call the PRINTMATIC contour initialization routines.
9. Draw the contours.
10. Draw the coils and the plate.

Variables

Starred variables must be set by the user.

CNM*	Real array which contains the values of the magnitude of the defect sensitivity factor on the contour lines. It is used as input for routine DCNTOUR.
DELTAR	The normalized distance in the radial direction between adjacent data points.
DELTAZ	The normalized distance in the axial direction between adjacent data points.
DSFMA	Two-dimensional real array which is used to hold the values of the magnitude of the defect sensitivity factor which are read in from a data file. Array DSFMA is used as input for routine DINIT.
L3	The normalized length of the driver coil.
L4	The normalized length of the pickup coil.
L5	The normalized distance of recess of the pickup coil.
L6	The normalized lift-off of the driver coil.
LBM*	Integer array that specifies which of the contours are to be labeled with their values. If all elements of LBM are zero, none of the contours will be labeled. Array LBM is used as input for routine DCNTOUR.
LOE*	The channel on which the file created by program

	RFDSF is opened.
NAME'	Character variable which contains the name of the file that program RFDSFPLT uses for output. NAME is used as input for routine DCNTOUR.
NC'	Specifies the number of contours to be drawn. The value of NC must be less than or equal to 10. It is used as input for routine DCNTOUR.
NRT	The number of points in the radial direction at which calculations were performed by program RFDSF.
NZT	The number of points in the axial direction at which calculations were performed by program RFDSF.
R1	The normalized inner radius of the driver coil.
R2	The normalized outer radius of the driver coil.
R3	The normalized inner radius of the pickup coil.
R4	The normalized outer radius of the pickup coil.
T1	The normalized thickness of the plate.
XX	Real array which describes the radial position of the data points in array DSFMA in the normalized coordinate system. It is used as input for routine DCNTOUR.
YY	Real array which describes the axial position of the data points in array DSFMA in the normalized coordinate system. It is used as input for routine DCNTOUR.

Notes

1. The coordinate system set up and used by this program has its origin at the intersection of the coil axis and the near side of the plate. One unit of distance in the coordinate system is equal to one mean radius of the driver coil.

2. The array DSFMA must be dimensioned to exactly NRT by NZT. Each time the value of NRT or NZT is changed in program RFDSF, the statement dimensioning the array in program RFDSFPLT must be changed also.

3. The statements in this program which seem to do nothing but write variables to the screen actually have a more important function. Due to a bug in either the PRINTMATIC routines or in RM/FORTRAN, the PRINTMATIC routine DLINE, which is supposed to draw a straight line, sometimes refuses to work. It was discovered by accident that putting a WRITE statement near the call to the routine corrects the problem.

4. Program RFDSFPLT does not actually send anything to the printer; it merely creates a file whose name is given by the program variable NAME. If the value of NAME is 'filename.ext', then to print the file created by program RFDSFPLT, enter

```
DPRINT filename.ext
```

DPRINT.EXE is a program supplied by PRINTMATIC.

Sample Output

Fig. 12 shows a typical plot generated by program RFDSFPLT.

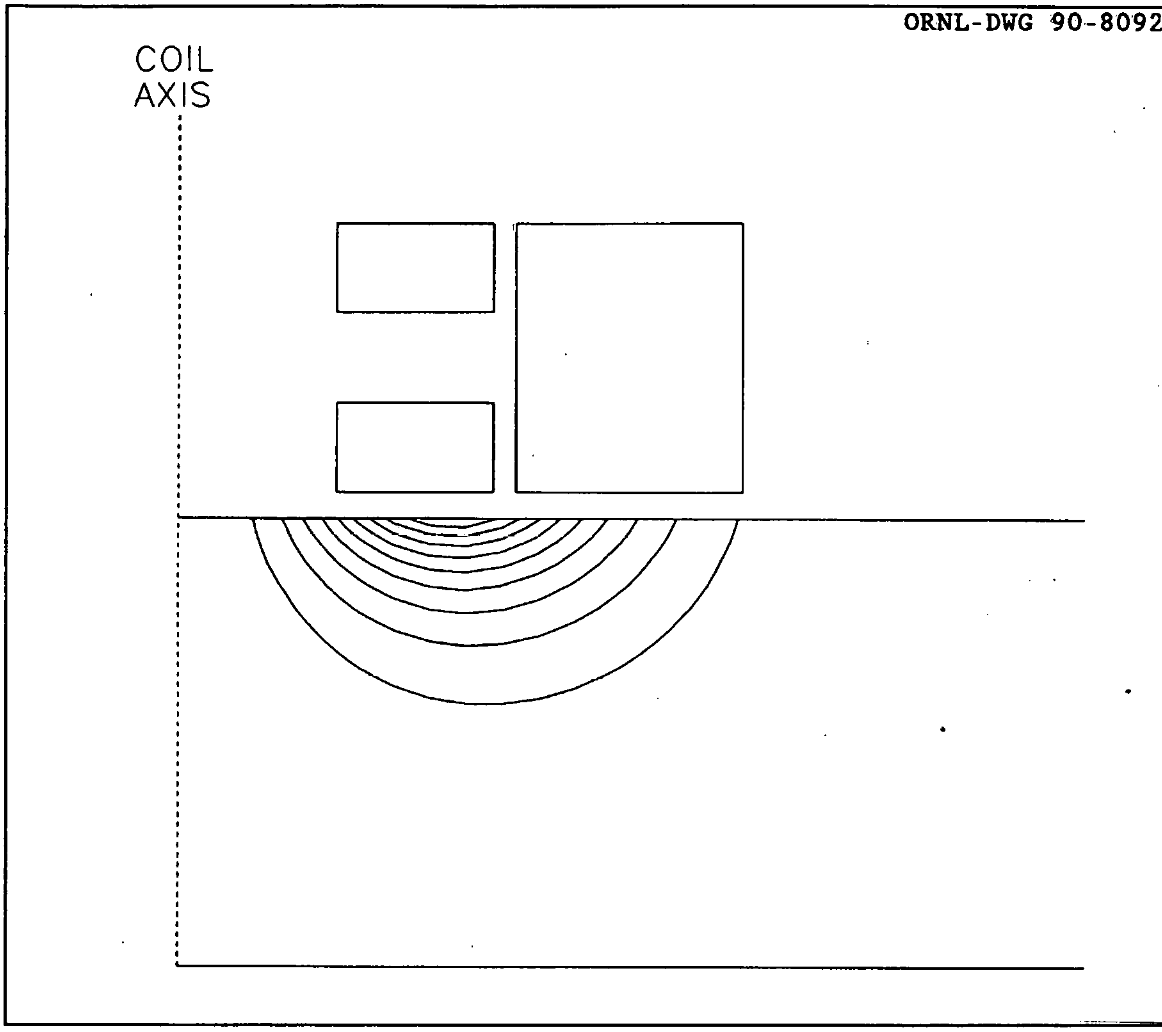

Fig. 12. Contour map of a the defect sensitivity factor for a reflection probe above a conducting plate.

Listing

```
      PROGRAM RFDSFPLT
C      VERSION October 31, 1988
C      Program to generate a contour plot of the magnitude of the
C      defect sensitivity factor of a reflection coil.
C
      CHARACTER*80 NAME
      IMPLICIT REAL*4 (A-H,O-Z)
      REAL*4 DSFMA(25,20)
      REAL*4 XX(25),YY(20)
      REAL*4 CNM(10)
      REAL*4 L3,L4,L5,L6
      INTEGER*2 LBM(10)
      INTEGER*2 I1,J1,I2,J2
      DATA XSCALE/1.0/,NC/9/
      DATA IDEF/2/,LOE/40/
C
C   Open the file created by program RFDSF and read in the
C    coil and plate information.
C
      OPEN(LOE,FILE='FORT40',STATUS='OLD')
      READ(LOE,*)NRT,NZT
      READ(LOE,*)DELTAR,DELTAZ
      READ(LOE,*)R1,R2
      READ(LOE,*)R3,R4
      READ(LOE,*)L3,L4
      READ(LOE,*)L6,L5
      READ(LOE,*)T1
C
C   Calculate the position of the data points in the
C    normalized coordinate system.
C
      DO 110 I=1,NRT
      XX(I)=REAL(I)*DELTAR
  110 CONTINUE
      DO 120 I=0,NZT-1
      YY(I+1)=-(((NZT-1)-REAL(I))*DELTAZ)
  120 CONTINUE
C
C   Set the label flags for the contours.
C
      DO 130 I=1,10
      LBM(I)=0
  130 CONTINUE
C
C   Read in the data stored by program RFDSF.
C
  140 READ(LOE,*,END=150)NR,NZ,DSFM
      NZ=NZT-NZ+1
      IF(DSFM.GT.DSFMMAX)DSFMMAX=DSFM
```

```
      IF(DSFMMIN.EQ.0.)DSFMMIN=DSFM
      IF(DSFM.LT.DSFMMIN)DSFMMIN=DSFM
      DSFMA(NR,NZ)=DSFM
      GO TO 140
C
C   Specify the values at which the contours are to be drawn
C
  150 VARMAG=DSFMMAX-DSFMMIN
      CNTDIF=VARMAG/(NC+1)
      DO 160 I=1,NC
      CNM(I)=DSFMMAX-I*CNTDIF
  160 CONTINUE
C
C   Call the necessary initialization routines.
C
      NAME = 'RFDSF.FIL'
      CALL DINIT(NAME)
      CALL DPLOT(0.7,1.,6.3,6.,-0.1,2.,-1.,1.,0.,0.)
      CALL DCTRDEF(1,1,1,1,1)
C
C   Draw the contours.
C
      CALL DCNTOUR(XSCALE,XX,YY,DSFMA,CNM,LBM,NRT,NZT,NC,IDEF)
C
C   Draw the plate.
C
      write(0,*)j2
      X1=0.
      Y1=0.
      X2=2.
      Y2=-T1
      CALL DRTOI(X1,Y1,I1,J1)
      CALL DRTOI(X2,Y2,I2,J2)
      write(0,*)j2
      CALL DLINE(I1,J1,I2,J1)
      write(0,*)j2
      CALL DLINE(I1,J2,I2,J2)
      write(0,*)j2
C
C   Draw the driver coil.
C
      X1=R1
      Y1=L6
      X2=R2
      Y2=L3+L6
      CALL DRTOI(X1,Y1,I1,J1)
      CALL DRTOI(X2,Y2,I2,J2)
      write(0,*)j2
      CALL DLINE(I1,J1,I2,J1)
      CALL DLINE(I1,J2,I2,J2)
      CALL DLINE(I1,J1,I1,J2)
```

```
      WRITE(0,*)I2,J1,I2,J2
      CALL DLINE(I2,J1,I2,J2)
      WRITE(0,*)I2,J1,I2,J2
C
C   Draw the pickup coils.
C
      X1=R3
      Y1=L6+L5
      X2=R4
      Y2=L4+L6+L5
      CALL DRTOI(X1,Y1,I1,J1)
      CALL DRTOI(X2,Y2,I2,J2)
      CALL DLINE(I1,J1,I2,J1)
      CALL DLINE(I1,J2,I2,J2)
      CALL DLINE(I1,J1,I1,J2)
      WRITE(0,*)I2,J1,I2,J2
      CALL DLINE(I2,J1,I2,J2)
      WRITE(0,*)I2,J1,I2,J2
      X1=R3
      Y1=L6+L3-L5-L4
      X2=R4
      Y2=L6+L3-L5
      CALL DRTOI(X1,Y1,I1,J1)
      CALL DRTOI(X2,Y2,I2,J2)
      CALL DLINE(I1,J1,I2,J1)
      CALL DLINE(I1,J2,I2,J2)
      CALL DLINE(I1,J1,I1,J2)
      WRITE(0,*)I2,J1,I2,J2
      CALL DLINE(I2,J1,I2,J2)
      WRITE(0,*)I2,J1,I2,J2
C
C   Draw the coil axis.
C
      X1=0.
      Y1=-1.
      X2=0.
      Y2=0.9
      CALL DRTOI(X1,Y1,I1,J1)
      CALL DRTOI(X2,Y2,I2,J2)
      CALL DDASH(I1,J1,I2,J2,1,10,10)
      CALL DRTOI(-0.1,1.0,I1,I2)
      CALL DFONT(4,'COIL',I1,I2,1)
      CALL DRTOI(-0.1,0.92,I1,I2)
      CALL DFONT(4,'AXIS',I1,I2,1)
C
C   Terminate the program.
C
      CALL DFINIS
      write(0,*)j2
      stop
      END
```

RFAVZSCN calculates defect voltage change, average over depth

Program RFAVZSCN calculates the change in the induced voltage in a pickup coil due to the presence of a defect in a plate. It does the calculations for a number of different coil-to-defect distances (see note #1), and it has the ability to divide the defect into a number of parts along the depth of the defect, centered on the axis of symmetry of the defect and to perform the calculations for each part separately, averaging the results to achieve better agreement with experimental results than if the defect were treated as a whole (see note #2). The program will scan the coil across the defect in the r direction and calculate the magnitude and phase of the defect. It is now set up to scan from the inner edge of the pickup coil, r_3, to the outer edge of the driver coil, r_2. It can be easily changed to any other set of values that are desired. The statement that controls the defect starting and ending point is:

```
RD=R3+DELTR*FLOAT(NR)
```

This statement must be changed at both its occurrences in the program. The step size, DELTR, is calculated by:

```
DELTR=(R2-R3)/NRT
```

The new values for the start and end of the scan should be placed in these equations.

A large section of the program is concerned with doing integrations to find the impedance of the coils. The details of the integration have been placed in a separate section at the end of the discussion.

The output from this program is stored in the file FORT39.

<u>Summary</u>

1. Dimension arrays and declare variable types.
2. Initialize variables.
3. Calculate the coil impedances in the absence of defects.
4. Select a value for RD, the radial distance between the coil axis and the center of the defect.
5. Do the integrals to calculate the change in coil impedance due to the defect.
6. Calculate the change in voltage due to the defect from the impedance changes.
7. Write the results to a data file.
8. Loop to 4 until done.

Variables

Starred variables must be set by the user.

C6*	The total shunt capacitance in farads of the driving circuit.
C7*	The total shunt capacitance in farads of the pickup circuit.
DELTR*	The normalized distance in the radial direction between adjacent data points.
DFDEP*	The depth to the bottom of the defect in inches.
DFDIAM*	The diameter of the defect in inches.
DVVI	The imaginary part of the change in induced voltage in the pickup coil due to the defect.
DVVR	The real part of the change in induced voltage in the pickup coil due to the defect.
FNZT	Same as variable NZT, but a real variable instead of an integer.
FREQ*	The operating frequency in hertz.
GAIN*	Gain of pickup amplifier.
L3	The normalized length of the driver coil.
L4	The normalized length of the pickup coil.
L5	The normalized distance of recess of the pickup coil.
L6	The normalized lift-off of the driver coil.
LOD*	The channel on which the output data file is opened.
NPROBE*	Character variable which contains the name of the reflection probe which is to be used in the calculations.
NRT*	The total number of points in the radial direction at which the defect sensitivity factor is calculated.
NS*	The side of the plate where the defect is located. If NS = 1, the defect is on the near side; if NS = 2, the defect is on the far side.
NZT*	The number of parts into which the defect is divided along its axis to do the calculations.
PHA	The phase of the change (not the change of the phase) of the induced voltage in the pickup coil due to the defect.
R0*	The output series resistance of the driving amplifier in ohms.
R1	The normalized inner radius of the driver coil.
R2	The normalized outer radius of the driver coil.
R3	The normalized inner radius of the pickup coil.
R4	The normalized outer radius of the pickup coil.
R5	The mean radius of the driver coil in inches.
R6	DC resistance of the driver coil in ohms.
R7	DC resistance of the pickup coil in ohms.
R9*	The input shunt resistance of pickup amplifier in

	ohms.
RD	The normalized distance in the radial direction from the axis of the coil to the center of the defect. (See note #1.)
RHO1*	The resistivity of the plate in μΩ-cm.
T1*	The thickness of the plate. It is input in inches and normalized by the program.
TD	The density of turns in the driver coil.
TMAG	The magnitude of the change (not the change of the magnitude) in induced voltage in the pickup coil due to the defect.
TNDR	The number of turns in the driver coil.
TNPU	The number of turns in the pickup coil.
TP	The density of turns in each pickup coil.
U1*	The relative magnetic permeability of the plate.
VIN*	Output voltage of driving amplifier in volts.
VOLN	The normalized volume of the defect.
VV0I	The imaginary part of the induced voltage in the pickup coil in the absence of defects.
VV0R	The real part of the induced voltage in the pickup coil in the absence of defects.
VV1I	The imaginary part of the induced voltage in the pickup coil in the presence of a defect.
VV1R	The real part of the induced voltage in the pickup coil in the presence of a defect.
W	The angular frequency at which the circuit is driven.
ZD	The normalized distance from the near surface of the plate to the center of the defect. It is a negative number.
ZDTI	The imaginary part of the total self impedance of the driver coil.
ZDTR	The real part of the total self impedance of the driver coil.
ZMTI	The imaginary part of the total mutual impedance between the driver and pickup coil.
ZMTR	The real part of the total mutual impedance between the driver and pickup coil.
ZPTI	The imaginary part of the total self impedance of the pickup coil.
ZPTR	The real part of the total self impedance of the pickup coil.

Notes

1. When this program begins, the radial distance RD between the coil axis and the center of the defect is initialized to the value of R3, the pickup coil inner radius, and when the calculations for this value of RD have been completed, the program increases the value of RD and repeats the calculations until RD is equal to R2, the driver coil outer radius. The

reason that the calculations are done over this interval is that this is the region where the defect signal is strongest, so the signal-to-noise ratio for experimental readings is highest in this region, and, therefore, the experimental readings and calculated readings agree most closely in this region.

2. Variable NZT controls the number of parts into which the defect is divided to perform the calculations. Since the theory upon which this program is based is more accurate for small defects, it is desirable to work with the defect in parts rather than as a whole.

Integration Section of Program RFAVZSCN

Symbol definitions

The following are definitions of the symbols used to describe the program variables appearing in the integration section of this program. All lengths are normalized by dividing by the mean radius of the driver coil unless otherwise noted.

α	Integration variable
α_1	$(\alpha^2+j\omega\mu\sigma_1\bar{r}^2)^{1/2}$
α_{22}	Defect shape and orientation factor
β_1	$(\alpha^2+j\omega\mu\sigma_1\bar{r}^2)^{1/2}/\mu$
c	Plate thickness
$J(x_2,x_1)$	Integral of $xJ_1(x)$ with respect to x from αx_1 to αx_2
$J_1(x)$	Bessel function of the first kind of order 1
ℓ_3	Length of driver coil
ℓ_4	Length of pickup coil
ℓ_5	Distance of recess of pickup coil
ℓ_6	Lift-off of driver coil
μ	Relative magnetic permeability of plate
N_3	Number of turns in the driver coil
N_4	Number of turns in each pickup coil
r	Coil-to-defect radial distance
$\bar{r}$	Mean radius of driver coil in inches
r_1	Inner radius of driver coil
r_2	Outer radius of driver coil
r_3	Inner radius of pickup coil
r_4	Outer radius of pickup coil
σ_1	Conductivity of plate
Vol_n	Normalized volume of defect
ω	Angular frequency at which circuit is driven
z	Depth to center of defect

Variables appearing in the integration section

Program variable	Symbolic equivalent
A1	$\alpha\ell_3+\exp(-\alpha\ell_3)-1$ (See note I1.)
A13	$\exp(-\alpha\ell_3)$
A14	$\exp(-\alpha\ell_4)$
A2	$\alpha\ell_4+\exp(-\alpha\ell_4)-1$ (See note I1.)
A2BI	$\mathrm{Im}[\alpha^2-\beta_1^2]$
A2BR	$\mathrm{Re}[\alpha^2-\beta_1^2]$
A3	$1-\exp(-\alpha\ell_3)$

A4	$1-\exp(-\alpha\ell_4)$
A5	$\exp(-\alpha\ell_5)$
A6	$\exp(-\alpha\ell_6)$
A7	$1-\exp[-\alpha(\ell_3-\ell_4-2\ell_5)]$ {See note I1.}
A8	$\exp[-\alpha(\ell_3-2\ell_4-2\ell_5)]$ {See note I1.}
A9	$[1-\exp(-\alpha\ell_4)]\exp(-\alpha\ell_5-\alpha\ell_6)[1-\exp(-\alpha(\ell_3-\ell_4-2\ell_5)]$
ADDI	{See note #I3.}
ADDR	{See note #I3.}
ADPI	{See note #I3.}
ADPR	{See note #I3.}
AMBI	$\mathrm{Im}[(\alpha-\beta_1)^2]$
AMBR	$\mathrm{Re}[(\alpha-\beta_1)^2]$
APBI	$\mathrm{Im}[(\alpha+\beta_1)^2]$
APBR	$\mathrm{Re}[(\alpha+\beta_1)^2]$
B1	{See note I2.}
B2	{See note I2.}
CHECK	{See note I2.}
CSDI	$\mathrm{Re}[\exp(\alpha_1 z)]$
CSTI	$\mathrm{Re}[\exp(2\alpha_1 c)]$
CSZI	$\mathrm{Re}[\exp(-\alpha_1(z+2c)]$
DDF	$\frac{3}{8\pi}\ \alpha_{22}\ \mathrm{Vol}_n\ \omega\mu\sigma_1\overline{r}^2$
DENI	$\mathrm{Im}[(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)]$
DENR	$\mathrm{Re}[(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)]$
DMF	$\frac{3}{8\pi}\ \alpha_{22}\ \mathrm{Vol}_n\ \omega\mu\sigma_1\overline{r}^2$
DPF	$\frac{3}{8\pi}\ \alpha_{22}\ \mathrm{Vol}_n\ \omega\mu\sigma_1\overline{r}^2$
ERR	{See note I2.}
ISTEPS	{See note I2.}
R21	$J(r_2,r_1)/\alpha^3$
R43	$J(r_4,r_3)/\alpha^3$
RI9	{See note I2.}

RJ1 $J_1(\alpha r)$

S1 $d\alpha$

S2 {See note I2.}

SAIR1 $$\int_0^\infty \frac{1}{\alpha^6} [J(r_2,r_1)]^2 2[\alpha\ell_3+\exp(-\alpha\ell_3)-1]\ d\alpha$$

SAIR2 $$\int_0^\infty \frac{1}{\alpha^6} [J(r_4,r_3)]^2[4(\alpha\ell_4+\exp(-\alpha\ell_4)-1)+$$
$$-2(1-\exp(-\alpha\ell_4))^2\exp(-\alpha(\ell_3-2\ell_4-2\ell_5))]\ d\alpha$$

SDDI $$\mathrm{Im}\left[\int_0^\infty \frac{1}{\alpha^3}J(r_2,r_1)(1-\exp(-\alpha\ell_3))J_1(\alpha r)(\exp(-\alpha\ell_6))\right.$$
$$\left.2\alpha\left[\frac{(\alpha+\beta_1)\exp(\alpha_1 z)-(\alpha-\beta_1)\exp(-\alpha_1(2c+z))}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]d\alpha\right]$$

SDDR $$\mathrm{Re}\left[\int_0^\infty \frac{1}{\alpha^3}J(r_2,r_1)(1-\exp(-\alpha\ell_3))J_1(\alpha r)(\exp(-\alpha\ell_6))\right.$$
$$\left.2\alpha\left[\frac{(\alpha+\beta_1)\exp(\alpha_1 z)-(\alpha-\beta_1)\exp(-\alpha_1(2c+z))}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]d\alpha\right]$$

SDPI $$\mathrm{Im}\left[\int_0^\infty \frac{1}{\alpha^3}\left[J(r_4,r_3)(1-\exp(-\alpha\ell_4))\exp(-\alpha(\ell_5+\ell_6))\right.\right.$$
$$\left.2\alpha(1-\exp(-\alpha(\ell_3-\ell_4-2\ell_5))J_1(\alpha r)\right]$$
$$\left.\left[\frac{(\alpha+\beta_1)\exp(\alpha_1 z)-(\alpha-\beta_1)\exp(-\alpha_1(2c+z))}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]d\alpha\right]$$

SDPR

$$\mathrm{Re}\left[\int_0^\infty \frac{1}{\alpha^3}\left[J(r_4,r_3)(1-\exp(-\alpha\ell_4))\exp(-\alpha(\ell_5+\ell_6))\right.\right.$$

$$\left. 2\alpha(1-\exp(-\alpha(\ell_3-\ell_4-2\ell_5))J_1(\alpha r)\right]$$

$$\left.\left[\frac{(\alpha+\beta_1)\exp(\alpha_1 z)-(\alpha-\beta_1)\exp(-\alpha_1(2c+z))}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right] d\alpha\right]$$

SFD $\quad \frac{1}{\alpha^6}[J(r_2,r_1)]^2\ d\alpha$

SFM $\quad \frac{1}{\alpha^6}J(r_4,r_3)J(r_2,r_1)\ d\alpha$

SFP $\quad \frac{1}{\alpha^6}[J(r_4,r_3)]^2\ d\alpha$

SNDI $\quad \mathrm{Im}[\exp(\alpha_1 z)]$

SNTI $\quad \mathrm{Im}[\exp(2\alpha_1 c)]$

SNZI $\quad -\mathrm{Im}[\exp(-\alpha_1(z+2c))]$

SZDI

$$\mathrm{Im}\left[\int_0^\infty \frac{1}{\alpha^6}\ (1-\exp(-\alpha\ell_3))^2\exp(-2\alpha\ell_6)(J(r_2,r_1))^2\right.$$

$$\left.\left[\frac{(\alpha-\beta_1)(\alpha+\beta_1)-(\alpha-\beta_1)(\alpha+\beta_1)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right] d\alpha\right]$$

SZDR

$$-\mathrm{Re}\left[\int_0^\infty \frac{1}{\alpha^6}\ (1-\exp(-\alpha\ell_3))^2\exp(-2\alpha\ell_6)(J(r_2,r_1))^2\right.$$

$$\left.\left[\frac{(\alpha-\beta_1)(\alpha+\beta_1)-(\alpha-\beta_1)(\alpha+\beta_1)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right] d\alpha\right]$$

SZMI $$\mathrm{Im}\left[\int_0^\infty \frac{1}{\alpha^6}\left[J(r_4,r_3)J(r_2,r_1)(1-\exp(-\alpha\ell_3))\exp(-\alpha(2\ell_6+\ell_5))\right.\right.$$

$$\left.(1-\exp(-\alpha\ell_4))(1-\exp(-\alpha(\ell_3-\ell_4-2\ell_5)\right]$$

$$\left.\left[\frac{(\alpha-\beta_1)(\alpha+\beta_1)-(\alpha-\beta_1)(\alpha+\beta_1)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]d\alpha\right]$$

SZMR $$-\mathrm{Re}\left[\int_0^\infty \frac{1}{\alpha^6}\left[J(r_4,r_3)J(r_2,r_1)(1-\exp(-\alpha\ell_3))\exp(-\alpha(2\ell_6+\ell_5))\right.\right.$$

$$\left.(1-\exp(-\alpha\ell_4))(1-\exp(-\alpha(\ell_3-\ell_4-2\ell_5)\right]$$

$$\left.\left[\frac{(\alpha-\beta_1)(\alpha+\beta_1)-(\alpha-\beta_1)(\alpha+\beta_1)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]d\alpha\right]$$

SZPI $$\mathrm{Im}\left[\int_0^\infty \frac{1}{\alpha^6}\,[J(r_4,r_3)]^2(1-\exp(-\alpha\ell_4))^2\exp(-2\alpha(\ell_5+\ell_6))\right.$$

$$\left.(1-\exp(-\alpha(\ell_3-\ell_4-2\ell_5)))^2\left[\frac{(\alpha-\beta_1)(\alpha+\beta_1)-(\alpha-\beta_1)(\alpha+\beta_1)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]d\alpha\right]$$

SZPR $$-\mathrm{Re}\left[\int_0^\infty \frac{1}{\alpha^6}\,[J(r_4,r_3)]^2(1-\exp(-\alpha\ell_4))^2\exp(-2\alpha(\ell_5+\ell_6))\right.$$

$$\left.(1-\exp(-\alpha(\ell_3-\ell_4-2\ell_5)))^2\left[\frac{(\alpha-\beta_1)(\alpha+\beta_1)-(\alpha-\beta_1)(\alpha+\beta_1)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]d\alpha\right]$$

TI $\mathrm{Im}[2\alpha_1 c]$

TR $\mathrm{Re}[2\alpha_1 c]$

TZI $\mathrm{Im}[\alpha_1(2c+z)]$

TZR $\mathrm{Re}[\alpha_1(2c+z)]$

X α

X1 $\mathrm{Re}(\beta_1)$

X1X	$\mathrm{Re}[\alpha(\beta_1-\alpha)]$
XJR1	$J(r_1,0)/\alpha^3$
XJR2	$J(r_2,0)/\alpha^3$
XJR3	$J(r_3,0)/\alpha^3$
XJR4	$J(r_4,0)/\alpha^3$
XL3	$\alpha\ell_3$
XL4	$\alpha\ell_4$
XL7	$\alpha(\ell_3-\ell_4-2\ell_5)$
XL8	$\alpha(\ell_3-2\ell_4-2\ell_5)$
XPDR	$\exp[\mathrm{Re}(\alpha_1 z)]$
XPTR	$\exp[\mathrm{Re}(2\alpha_1 c)]$
XPZR	$\exp[\mathrm{Re}(-\alpha_1(2c+z))]$
XRD	αr
XX	α^2
XX1	$\mathrm{Re}[\alpha(\beta_1+\alpha)]$
XY1	$\mathrm{Im}[\alpha(\beta_1+\alpha)]=\mathrm{Im}[\alpha(\beta_1-\alpha)]$
Y1	$\mathrm{Im}(\beta_1)$
ZDF	$\dfrac{\omega\pi\mu_0 N_3^{\,2}}{(r_2-r_1)^2\ell_3^{\,2}}$
ZDI	$\mathrm{Im}(\alpha_1 z)$
ZDIM	$\mathrm{Im}\left[\alpha\dfrac{(\alpha+\beta_1)\exp(\alpha_1 z)-(\alpha-\beta_1)\exp(-\alpha_1(2c+z))}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]$
ZDR	$\mathrm{Re}(\alpha_1 z)$
ZDRL	$\mathrm{Re}\left[\alpha\dfrac{(\alpha+\beta_1)\exp(\alpha_1 z)-(\alpha-\beta_1)\exp(-\alpha_1(2c+z))}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]$
ZIM	$\mathrm{Im}\left[\dfrac{(\alpha-\beta_1)(\alpha+\beta_1)-(\alpha-\beta_1)(\alpha+\beta_1)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]$
ZMF	$\dfrac{\omega\pi\mu_0 N_3 N_4}{(r_2-r_1)(r_4-r_3)\ell_3\ell_4}$
ZNDI	$\mathrm{Im}[\alpha(\beta_1+\alpha)\exp(\alpha_1 z)+\alpha(\beta_1-\alpha)\exp(-\alpha_1(2c+z))]$
ZNDR	$\mathrm{Re}[\alpha(\beta_1+\alpha)\exp(\alpha_1 z)+\alpha(\beta_1-\alpha)\exp(-\alpha_1(2c+z))]$
ZNUI	$\mathrm{Im}[(\alpha^2-\beta_1^{\,2})-(\alpha^2-\beta_1^{\,2})\exp(-2\alpha_1 c)]$ {See note I1.}
ZNUR	$\mathrm{Re}[(\alpha^2-\beta_1^{\,2})-(\alpha^2-\beta_1^{\,2})\exp(-2\alpha_1 c)]$ {See note I1.}

ZPF $$\frac{\omega\pi\mu_0 N_4^2}{(r_4-r_3)^2\ell_4^2}$$

ZRL $$\mathrm{Re}\left[\frac{(\alpha-\beta_1)(\alpha+\beta_1)-(\alpha-\beta_1)(\alpha+\beta_1)\exp(-2\alpha_1 c)}{(\alpha+\beta_1)^2-(\alpha-\beta_1)^2\exp(-2\alpha_1 c)}\right]$$

Notes for the integration section

I1. A number of the variables in the integration section are not always assigned their exact values but are approximated in certain cases to save time. For example, rather than calculate the exponential of a very small number, the Maclaurin series expansion is sometimes used. Also, the exponential of a very large negative number is usually treated as zero.

I2. Several variables appear in the integration section of the program which play no part in the calculations being done. They are merely there to do such things as to determine the maximum step size which can be used while still accurately calculating the integrals.

I3. Variables ADDR, ADDI, ADPR, and ADPI are the averages of the elements in arrays SDDR, SDDI, SDPR, and SDPI, respectively. For improved accuracy, the defect is divided into NZT parts to perform the calculations. The NZT elements of each array contain the calculations for these NZT parts. These elements are averaged to give the total effect of all of these parts on the impedances of the coils.

Sample output

Printer output of program RFAVZSCN:

```
RFAVZSCN  20 POINTS USED   TIME  9:42:19  DATE  8/ 9/89
PROBE 250A   PLATE THIK 0.2500
COIL    IN RAD OT RAD LENGTH OLO/REC TURNS  COIL RES CKT: RES   CAP
DRIVER 0.7500 1.2500 0.6000 0.0600 2350.0 5.310E+02 3.050E+03 1.500E-10
PICKUP 0.3500 0.7000 0.2000 0.0000 3450.0 5.147E+03 1.000E+06 1.500E-10

RBAR 0.2500 FREQ=  5.000000E+02 RHO=  4.0540 PERM= 1.000 WUSRR=  3.9267

NEAR SIDE DEFECT:  DIAM= 0.2215, DEPTH= 0.2215
GAIN 1. DVR VOLT 1.1000 NOR DEF VOL 5.4625E-01 DVR AIR IND 6.734812E-02
```

Partial listing of file FORT39:

```
 0.359   0.212D-03  125.01
 0.368   0.221D-03  125.06
 0.377   0.230D-03  125.11
 0.386   0.239D-03  125.15
 0.395   0.248D-03  125.19
 0.404   0.256D-03  125.23
 0.413   0.265D-03  125.26
```

0.422	0.273D-03	125.28
0.431	0.282D-03	125.30
0.440	0.290D-03	125.31

<u>Listing</u>

```
      PROGRAM RFAVZSCN
C     VERSION August 8, 1989
C      Program to calculate the change in induced voltage in
C      a pickup coil due to a defect in a conducting plate.
      CHARACTER SIDE(2)*4,NPROBE*6,COIL*6
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 L3,L4,L5,L6
      COMPLEX ZMT,ZDT,ZPT,Z0,Z6,Z7,Z9,VOUT,V0,V1
      DIMENSION S1(6),S2(6),ERR(6),RJ(120)
      DIMENSION SDDR(120,40),SDDI(120,40),SDPR(120,40),SDPI(120,40)
      DATA LOU/8/,PI/3.141592653/,LOD/39/
      DATA S1/.005,.02,.05,.1,.5,2./
      DATA S2/1.0,2.0,5.0,10.0,50.0,200.0/
      DATA ERR/.1,.01,.001,1.E-4,1.E-5,1.E-6/
      DATA RHO1/4.054/,U1/1.0/,DFDIAM/0.2215/,DFDEP/0.2215/
      DATA NRT/100/,NZT/20/
      DATA T1/0.250/,NPROBE/'250A  '/,SIDE/'NEAR',' FAR'/,NS/1/
      DATA R0/3050./,R9/1.0D6/,C6/15.E-11/,C7/15.E-11/
      DATA FREQ/500./,GAIN/1./,VIN/1.100/,RSUM/0.0/,DELTR/0.005/
C     TIME AND DATE ARE PRINTED
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,2) NZT,IHR,IMN,ISE,IMO,IDA,IYR
   2  FORMAT('RFAVZSCN ',I3,' POINTS USED    TIME ',I2,':',I2,':',I2
     *,'  DATE ',I2,'/',I2,'/',I2)
      W=2.0*PI*FREQ
      FNZT=FLOAT(NZT)
      OPEN(28,FILE='REF.DAT',STATUS='OLD')
  10  READ(28,11)COIL,R5,R1,R2,L3,R3,R4,L4,L5,L6
     *,R6,R7,TNDR,TNPU
  11  FORMAT(A6,9F8.4,F10.4,F11.4,2F8.1)
      IF(COIL.EQ.'END   ')WRITE(0,*)' COIL NOT FOUND'
      IF(COIL.EQ.'END   ')GO TO 1020
      IF(COIL.NE.NPROBE)GO TO 10
      L6=L6+0.010/R5
      WRITE(LOU,3)NPROBE,T1
   3  FORMAT('PROBE ',A6,' PLATE THIK',F7.4)
      WRITE(LOU,5)
   5  FORMAT('COIL   IN RAD  OT RAD  LENGTH OLO/REC   TURNS',
     *'  COIL RES  CKT: RES        CAP')
      WRITE(LOU,14)R1,R2,L3,L6,TNDR,R6,R0,C6
      WRITE(LOU,15)R3,R4,L4,L5,TNPU,R7,R9,C7
  14  FORMAT('DRIVER ',4(F7.4,1X),F8.1,3(1PE10.3))
  15  FORMAT('PICKUP ',4(F7.4,1X),F8.1,3(1PE10.3))
      WUSRR=0.5093979*U1*R5*R5*FREQ/RHO1
      WRITE(LOU,20)R5,FREQ,RHO1,U1,WUSRR
  20  FORMAT('RBAR',F7.4,' FREQ= ',1PE13.6,' RHO-',0PF9.4,
     *' PERM=',F7.3,' WUSRR=',F9.4)
```

```
C      WRITE(LOD,23)NPROBE,T1,FREQ,SIDE(NS),DFDIAM,DFDEP
       WRITE(0,23)NPROBE,T1,FREQ,SIDE(NS),DFDIAM,DFDEP
   23  FORMAT('PROBE ',A6,' PLATE THK',F7.4,' FREQ=',1PE8.1,1X,A4,
      *' SIDE',0PF6.4,' DIA',F6.4,' DEEP')
       DELTR=(R2-R3)/NRT
       RD=R3
       ZD=-DFDEP
       TD=TNDR/((R2-R1)*L3)
       TP=TNPU/((R4-R3)*L4)
       ZDF=6.300475E-7*FREQ*TD*TD*R5
       ZPF=6.300475E-7*FREQ*TP*TP*R5
       ZMF=6.300475E-7*FREQ*TD*TP*R5
C      VOLN=0.1666667*PI*(DFDIAM(NRUN)/R5)*(DFDIAM(NRUN)/R5)*(DFDIAM/R5)
       VOLN=PI*DFDIAM*DFDIAM*DFDEP/(4.*R5*R5*R5)
       DDF=0.1193662*VOLN*WUSRR
       DPF=0.1193662*VOLN*WUSRR
       DMF=0.1193662*VOLN*WUSRR
       WRITE(LOU,18)SIDE(NS),DFDIAM,DFDEP
   18  FORMAT(A4,' SIDE DEFECT:  DIAM=',F7.4,', DEPTH=',F7.4)
       ZD=-DFDEP/R5
       T1=T1/R5
       SAIR1=0.0
       SAIR2=0.0
       SZDR=0.0
       SZDI=0.0
       SZPR=0.0
       SZPI=0.0
       SZMR=0.0
       SZMI=0.0
       DO 27 NR=1,NRT
       DO 25 NZ=1,NZT
       SDDR(NR,NZ)=0.0
       SDDI(NR,NZ)=0.0
       SDPR(NR,NZ)=0.0
   25  SDPI(NR,NZ)=0.0
   27  CONTINUE
       B1=0.0
       B2=S2(1)
       DO 100 JKL=1,6
   30  RI9=SAIR1
       X=B1-0.5*S1(JKL)
C      DETERMINE NUMBER OF STEPS,ISTEPS,BASED ON STEP SIZE&INTERVAL
       ISTEPS=DNINT((B2-B1)/S1(JKL))
       DO 95 I=1,ISTEPS
       X=X+S1(JKL)
       CALL BESSEL(XJR2,X,R2)
       CALL BESSEL(XJR1,X,R1)
       CALL BESSEL(XJR4,X,R4)
       CALL BESSEL(XJR3,X,R3)
       R21=XJR2-XJR1
       R43=XJR4-XJR3
```

```
      XL3=X*L3
      IF(XL3.GT.5.0E-3) GO TO 60
      A1=XL3*XL3*(0.5-XL3/6.0)
      GO TO 80
   60 IF(XL3.GT.75.0) GO TO 70
      A1=XL3+DEXP(-XL3)-1.0
      GO TO 80
   70 A1=XL3-1.0
   80 A3=XL3-A1
      SFD=S1(JKL)*R21*R21
      SFP=S1(JKL)*R43*R43
      SFM=S1(JKL)*R21*R43
      SAIR1=SAIR1+SFD*2.0*A1
      XL4=X*L4
      IF(XL4.GT.5.0E-3) GO TO 81
      A2=XL4*XL4*(0.5-XL4/6.0)
      GO TO 83
   81 IF(XL4.GT.75.0) GO TO 82
      A2=XL4+DEXP(-XL4)-1.0
      GO TO 83
   82 A2=XL4-1.0
   83 A4=XL4-A2
      A13=1.0-A3
      A14=1.0-A4
      A5=DEXP(-X*L5)
      XL8=X*(L3-2.*L4-2.*L5)
      IF(XL8.GT.75.)A8=0.
      IF(XL8.LE.75.)A8=DEXP(-XL8)
      SAIR2=SAIR2+SFP*(4.0*A2-2.0*A4*A4*A8)
      IF(X*L6.GT.75.)GO TO 95
C        CALCULATION OF GAMMA FACTORS
      XX=X*X
      X1=DSQRT(0.5*(XX+DSQRT(XX*XX+WUSRR*WUSRR)))/U1
      Y1=WUSRR/(2.*X1*U1*U1)
      A6=DEXP(-X*L6)
      XL7=X*(L3-L4-2.*L5)
      IF(XL7.GT.75.)A7=1.0
      IF(XL7.LE.75.)A7=1.0-DEXP(-XL7)
      A9=A4*A5*A6*A7
      APBR=(X+X1)*(X+X1)-Y1*Y1
      APBI=2.*Y1*(X+X1)
      AMBR=(X-X1)*(X-X1)-Y1*Y1
      AMBI=-2.*Y1*(X-X1)
      A2BR=0.0
      A2BI=-2.*X1*Y1
      ZNUR=A2BR
      ZNUI=A2BI
      DENR=APBR
      DENI=APBI
      DNCJ=DENR*DENR+DENI*DENI
C     SECTION THAT MULTIPLIES BY DEXP(ALPHA1*ZDEFECT)
```

```
      DO 91 NZ=1,NZT
C     NEAR SIDE DEFECT CALCULATION
      FZD=(FLOAT(NZ)-.5)*ZD/FNZT
C     FAR SIDE DEFECT CALCULATION
      IF(NS.EQ.2)FZD=-T1-FZD
      ZDR=X1*U1*FZD
      IF(ZDR.LT.-60.0)GO TO 93
      ZDI=Y1*U1*FZD
      XPDR=DEXP(ZDR)
      CSDI=DCOS(ZDI)*XPDR
      SNDI=DSIN(ZDI)*XPDR
      XX1=X*X1+XX
      XY1=X*Y1
      X1X=X*X1-XX
      ZNDR=XX1*CSDI-XY1*SNDI
      ZNDI=XX1*SNDI+XY1*CSDI
C     SECTION THAT MULTIPLIES BY DEXP(ALPHA1*(2*TH+ZDEFECT))
      TZR=X1*U1*(2.*T1+FZD)
      IF(TZR.GT.60.)GO TO 87
      TZI=Y1*U1*(2.*T1+FZD)
      XPZR=DEXP(-TZR)
      CSZI=DCOS(TZI)*XPZR
      SNZI=DSIN(TZI)*XPZR
      ZNDR=XX1*CSDI-XY1*SNDI+X1X*CSZI+XY1*SNZI
      ZNDI=XX1*SNDI+XY1*CSDI+XY1*CSZI-X1X*SNZI
C     SECTION THAT MULTIPLIES BY DEXP(-ALPHA1*2*CLADTH)
      TR=2.*X1*U1*T1
      IF(TR.GT.60.)GO TO 87
      TI=2.*Y1*U1*T1
      XPTR=DEXP(-TR)
      CSTI=DCOS(TI)*XPTR
      SNTI=DSIN(TI)*XPTR
      DENR=APBR-AMBR*CSTI-AMBI*SNTI
      DENI=APBI-AMBI*CSTI+AMBR*SNTI
      ZNUR=A2BR-A2BR*CSTI-A2BI*SNTI
      ZNUI=A2BI-A2BI*CSTI+A2BR*SNTI
      DNCJ=DENR*DENR+DENI*DENI
   87 ZDRL=(ZNDR*DENR+ZNDI*DENI)/DNCJ
      ZDIM=(DENR*ZNDI-ZNDR*DENI)/DNCJ
C     LOOP OVER THE R VARIATION FOR THE DEFECT
      DO 90 NR=1,NRT
      IF(NZ.GT.1) GO TO 89
      RD=R3+DELTR*FLOAT(NR)
      XRD=X*RD
      CALL BESEL1(XRD,RJ1)
      RJ(NR)=RJ1
C     SUM DEFECT SENSITIVITY FACTORS FOR THE DRIVER & PICKUP COILS
   89 SDDR(NR,NZ)=SDDR(NR,NZ)+A3*A6*RJ(NR)*2*ZDRL*R21*S1(JKL)
      SDDI(NR,NZ)=SDDI(NR,NZ)+A3*A6*RJ(NR)*2*ZDIM*R21*S1(JKL)
      SDPR(NR,NZ)=SDPR(NR,NZ)+A9*RJ(NR)*2*ZDRL*R43*S1(JKL)
   90 SDPI(NR,NZ)=SDPI(NR,NZ)+A9*RJ(NR)*2*ZDIM*R43*S1(JKL)
```

```
   91 CONTINUE
   93 ZRL=(ZNUR*DENR+ZNUI*DENI)/DNCJ
      ZIM=(DENR*ZNUI-ZNUR*DENI)/DNCJ
      SZDI=SZDI+A3*A3*A6*A6*ZRL*SFD
      SZDR=SZDR-A3*A3*A6*A6*ZIM*SFD
      SZPI=SZPI+A9*A9*ZRL*SFP
      SZPR=SZPR-A9*A9*ZIM*SFP
      SZMI=SZMI+A3*A6*A9*ZRL*SFM
      SZMR=SZMR-A3*A6*A9*ZIM*SFM
   95 CONTINUE
      B1=B2
      B2=B2+S2(JKL)
      CHECK=(SAIR1-RI9)/SAIR1
      IF(ABS(CHECK).GT.ERR(JKL)) GO TO 30
  100 CONTINUE
C     COMPUTATION OF DRIVER INDUCTANCE
      Q6=ZDF*SAIR1/W
      WRITE(LOU,120)GAIN,VIN,VOLN,Q6
  120 FORMAT('GAIN',F10.0,' DVR VOLT',F7.4,' NOR DEF VOL',1PE11.4,
     *' DVR AIR IND',1PE13.6)
C     DEFINE COMPLEX QUANTITIES THAT ARE CONSTANT
      Z0=DCMPLX(0.0D0,-R0)
      Z6=DCMPLX(W*C6*R0,-1.0D0)
      Z7=DCMPLX(W*C7*R9,-1.0D0)
      Z9=DCMPLX(0.0D0,-R9)
C     SCAN PAST THE DEFECT IN THE R DIRECTION
      DO 1010 NR=1,NRT
      RD=R3+DELTR*FLOAT(NR)
      D=0.
C     AVERAGE DEFECT VALUES OVER DEPTH
      ADDR=0.0
      ADDI=0.0
      ADPR=0.0
      ADPI=0.0
      DO 125 NZ=1,NZT
      ADDR=ADDR+SDDR(NR,NZ)/FNZT
      ADDI=ADDI+SDDI(NR,NZ)/FNZT
      ADPR=ADPR+SDPR(NR,NZ)/FNZT
  125 ADPI=ADPI+SDPI(NR,NZ)/FNZT
  135 ZDTR=ZDF*(SZDR-D*DDF*(ADDR*ADDR-ADDI*ADDI))
      ZDTI=ZDF*(SAIR1+SZDI-D*DDF*2.*ADDR*ADDI)
      ZPTR=ZPF*(SZPR-D*DPF*(ADPR*ADPR-ADPI*ADPI))
      ZPTI=ZPF*(SAIR2+SZPI-D*DPF*2.*ADPR*ADPI)
      ZMTR=ZMF*(SZMR-D*DMF*(ADDR*ADPR-ADDI*ADPI))
      ZMTI=ZMF*(SZMI-D*DMF*(ADDI*ADPR+ADDR*ADPI))
C     DEFINE COMPLEX QUANTITIES, DO COMPLEX CIRCUIT CALCULATIONS
      ZDT=DCMPLX(ZDTR,ZDTI)
      ZPT=DCMPLX(ZPTR,ZPTI)
      ZMT=DCMPLX(ZMTR,ZMTI)
      VOUT=(-ZMT*VIN*GAIN*R9)/
     *(Z6*Z7*ZMT*ZMT+(Z6*(ZDT+R6)+Z0)*(Z7*(ZPT+R7)+Z9))
```

```
      IF(D.EQ.0)V0=VOUT
      IF(D.EQ.1)V1=VOUT
      IF(D.EQ.1.)GO TO 1000
C     REPEAT CALCULATIONS WITH DEFECT
      D=1.0
      GO TO 135
 1000 CONTINUE
      VV0R=REAL(V0)
      VV0I=AIMAG(V0)
      VV1R=REAL(V1)
      VV1I=AIMAG(V1)
      DVVR=VV1R-VV0R
      DVVI=VV1I-VV0I
      TTSR=TTSR-DVVR*DELTR
      TTSI=TTSI-DVVI*DELTR
      TMAG=DSQRT(DVVR*DVVR+DVVI*DVVI)
      PHA=(ATAN2(DVVI,DVVR))*180./PI
C     WRITE(0,180)RD,TMAG,PHA
      WRITE(LOD,180)RD,TMAG,PHA
  180 FORMAT(F6.3,1X,D11.3,1X,F7.2)
 1010 CONTINUE
C     WRITE(LOU,*)TTSR,TTSI
      TTMAG=DSQRT(TTSR*TTSR+TTSI*TTSI)
      TTPHA=DATAN2(TTSI,TTSR)*180./PI
C     WRITE(LOU,*)'MAG ',TTMAG
C     WRITE(LOU,*)'PHA ',TTPHA
 1020 STOP 'JOB '
      END
```

RFGRAPH plots two sets of data on same graph

Program RFGRAPH plots two sets of data on the same graph and sends the output to the screen and to the printer. It is normally used to compare the calculated and experimental change in the induced voltage in a pickup coil due to a defect as the coil is scanned past the defect.

Summary

1. Dimension arrays and declare variable types.
2. Initialize variables.
3. Open the data files containing the input data.
4. Read in and scale the data.
5. Plot the data to the screen.
6. Send the contents of the screen to the printer.

Variables

C6*	The total shunt capacitance in farads of the driving circuit.
C7*	The total shunt capacitance in farads of the pickup circuit.
CGIM	The factor by which the imaginary parts of the calculated data points can be multiplied to make the graph as large as possible on the screen.
CGRL	The factor by which the real parts of the calculated data points can be multiplied to make the graph as large as possible on the screen.
CMMAX	The maximum value of the magnitude of the calculated data points.
CPMMAX	The value of the phase for the experimental data point which has the largest magnitude.
CXMAX	The maximum value of the real parts of the calculated data points.
CYMAX	The maximum value of the imaginary parts of the calculated data points.
DFDEP*	The depth to the bottom of the defect in inches.
DFDIAM*	The diameter of the defect in inches.
EGIM	The factor by which the imaginary parts of the experimental data points can be multiplied to make the graph as large as possible on the screen.
EGRL	The factor by which the real parts of the experimental data points can be multiplied to make the graph as large as possible on the screen.
EMMAX	The maximum value of the magnitude of the experimental data points.
EPMMAX	The value of the phase for the experimental data point which has the largest magnitude.
EXMAX	The maximum value of the real parts of the experimental data points.

EYMAX	The maximum value of the imaginary parts of the experimental data points.
FREQ*	The operating frequency in hertz.
GAIN*	Gain of pickup amplifier.
GIM	The factor by which the imaginary parts of both sets of data are multiplied to obtain the largest possible graph.
GRL	The factor by which the real parts of both sets of data are multiplied to obtain the largest possible graph.
L3	The normalized length of the driver coil.
L4	The normalized length of the pickup coil.
L5	The normalized distance of recess of the pickup coil.
L6	The normalized lift-off of the driver coil.
LOEC*	The number of the I/O unit connected to the file containing the calculated data.
LOEE*	The number of the I/O unit connected to the file containing the experimental data.
LOU	The number of the I/O unit connected to the printer.
MODE	The screen mode to be used. Mode 16 is the EGA high resolution mode.
NPROBE*	Character variable which contains the name of the reflection probe which is to be used in the calculations.
NS*	The side of the plate where the defect is located. If NS = 1, the defect is on the near side; if NS = 2, the defect is on the far side.
OIM	The number which is added to the imaginary parts of all data points to move the origin to the desired location.
ORL	The number which is added to the real parts of all data points to move the origin to the desired location.
R0*	The output series resistance of the driving amplifier in ohms.
R1	The normalized inner radius of the driver coil.
R2	The normalized outer radius of the driver coil.
R3	The normalized inner radius of the pickup coil.
R4	The normalized outer radius of the pickup coil.
R5	The mean radius of the driver coil in inches.
R6	DC resistance of the driver coil in ohms.
R7	DC resistance of the pickup coil in ohms.
R9*	The input shunt resistance of pickup amplifier in ohms.
RHO1*	The resistivity of the plate in $\mu\Omega$-cm.
T1*	The thickness of the plate. It is input in inches and normalized by the program.
TNDR	The number of turns in the driver coil.
TNPU	The number of turns in the pickup coil.

U1'	The relative magnetic permeability of the plate.
VIN'	Output voltage of driving amplifier in volts.
WUSRR	The product of the angular frequency, the magnetic permeability, the electrical conductivity, and the square of the mean coil radius.

Listing

```
      PROGRAM RFGRAPH
C      VERSION November 10, 1988
C      Program to graph experimental and calculated reflection
C      coil data.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 GRL,GIM,L3,L4,L5,L6
      REAL CX(200),CY(200),EX(200),EY(200)
      CHARACTER SIDE(2)*4,NPROBE*6,COIL*6
      CHARACTER*1 FF
      DATA LOU/8/,PI/3.141592653/,LOEE/38/,LOEC/39/
      DATA RHO1/4.054/,U1/1.0/,DFDIAM/0.2215/,DFDEP/0.2215/,NS/1/
      DATA T1/0.250/,NPROBE/'250A  '/,SIDE/'NEAR',' FAR'/
      DATA R0/3050./,R9/1.0D6/,C6/15.E-11/,C7/15.E-11/
      DATA FREQ/500./,GAIN/1./,VIN/1.100/
      DATA MODE/16/,ORL/330/,OIM/50/
      FF=CHAR(12)
C     TIME AND DATE ARE PRINTED
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,102) IHR,IMN,ISE,IMO,IDA,IYR
  102 FORMAT('RFGRAPH     TIME ',I2,':',I2,':',I2
     *,'  DATE ',I2,'/',I2,'/',I2)
      OPEN(28,FILE='REF.DAT',STATUS='OLD')
  110 READ(28,111)COIL,R5,R1,R2,L3,R3,R4,L4,L5,L6
     *,R6,R7,TNDR,TNPU
  111 FORMAT(A6,9F8.4,F10.4,F11.4,2F8.1)
      IF(COIL.EQ.'END   ')WRITE(0,*)' COIL NOT FOUND'
      IF(COIL.EQ.'END   ')GO TO 1000
      IF(COIL.NE.NPROBE)GO TO 110
      L6=L6+0.010/R5
      WRITE(LOU,103)NPROBE,T1
  103 FORMAT('PROBE ',A6,' PLATE THICKNESS',F7.4)
      WRITE(LOU,105)
  105 FORMAT('COIL   IN RAD  OT RAD  LENGTH OLO/REC   TURNS',
     *'  COIL RES  CKT: RES       CAP')
      WRITE(LOU,114)R1,R2,L3,L6,TNDR,R6,R0,C6
      WRITE(LOU,115)R3,R4,L4,L5,TNPU,R7,R9,C7
  114 FORMAT('DRIVER ',4(F7.4,1X),F8.1,3(1PE10.3))
  115 FORMAT('PICKUP ',4(F7.4,1X),F8.1,3(1PE10.3))
      WUSRR=0.5093979*U1*R5*R5*FREQ/RHO1
      WRITE(LOU,120)R5,FREQ,RHO1,U1,WUSRR
  120 FORMAT('RBAR',F7.4,' FREQ= ',1PE13.6,' RHO=',0PF9.4,
     *' PERM=',F7.3,' WUSRR=',F9.4)
C  Open the files containing the data.
      OPEN(LOEE,FILE='RFF3EX2.500',STATUS='OLD')
      OPEN(LOEC,FILE='RFF3CAL.500',STATUS='OLD')
      CALL QSMODE(MODE)
      CALL GRID
```

```
11 CXMAX=0.
   CYMAX=0.
   EXMAX=0.
   EYMAX=0.
   EMMAX=0.
   CMMAX=0.
   I=1
20 READ(LOEE,*,END=29)EMAG,EPHA
   EPHA=EPHA*PI/180.
   EX(I)=EMAG*COS(EPHA)
   EY(I)=EMAG*SIN(EPHA)
   IF(ABS(EX(I)).GT.EXMAX)EXMAX=ABS(EX(I))
   IF(ABS(EY(I)).GT.EYMAX)EYMAX=ABS(EY(I))
   IF(EMAG.GT.EMMAX) THEN
   EMMAX=EMAG
   EPMMAX=EPHA
   END IF
   I=I+1
   GO TO 20
29 EX(I)=999.
   I=1
30 READ(LOEC,*,END=40)XTMP,CMAG,CPHA
   CPHA=CPHA*PI/180.
   CX(I)=CMAG*COS(CPHA)
   CY(I)=CMAG*SIN(CPHA)
   IF(ABS(CX(I)).GT.CXMAX)CXMAX=ABS(CX(I))
   IF(ABS(CY(I)).GT.CYMAX)CYMAX=ABS(CY(I))
   IF(CMAG.GT.CMMAX) THEN
   CMMAX=CMAG
   CPMMAX=CPHA
   END IF
   I=I+1
   GO TO 30
40 CX(I)=999.
   EGIM=300./EXMAX
   EGRL=300./EYMAX
   IF(EGIM.GT.EGRL) THEN
   EGIM=EGRL
   ELSE
   EGRL=EGIM
   END IF
   CGIM=300./CXMAX
   CGRL=300./CYMAX
   IF(CGIM.GT.CGRL) THEN
   CGIM=CGRL
   ELSE
   CGRL=CGIM
   END IF
   IF(EGIM.GT.CGIM) THEN
   GIM=CGIM
   GRL=CGRL
```

```
      ELSE
      GIM=EGIM
      GRL=EGRL
      END IF
      EPMMAX=EPMMAX*180./PI
      CPMMAX=CPMMAX*180./PI
      WRITE(LOU,*)
      WRITE(LOU,*)'           MAX MAG      PHA AT MAX MAG'
      WRITE(LOU,52)EMMAX,EPMMAX
      WRITE(LOU,53)CMMAX,CPMMAX
   52 FORMAT(' EXP:   ',D11.4,7X,F7.3)
   53 FORMAT(' CAL:   ',D11.4,7X,F7.3)
      IR1=GRL*EX(1)+ORL
      IM1=GIM*EY(1)+OIM
      I=2
   50 IF(EX(I).GT.900) GO TO 59
      IR2=GRL*EX(I)+ORL
      IM2=GIM*EY(I)+OIM
      IF((IR1.GT.0).AND.(IR1.LT.600)) THEN
      IF((IR2.GT.0).AND.(IR2.LT.600)) THEN
      IF((IM1.GT.0).AND.(IM1.LT.350)) THEN
      IF((IM2.GT.0).AND.(IM2.LT.350)) THEN
      CALL QLINE(IR1,IM1,IR2,IM2,15)
      END IF
      END IF
      END IF
      END IF
      IR1=IR2
      IM1=IM2
      I=I+1
      GO TO 50
   59 IR1=GRL*CX(1)+ORL
      IM1=GIM*CY(1)+OIM
      I=2
   60 IF(CX(I).GT.900) GO TO 70
      IR2=GRL*CX(I)+ORL
      IM2=GIM*CY(I)+OIM
      CALL QLINE(IR1,IM1,IR2,IM2,11)
      IR1=IR2
      IM1=IM2
      I=I+1
      GO TO 60
   70 CONTINUE
      WRITE(LOU,*)
      WRITE(LOU,*)
      WRITE(LOU,*)
      CALL PRTSC
      WRITE(LOU,*)FF
 1000 END
```

RFINV inverts scan of defect using reflection coil data

Program RFINV calculates the depth and volume of a defect given the change in induced voltage in a pickup coil caused by the defect. The program works with either experimental data stored in a file by program RFFIX or with calculated data stored by program RFAVZSCN, and it uses a lookup file built by program RFBLDF. The program calculates the integral of the change in induced voltage in the pickup coil due to the presence of a defect with respect to the radial distance between the coil axis and the center of the defect from the inner radius of the pickup coil to the outer radius of the driver coil and then finds the magnitude and phase of this complex integral. It opens a lookup file which contains a list of phases along with the magnitude and depth corresponding to these phases. When the program finds in the lookup file the phase it calculated when doing the integral, the depth corresponding to this phase is the depth of the defect, and the magnitude in the lookup file can be used with the magnitude obtained by doing the integral to find the volume of the defect.

<u>Summary</u>

1. Declare variable types.
2. Initialize variables.
3. Open the file containing the experimental data, read in the data, and calculate the integral.
4. Find the depth and magnitude in the lookup file corresponding to the phase of the integral.
5. Calculate the inverted defect depth and volume based on the experimental data.
6. Open the file containing the calculated data, read in the data, and calculate the integral.
7. Find the depth and magnitude in the lookup file corresponding to the phase of the integral.
8. Calculate the inverted defect depth and volume based on the calculated data.

<u>Variables</u>

Starred variables must be set by the user.

C6*	The total shunt capacitance in farads of the driving circuit.
C7*	The total shunt capacitance in farads of the pickup circuit.
DELRDC*	The normalized distance in the radial direction between data points from the calculated data file.
DELRDE*	The normalized distance in the radial direction between data points from the experimental data file.
DELTR*	The radial distance in inches between points in the experimental data file.

DEPTH	The inverted normalized depth of the center of the defect.
DFDEP*	The actual depth to the bottom of the defect in inches.
DFDIAM*	The actual diameter of the defect in inches.
DFM	The magnitude of the change in induced voltage in the pickup coil at a point due to the defect.
DFP	The phase of the change in induced voltage in the pickup coil at a point due to the defect.
FREQ*	The frequency in hertz at which the circuit is driven.
L3	The normalized length of the driver coil.
L4	The normalized length of the pickup coil.
L5	The normalized distance of recess of the pickup coil.
L6	The normalized lift-off of the driver coil.
LHSMAG	The magnitude of the integral calculated by the program.
LHSPHA	The phase of the integral calculated by the program.
LOEC*	The channel on which the file containing the calculated data is opened.
LOEE*	The channel on which the file containing the experimental data is opened.
NPROBE*	Character variable containing the name of the reflection coil being used.
NS*	The side of the plate where the defect is located. If NS=1, the defect is on the near side; if NS=2, the defect is on the far side.
R0*	The series resistance of the driver circuit.
R1	The normalized inner radius of the driver coil.
R2	The normalized outer radius of the driver coil.
R3	The normalized inner radius of the pickup coil.
R4	The normalized outer radius of the pickup coil.
R5	The mean radius of the driver coil in inches.
R6	DC resistance of the driver coil in ohms.
R7	DC resistance of the pickup coil in ohms.
R9*	The amplifier input impedance.
RDC	The normalized radial distance between the coil and defect for the calculated inversion section of the program.
RDE	The normalized radial distance between the coil and defect for the experimental inversion section of the program.
RHO1*	The resistivity of the plate in $\mu\Omega$-cm.
RHSMAG	The magnitude in the lookup file corresponding to the phase of the integral calculated by the program.
T1*	The thickness of the plate. It is input in inches and then normalized by the program.
TNDR	The number of turns in the driver coil.

TNPU	The number of turns in each pickup coil.
U1'	The relative magnetic permeability of the plate.
VOL1	The inverted normalized volume of the defect.
VOLN	The actual normalized volume of the defect.
ZD	The actual normalized depth of the center of the defect.

Notes

1. If the user wants the program to perform only the inversion of calculated data, he should remove the 'c' which comments out the statement 'goto 79' just before statement number 70 in the program. If the user wants the program to perform only the inversion of experimental data, he should remove the 'c' which comments out the statement 'goto 89' just before statement number 80 in the program.

Listing

```
      PROGRAM RFINV
C     VERSION November 7, 1988
      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER SIDE(2)*4,NPROBE*6,COIL*6
      REAL*8 L,L1,L2,LHSPHA
      REAL*8 L3,L4,L5,L6
      DATA LOEE/38/,LOEC/39/,LOU/8/,PI/3.141592653/
      DATA RHO1/5.72/,U1/1.0/
      DATA DFDIAM/0.0765/,DFDEP/0.0780/,NS/1/
      DATA NPROBE/'250A  '/,SIDE/'NEAR',' FAR'/
      DATA R0/3050./,R9/1.0D6/,C6/8.47E-11/,C7/8.45E-11/
      DATA FREQ/1.0E3/,DELTR/0.005/
      DATA T1/0.2500/
      OPEN(LOEE,FILE='RFN6EX2.500',STATUS='OLD')
      OPEN(LOEC,FILE='RFN6CAL.500',STATUS='OLD')
C     TIME AND DATE ARE PRINTED
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT(' RFINV      TIME ',I2,':',I2,':',I2
     *,'  DATE ',I2,'/',I2,'/',I2)
      OPEN(28,FILE='REF.DAT',STATUS='OLD')
   10 READ(28,11)COIL,R5,R1,R2,L3,R3,R4,L4,L5,L6
     *,R6,R7,TNDR,TNPU
   11 FORMAT(A6,9F8.4,F10.4,F11.4,2F8.1)
      IF(COIL.EQ.'END   ')WRITE(0,*)' COIL NOT FOUND'
      IF(COIL.EQ.'END   ')GO TO 89
      IF(COIL.NE.NPROBE)GO TO 10
      L6=L6+0.010/R5
      WRITE(LOU,3)NPROBE,T1
    3 FORMAT('PROBE ',A6,' PLATE THICKNESS',F7.4)
      WRITE(LOU,5)
    5 FORMAT('COIL   IN RAD  OT RAD  LENGTH OLO/REC   TURNS',
     *'  COIL RES  CKT: RES       CAP')
      WRITE(LOU,14)R1,R2,L3,L6,TNDR,R6,R0,C6
      WRITE(LOU,15)R3,R4,L4,L5,TNPU,R7,R9,C7
   14 FORMAT('DRIVER ',4(F7.4,1X),F8.1,3(1PE10.3))
   15 FORMAT('PICKUP ',4(F7.4,1X),F8.1,3(1PE10.3))
      WUSRR=0.5093979*U1*R5*R5*FREQ/RHO1
      WRITE(LOU,20)R5,FREQ,RHO1,U1,WUSRR
   20 FORMAT('RBAR',F7.4,' FREQ= ',1PE13.6,' RHO=',0PF9.4,
     *' PERM=',F7.3,' WUSRR=',F9.4)
      WRITE(LOD,23)NPROBE,T1,FREQ,SIDE(NS),DFDIAM,DFDEP
      WRITE(0,23)NPROBE,T1,FREQ,SIDE(NS),DFDIAM,DFDEP
   23 FORMAT('PROBE ',A6,' PLATE THK',F7.4,' FREQ=',1PE8.1,1X,A4,
     *' SIDE',0PF6.4,' DIA',F6.4,' DEEP')
      DELRDC=0.01*(R2-R3)
      DELRDE=DELTR/R5
```

```
      RDE=0.
      T1=T1/R5
      ZD=-DFDEP/(2*R5)
      IF(NS.EQ.2) ZD=-T1-ZD
      RDE=RDE/R5
      L2=L+L1
      VOLN=PI*DFDIAM*DFDIAM*DFDEP/(4.*R5*R5*R5)
      WUSRR=0.5093979*U1*R5*R5*FREQ/RHO1
      WRITE(LOU,*)
      WRITE(LOU,*)
      WRITE(LOU,26)
      WRITE(LOU,27)ZD,VOLN
   26 FORMAT('                DEPTH    VOLUME')
   27 FORMAT('  Actual:  ',F8.5,2X,F8.5)
   28 CONTINUE
      SIVR=0.0
      SIVI=0.0
      M=0
c     GO TO 79
   70 READ(LOEE,*,END=78)DFM,DFP
      IF(RDE.LT.R3)GOTO 76
      IF(RDE.GT.R2)GOTO 78
      DFP=DFP*(PI/180.)
      XFACT=DFM*DELRDE
      SIVI=SIVI-XFACT*DSIN(DFP)
      SIVR=SIVR-XFACT*DCOS(DFP)
      LHSPHA=ATAN2(SIVI,SIVR)
      XMAG=SQRT(SIVR*SIVR+SIVI*SIVI)
   76 RDE=RDE+DELRDE
      GO TO 70
   77 FORMAT('  Exp Inv: ',F8.5,2X,F8.5)
   78 CONTINUE
      LHSPHA=LHSPHA*180./PI
      CALL RFLKUP(DEPTH,RHSMAG,LHSPHA)
      VOL1=XMAG/RHSMAG
      WRITE(LOU,77)DEPTH,VOL1
   79 SIVR=0.0
      SIVI=0.0
      M=0
c     GO TO 89
   80 READ(LOEC,*,END=88)RDC,DFM,DFP
      IF(RDC.LT.R3)GOTO 86
      IF(RDC.GT.R2)GOTO 88
      DFP=DFP*(PI/180.)
      M=M+1
      XFACT=DFM*DELRDC
      SIVI=SIVI-XFACT*DSIN(DFP)
      SIVR=SIVR-XFACT*DCOS(DFP)
      LHSPHA=ATAN2(SIVI,SIVR)
      XMAG=SQRT(SIVR*SIVR+SIVI*SIVI)
   86 GO TO 80
```

```
87 FORMAT('  Cal Inv: ',F8.5,2X,F8.5)
88 CONTINUE
   LHSPHA=LHSPHA*180./PI
   CALL RFLKUP(DEPTH,RHSMAG,LHSPHA)
   VOL1=XMAG/RHSMAG
   WRITE(LOU,87)DEPTH,VOL1
89 END
```

CIRCUMFERENTIAL BORESIDE COIL PROGRAMS

The programs in this section perform functions relating to the effect of a defect in a single conducting tube on a circumferential coil. The types of circumferential coils dealt with are absolute boreside and differential boreside. Fig. 13 shows a differential probe in a tube.

ORNL-DWG 90-8093

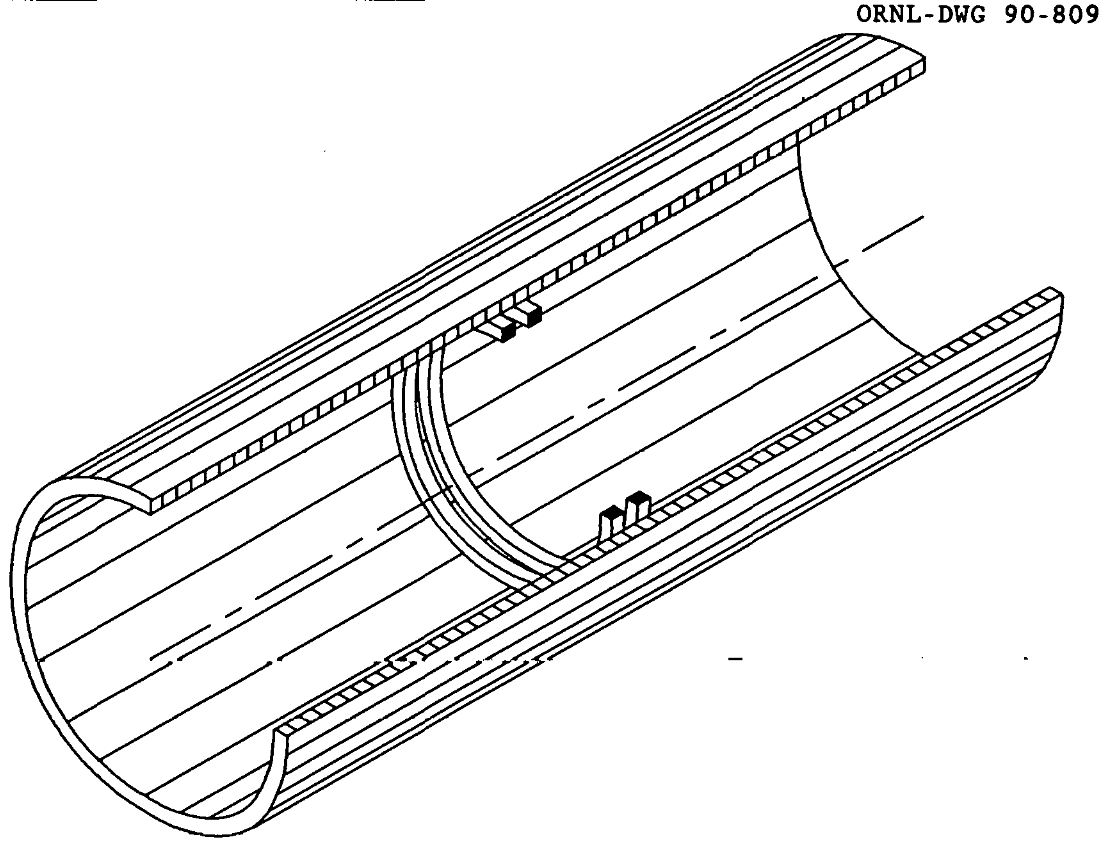

Fig. 13. Differential probe in the bore of a tube.

Calculations of the normalized impedance change in these coils due to a defect in a single conducting tube are done by programs ABBORAR and DFBORAR. The other pair of programs in this section, DBDSF and DBDSFPLT, work together to calculate and plot the contours of the magnitude of the defect sensitivity factor of a differential boreside coil.

ABBORAR calculates defect impedance change for absolute coil

The program ABBORAR calculates the change in the normalized impedance of an absolute boreside coil due to the presence of a defect in a single conducting tube, as shown in Fig. 14. It performs the calculations for a number of different axial distances between the center of the coil and the defect, and it can divide the defect into a number of parts and calculate the effect of each part separately.

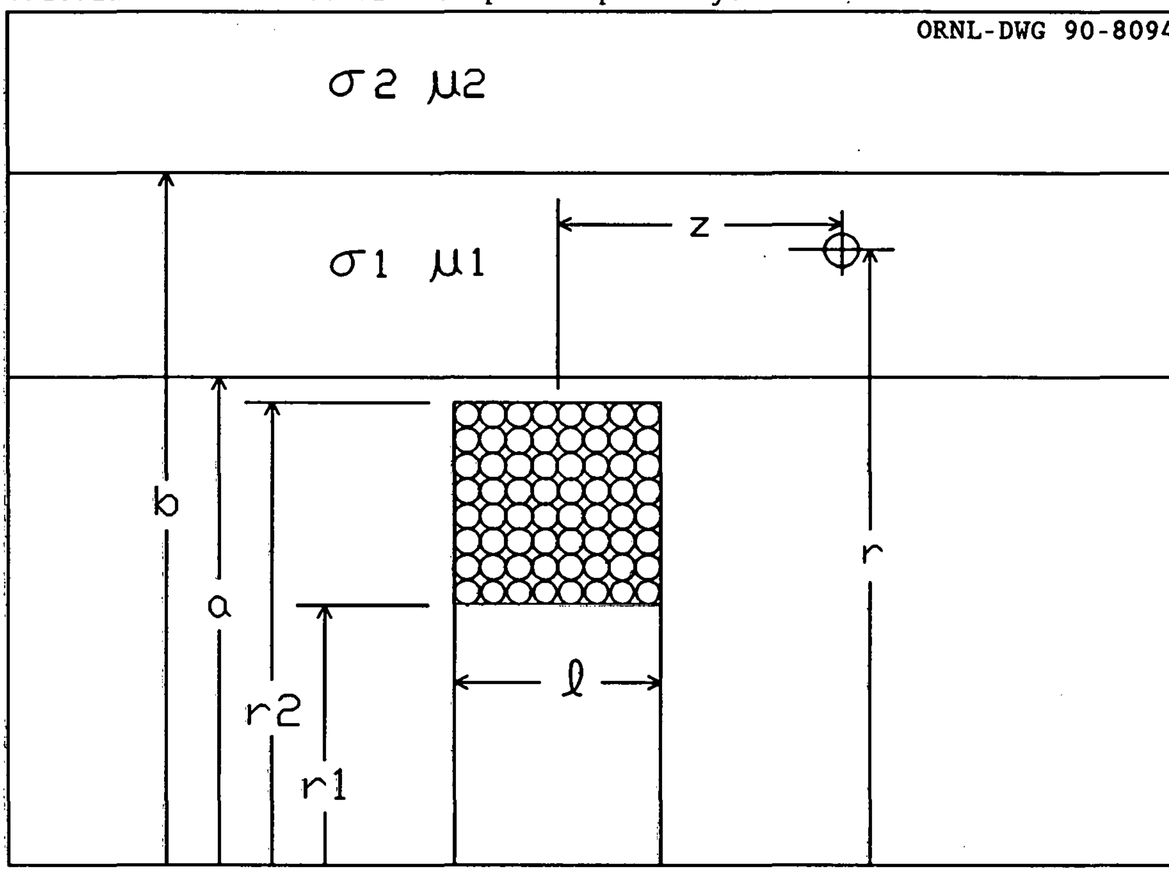

Fig. 14. Cross section of a coil in the bore of a tube with a defect present.

The normalized impedance for a coil inside a cylindrical conductor, with no defect present, is:

$$Z_n = \frac{j}{I_{air}} \int_0^\infty \left[\frac{8}{\pi\alpha^6} I^2(r_2, r_1) \left\{ \frac{K_1(\alpha_1 a)D_4 - I_1(\alpha_1 a)D_1}{I_1(\alpha a)(D_1D_2 + D_3D_4)} - \frac{K_1(\alpha a)}{I_1(\alpha a)} \right\} \sin^2\left(\frac{\alpha\ell}{2}\right) + I_{air} \right] d\alpha \quad (28)$$

and for the change in the normalized impedance due to the defect we have:

$$Z_{nd}(r,z) = \frac{-3(\omega\mu\sigma_1\bar{r}^2)Vol_n}{2\pi I_{air}} \left[\int_0^\infty \frac{I(r_2, r_1)}{\pi\alpha^3} \left\{ \frac{K_1(\alpha_1 r)D_4 - I_1(\alpha_1 r)D_1}{(D_1D_2 + D_3D_4)} \right\} \sin\left(\frac{\alpha\ell}{2}\right) 2\cos(\alpha z)\, d\alpha \right]^2$$

(29)

where:

$$D_1 = \beta_2 bK_0(\alpha_2 b)K_1(\alpha_1 b) - \beta_1 bK_0(\alpha_1 b)K_1(\alpha_2 b) \quad (30)$$

$$D_2 = \beta_1 aI_0(\alpha_1 a)I_1(\alpha a) - \alpha aI_0(\alpha a)I_1(\alpha_1 a) \quad (31)$$

$$D_3 = \alpha aI_0(\alpha a)K_1(\alpha_1 a) + \beta_1 aK_0(\alpha_1 a)I_1(\alpha a) \quad (32)$$

$$D_4 = \beta_1 bI_0(\alpha_1 b)K_1(\alpha_2 b) + \beta_2 bK_0(\alpha_2 b)I_1(\alpha_1 b) \quad (33)$$

$$I(r_2, r_1) = \int_{\alpha r_1}^{\alpha r_2} xI_1(x)dx \quad (34)$$

$$\alpha_i = (\alpha^2 + j\omega\mu\sigma_i\bar{r}^2)^{1/2} \quad (35)$$

and $$\beta_i = (\mu_0/\mu_i)(\alpha^2 + j\omega\mu_i\sigma_i\bar{r}^2)^{1/2} \quad (36)$$

The term I_{air} is related to the air inductance of the coil and is:

$$I_{air} = \int_0^\infty \frac{1}{\alpha^6} [J(r_2, r_1)]^2 \, 2[\alpha\ell + \exp(-\alpha\ell) - 1] \, d\alpha \quad (37)$$

where $$J(r_2, r_1) = \int_{\alpha r_1}^{\alpha r_2} xJ_1(x)dx \quad (38)$$

In the computer program, the outer conductor is taken to be air and given a conductivity of zero and a relative permeability of unity. The term β_2 therefore reduces to α. The term is carried as β_2 in the derivation and equations for completeness.

<u>Variables</u>

A number of the variables which occur mainly inside the integration loops do not correspond to anything physical. These variables are described in a section at the end. Starred variables must be assigned by the user.

A	The normalized inner radius of the tube.
B	The normalized outer radius of the tube.
DELTAZ*	The normalized axial distance between the points at which the calculations are done.
DFDEP*	The depth of the defect in the tube in inches.
DFDIAM*	The diameter of the defect in the tube in inches.
DFM	The magnitude of the normalized impedance change in the coil due to the defect.
DFP	The phase of the normalized impedance change in

	the coil due to the defect.
DSFI	The imaginary part of the defect sensitivity factor of the coil.
DSFR	The real part of the defect sensitivity factor of the coil.
FREQ*	The operating frequency in hertz.
ISIDE*	The side of the tube where the defect is located. If ISIDE = 1, the defect is on the outside of the tube; if ISIDE = 2, the defect is on the inside of the tube.
L*	The length of the coil. The value is input in inches and normalized by the program.
LOD*	The number of the I/O unit connected to the output data file.
LOE*	The number of the I/O unit connected to a file which contains the diameters and depths of the defects for which calculations are to be performed.
LOU	The number of the I/O unit connected to the printer.
NRT*	The total number of parts centered along the axis of the defect into which the defect is divided to perform the calculations.
NZT*	The total number of different values of the axial distance between the center of the coil and the defect for which calculations are performed.
POW	The depth of the defect expressed as a percentage of wall thickness.
Q6	The inductance in henries of the coil in air.
R1*	The inner radius of the coil. The value is input in inches and normalized by the program.
R2*	The outer radius of the coil. The value is input in inches and normalized by the program.
R3	The mean radius of the coil in inches.
RCL*	The distance between the outside of the coil and the inside wall of the tube. The value is input in inches and normalized by the program.
RD	The radial distance between the center of the coil and the defect (see note 1).
RDT	The normalized depth to the bottom of the defect. A negative number.
RHO1*	The electrical resistivity of the tube in $\mu\Omega$-cm.
T1*	The thickness of the tube wall. The value is input in inches and normalized by the program.
TRN*	The number of turns in the coil.
U1*	The relative magnetic permeability of the tube.
VOLN	The normalized volume of the defect.
WUSRR	The product of the angular operating frequency, the magnetic permeability of the tube, the electrical conductivity of the tube, and the square of the mean radius of the coil.

ZD	The axial distance between the center of the coil and the defect.
ZNDFI	The imaginary part of the normalized impedance change in the coil due to the defect.
ZNDFR	The real part of the normalized impedance change in the coil due to the defect.
ZNIM	The imaginary part of the normalized impedance of the coil when no defects are present.
ZNRL	The real part of the normalized impedance of the coil when no defects are present.

Integration Section of Program ABBORAR

Symbol definitions

The following are definitions of the symbols used to describe the program variables which appear in the integration section of the program. All lengths are normalized unless otherwise noted.

α	Integration variable
a	Inner radius of the tube
β_1	$(\alpha^2+j\omega\mu\sigma_1\overline{r}^2)^{1/2}\mu$
b	Outer radius of the tube
$I(x_2,x_1)$	Integral of $xI_1(x)$ with respect to x from αx_1 to αx_2
$I_0(x)$	Modified Bessel function of the first kind of order 0
$I_1(x)$	Modified Bessel function of the first kind of order 1
$J(x_2,x_1)$	Integral of $xJ_1(x)$ with respect to x from αx_1 to αx_2
$J_1(x)$	Bessel function of the first kind of order 1
$K_0(x)$	Modified Bessel function of the second kind of order 0
$K_1(x)$	Modified Bessel function of the second kind of order 1
ℓ	Length of the coil
μ	Relative magnetic permeability of the tube
σ_1	Electrical conductivity of the tube
r	Radial distance between center of coil and defect
$\overline{r}$	Coil mean radius in inches
r_1	Inner radius of coil
r_2	Outer radius of coil
z	Axial distance between defect and coil center
ω	Angular operating frequency

Variables appearing in the integration section

Program variable	Symbolic equivalent
A1	$\alpha\ell + \exp(-\alpha\ell) - 1$
BI0A	$I_0(\alpha a)$
BI0B	$I_0(\alpha b)$
BI1A	$I_1(\alpha a)$
BI1B	$I_1(\alpha b)$
BI1IA	$\mathrm{Im}[I_1(\alpha_1 a)]$
BI1IB	$\mathrm{Im}[I_1(\alpha_1 b)]$
BI1IR	$\mathrm{Im}[I_1(\alpha_1 r)]$
BI1RA	$\mathrm{Re}[I_1(\alpha_1 a)]$
BI1RB	$\mathrm{Re}[I_1(\alpha_1 b)]$
BI1RR	$\mathrm{Re}[I_1(\alpha_1 r)]$
BK0A	$K_0(\alpha a)$

BK0B	$K_0(\alpha b)$
BK1A	$K_1(\alpha a)$
BK1B	$K_1(\alpha b)$
BK1IA	$\mathrm{Im}[K_1(\alpha_1 a)]$
BK1IB	$\mathrm{Im}[K_1(\alpha_1 b)]$
BK1IR	$\mathrm{Im}[K_1(\alpha_1 r)]$
BK1RA	$\mathrm{Re}[K_1(\alpha_1 a)]$
BK1RB	$\mathrm{Re}[K_1(\alpha_1 b)]$
BK1RR	$\mathrm{Re}[K_1(\alpha_1 r)]$

DDI

$$\mathrm{Im}\{[\beta_2 bK_0(\alpha_2 b)K_1(\alpha_1 b)-\beta_1 bK_0(\alpha_1 b)K_1(\alpha_2 b)] [\beta_1 aI_0(\alpha_1 a)I_1(\alpha a)-\alpha aI_0(\alpha a)I_1(\alpha_1 a)] + [\alpha aI_0(\alpha a)K_1(\alpha_1 a)+\beta_1 aK_0(\alpha_1 a)I_1(\alpha a)] [\beta_1 bI_0(\alpha_1 b)K_1(\alpha_2 b)+\beta_2 bK_0(\alpha_2 b)I_1(\alpha_1 b)]\}$$

DDR

$$\mathrm{Re}\{[\beta_2 bK_0(\alpha_2 b)K_1(\alpha_1 b)-\beta_1 bK_0(\alpha_1 b)K_1(\alpha_2 b)] [\beta_1 aI_0(\alpha_1 a)I_1(\alpha a)-\alpha aI_0(\alpha a)I_1(\alpha_1 a)] + [\alpha aI_0(\alpha a)K_1(\alpha_1 a)+\beta_1 aK_0(\alpha_1 a)I_1(\alpha a)] [\beta_1 bI_0(\alpha_1 b)K_1(\alpha_2 b)+\beta_2 bK_0(\alpha_2 b)I_1(\alpha_1 b)]\}$$

DFR

$$\mathrm{Re}\ \{\ K_1(\alpha_1 r)[\beta_1 bI_0(\alpha_1 b)K_1(\alpha_2 b)+\beta_2 bK_0(\alpha_2 b)I_1(\alpha_1 b)] - I_1(\alpha_1 r)[\beta_2 bK_0(\alpha_2 b)K_1(\alpha_1 b)-\beta_1 bK_0(\alpha_1 b)K_1(\alpha_2 b)]\} \div \{[\beta_2 bK_0(\alpha_2 b)K_1(\alpha_1 b)-\beta_1 bK_0(\alpha_1 b)K_1(\alpha_2 b)] [\beta_1 aI_0(\alpha_1 a)I_1(\alpha a)-\alpha aI_0(\alpha a)I_1(\alpha_1 a)] + [\alpha aI_0(\alpha a)K_1(\alpha_1 a)+\beta_1 aK_0(\alpha_1 a)I_1(\alpha a)] [\beta_1 bI_0(\alpha_1 b)K_1(\alpha_2 b)+\beta_2 bK_0(\alpha_2 b)I_1(\alpha_1 b)]\}$$

DI1	$\mathrm{Im}\ [\beta_2 bK_0(\alpha_2 b)K_1(\alpha_1 b)-\beta_1 bK_0(\alpha_1 b)K_1(\alpha_2 b)]$
DI2	$\mathrm{Im}\ [\beta_1 aI_0(\alpha_1 a)I_1(\alpha a)-\alpha aI_0(\alpha a)I_1(\alpha_1 a)]$
DI3	$\mathrm{Im}\ [\alpha aI_0(\alpha a)K_1(\alpha_1 a)+\beta_1 aK_0(\alpha_1 a)I_1(\alpha a)]$
DI4	$\mathrm{Im}\ [\beta_1 bI_0(\alpha_1 b)K_1(\alpha_2 b)+\beta_2 bK_0(\alpha_2 b)I_1(\alpha_1 b)]$
DNI	$\mathrm{Im}\ \{\ K_1(\alpha_1 r)[\beta_1 bI_0(\alpha_1 b)K_1(\alpha_2 b)+\beta_2 bK_0(\alpha_2 b)I_1(\alpha_1 b)] - I_1(\alpha_1 r)[\beta_2 bK_0(\alpha_2 b)K_1(\alpha_1 b)-\beta_1 bK_0(\alpha_1 b)K_1(\alpha_2 b)]\}$
DNR	$\mathrm{Re}\ \{\ K_1(\alpha_1 r)[\beta_1 bI_0(\alpha_1 b)K_1(\alpha_2 b)+\beta_2 bK_0(\alpha_2 b)I_1(\alpha_1 b)] - I_1(\alpha_1 r)[\beta_2 bK_0(\alpha_2 b)K_1(\alpha_1 b)-\beta_1 bK_0(\alpha_1 b)K_1(\alpha_2 b)]\}$

DR1	Re $[\beta_2 b K_0(\alpha_2 b) K_1(\alpha_1 b) - \beta_1 b K_0(\alpha_1 b) K_1(\alpha_2 b)]$
DR2	Re $[\beta_1 a I_0(\alpha_1 a) I_1(\alpha a) - \alpha a I_0(\alpha a) I_1(\alpha_1 a)]$
DR3	Re $[\alpha a I_0(\alpha a) K_1(\alpha_1 a) + \beta_1 a K_0(\alpha_1 a) I_1(\alpha a)]$
DR4	Re $[\beta_1 b I_0(\alpha_1 b) K_1(\alpha_2 b) + \beta_2 b K_0(\alpha_2 b) I_1(\alpha_1 b)]$
S1	$d\alpha$
SNI	Im$\{-I_1(\alpha_1 a)[\beta_2 b K_0(\alpha_2 b) K_1(\alpha_1 b) - \beta_1 b K_0(\alpha_1 b) K_1(\alpha_2 b)] + K_1(\alpha_1 a)[\beta_1 b I_0(\alpha_1 b) K_1(\alpha_2 b) + \beta_2 b K_0(\alpha_2 b) I_1(\alpha_1 b)]\}$
SNR	Re$\{-I_1(\alpha_1 a)[\beta_2 b K_0(\alpha_2 b) K_1(\alpha_1 b) - \beta_1 b K_0(\alpha_1 b) K_1(\alpha_2 b)] + K_1(\alpha_1 a)[\beta_1 b I_0(\alpha_1 b) K_1(\alpha_2 b) + \beta_2 b K_0(\alpha_2 b) I_1(\alpha_1 b)]\}$

SSR

$$\mathrm{Re}\ \frac{1}{I_1(\alpha a)}\Big[-I_1(\alpha_1 a)[\beta_2 b K_0(\alpha_2 b) K_1(\alpha_1 b) - \beta_1 b K_0(\alpha_1 b) K_1(\alpha_2 b)] + K_1(\alpha_1 a)[\beta_1 b I_0(\alpha_1 b) K_1(\alpha_2 b) + \beta_2 b K_0(\alpha_2 b) I_1(\alpha_1 b)]\Big]$$

$$\Big[\ [\beta_2 b K_0(\alpha_2 b) K_1(\alpha_1 b) - \beta_1 b K_0(\alpha_1 b) K_1(\alpha_2 b)]$$

$$[\beta_1 a I_0(\alpha_1 a) I_1(\alpha a) - \alpha a I_0(\alpha a) I_1(\alpha_1 a)]\ +$$

$$[\alpha a I_0(\alpha a) K_1(\alpha_1 a) + \beta_1 a K_0(\alpha_1 a) I_1(\alpha a)]$$

$$[\beta_1 b I_0(\alpha_1 b) K_1(\alpha_2 b) + \beta_2 b K_0(\alpha_2 b) I_1(\alpha_1 b)]\Big]^{-1} - \frac{K_1(\alpha a)}{I_1(\alpha a)}$$

X	α
X1	Re$[\alpha_1]$
X1A	Re$[\alpha_1 a]$
X1B	Re$[\alpha_1 b]$
X1R	Re$[\alpha_1 r]$
XA	αa
XB	αb
XF2	$\frac{2}{\pi}\frac{1}{\alpha^3}\ I(r_2, r_1)\ \sin(\alpha \ell/2)\ d\alpha$
XFACT	$\frac{1}{\alpha^3}\ I(r_2, r_1)\sin(\alpha \ell/2)$

XFACT2	$\frac{2}{\pi}\frac{1}{\alpha^3} I(r_2,r_1)\ \sin(\alpha\ell/2)\ \cos(\alpha z)d\alpha$
XIR21	$\frac{1}{\alpha^3} I(r_2,r_1)$
XJR21	$\frac{1}{\alpha^3} J(r_2,r_1)$
XL	$\alpha\ell$
XX	α^2
XXXX	α^4
Y1	$\mathrm{Im}[\alpha_1]$
Y1A	$\mathrm{Im}[\alpha_1 a]$
Y1B	$\mathrm{Im}[\alpha_1 b]$
Y1R	$\mathrm{Im}[\alpha_1 r]$
ZI0IA	$\mathrm{Im}[\alpha_1 a I_0(\alpha_1 a)]$
ZI0IB	$\mathrm{Im}[\alpha_1 b I_0(\alpha_1 b)]$
ZI0IR	$\mathrm{Im}[\alpha_1 r I_0(\alpha_1 r)]$
ZI0RA	$\mathrm{Re}[\alpha_1 a I_0(\alpha_1 a)]$
ZI0RB	$\mathrm{Re}[\alpha_1 b I_0(\alpha_1 b)]$
ZI0RR	$\mathrm{Re}[\alpha_1 r I_0(\alpha_1 r)]$
ZK0IA	$\mathrm{Im}[\alpha_1 a K_0(\alpha_1 a)]$
ZK0IB	$\mathrm{Im}[\alpha_1 b K_0(\alpha_1 b)]$
ZK0IR	$\mathrm{Im}[\alpha_1 r K_0(\alpha_1 r)]$
ZK0RA	$\mathrm{Re}[\alpha_1 a K_0(\alpha_1 a)]$
ZK0RB	$\mathrm{Re}[\alpha_1 b K_0(\alpha_1 b)]$
ZK0RR	$\mathrm{Re}[\alpha_1 r K_0(\alpha_1 r)]$

<u>Sample output</u>

The program ABBORAR calculates the defect signal averaged over the depth of the defect at different distances along the tube. The program can plot the defect impedance as the tube is scanned and pick out the maximum amplitude. Below we show a sample run where the maximum signal is printed.

```
ABBORAR     TIME  8:53:41  DATE  8/16/89
      IN RAD  OT RAD  LENGTH  RAD CLR WALLTH  % WALL  TUB IR  TUB OR
ACT  1.2000  1.5000  0.2650  0.0575  0.2200   45.45  1.5575  1.7775
NOR  0.8889  1.1111  0.1963  0.0426  0.1630   45.45  1.1537  1.3167
RBAR 1.3500 FREQ= 4.000000E+02 RHO= 3.8400 PERM=1.000 WUSRR=96.7060
```

```
NORM IMPD:RL  0.110847 IM  0.589927 AIR IND 1.693801E-02
NORM DSF:RL 1.4332E-01 IM 3.3396E-02 VOLN 3.1922E-04
MAXIMUM MAG  0.5544D-04   PHA AT MAX MAG   18.91   OD DEFECT
```

Partial listing of file FORT39:

0.000	0.5544D-04	18.91
0.001	0.5544D-04	18.91
0.002	0.5544D-04	18.91
0.003	0.5543D-04	18.91
0.004	0.5543D-04	18.90
0.005	0.5542D-04	18.90
0.006	0.5541D-04	18.89
0.007	0.5540D-04	18.88
0.008	0.5538D-04	18.87
0.009	0.5536D-04	18.86
0.010	0.5535D-04	18.85

Listing

```
      PROGRAM ABBORAR
C     VERSION August 16, 1989
C
C     Program to calculate the normalized impedance change in
C      an absolute boreside circumferential coil due to a defect
C      in a single tube as the coil scans past the defect.  The
C      program averages the effect of the defect over the depth
C      of the defect.
C
C     Z=0.0 AT CENTER OF COIL.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 L
      CHARACTER*1 FF
      DIMENSION S1(6),S2(6),ERR(6)
      DIMENSION CX(0:200),CY(0:200),XFACT2(200)
      DIMENSION SMZDFRA(0:200,30),SMZDFIA(0:200,30)
      DATA LOU/8/,PI/3.141592653/,LOD/39/,LOE/38/
      DATA S1/.005,.02,.05,.1,.1,.5/
      DATA S2/1.0,2.0,5.0,10.0,50.0,200.0/
      DATA ERR/.1,.01,.001,1.E-4,1.E-5,1.E-10/
      DATA FREQ/4.0E2/,RHO1/3.84/,U1/1.0/
      DATA TRN/400./,ISIDE/1/
      DATA MODE/16/NZT/100/,NRT/20/
      DATA OIM/50/,ORL/330/
      FF=CHAR(12)
C      OPEN(LOD,FILE='BORSCN.DAT',STATUS='NEW')
c      OPEN(LOE,FILE='BORDAT.DAT',STATUS='OLD')
   11 XMAX=0.
      YMAX=0.
      DFMMAX=0.
      DO 14 NZ=1,NZT
      DO 12 NR=1,NRT
      SMZDFRA(NZ,NR)=0.
      SMZDFIA(NZ,NR)=0.
   12 CONTINUE
   14 CONTINUE
c      READ(LOE,*,END=1001)DFDIAM,DFDEP
C     TIME AND DATE ARE PRINTED
      dfdiam=0.1
      dfdep=0.1
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT(' ABBORAR    TIME ',I2,':',I2,':'
     *,I2,'  DATE ',I2,'/',I2,'/',I2)
      WRITE(LOU,5)
    5 FORMAT(5X,'IN RAD',2X,'OT RAD',2X,'LENGTH',2X,'RAD CLR'
     *,1X,'WALLTH',2X,'% WALL',2X,'TUB IR',2X,'TUB OR')
```

```
      R1=1.2
      R2=1.5
      L=0.265
      RCL=0.0575
      T1=.220
      POW=(DFDEP/T1)*100.
      A=R2+RCL
      B=A+T1
      RDT=-DFDEP
      R3=0.5*(R1+R2)
      WRITE(LOU,10)R1,R2,L,RCL,T1,POW,A,B
      R1=R1/R3
      R2=2.0-R1
      L=L/R3
      RCL=RCL/R3
      RDT=RDT/R3
      T1=T1/R3
      A=A/R3
      B=B/R3
      DELTAZ=0.5*L/NZT
C      VOLN=0.1666667*PI*(DFDIAM/R3)*(DFDIAM/R3)*(DFDIAM/R3)
      VOLN=PI*DFDIAM*DFDIAM*DFDEP/(4.*R3*R3*R3)
      WUSRR=0.5093979*U1*R3*R3*FREQ/RHO1
      WRITE(LOU,15)R1,R2,L,RCL,T1,POW,A,B
   10 FORMAT('ACT ',5(F7.4,1X),F7.2,1X,3(F7.4,1X))
   15 FORMAT('NOR ',5(F7.4,1X),F7.2,1X,3(F7.4,1X))
      WRITE(LOU,20)R3,FREQ,RHO1,U1,WUSRR
   20 FORMAT(' RBAR',F7.4,' FREQ= ',1PE13.6,' RHO=',0PF9.4,
     *' PERM=',F7.3,' WUSRR=',F11.4)
      CALL QSMODE(MODE)
      CALL GRID
      SMAIR=0.0
      SMIMPR=0.0
      SMIMPI=0.0
      SMZDFR=0.0
      SMZDFI=0.0
C     AIR=0.
C     AII=0.
C     AUR=0.
C     AUI=0.
      B1=0.0
      B2=S2(1)
      DO 100 JKL=1,6
   30 RI9=SMAIR
      X=B1-0.5*S1(JKL)
C     DETERMINE NUMBER OF STEPS,ISTEPS,BASED ON STEP SIZE&INTERVAL
      ISTEPS=DNINT((B2-B1)/S1(JKL))
      DO 90 I=1,ISTEPS
      X=X+S1(JKL)
      CALL IJBSSL(X,R1,R2,XIR21,XJR21)
      XL=X*L
```

```
      IF(XL.GT.5.0E-3) GO TO 60
      A1=XL*XL*(0.5-XL/6.0)
      GO TO 80
   60 IF(XL.GT.75.0) GO TO 70
      A1=XL+DEXP(-XL)-1.0
      GO TO 80
   70 A1=XL-1.0
   80 SMAIR=SMAIR+2.*XJR21*XJR21*A1*S1(JKL)
      IF(X.GT.160.)GO TO 90
      XX=X*X
      XXXX=XX*XX
      X1=DSQRT(0.5*(XX+DSQRT(XXXX+WUSRR*WUSRR)))/U1
      Y1=WUSRR/(2*X1*U1*U1)
      XA=X*A
      XB=X*B
      X1A=X1*A
      Y1A=Y1*A
      X1B=X1*B
      Y1B=Y1*B
      CALL CMDBES(X1A,Y1A,ZIORA,ZIOIA,ZKORA,ZKOIA,BI1RA,BI1IA
     *,BK1RA,BK1IA)
      CALL CMDBES(X1B,Y1B,ZIORB,ZIOIB,ZKORB,ZKOIB,BI1RB,BI1IB
     *,BK1RB,BK1IB)
      CALL BESI(XA,BIOA,BI1A)
      CALL BESK(XA,BKOA,BK1A)
      CALL BESI(XB,BIOB,BI1B)
      CALL BESK(XB,BKOB,BK1B)
      DR1=XB*BKOB*BK1RB-ZKORB*BK1B/U1
      DI1=XB*BKOB*BK1IB-ZKOIB*BK1B/U1
      DR2=ZIORA*BI1A/U1-XA*BIOA*BI1RA
      DI2=ZIOIA*BI1A/U1-XA*BIOA*BI1IA
      DR3=XA*BIOA*BK1RA+ZKORA*BI1A/U1
      DI3=XA*BIOA*BK1IA+ZKOIA*BI1A/U1
      DR4=ZIORB*BK1B/U1+XB*BKOB*BI1RB
      DI4=ZIOIB*BK1B/U1+XB*BKOB*BI1IB
      DDR=DR1*DR2-DI1*DI2+DR3*DR4-DI3*DI4
      DDI=DR1*DI2+DR2*DI1+DR3*DI4+DR4*DI3
      SNR=-BI1RA*DR1+BI1IA*DI1+BK1RA*DR4-BK1IA*DI4
      SNI=-BI1IA*DR1-BI1RA*DI1+BK1RA*DI4+BK1IA*DR4
      DEN=DDR*DDR+DDI*DDI
      SSR=((SNR*DDR+SNI*DDI)/DEN-BK1A)/BI1A
      SSI=(SNI*DDR-SNR*DDI)/(BI1A*DEN)
      XFACT=XIR21*DSIN(XL/2.)
      SMIMPR=SMIMPR+8.*XFACT*XFACT*S1(JKL)*SSR/PI
      SMIMPI=SMIMPI+8.*XFACT*XFACT*S1(JKL)*SSI/PI
      XF2=2.*XFACT*S1(JKL)/PI

      DO 89 NR=1,NRT
c       write(0,*)rdt
      RD=(REAL(NR)-0.5)*(RDT/REAL(NRT))
c       write(0,*)rd,rdt
```

```
c       pause
        IF(ISIDE.EQ.1) THEN
        RD=B+RD
        ELSE
        RD=A-RD
        END IF
c       rd=1.3333
        X1R=X1*RD
        Y1R=Y1*RD

        CALL CMDBES(X1R,Y1R,ZI0RR,ZI0IR,ZK0RR,ZK0IR,BI1RR,BI1IR
       *,BK1RR,BK1IR)

        DNR=BK1RR*DR4-BK1IR*DI4-BI1RR*DR1+BI1IR*DI1
        DNI=BK1IR*DR4+BK1RR*DI4-BI1IR*DR1-BI1RR*DI1
        DFR=(DNR*DDR+DNI*DDI)/DEN
        DFI=(DNI*DDR-DNR*DDI)/DEN

        DO 88 NZ=0,NZT
        IF(NR.GT.1)GO TO 87
        ZD=DELTAZ*NZ
        XFACT2(NZ)=XF2*DCOS(X*ZD)
     87 SMZDFRA(NZ,NR)=SMZDFRA(NZ,NR)+XFACT2(NZ)*DFR
        SMZDFIA(NZ,NR)=SMZDFIA(NZ,NR)+XFACT2(NZ)*DFI
     88 CONTINUE
     89 CONTINUE
     90 CONTINUE
        B1=B2
        B2=B2+S2(JKL)
        CHECK=(SMAIR-RI9)/SMAIR
        IF(ABS(CHECK).GT.ERR(JKL)) GO TO 30
    100 CONTINUE
        DO 990 NZ=0,NZT
        ZD=DELTAZ*NZ
        SMZDFR=0.
        SMZDFI=0.
        DO 120 NR=1,NRT
        SMZDFR=SMZDFR+SMZDFRA(NZ,NR)
        SMZDFI=SMZDFI+SMZDFIA(NZ,NR)
    120 CONTINUE
        SMZDFR=SMZDFR/NRT
        SMZDFI=SMZDFI/NRT
C       WRITE(LOU,*)NZ,SMZDFR,SMZDFI
        DSFR=-1.5*WUSRR*(SMZDFR*SMZDFR-SMZDFI*SMZDFI)/(SMAIR*PI)
        DSFI=-1.5*WUSRR*2.0*SMZDFR*SMZDFI/(SMAIR*PI)
    135 Q6=0.0254*4.0E-7*TRN*TRN*R3*PI*PI*SMAIR/(L*(R2-R1))**2
        ZNIM=SMIMPR/SMAIR+1.0
        ZNRL=-SMIMPI/SMAIR
        ZNDFR=VOLN*DSFR
        ZNDFI=VOLN*DSFI
        IF(NZ.EQ.0) WRITE(LOU,140)ZNRL,ZNIM,Q6
```

```
      DFM=VOLN*SQRT(DSFR*DSFR+DSFI*DSFI)
      DFP=ATAN2(DSFI,DSFR)
      CX(NZ)=DFM*COS(DFP)
      CY(NZ)=DFM*SIN(DFP)
      DFP=DFP*(180./PI)
      IF(ABS(CX(NZ)).GT.XMAX) XMAX=ABS(CX(NZ))
      IF(ABS(CY(NZ)).GT.YMAX) YMAX=ABS(CY(NZ))
      IF(DFM.GT.DFMMAX) THEN
      DFMMAX=DFM
      DFPMMAX=DFP
      END IF
      WRITE(LOD,162)ZD,DFM,DFP
  990 CONTINUE
c      write(0,*)'rd ',rd
      GIM=300./YMAX
      GRL=300./XMAX
      IF(GIM.GT.GRL) THEN
      GIM=GRL
      ELSE
      GRL=GIM
      END IF
      IM1=GIM*CY(0)+OIM
      IR1=GRL*CX(0)+ORL
C      WRITE(LOU,150)ZNDFR,ZNDFI,DFM,DFP
      WRITE(LOU,160)DSFR,DSFI,VOLN
      WRITE(LOU,164)DFMMAX,DFPMMAX
      IF(ISIDE.EQ.1) THEN
      WRITE(LOU,*)'  OD DEFECT'
      ELSE
      WRITE(LOU,*)'  ID DEFECT'
      END IF

      DO 1000 NZ=1,NZT
      IM2=GIM*CY(NZ)+OIM
      IR2=GRL*CX(NZ)+ORL
      CALL QLINE(IR1,IM1,IR2,IM2,15)
      IR1=IR2
      IM1=IM2
 1000 CONTINUE
      WRITE(LOU,*)
      WRITE(LOU,*)
      CALL PRTSC
      WRITE(LOU,*)FF
c      GO TO 11

  140 FORMAT(' NORM IMPD:RL',F10.6,' IM',F10.6,
     *' AIR IND',1PE13.6)
C 150 FORMAT(' NORM CHG:RL ',0PF10.6,' IM',0PF10.6,
C    *' MAG',0PF10.6,' PHA ',0PF7.2)
  160 FORMAT(' NORM DSF:RL',1PE11.4,' IM',1PE11.4,' VOLN',1PE11.4)
  161 FORMAT('    ZD        MAG        PHA')
```

```
 162 FORMAT(F6.3,5X,D11.4,5X,F7.2)
 164 FORMAT(' MAXIMUM MAG ',D11.4,'    PHA AT MAX MAG ',F7.2,\)
1001 END
```

DBDSF calculates DSF at lattice of points for differential coil

Program DBDSF calculates the defect sensitivity factor of a differential boreside coil at a two-dimensional lattice of points throughout the wall of a conducting tube. The differential probe is shown in Fig. 15.

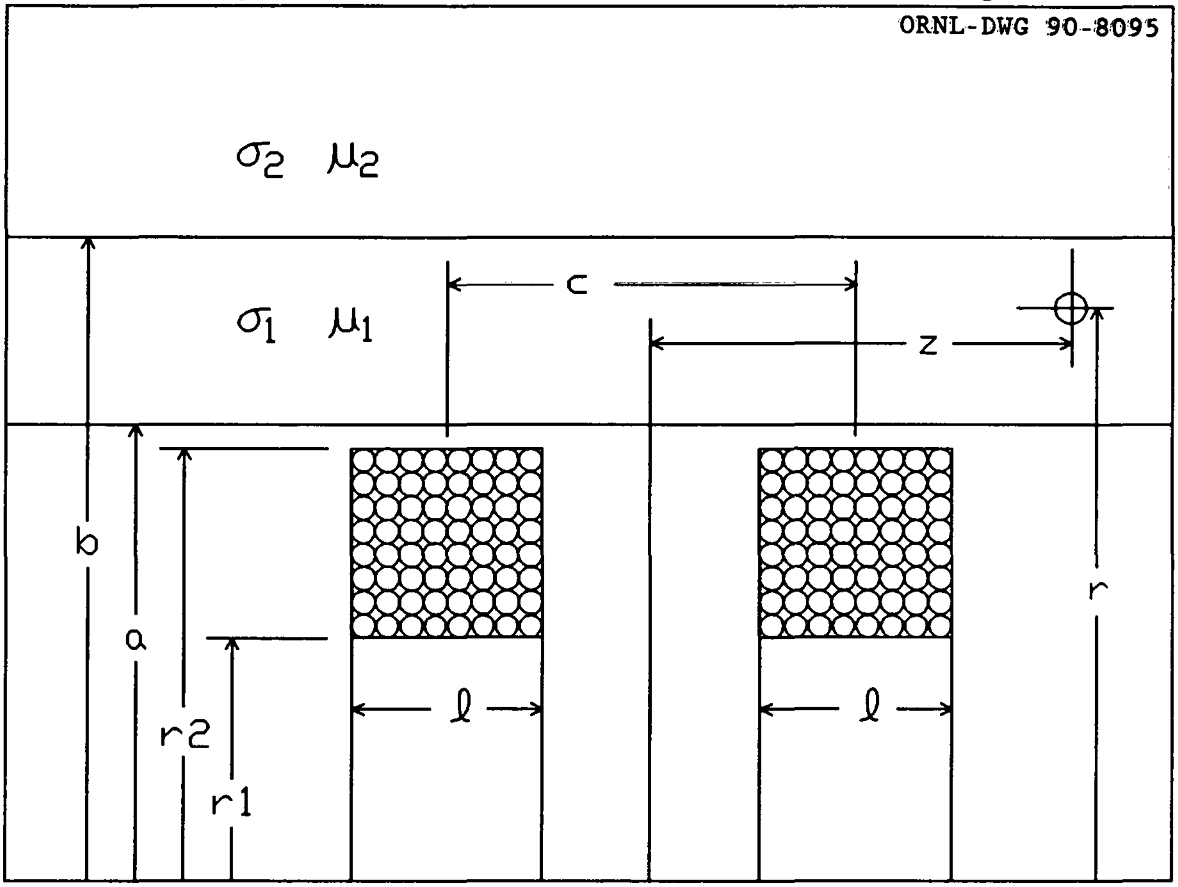

Fig. 15. Cross section of a differential coil in the bore of a tube with a defect present.

The distance to the defect is measured from the center of the coil assembly, and the center-to-center distance of the coils is denoted *c*. The impedance difference between the two matched coils is:

$$Z_{nd} = Z_{1d} - Z_{2d} \tag{39}$$

Substituting in from Eq. (29) for the impedance change of each coil, removing the normalized defect volume to get the defect sensitivity factor and using some trigonometric identities gives for the defect sensitivity factor:

$$DSF(r,z) = \frac{-3(\omega\mu\sigma_1\bar{r}^2)}{2\pi I_{air}} \tag{40}$$

$$\times \left[\int_0^\infty \frac{I(r_2,r_1)}{\pi\alpha^3} \left\{ \frac{K_1(\alpha_1 r)D_4 - I_1(\alpha_1 r)D_1}{(D_1D_2+D_3D_4)} \right\} sin(\frac{\alpha\ell}{2}) sin(\frac{\alpha c}{2}) sin(\alpha z) 4d\alpha \right]$$

$$\times \left[\int_0^\infty \frac{I(r_2,r_1)}{\pi\alpha^3} \left\{ \frac{K_1(\alpha_1 r)D_4 - I_1(\alpha_1 r)D_1}{(D_1D_2+D_3D_4)} \right\} sin(\frac{\alpha\ell}{2}) cos(\frac{\alpha c}{2}) cos(\alpha z) 4d\alpha \right]$$

The definitions of the various terms in Eq. (40) are given in Eqs. (30) through (38) in the discussion of the absolute boreside coil. The function and variable names are essentially the same as they are for the absolute coil and will not be repeated here. The program DBDSF stores the calculated values in the data file FORT40 so they can be plotted by program DBDSFPLT.

Variables

A	The normalized inner radius of the tube.
B	The normalized outer radius of the tube.
C*	The axial distance between the centers of the two coils. The value is input in inches and normalized by the program.
DELTAR	The normalized radial distance between the points at which the calculations are done.
DELTAZ*	The normalized axial distance between the points at which the calculations are done.
DSFI	The imaginary part of the defect sensitivity factor of the coil.
DSFM	The magnitude of the defect sensitivity factor of the coil.
DSFP	The phase of the defect sensitivity factor of the coil.
DSFR	The real part of the defect sensitivity factor of the coil.
FREQ*	The operating frequency in hertz.
L*	The length of each coil. The value is input in inches and normalized by the program.
LOD*	The number of the I/O unit connected to the output data file.
LOU	The number of the I/O unit connected to the printer.
NRT*	The total number of points in the radial direction at which the defect sensitivity factor is calculated.
NZT*	The total number of points in the axial direction at which the defect sensitivity factor is

	calculated.
R1*	The inner radius of each coil. The value is input in inches and normalized by the program.
R2*	The outer radius of each coil. The value is input in inches and normalized by the program.
R3	The mean radius of each coil in inches.
RCL*	The distance between the outside of the coil and the inside wall of the tube. The value is input in inches and normalized by the program.
RD	The radial distance between the center of the coil and the point where the calculations are being performed.
RDT	The normalized thickness of the tube. A negative number.
RHO1*	The electrical resistivity of the tube in μΩ-cm.
T1*	The thickness of the tube wall. The value is input in inches and normalized by the program.
TRN*	The number of turns in each coil.
U1*	The relative magnetic permeability of the tube.
WUSRR	The product of the angular operating frequency, the magnetic permeability of the tube, the electrical conductivity of the tube, and the square of the mean radius of the coil.
ZD	The axial distance between the center of the coil and the point where the calculations are being performed.

Sample output

A listing of the printer output is shown below:

```
DBDSF    TIME  9:59:28  DATE  8/16/89
     IN RAD  OT RAD  LENGTH  RAD CLR WALLTH  C TO C  TUB IR  TUB OR
ACT  1.2400  1.4900  0.2650  0.0575  0.2200  0.5150  1.5475  1.7675
NOR  0.9084  1.0916  0.1941  0.0421  0.1612  0.3773  1.1337  1.2949
RBAR 1.3650 FREQ=  4.000000E+02 RHO=   3.8400 PERM=  1.000 WUSRR= 98.8670
```

A partial listing of the file FORT40 is given below:

```
      50       40
     0.02500      0.00403
     0.04212
     0.90842      1.09158
     0.37729      0.19414
     1.13370      1.29487
    0     0  0.00000D+00  0.00000D+00
    0     1  0.00000D+00  0.00000D+00
    ........... (zero for all of first row)....
    1     0  0.24266D-01  0.33487D+00
    1     1  0.25157D-01  0.33978D+00
    1     2  0.26086D-01  0.34774D+00
    1     3  0.27053D-01  0.35864D+00
```

1	4	0.28061D-01	0.37238D+00
1	5	0.29113D-01	0.38887D+00
1	6	0.30213D-01	0.40800D+00
1	7	0.31364D-01	0.42968D+00
1	8	0.32570D-01	0.45382D+00
1	9	0.33836D-01	0.48031D+00
1	10	0.35166D-01	0.50906D+00

Listing

```
      PROGRAM DBDSF
C     VERSION August 16, 1989
C     PROGRAM TO CALCULATE THE DEFECT SENSITIVITY FACTOR OF A
C     DIFFERENTIAL BORESIDE COIL AT AN ARRAY OF POINTS THROUGHOUT
C     THE CROSS SECTION OF A TUBE WALL.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 L
      DIMENSION S1(6),S2(6),ERR(6),XFACT1(0:50),XFACT2(0:50)
      DIMENSION SMZDFR1A(0:50,0:40),SMZDFI1A(0:50,0:40)
      DIMENSION SMZDFR2A(0:50,0:40),SMZDFI2A(0:50,0:40)
      DATA LOU/8/,PI/3.141592653/,LOD/40/
      DATA S1/.005,.02,.05,.1,.1,.5/
      DATA S2/1.0,2.0,5.0,10.0,50.0,200.0/
      DATA ERR/.1,.01,.001,1.E-4,1.E-5,1.E-10/
      DATA FREQ/4.0E2/,RHO1/3.84/,U1/1.0/
      DATA TRN/400./,DELTAZ/0.025/
C  NOTE:  If the value of either NZT or NRT is changed, the
C   statements to dimension the arrays DSFMA, XX, and YY in
C   program DBDSFPLT must be changed so that the arrays are
C   dimensioned to exactly the new values.
      DATA NZT/50/,NRT/40/
   11 DO 14 NZ=0,NZT
      DO 12 NR=0,NRT
      SMZDFR1A(NZ,NR)=0.
      SMZDFI1A(NZ,NR)=0.
      SMZDFR2A(NZ,NR)=0.
      SMZDFI2A(NZ,NR)=0.
   12 CONTINUE
   14 CONTINUE
C     TIME AND DATE ARE PRINTED
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT(' DBDSF    TIME ',I2,':',I2,':'
     *,I2,'  DATE ',I2,'/',I2,'/',I2)
      WRITE(LOU,5)
    5 FORMAT(5X,'IN RAD',2X,'OT RAD',2X,'LENGTH',2X,'RAD CLR'
     *,1X,'WALLTH',2X,'C TO C',2X,'TUB IR',2X,'TUB OR')
      R1=1.24
      R2=1.49
      L=0.265
      C=0.515
      RCL=0.0575
      T1=0.22
      A=R2+RCL
      B=A+T1
      RDT=-T1
      R3=0.5*(R1+R2)
```

```
      WRITE(LOU,10)R1,R2,L,RCL,T1,C,A,B
      R1=R1/R3
      R2=2.0-R1
      L=L/R3
      C=C/R3
      RCL=RCL/R3
      RDT=RDT/R3
      T1=T1/R3
      DELTAR=T1/NRT
      A=A/R3
      B=B/R3
      WRITE(LOD,7)NZT,NRT
      WRITE(LOD,8)DELTAZ,DELTAR
      WRITE(LOD,9)RCL
      WRITE(LOD,8)R1,R2
      WRITE(LOD,8)C,L
      WRITE(LOD,8)A,B
    7 FORMAT(I8,1X,I8)
    8 FORMAT(F12.5,1X,F12.5)
    9 FORMAT(F12.5)
      WUSRR=0.5093979*U1*R3*R3*FREQ/RHO1
      WRITE(LOU,15)R1,R2,L,RCL,T1,C,A,B
   10 FORMAT('ACT ',5(F7.4,1X),3(F7.4,1X))
   15 FORMAT('NOR ',5(F7.4,1X),3(F7.4,1X))
      WRITE(LOU,20)R3,FREQ,RHO1,U1,WUSRR
   20 FORMAT(' RBAR',F7.4,' FREQ= ',1PE13.6,' RHO=',0PF9.4,
     *' PERM=',F7.3,' WUSRR=',F11.4)
      SMAIR=0.0
      SMIMPR=0.0
      SMIMPI=0.0
      SMZDFR1=0.0
      SMZDFI1=0.0
      SMZDFR2=0.0
      SMZDFI2=0.0
      B1=0.0
      B2=S2(1)
      DO 100 JKL=1,6
   30 RI9=SMAIR
      X=B1-0.5*S1(JKL)
C     DETERMINE NUMBER OF STEPS,ISTEPS,BASED ON STEP SIZE&INTERVAL
      ISTEPS=DNINT((B2-B1)/S1(JKL))
      DO 90 I=1,ISTEPS
      X=X+S1(JKL)
      CALL IJBSSL(X,R1,R2,XIR21,XJR21)
      XL=X*L
      IF(XL.GT.5.0E-3) GO TO 60
      A1=XL*XL*(0.5-XL/6.0)
      GO TO 80
   60 IF(XL.GT.75.0) GO TO 70
      A1=XL+DEXP(-XL)-1.0
      GO TO 80
```

```
70 A1=XL-1.0
80 SMAIR=SMAIR+2.*XJR21*XJR21*A1*S1(JKL)
   IF(X.GT.160.)GO TO 90
   XX=X*X
   XXXX=XX*XX
   X1=DSQRT(0.5*(XX+DSQRT(XXXX+WUSRR*WUSRR)))/U1
   Y1=WUSRR/(2*X1*U1*U1)
   XA=X*A
   XB=X*B
   X1A=X1*A
   Y1A=Y1*A
   X1B=X1*B
   Y1B=Y1*B
   CALL CMDBES(X1A,Y1A,ZIORA,ZIOIA,ZKORA,ZKOIA,BI1RA,BI1IA
  *,BK1RA,BK1IA)
   CALL CMDBES(X1B,Y1B,ZIORB,ZIOIB,ZKORB,ZKOIB,BI1RB,BI1IB
  *,BK1RB,BK1IB)
   CALL BESI(XA,BIOA,BI1A)
   CALL BESK(XA,BKOA,BK1A)
   CALL BESI(XB,BIOB,BI1B)
   CALL BESK(XB,BKOB,BK1B)
   DR1=XB*BKOB*BK1RB-ZKORB*BK1B/U1
   DI1=XB*BKOB*BK1IB-ZKOIB*BK1B/U1
   DR2=ZIORA*BI1A/U1-XA*BIOA*BI1RA
   DI2=ZIOIA*BI1A/U1-XA*BIOA*BI1IA
   DR3=XA*BIOA*BK1RA+ZKORA*BI1A/U1
   DI3=XA*BIOA*BK1IA+ZKOIA*BI1A/U1
   DR4=ZIORB*BK1B/U1+XB*BKOB*BI1RB
   DI4=ZIOIB*BK1B/U1+XB*BKOB*BI1IB
   DDR=DR1*DR2-DI1*DI2+DR3*DR4-DI3*DI4
   DDI=DR1*DI2+DR2*DI1+DR3*DI4+DR4*DI3
   SNR=-BI1RA*DR1+BI1IA*DI1+BK1RA*DR4-BK1IA*DI4
   SNI=-BI1IA*DR1-BI1RA*DI1+BK1RA*DI4+BK1IA*DR4
   DEN=DDR*DDR+DDI*DDI
   SSR=((SNR*DDR+SNI*DDI)/DEN-BK1A)/BI1A
   SSI=(SNI*DDR-SNR*DDI)/(BI1A*DEN)
   XFACT=XIR21*DSIN(XL/2.)
   SMIMPR=SMIMPR+8.*XFACT*XFACT*S1(JKL)*SSR/PI
   SMIMPI=SMIMPI+8.*XFACT*XFACT*S1(JKL)*SSI/PI
   XF=4.*XFACT*S1(JKL)/PI
   XF1=XF*DSIN(0.5*X*C)
   XF2=XF*DCOS(0.5*X*C)
   DO 89 NR=0,NRT
   RD=(REAL(NR))*(RDT/REAL(NRT))
   RD=B+RD
   X1R=X1*RD
   Y1R=Y1*RD

   CALL CMDBES(X1R,Y1R,ZIORR,ZIOIR,ZKORR,ZKOIR,BI1RR,BI1IR
  *,BK1RR,BK1IR)
```

```
      DNR=BK1RR*DR4-BK1IR*DI4-BI1RR*DR1+BI1IR*DI1
      DNI=BK1IR*DR4+BK1RR*DI4-BI1IR*DR1-BI1RR*DI1
      DFR=(DNR*DDR+DNI*DDI)/DEN
      DFI=(DNI*DDR-DNR*DDI)/DEN

      DO 88 NZ=0,NZT
      IF(NR.GT.1)GO TO 87
      ZD=DELTAZ*NZ
      XFACT1(NZ)=XF1*DSIN(X*ZD)
      XFACT2(NZ)=XF2*DCOS(X*ZD)
   87 SMZDFR1A(NZ,NR)=SMZDFR1A(NZ,NR)+XFACT1(NZ)*DFR
      SMZDFI1A(NZ,NR)=SMZDFI1A(NZ,NR)+XFACT1(NZ)*DFI
      SMZDFR2A(NZ,NR)=SMZDFR2A(NZ,NR)+XFACT2(NZ)*DFR
      SMZDFI2A(NZ,NR)=SMZDFI2A(NZ,NR)+XFACT2(NZ)*DFI
   88 CONTINUE
   89 CONTINUE
   90 CONTINUE
      B1=B2
      B2=B2+S2(JKL)
      CHECK=(SMAIR-RI9)/SMAIR
      IF(ABS(CHECK).GT.ERR(JKL)) GO TO 30
  100 CONTINUE
      DO 990 NZ=0,NZT
      ZD=DELTAZ*NZ
      SMZDFR1=0.
      SMZDFI1=0.
      SMZDFR2=0.
      SMZDFI2=0.
      DO 120 NR=0,NRT
      SMZDFR1=SMZDFR1A(NZ,NR)
      SMZDFI1=SMZDFI1A(NZ,NR)
      SMZDFR2=SMZDFR2A(NZ,NR)
      SMZDFI2=SMZDFI2A(NZ,NR)
      DSFR=-1.5*WUSRR*(SMZDFR1*SMZDFR2-SMZDFI1*SMZDFI2)/(SMAIR*PI)
      DSFI=-1.5*WUSRR*(SMZDFR1*SMZDFI2+SMZDFR2*SMZDFI1)/(SMAIR*PI)
      DSFM=DSQRT(DSFR*DSFR+DSFI*DSFI)
      DSFP=DATAN2(DSFI,DSFR)
      WRITE(LOD,126)NZ,NR,DSFM,DSFP
c       WRITE(0,126)NZ,NR,DSFM,DSFP
  120 CONTINUE
  126 FORMAT(I5,1X,I5,1X,D12.5,1X,D12.5)
  990 CONTINUE
 1000 CONTINUE
 1001 END
```

DBDSFPLT generates a contour plot from DBDSF data

Program DBDSFPLT generates a contour plot of the magnitude of the defect sensitivity factor for a differential boreside coil using calculations performed and stored by the program DBDSF in the file FORT40.

Summary

1. Dimension arrays and declare variable types.
2. Initialize variables.
3. Open the file created by program DBDSF.
4. Read in the information about the coil and tube from the file.
5. Calculate the position of the data points in the normalized coordinate system.
6. Set the label flags for the contours.
7. Read the data stored by program DBDSF into array DSFMA.
8. Specify the values of the magnitude of the defect sensitivity factor where the contours are to be drawn.
9. Call the PRINTMATIC contour initialization routines.
10. Draw the contours.
11. Draw the coil and tube.

Variables

A	The normalized inner radius of the tube.
B	The normalized outer radius of the tube.
C	The center-to-center spacing between the coils.
CNM*	Array giving the values of the magnitude of the defect sensitivity factor at which contours are to be drawn.
DELTAR	The normalized distance in the radial direction between adjacent data points.
DELTAZ	The normalized distance in the axial direction between adjacent data points.
DSFMA	Array containing the values of the magnitude of the defect sensitivity factor which were read in from the data file.
L	The normalized length of the coil.
LBM*	Array which tells the program which of the contours are to be labeled with their values. If all elements of LBM are zero, none of the contours will be labeled.
LOE*	The number of the I/O unit connected to the input data file.
NAME*	Character variable which contains the name of the file which this program uses for output.
NC*	The number of contours to be drawn.
NRT	The total number of points in the radial direction at which calculations were performed.
NZT	The total number of points in the axial direction

	at which calculations were performed.
R1	The normalized inner coil radius.
R2	The normalized outer coil radius.
RCL	The normalized distance between the outside of the coil and the inside of the tube.
XX	Array describing the axial location of the data points in the normalized coordinate system.
YY	Array describing the radial location of the data points in the normalized coordinate system.

Notes

1. Program DBDSFPLT does not actually send anything to the printer; it merely creates a file whose name is given by the program variable NAME. If the value of NAME is 'filename.ext', then to print the file created by program DBDSFPLT, enter

```
DPRINT filename.ext
```

DPRINT.EXE is a program supplied by PRINTMATIC. For this particular program the variable NAME is set to PCDSF.FIL so to make a plot one would type:

```
DPRINT DBDSF.FIL
```

Sample output

We show a plot of the data from DBDSF plotted using DBDSFPLT in Fig. 16. The phase contours can be plotted rather than the magnitude by using the second statement with label 140 and commenting the first one out.

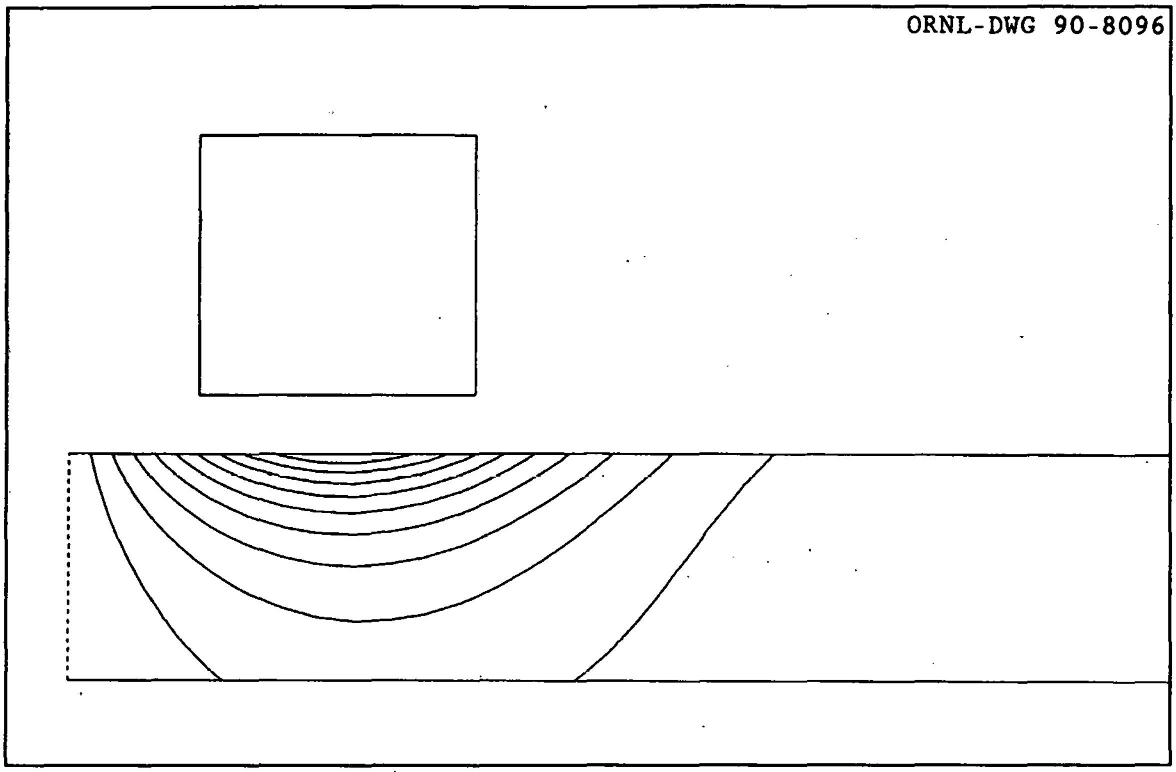

Fig. 16. Contour plot of the magnitude of the defect sensitivity factor for a differential boreside probe.

Listing

```
      PROGRAM DBDSFPLT
C     VERSION October 25, 1988
C     Program to generate a contour plot of the magnitude of the
C     defect sensitivity factor of a differential boreside coil.
C
      CHARACTER*80 NAME
      IMPLICIT REAL*4 (A-H,O-Z)
      REAL*4 DSFMA(51,41)
      REAL*4 XX(51),YY(41)
      REAL*4 CNM(10)
      REAL*4 L
      INTEGER*2 LBM(10)
      INTEGER*2 I1,J1,I2,J2
      DATA XSCALE/1.0/,NC/9/
      DATA IDEF/2/,LOE/40/
C
C   Open the file created by program DBDSF and read in
C    the coil and tube information.
C
      OPEN(LOE,FILE='FORT40',STATUS='OLD')
      READ(LOE,*)NZT,NRT
```

```
      write(0,*)nzt,nrt
      READ(LOE,*)DELTAZ,DELTAR
      READ(LOE,*)RCL
      READ(LOE,*)R1,R2
      READ(LOE,*)C,L
      READ(LOE,*)A,B
C
C  Calculate the position of the data points in normalized units
C
      DO 110 I=1,NZT+1
      XX(I)=REAL(I-1)*DELTAZ
  110 CONTINUE
      DO 120 I=0,NRT
      YY(I+1)=-((REAL(NRT)-REAL(I))*DELTAR)
  120 CONTINUE
C
C  Set the label flags for the contours
C
      DO 130 I=1,10
      LBM(I)=0
  130 CONTINUE
C
C  Read the data stored by program DBDSF
C
      NI=0
      DSFMMAX=0.
  140 READ(LOE,*,END=150)NZ,NR,DSFM
c 140 READ(LOE,*,END=150)NZ,NR,DUM,DSFM
      NI=NI+1
      IF(DSFM.GT.DSFMMAX)DSFMMAX=DSFM
      IF(NI.EQ.1)DSFMMIN=DSFM
      IF(DSFM.LT.DSFMMIN)DSFMMIN=DSFM
      DSFMA(NZ+1,NR+1)=DSFM
      GO TO 140
C
C  Specify the values of the magnitude of the defect
C   sensitivity factor where the contours are to be drawn.
C
  150 VMG=DSFMMAX-DSFMMIN
      CNDF=VMG/(NC+1)
      WRITE(0,*)DSFMMAX
      WRITE(0,*)DSFMMIN
      WRITE(0,*)VMG
      WRITE(0,*)CNDF
      DO 160 I=1,NC
      CNM(I)=DSFMMAX-I*CNDF
      WRITE(0,*)CNM(I)
  160 CONTINUE
      PAUSE
C
C  Call the necessary initialization routines
```

```
C
      NAME='DBDSF.FIL'
      CALL DINIT(NAME)
      CALL DPLOT(1.,1.,8.,4.,0.,0.8,-0.2,0.2,0.,0.)
      CALL DCTRDEF(1,1,1,1,1)
C
C  Draw the contours
C
      nzt=nzt+1
      nrt=nrt+1
      CALL DCNTOUR(XSCALE,XX,YY,DSFMA,CNM,LBM,NZT,NRT,NC,IDEF)
C
C  Draw the tube
C
      X1=0.
      Y1=0.
      X2=0.8
      Y2=-(B-A)
      write(0,*)j2
      CALL DRTOI(X1,Y1,I1,J1)
      CALL DRTOI(X2,Y2,I2,J2)
      write(0,*)I1,J1,I2,J1
      CALL DLINE(I1,J1,I2,J1)
      write(0,*)I1,J1,I2,J1
      CALL DLINE(I1,J2,I2,J2)
      write(0,*)j2
C
C  Draw the coil
C
      X1=0.5*(C-L)
      Y1=RCL
      X2=0.5*(C+L)
      Y2=RCL+R2-R1
      CALL DRTOI(X1,Y1,I1,J1)
      CALL DRTOI(X2,Y2,I2,J2)
      write(0,*)j2
      CALL DLINE(I1,J1,I2,J1)
      CALL DLINE(I1,J2,I2,J2)
      CALL DLINE(I1,J1,I1,J2)
      write(0,*)I2,J1,I2,J2
      CALL DLINE(I2,J1,I2,J2)
      write(0,*)I2,J1,I2,J2
C
C  Draw a dotted line on the plane between the coils
C
      X1=0.
      Y1=0.
      X2=0.
      Y2=-(B-A)
      CALL DRTOI(X1,Y1,I1,J1)
      CALL DRTOI(X2,Y2,I2,J2)
```

```
CALL DDASH(I1,J1,I2,J2,1,10,10)
CALL DFINIS
write(0,*)j2
STOP
END
```

DFBORAR calculates defect impedance change, average over depth

Program DFBORAR calculates the change in the normalized impedance of a differential boreside coil due to the presence of a defect in a single conducting tube, as shown in Fig. 15. The equations computed are the same as those for the defect sensitivity factor, with the defect volume included. The distance to the defect is measured from the center of the coil assembly, and the center-to-center distance of the coils is denoted c. The impedance difference between the two matched coils is:

$$Z_{nd} = Z_{1d} - Z_{2d} \tag{41}$$

Substituting in from Eq. (29) for the impedance change of each coil and using some trigonometric identities gives for the normalized impedance difference:

$$Z_{nd}(r,z) = \frac{-3(\omega\mu\sigma_1\bar{r}^2)Vol_n}{2\pi I_{air}} \tag{42}$$

$$\times \left[\int_0^\infty \frac{I(r_2,r_1)}{\pi\alpha^3}\left\{\frac{K_1(\alpha_1 r)D_4 - I_1(\alpha_1 r)D_1}{(D_1D_2+D_3D_4)}\right\} sin(\frac{\alpha\ell}{2}) sin(\frac{\alpha c}{2}) sin(\alpha z) 4d\alpha\right]$$

$$\times \left[\int_0^\infty \frac{I(r_2,r_1)}{\pi\alpha^3}\left\{\frac{K_1(\alpha_1 r)D_4 - I_1(\alpha_1 r)D_1}{(D_1D_2+D_3D_4)}\right\} sin(\frac{\alpha\ell}{2}) cos(\frac{\alpha c}{2}) cos(\alpha z) 4d\alpha\right]$$

The definition of the various terms in Eq. (42) are given in Eqs. (30) through (38) in the discussion of the absolute boreside coil. It performs the calculations for a number of different axial distances between the center of the coil and the defect, and it can divide the defect into a number of parts and calculate the effect of each part separately. The effect of the defect is averaged over the depth of the defect, and the defect is moved from the probe center ($z = 0$) outward in the positive z direction.

<u>Summary</u>

1. Dimension arrays and declare variable types.
2. Initialize variables.
3. Begin the integration. Calculate the expressions that are independent of the position of the defect.
4. Select a value for the radial distance between the defect and the center of the coil. Do the calculations which depend only upon this component of the position.
5. Select a value for the axial distance between the defect and the center of the coil. Complete the calculations.
6. Loop to 5 until done.
7. Loop to 4 until done.

8. Output the results.

Variables

A number of the variables which occur mainly inside the integration loops do not correspond to anything physical. These variables are described in a section at the end. Starred variables must be assigned by the user.

A	The normalized inner radius of the tube.
B	The normalized outer radius of the tube.
C*	The axial distance between the centers of the two coils. The value is input in inches and normalized by the program.
DELTAZ*	The normalized axial distance between the points at which the calculations are done.
DFDEP*	The depth of the defect in the tube in inches.
DFDIAM*	The diameter of the defect in the tube in inches.
DFM	The magnitude of the normalized impedance change in the coil due to the defect.
DFP	The phase of the normalized impedance change in the coil due to the defect.
DSFI	The imaginary part of the defect sensitivity factor of the coil.
DSFR	The real part of the defect sensitivity factor of the coil.
FREQ*	The operating frequency in hertz.
ISIDE*	The side of the tube where the defect is located. If ISIDE = 1, the defect is on the outside of the tube; if ISIDE = 2, the defect is on the inside of the tube.
L*	The length of each coil. The value is input in inches and normalized by the program.
LOD*	The number of the I/O unit connected to the output data file.
LOE*	The number of the I/O unit connected to a file which contains the diameters and depths of the defects for which calculations are to be performed.
LOU	The number of the I/O unit connected to the printer.
NRT*	The total number of parts centered along the axis of the defect into which the defect is divided to perform the calculations.
NZT*	The total number of different values of the axial distance between the center of the coil and the defect for which calculations are performed.
POW	The depth of the defect expressed as a percentage of wall thickness.
Q6	The inductance in henries of the coil in air.
R1*	The inner radius of each coil. The value is input in inches and normalized by the program.

R2*	The outer radius of each coil. The value is input in inches and normalized by the program.
R3	The mean radius of each coil in inches.
RCL*	The distance between the outside of the coil and the inside wall of the tube. The value is input in inches and normalized by the program.
RD	The radial distance between the center of the coil and the defect (see note 1).
RDT	The normalized depth to the bottom of the defect. A negative number.
RHO1*	The electrical resistivity of the tube in μΩ-cm.
T1*	The thickness of the tube wall. The value is input in inches and normalized by the program.
TRN*	The number of turns in each coil.
U1*	The relative magnetic permeability of the tube.
VOLN	The normalized volume of the defect.
WUSRR	The product of the angular operating frequency, the magnetic permeability of the tube, the electrical conductivity of the tube, and the square of the mean radius of the coil.
ZD	The axial distance between the center of the coil and the defect.
ZNDFI	The imaginary part of the normalized impedance change in the coil due to the defect.
ZNDFR	The real part of the normalized impedance change in the coil due to the defect.
ZNIM	The imaginary part of the normalized impedance of the coil when no defects are present.
ZNRL	The real part of the normalized impedance of the coil when no defects are present.

<u>Notes</u>

1. The program variable RD does not give the radial distance between the actual defect and the center of the coil; it gives the radial distance between the part of the defect with which the program is working at any time and the center of the coil.

Integration Section of Program DFBORAR

Symbol definitions

The following are definitions of the symbols used to describe the program variables which appear in the integration section of the program. All lengths are normalized unless otherwise noted.

α	Integration variable
a	Inner radius of the tube
β_1	$(\alpha^2+j\omega\mu\sigma_1\bar{r}^2)^{1/2}/\mu$
b	Outer radius of the tube
c	Distance between the coil centers
$I(x_2,x_1)$	Integral of $xI_1(x)$ with respect to x from αx_1 to αx_2
$I_0(x)$	Modified Bessel function of the first kind of order 0
$I_1(x)$	Modified Bessel function of the first kind of order 1
$J(x_2,x_1)$	Integral of $xJ_1(x)$ with respect to x from αx_1 to αx_2
$J_1(x)$	Bessel function of the first kind of order 1
$K_0(x)$	Modified Bessel function of the second kind of order 0
$K_1(x)$	Modified Bessel function of the second kind of order 1
ℓ	Length of the coil
μ	Relative magnetic permeability of the tube
r	Radial distance between center of coil and defect
$\bar{r}$	Coil mean radius in inches
r_1	Inner radius of coil
r_2	Outer radius of coil
σ_1	Electrical conductivity of the tube
t_1	Thickness of the tube
z	Axial distance between defect and probe center
ω	Angular operating frequency

Variables appearing in the integration section

Program variable	Symbolic equivalent
A1	$\alpha\ell + \exp(-\alpha\ell) - 1$
BI0A	$I_0(\alpha a)$
BI0B	$I_0(\alpha b)$
BI1A	$I_1(\alpha a)$
BI1B	$I_1(\alpha b)$
BI1IA	$\mathrm{Im}[I_1(\alpha_1 a)]$
BI1IB	$\mathrm{Im}[I_1(\alpha_1 b)]$
BI1IR	$\mathrm{Im}[I_1(\alpha_1 r)]$
BI1RA	$\mathrm{Re}[I_1(\alpha_1 a)]$
BI1RB	$\mathrm{Re}[I_1(\alpha_1 b)]$

BI1RR	$\mathrm{Re}[I_1(\alpha_1 r)]$
BK0A	$K_0(\alpha a)$
BK0B	$K_0(\alpha b)$
BK1A	$K_1(\alpha a)$
BK1B	$K_1(\alpha b)$
BK1IA	$\mathrm{Im}[K_1(\alpha_1 a)]$
BK1IB	$\mathrm{Im}[K_1(\alpha_1 b)]$
BK1IR	$\mathrm{Im}[K_1(\alpha_1 r)]$
BK1RA	$\mathrm{Re}[K_1(\alpha_1 a)]$
BK1RB	$\mathrm{Re}[K_1(\alpha_1 b)]$
BK1RR	$\mathrm{Re}[K_1(\alpha_1 r)]$
DDI	$\mathrm{Im}\{[\beta_2 bK_0(\alpha_2 b)K_1(\alpha_1 b)-\beta_1 bK_0(\alpha_1 b)K_1(\alpha_2 b)]$ $[\beta_1 aI_0(\alpha_1 a)I_1(\alpha a)-\alpha aI_0(\alpha a)I_1(\alpha_1 a)]$ + $[\alpha aI_0(\alpha a)K_1(\alpha_1 a)+\beta_1 aK_0(\alpha_1 a)I_1(\alpha a)]$ $[\beta_1 bI_0(\alpha_1 b)K_1(\alpha_2 b)+\beta_2 bK_0(\alpha_2 b)I_1(\alpha_1 b)]\}$
DDR	$\mathrm{Re}\{[\beta_2 bK_0(\alpha_2 b)K_1(\alpha_1 b)-\beta_1 bK_0(\alpha_1 b)K_1(\alpha_2 b)]$ $[\beta_1 aI_0(\alpha_1 a)I_1(\alpha a)-\alpha aI_0(\alpha a)I_1(\alpha_1 a)]$ + $[\alpha aI_0(\alpha a)K_1(\alpha_1 a)+\beta_1 aK_0(\alpha_1 a)I_1(\alpha a)]$ $[\beta_1 bI_0(\alpha_1 b)K_1(\alpha_2 b)+\beta_2 bK_0(\alpha_2 b)I_1(\alpha_1 b)]\}$
DFR	$\mathrm{Re}\ \{\ K_1(\alpha_1 r)[\beta_1 bI_0(\alpha_1 b)K_1(\alpha_2 b)+\beta_2 bK_0(\alpha_2 b)I_1(\alpha_1 b)]$ − $I_1(\alpha_1 r)[\beta_2 bK_0(\alpha_2 b)K_1(\alpha_1 b)-\beta_1 bK_0(\alpha_1 b)K_1(\alpha_2 b)]\}$ $\div\ \{[\beta_2 bK_0(\alpha_2 b)K_1(\alpha_1 b)-\beta_1 bK_0(\alpha_1 b)K_1(\alpha_2 b)]$ $[\beta_1 aI_0(\alpha_1 a)I_1(\alpha a)-\alpha aI_0(\alpha a)I_1(\alpha_1 a)]$ + $[\alpha aI_0(\alpha a)K_1(\alpha_1 a)+\beta_1 aK_0(\alpha_1 a)I_1(\alpha a)]$ $[\beta_1 bI_0(\alpha_1 b)K_1(\alpha_2 b)+\beta_2 bK_0(\alpha_2 b)I_1(\alpha_1 b)]\}$
DI1	$\mathrm{Im}\ [\beta_2 bK_0(\alpha_2 b)K_1(\alpha_1 b)-\beta_1 bK_0(\alpha_1 b)K_1(\alpha_2 b)]$
DI2	$\mathrm{Im}\ [\beta_1 aI_0(\alpha_1 a)I_1(\alpha a)-\alpha aI_0(\alpha a)I_1(\alpha_1 a)]$
DI3	$\mathrm{Im}\ [\alpha aI_0(\alpha a)K_1(\alpha_1 a)+\beta_1 aK_0(\alpha_1 a)I_1(\alpha a)]$
DI4	$\mathrm{Im}\ [\beta_1 bI_0(\alpha_1 b)K_1(\alpha_2 b)+\beta_2 bK_0(\alpha_2 b)I_1(\alpha_1 b)]$
DNI	$\mathrm{Im}\ \{\ K_1(\alpha_1 r)[\beta_1 bI_0(\alpha_1 b)K_1(\alpha_2 b)+\beta_2 bK_0(\alpha_2 b)I_1(\alpha_1 b)]$ − $I_1(\alpha_1 r)[\beta_2 bK_0(\alpha_2 b)K_1(\alpha_1 b)-\beta_1 bK_0(\alpha_1 b)K_1(\alpha_2 b)]\}$

DNR	$\mathrm{Re}\ \{\ K_1(\alpha_1 r)\,[\beta_1 b I_0(\alpha_1 b) K_1(\alpha_2 b) + \beta_2 b K_0(\alpha_2 b) I_1(\alpha_1 b)]\ -\ I_1(\alpha_1 r)\,[\beta_2 b K_0(\alpha_2 b) K_1(\alpha_1 b) - \beta_1 b K_0(\alpha_1 b) K_1(\alpha_2 b)]\,\}$
DR1	$\mathrm{Re}\ [\beta_2 b K_0(\alpha_2 b) K_1(\alpha_1 b) - \beta_1 b K_0(\alpha_1 b) K_1(\alpha_2 b)]$
DR2	$\mathrm{Re}\ [\beta_1 a I_0(\alpha_1 a) I_1(\alpha a) - \alpha a I_0(\alpha a) I_1(\alpha_1 a)]$
DR3	$\mathrm{Re}\ [\alpha a I_0(\alpha a) K_1(\alpha_1 a) + \beta_1 a K_0(\alpha_1 a) I_1(\alpha a)]$
DR4	$\mathrm{Re}\ [\beta_1 b I_0(\alpha_1 b) K_1(\alpha_2 b) + \beta_2 b K_0(\alpha_2 b) I_1(\alpha_1 b)]$
S1	$d\alpha$
SNI	$\mathrm{Im}\{-I_1(\alpha_1 a)\,[\beta_2 b K_0(\alpha_2 b) K_1(\alpha_1 b) - \beta_1 b K_0(\alpha_1 b) K_1(\alpha_2 b)] + K_1(\alpha_1 a)\,[\beta_1 b I_0(\alpha_1 b) K_1(\alpha_2 b) + \beta_2 b K_0(\alpha_2 b) I_1(\alpha_1 b)]\}$
SNR	$\mathrm{Re}\{-I_1(\alpha_1 a)\,[\beta_2 b K_0(\alpha_2 b) K_1(\alpha_1 b) - \beta_1 b K_0(\alpha_1 b) K_1(\alpha_2 b)] + K_1(\alpha_1 a)\,[\beta_1 b I_0(\alpha_1 b) K_1(\alpha_2 b) + \beta_2 b K_0(\alpha_2 b) I_1(\alpha_1 b)]\}$

SSR

$$\mathrm{Re}\ \frac{1}{I_1(\alpha a)}\Bigg[-I_1(\alpha_1 a)\,[\beta_2 b K_0(\alpha_2 b) K_1(\alpha_1 b) - \beta_1 b K_0(\alpha_1 b) K_1(\alpha_2 b)] + K_1(\alpha_1 a)\,[\beta_1 b I_0(\alpha_1 b) K_1(\alpha_2 b) + \beta_2 b K_0(\alpha_2 b) I_1(\alpha_1 b)]\Bigg]$$

$$\Bigg[\ [\beta_2 b K_0(\alpha_2 b) K_1(\alpha_1 b) - \beta_1 b K_0(\alpha_1 b) K_1(\alpha_2 b)]$$

$$[\beta_1 a I_0(\alpha_1 a) I_1(\alpha a) - \alpha a I_0(\alpha a) I_1(\alpha_1 a)]$$

$$[\alpha a I_0(\alpha a) K_1(\alpha_1 a) + \beta_1 a K_0(\alpha_1 a) I_1(\alpha a)]$$

$$[\beta_1 b I_0(\alpha_1 b) K_1(\alpha_2 b) + \beta_2 b K_0(\alpha_2 b) I_1(\alpha_1 b)]\Bigg]^{-1} - \frac{K_1(\alpha a)}{I_1(\alpha a)}$$

X	α
X1	$\mathrm{Re}[\alpha_1]$
X1A	$\mathrm{Re}[\alpha_1 a]$
X1B	$\mathrm{Re}[\alpha_1 b]$
X1R	$\mathrm{Re}[\alpha_1 r]$
XA	αa
XB	αb
XF	$\frac{4}{\pi}\,\frac{1}{\alpha^3}\ I(r_2, r_1)\ \sin(\alpha \ell/2)\ d\alpha$

XF1	$\frac{4}{\pi}\frac{1}{\alpha^3} I(r_2,r_1)\ \sin(\alpha\ell/2)\ sin(\frac{\alpha c}{2})\ d\alpha$
XF2	$\frac{4}{\pi}\frac{1}{\alpha^3} I(r_2,r_1)\ \sin(\alpha\ell/2)\ cos(\frac{\alpha c}{2})\ d\alpha$
XFACT	$\frac{1}{\alpha^3} I(r_2,r_1)\sin(\alpha\ell/2)$
XFACT1	$\frac{4}{\pi}\frac{1}{\alpha^3} I(r_2,r_1)\ \sin(\alpha\ell/2)\ sin(\frac{\alpha c}{2})sin(\alpha z)\ d\alpha$
XFACT2	$\frac{4}{\pi}\frac{1}{\alpha^3} I(r_2,r_1)\ \sin(\alpha\ell/2)\ cos(\frac{\alpha c}{2})cos(\alpha z)\ d\alpha$
XIR21	$\frac{1}{\alpha^3} I(r_2,r_1)$
XJR21	$\frac{1}{\alpha^3} J(r_2,r_1)$
XL	$\alpha\ell$
XX	α^2
XXXX	α^4
Y1	$\mathrm{Im}[\alpha_1]$
Y1A	$\mathrm{Im}[\alpha_1 a]$
Y1B	$\mathrm{Im}[\alpha_1 b]$
Y1R	$\mathrm{Im}[\alpha_1 r]$
ZI0IA	$\mathrm{Im}[\alpha_1 a I_0(\alpha_1 a)]$
ZI0IB	$\mathrm{Im}[\alpha_1 b I_0(\alpha_1 b)]$
ZI0IR	$\mathrm{Im}[\alpha_1 r I_0(\alpha_1 r)]$
ZI0RA	$\mathrm{Re}[\alpha_1 a I_0(\alpha_1 a)]$
ZI0RB	$\mathrm{Re}[\alpha_1 b I_0(\alpha_1 b)]$
ZI0RR	$\mathrm{Re}[\alpha_1 r I_0(\alpha_1 r)]$
ZK0IA	$\mathrm{Im}[\alpha_1 a K_0(\alpha_1 a)]$
ZK0IB	$\mathrm{Im}[\alpha_1 b K_0(\alpha_1 b)]$
ZK0IR	$\mathrm{Im}[\alpha_1 r K_0(\alpha_1 r)]$
ZK0RA	$\mathrm{Re}[\alpha_1 a K_0(\alpha_1 a)]$
ZK0RB	$\mathrm{Re}[\alpha_1 b K_0(\alpha_1 b)]$
ZK0RR	$\mathrm{Re}[\alpha_1 r K_0(\alpha_1 r)]$

Sample output

The program DFBORAR calculates the defect signal averaged over the depth of the defect at different distances along the tube. The program can plot the defect impedance as the tube is scanned and pick out the maximum magnitude. Below we show the printer output of a sample run where the maximum signal is printed.

```
 DFBORAR     TIME 15:50: 5  DATE  8/15/89
    IN RAD  OT RAD  LENGTH  RAD CLR WALLTH  % WALL  C TO C  TUB IR TUB OR
ACT 1.2400  1.4900  0.2650  0.0575  0.2200   45.45  0.5150  1.5475 1.7675
NOR 0.9084  1.0916  0.1941  0.0421  0.1612   45.45  0.3773 1.1337  1.2949
 RBAR 1.3650 FREQ= 4.000000E+02 RHO=  3.8400 PERM= 1.000 WUSRR= 98.8670
 NORM IMPD:RL  0.116284 IM  0.580440 AIR IND 1.784137E-02
 NORM DSF:RL-1.0456E-04 IM-2.8334E-04 VOLN 3.0881E-04
 MAXIMUM MAG  0.5610D-04   PHA AT MAX MAG   27.37   OD DEFECT
```

Partial listing of output defect axial position (ZD), defect magnitude (DFM) and defect phase (DFP) that is sent to LOD file (either FORT39 or BORSCN.DAT):

```
 0.010       0.4270D-05       30.24
 0.020       0.8517D-05       30.27
 0.030       0.1272D-04       30.31
 0.040       0.1684D-04       30.36
 0.050       0.2088D-04       30.42
 0.060       0.2479D-04       30.49
 0.070       0.2857D-04       30.55
 0.080       0.3218D-04       30.60
 0.090       0.3560D-04       30.64
 0.100       0.3881D-04       30.66
 ......
```

If the proper plotting software has been installed, a plot of this data is made on the CRT and a hard copy can be obtained, as shown in Fig. 17. The plot forms one-half the normal Lissajous pattern one gets in an eddy-current test with a differential bobbin coil. The second half can be obtained by reflecting the signal in the -*z* direction, since the signal is anti-symmetric. The phases and magnitudes given in this report are referenced to the X axis being zero phase and measured counterclockwise from X axis, which is standard for mathematics and electrical engineering. Standard eddy-current practice is to measure the phase shift in a clockwise direction from the -X axis. Furthermore, the phase is rotated until the probe wobble/fill factor/lift-off variations lie in a horizontal direction.

ORNL-DWG 90-8097

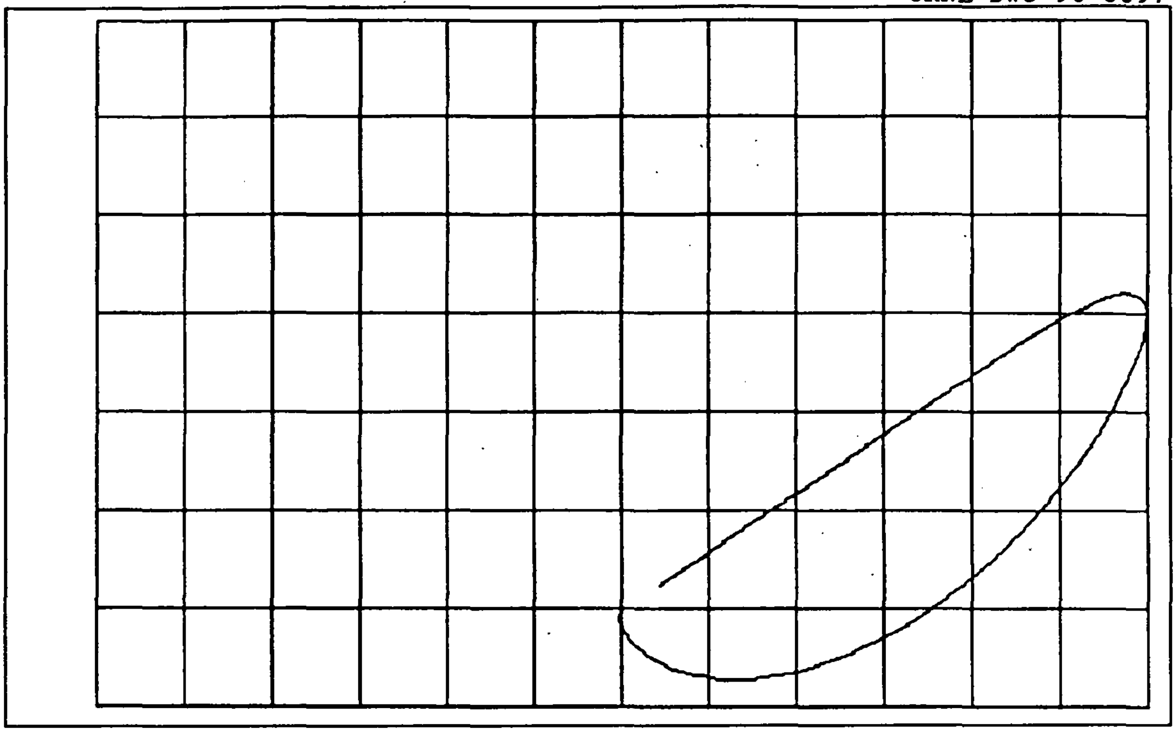

Fig. 17. Plot of defect signal on complex impedance plane as the defect is moved from the probe center in the plus *z* direction.

Listing

```
      PROGRAM DFBORAR
C     VERSION August 16, 1989
C     Program to calculate the normalized impedance change
C     for a defect in a single tube on a differential boreside probe
C     as the probe scans past the defect.  The program averages the effect
C       of the defect over the depth of the defect.
C     Z=0.0 AT CENTER OF PROBE.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 L
      CHARACTER*1 FF
      DIMENSION S1(6),S2(6),ERR(6)
      DIMENSION CX(200),CY(200),XFACT1(200),XFACT2(200)
      DIMENSION SMZDFR1A(200,30),SMZDFI1A(200,30)
      DIMENSION SMZDFR2A(200,30),SMZDFI2A(200,30)
      DATA LOU/8/,PI/3.141592653/,LOD/39/,LOE/38/
      DATA S1/.005,.02,.05,.1,.1,.5/
      DATA S2/1.0,2.0,5.0,10.0,50.0,200.0/
      DATA ERR/.1,.01,.001,1.E-4,1.E-5,1.E-10/
      DATA FREQ/ 4.0E2/,RHO1/3.84/,U1/1.0/
      DATA TRN/400./,ISIDE/1/,DELTAZ/0.01/
      DATA MODE/16/,NZT/100/,NRT/20/
      DATA OIM/50/,ORL/330/
      FF=CHAR(12)
C      OPEN(LOD,FILE='BORSCN.DAT',STATUS='NEW')
c      OPEN(LOE,FILE='BORDAT.DAT',STATUS='OLD')
   11 XMAX=0.
      YMAX=0.
      DFMMAX=0.
      DO 14 NZ=1,NZT
      DO 12 NR=1,NRT
      SMZDFR1A(NZ,NR)=0.
      SMZDFI1A(NZ,NR)=0.
      SMZDFR2A(NZ,NR)=0.
      SMZDFI2A(NZ,NR)=0.
   12 CONTINUE
   14 CONTINUE
c      READ(LOE,*,END=1001)DFDIAM,DFDEP
      dfdiam=0.1
      dfdep=0.1
C     TIME AND DATE ARE PRINTED
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT(' DFBORAR    TIME ',I2,':',I2,':'
     *,I2,'  DATE ',I2,'/',I2,'/',I2)
      WRITE(LOU,5)
    5 FORMAT(5X,'IN RAD',2X,'OT RAD',2X,'LENGTH',2X,'RAD CLR'
     *,1X,'WALLTH',2X,'% WALL',2X,'C TO C',2X,'TUB IR',2X,'TUB OR')
```

```
      R1=1.24
      R2=1.49
      L=0.265
      C=0.515
      RCL=0.0575
      T1=0.22
      POW=(DFDEP/T1)*100.
      A=R2+RCL
      B=A+T1
      RDT=-DFDEP
      R3=0.5*(R1+R2)
      WRITE(LOU,10)R1,R2,L,RCL,T1,POW,C,A,B
      R1=R1/R3
      R2=2.0-R1
      L=L/R3
      C=C/R3
      RCL=RCL/R3
      RDT=RDT/R3
      T1=T1/R3
      A=A/R3
      B=B/R3
C      VOLN=0.1666667*PI*(DFDIAM/R3)*(DFDIAM/R3)*(DFDIAM/R3)
      VOLN=PI*DFDIAM*DFDIAM*DFDEP/(4.*R3*R3*R3)
      WUSRR=0.5093979*U1*R3*R3*FREQ/RHO1
      WRITE(LOU,15)R1,R2,L,RCL,T1,POW,C,A,B
   10 FORMAT('ACT ',5(F7.4,1X),F7.2,1X,3(F7.4,1X))
   15 FORMAT('NOR ',5(F7.4,1X),F7.2,1X,3(F7.4,1X))
      WRITE(LOU,20)R3,FREQ,RHO1,U1,WUSRR
   20 FORMAT(' RBAR',F7.4,' FREQ= ',1PE13.6,' RHO=',0PF9.4,
     *' PERM=',F7.3,' WUSRR=',F11.4)
      CALL QSMODE(MODE)
      CALL GRID
C      DO 1000 NZ=1,NZT
C      ZD=DELTAZ*NZ
      SMAIR=0.0
      SMIMPR=0.0
      SMIMPI=0.0
      SMZDFR1=0.0
      SMZDFI1=0.0
      SMZDFR2=0.0
      SMZDFI2=0.0
C     AIR=0.
C     AII=0.
C     AUR=0.
C     AUI=0.
      B1=0.0
      B2=S2(1)
      DO 100 JKL=1,6
   30 RI9=SMAIR
      X=B1-0.5*S1(JKL)
C     DETERMINE NUMBER OF STEPS,ISTEPS,BASED ON STEP SIZE&INTERVAL
```

```
      ISTEPS=DNINT((B2-B1)/S1(JKL))
      DO 90 I=1,ISTEPS
      X=X+S1(JKL)
      CALL IJBSSL(X,R1,R2,XIR21,XJR21)
      XL=X*L
      IF(XL.GT.5.0E-3) GO TO 60
      A1=XL*XL*(0.5-XL/6.0)
      GO TO 80
   60 IF(XL.GT.75.0) GO TO 70
      A1=XL+DEXP(-XL)-1.0
      GO TO 80
   70 A1=XL-1.0
   80 SMAIR=SMAIR+2.*XJR21*XJR21*A1*S1(JKL)
      IF(X.GT.160.)GO TO 90
      XX=X*X
      XXXX=XX*XX
      X1=DSQRT(0.5*(XX+DSQRT(XXXX+WUSRR*WUSRR)))/U1
      Y1=WUSRR/(2*X1*U1*U1)
      XA=X*A
      XB=X*B
      X1A=X1*A
      Y1A=Y1*A
      X1B=X1*B
      Y1B=Y1*B
      CALL CMDBES(X1A,Y1A,ZIORA,ZIOIA,ZKORA,ZKOIA,BI1RA,BI1IA
     *,BK1RA,BK1IA)
      CALL CMDBES(X1B,Y1B,ZIORB,ZIOIB,ZKORB,ZKOIB,BI1RB,BI1IB
     *,BK1RB,BK1IB)
      CALL BESI(XA,BIOA,BI1A)
      CALL BESK(XA,BKOA,BK1A)
      CALL BESI(XB,BIOB,BI1B)
      CALL BESK(XB,BKOB,BK1B)
      DR1=XB*BKOB*BK1RB-ZKORB*BK1B/U1
      DI1=XB*BKOB*BK1IB-ZKOIB*BK1B/U1
      DR2=ZIORA*BI1A/U1-XA*BIOA*BI1RA
      DI2=ZIOIA*BI1A/U1-XA*BIOA*BI1IA
      DR3=XA*BIOA*BK1RA+ZKORA*BI1A/U1
      DI3=XA*BIOA*BK1IA+ZKOIA*BI1A/U1
      DR4=ZIORB*BK1B/U1+XB*BKOB*BI1RB
      DI4=ZIOIB*BK1B/U1+XB*BKOB*BI1IB
      DDR=DR1*DR2-DI1*DI2+DR3*DR4-DI3*DI4
      DDI=DR1*DI2+DR2*DI1+DR3*DI4+DR4*DI3
      SNR=-BI1RA*DR1+BI1IA*DI1+BK1RA*DR4-BK1IA*DI4
      SNI=-BI1IA*DR1-BI1RA*DI1+BK1RA*DI4+BK1IA*DR4
      DEN=DDR*DDR+DDI*DDI
      SSR=((SNR*DDR+SNI*DDI)/DEN-BK1A)/BI1A
      SSI=(SNI*DDR-SNR*DDI)/(BI1A*DEN)
      XFACT=XIR21*DSIN(XL/2.)
      SMIMPR=SMIMPR+8.*XFACT*XFACT*S1(JKL)*SSR/PI
      SMIMPI=SMIMPI+8.*XFACT*XFACT*S1(JKL)*SSI/PI
      XF=4.*XFACT*S1(JKL)/PI
```

```
      XF1=XF*DSIN(0.5*X*C)
      XF2=XF*DCOS(0.5*X*C)
      DO 89 NR=1,NRT
      RD=(REAL(NR)-0.5)*(RDT/REAL(NRT))
      IF(ISIDE.EQ.1) THEN
      RD=B+RD
      ELSE
      RD=A-RD
      END IF
      X1R=X1*RD
      Y1R=Y1*RD

      CALL CMDBES(X1R,Y1R,ZIORR,ZIOIR,ZKORR,ZKOIR,BI1RR,BI1IR
     *,BK1RR,BK1IR)

      DNR=BK1RR*DR4-BK1IR*DI4-BI1RR*DR1+BI1IR*DI1
      DNI=BK1IR*DR4+BK1RR*DI4-BI1IR*DR1-BI1RR*DI1
      DFR=(DNR*DDR+DNI*DDI)/DEN
      DFI=(DNI*DDR-DNR*DDI)/DEN

      DO 88 NZ=1,NZT
      IF(NR.GT.1)GO TO 87
      ZD=DELTAZ*NZ
      XFACT1(NZ)=XF1*DSIN(X*ZD)
      XFACT2(NZ)=XF2*DCOS(X*ZD)
   87 SMZDFR1A(NZ,NR)=SMZDFR1A(NZ,NR)+XFACT1(NZ)*DFR
      SMZDFI1A(NZ,NR)=SMZDFI1A(NZ,NR)+XFACT1(NZ)*DFI
      SMZDFR2A(NZ,NR)=SMZDFR2A(NZ,NR)+XFACT2(NZ)*DFR
      SMZDFI2A(NZ,NR)=SMZDFI2A(NZ,NR)+XFACT2(NZ)*DFI
   88 CONTINUE
   89 CONTINUE
   90 CONTINUE
      B1=B2
      B2=B2+S2(JKL)
      CHECK=(SMAIR-RI9)/SMAIR
      IF(ABS(CHECK).GT.ERR(JKL)) GO TO 30
  100 CONTINUE
      DO 990 NZ=1,NZT
      ZD=DELTAZ*NZ
      SMZDFR1=0.
      SMZDFI1=0.
      SMZDFR2=0.
      SMZDFI2=0.
      DO 120 NR=1,NRT
      SMZDFR1=SMZDFR1+SMZDFR1A(NZ,NR)
      SMZDFI1=SMZDFI1+SMZDFI1A(NZ,NR)
      SMZDFR2=SMZDFR2+SMZDFR2A(NZ,NR)
      SMZDFI2=SMZDFI2+SMZDFI2A(NZ,NR)
  120 CONTINUE
      SMZDFR1=SMZDFR1/NRT
      SMZDFI1=SMZDFI1/NRT
```

```
      SMZDFR2=SMZDFR2/NRT
      SMZDFI2=SMZDFI2/NRT
C      WRITE(LOU,*)NZ,SMZDFR1,SMZDFI1
C      WRITE(LOU,*)'      ',SMZDFR2,SMZDFI2
      DSFR=-1.5*WUSRR*(SMZDFR1*SMZDFR2-SMZDFI1*SMZDFI2)/(SMAIR*PI)
      DSFI=-1.5*WUSRR*(SMZDFR1*SMZDFI2+SMZDFR2*SMZDFI1)/(SMAIR*PI)
  135 Q6=0.0254*4.0E-7*TRN*TRN*R3*PI*PI*SMAIR/(L*(R2-R1))**2
      ZNIM=SMIMPR/SMAIR+1.0
      ZNRL=-SMIMPI/SMAIR
      ZNDFR=VOLN*DSFR
      ZNDFI=VOLN*DSFI
      IF(NZ.EQ.1.) WRITE(LOU,140)ZNRL,ZNIM,Q6
      DFM=VOLN*SQRT(DSFR*DSFR+DSFI*DSFI)
      IF(ZD.EQ.0.0) THEN
      DFP=0.
      ELSE
      DFP=ATAN2(DSFI,DSFR)
      END IF
      CX(NZ)=DFM*COS(DFP)
      CY(NZ)=DFM*SIN(DFP)
      DFP=DFP*(180./PI)
      IF(ABS(CX(NZ)).GT.XMAX) XMAX=ABS(CX(NZ))
      IF(ABS(CY(NZ)).GT.YMAX) YMAX=ABS(CY(NZ))
      IF(DFM.GT.DFMMAX) THEN
      DFMMAX=DFM
      DFPMMAX=DFP
      END IF
      WRITE(LOD,162)ZD,DFM,DFP
  990 CONTINUE

      GIM=300./YMAX
      GRL=300./XMAX
      IF(GIM.GT.GRL) THEN
      GIM=GRL
      ELSE
      GRL=GIM
      END IF
      IM1=GIM*CY(1)+OIM
      IR1=GRL*CX(1)+ORL
C      WRITE(LOU,150)ZNDFR,ZNDFI,DFM,DFP
      WRITE(LOU,160)DSFR,DSFI,VOLN
      WRITE(LOU,164)DFMMAX,DFPMMAX
      IF(ISIDE.EQ.1) THEN
      WRITE(LOU,*)'  OD DEFECT'
      ELSE
      WRITE(LOU,*)'  ID DEFECT'
      END IF

      DO 1000 NZ=2,NZT
      IM2=GIM*CY(NZ)+OIM
      IR2=GRL*CX(NZ)+ORL
```

```
      CALL QLINE(IR1,IM1,IR2,IM2,15)
      IR1=IR2
      IM1=IM2
 1000 CONTINUE
      WRITE(LOU,*)
      WRITE(LOU,*)
      CALL PRTSC
      WRITE(LOU,*)FF
c       GO TO 11

  140 FORMAT(' NORM IMPD:RL',F10.6,' IM',F10.6,
     *' AIR IND',1PE13.6)
C  150 FORMAT(' NORM CHG:RL ',0PF10.6,' IM',0PF10.6,
C     *' MAG',0PF10.6,' PHA ',0PF7.2,\)
  160 FORMAT(' NORM DSF:RL',1PE11.4,' IM',1PE11.4,' VOLN',1PE11.4)
  161 FORMAT('    ZD        MAG         PHA')
  162 FORMAT(F6.3,5X,D11.4,5X,F7.2)
  164 FORMAT(' MAXIMUM MAG ',D11.4,'   PHA AT MAX MAG ',F7.2,\)

 1001 END
```

CIRCUMFERENTIAL ENCIRCLING COIL PROGRAMS

The programs in this section perform functions relating to the effect on an encircling coil of a defect in a single conducting tube. The types of circumferential coils dealt with in this section are absolute encircling and differential encircling coils. Calculations of the normalized impedance change in these coils due to a defect in a single conducting tube are done by programs ABENCAR and DFENCAR. Fig. 18 shows a differential coil encircling a tube.

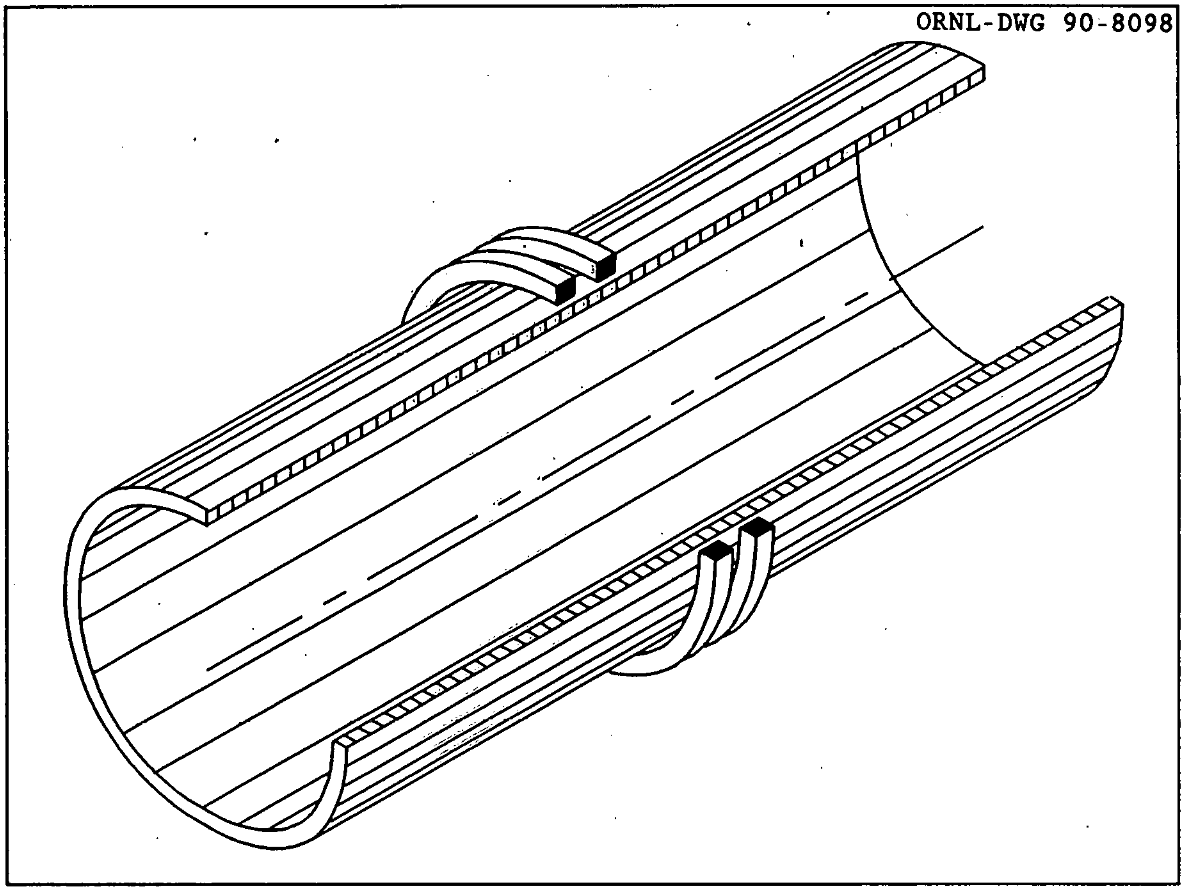

Fig. 18. Differential coil encircling a conducting tube.

ABENCAR calculates impedance change for absolute coil

Program ABENCAR calculates the change in the normalized impedance of an absolute encircling coil due to the presence of a defect in a single conducting tube, as shown in Fig. 19.

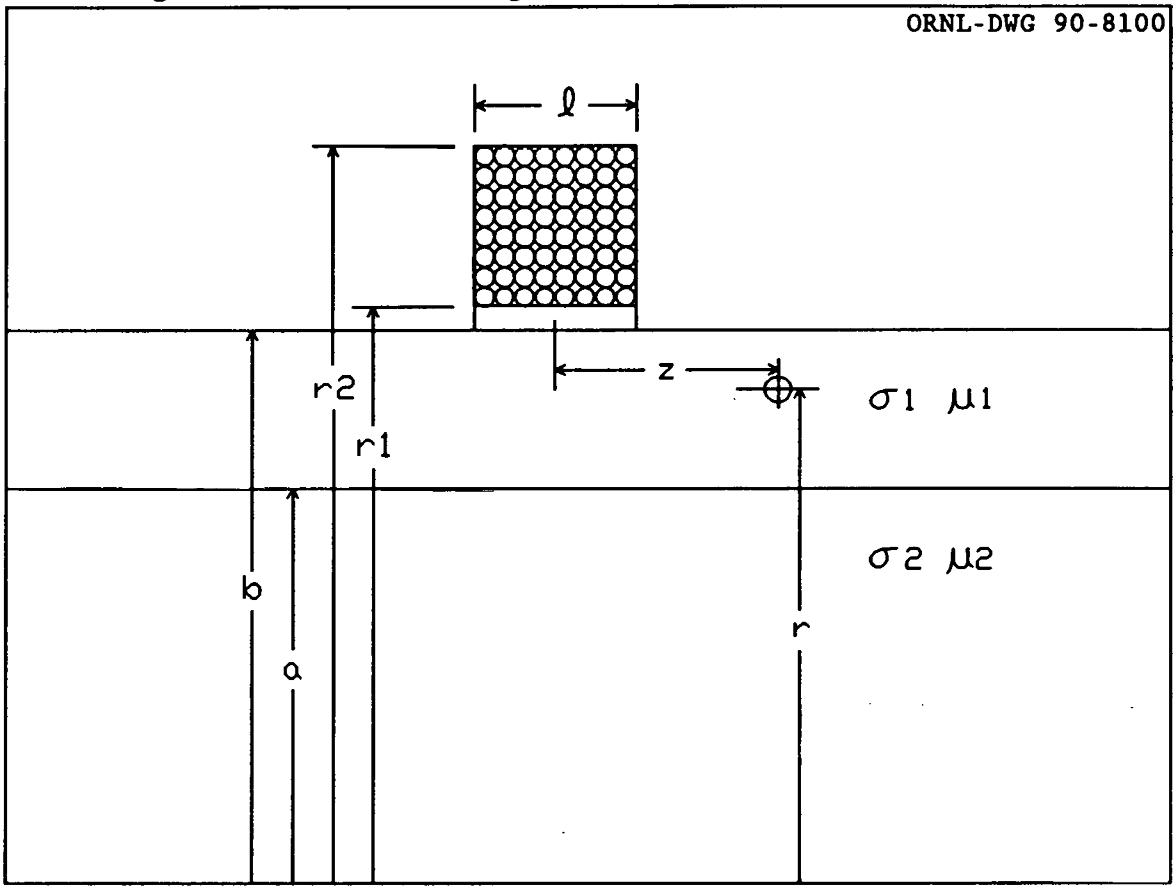

Fig. 19. Absolute coil encircling a tube in the presence of a defect.

The normalized impedance for a coil encircling a tube, without the defect, is:

$$Z_n = \frac{j}{I_{air}} \int_0^{\infty} \left[\frac{8}{\pi\alpha^6} K^2(r_2, r_1) \left\{ \frac{K_1(\alpha_1 b)D_2 + I_1(\alpha_1 b)D_3}{K_1(\alpha b)\,(D_1 D_2 + D_3 D_4)} - \frac{I_1(\alpha b)}{K_1(\alpha b)} \right\} sin^2\left(\frac{\alpha\ell}{2}\right) + I_{air} \right] d\alpha \quad (43)$$

and for the change in the normalized impedance due to the defect we have:

$$Z_{nd}(r,z) = \frac{-3\,(\omega\mu\sigma_1\bar{r}^2)\,Vol_n}{2\pi I_{air}} \left[\int_0^{\infty} \frac{K(r_2, r_1)}{\pi\alpha^3} \left\{ \frac{K_1(\alpha_1 r)D_2 + I_1(\alpha_1 r)D_3}{(D_1 D_2 + D_3 D_4)} \right\} sin\left(\frac{\alpha\ell}{2}\right) 2cos(\alpha z)\, d\alpha \right]^2 \quad (44)$$

where:

$$D_1 = \alpha b K_0(\alpha b) K_1(\alpha_1 b) - \beta_1 b K_0(\alpha_1 b) K_1(\alpha b) \tag{45}$$

$$D_2 = \beta_1 a I_0(\alpha_1 a) I_1(\alpha_2 a) - \beta_2 a I_0(\alpha_2 a) I_1(\alpha_1 a) \tag{46}$$

$$D_3 = \beta_2 a I_0(\alpha_2 a) K_1(\alpha_1 a) + \beta_1 a K_0(\alpha_1 a) I_1(\alpha_2 a) \tag{47}$$

$$D_4 = \beta_1 b I_0(\alpha_1 b) K_1(\alpha b) + \alpha b K_0(\alpha b) I_1(\alpha_1 b) \tag{48}$$

$$K(r_2, r_1) = \int_{\alpha r_1}^{\alpha r_2} x K_1(x) dx \tag{49}$$

$$\alpha_1 = (\alpha^2 + j\omega\mu\sigma_1 \overline{r}^2)^{1/2} \tag{50}$$

and

$$\beta_1 = (\mu_0/\mu_1)(\alpha^2 + j\omega\mu_1\sigma_1\overline{r}^2)^{1/2} \tag{51}$$

The term I_{air} is related to the air inductance of the coil and is:

$$I_{air} = \int_0^\infty \frac{1}{\alpha^6} [J(r_2, r_1)]^2 \, 2[\alpha\ell + \exp(-\alpha\ell) - 1] \, d\alpha \tag{52}$$

where

$$J(r_2, r_1) = \int_{\alpha r_1}^{\alpha r_2} x J_1(x) dx \tag{53}$$

The program performs the calculations for a number of different axial distances between the coil and the defect, and it can divide the defect into a number of parts and calculate the effect of each part separately. In the inner conductor, the value of the conductivity σ_2 is taken as zero, and the relative permeability μ_2 is taken as unity, so that β_2 becomes α. These values are used in the program, but not in the derivation.

<u>Variables</u>

A number of the variables which occur mainly inside the integration loops do not correspond to anything physical. These variables are described in a section at the end. Starred variables must be assigned by the user.

A	The normalized inner radius of the tube.
B	The normalized outer radius of the tube.
DELTAZ*	The normalized axial distance between the points at which the calculations are done.
DFDEP*	The depth of the defect in the tube in inches.
DFDIAM*	The diameter of the defect in the tube in inches.
DFM	The magnitude of the normalized impedance change in the coil due to the defect.
DFP	The phase of the normalized impedance change in the coil due to the defect.

DSFI	The imaginary part of the defect sensitivity factor of the coil.
DSFR	The real part of the defect sensitivity factor of the coil.
FREQ*	The operating frequency in hertz.
ISIDE*	The side of the tube where the defect is located. If ISIDE=1, the defect is on the outside of the tube; if ISIDE=2, the defect is on the inside of the tube.
L*	The length of the coil. The value is input in inches and normalized by the program.
LOD*	The number of the I/O unit connected to the output data file.
LOE*	The number of the I/O unit connected to a file which contains the diameters and depths of the defects for which calculations are to be performed.
LOU	The number of the I/O unit connected to the printer.
NRT*	The total number of parts centered along the axis of the defect into which the defect is divided to perform the calculations.
NZT*	The total number of different values of the axial distance between the center of the coil and the defect for which calculations are performed.
POW	The depth of the defect expressed as a percentage of wall thickness.
Q6	The inductance in henries of the coil in air.
R1*	The inner radius of the coil. The value is input in inches and normalized by the program.
R2*	The outer radius of the coil. The value is input in inches and normalized by the program.
R3	The mean radius of the coil in inches.
RCL*	The distance between the outside of the coil and the inside wall of the tube. The value is input in inches and normalized by the program.
RD	The radial distance between the center of the coil and the defect.(See note 1)
RDT	The normalized depth to the bottom of the defect. A negative number.
RHO1*	The electrical resistivity of the tube in $\mu\Omega$-cm.
T1*	The thickness of the tube wall. The value is input in inches and normalized by the program.
TRN*	The number of turns in the coil.
U1*	The relative magnetic permeability of the tube.
VOLN	The normalized volume of the defect.
WUSRR	The product of the angular operating frequency, the magnetic permeability of the tube, the electrical conductivity of the tube, and the square of the mean radius of the coil.
ZD	The axial distance between the center of the coil

	and the defect.
ZNDFI	The imaginary part of the normalized impedance change in the coil due to the defect.
ZNDFR	The real part of the normalized impedance change in the coil due to the defect.
ZNIM	The imaginary part of the normalized impedance of the coil when no defects are present.
ZNRL	The real part of the normalized impedance of the coil when no defects are present.

Integration Section of Program ABENCAR

Symbol definitions

The following are definitions of the symbols used to describe the program variables which appear in the integration section of the program. All lengths are normalized unless otherwise noted.

α	Integration variable
a	Inner radius of the tube
β_1	$(\alpha^2+j\omega\mu\sigma_1\bar{r}^2)^{1/2}/\mu$
b	Outer radius of the tube
$I_0(x)$	Modified Bessel function of the first kind of order 0
$I_1(x)$	Modified Bessel function of the first kind of order 1
$J(x_2,x_1)$	Integral of $xJ_1(x)$ with respect to x from αx_1 to αx_2
$J_1(x)$	Bessel function of the first kind of order 1
$K(x_2,x_1)$	Integral of $xK_1(x)$ with respect to x from αx_1 to αx_2
$K_0(x)$	Modified Bessel function of the second kind of order 0
$K_1(x)$	Modified Bessel function of the second kind of order 1
ℓ	Length of the coil
μ	Relative magnetic permeability of the tube
σ_1	Electrical conductivity of the tube
$\underline{r}$	Radial distance between center of coil and defect
$\bar{r}$	Coil mean radius in inches
r_1	Inner radius of coil
r_2	Outer radius of coil
z	Axial distance between defect and coil center
ω	Angular operating frequency

Variables appearing in the integration section

Program variable	Symbolic equivalent
A1	$\alpha\ell + \exp(-\alpha\ell) - 1$
BI0A	$I_0(\alpha a)$
BI0B	$I_0(\alpha b)$
BI1A	$I_1(\alpha a)$
BI1B	$I_1(\alpha b)$

BI1IA	$\mathrm{Im}[I_1(\alpha_1 a)]$
BI1IB	$\mathrm{Im}[I_1(\alpha_1 b)]$
BI1IR	$\mathrm{Im}[I_1(\alpha_1 r)]$
BI1RA	$\mathrm{Re}[I_1(\alpha_1 a)]$
BI1RB	$\mathrm{Re}[I_1(\alpha_1 b)]$
BI1RR	$\mathrm{Re}[I_1(\alpha_1 r)]$
BK0A	$K_0(\alpha a)$
BK0B	$K_0(\alpha b)$
BK1A	$K_1(\alpha a)$
BK1B	$K_1(\alpha b)$
BK1IA	$\mathrm{Im}[K_1(\alpha_1 a)]$
BK1IB	$\mathrm{Im}[K_1(\alpha_1 b)]$
BK1IR	$\mathrm{Im}[K_1(\alpha_1 r)]$
BK1RA	$\mathrm{Re}[K_1(\alpha_1 a)]$
BK1RB	$\mathrm{Re}[K_1(\alpha_1 b)]$
BK1RR	$\mathrm{Re}[K_1(\alpha_1 r)]$
DDI	$\mathrm{Im}\{[\alpha b K_0(\alpha b)K_1(\alpha_1 b)-\beta_1 b K_0(\alpha_1 b)K_1(\alpha b)]$ $[\beta_1 a I_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 a I_0(\alpha_2 a)I_1(\alpha_1 a)]\ +$ $[\beta_2 a I_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 a K_0(\alpha_1 a)I_1(\alpha_2 a)]$ $[\beta_1 b I_0(\alpha_1 b)K_1(\alpha b)+\alpha b K_0(\alpha b)I_1(\alpha_1 b)]\}$
DDR	$\mathrm{Re}\{[\alpha b K_0(\alpha b)K_1(\alpha_1 b)-\beta_1 b K_0(\alpha_1 b)K_1(\alpha b)]$ $[\beta_1 a I_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 a I_0(\alpha_2 a)I_1(\alpha_1 a)]\ +$ $[\beta_2 a I_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 a K_0(\alpha_1 a)I_1(\alpha_2 a)]$ $[\beta_1 b I_0(\alpha_1 b)K_1(\alpha b)+\alpha b K_0(\alpha b)I_1(\alpha_1 b)]\}$
DFR	$\mathrm{Re}\ \{\ K_1(\alpha_1 r)[\beta_1 a I_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 a I_0(\alpha_2 a)I_1(\alpha_1 a)]\ +$ $I_1(\alpha_1 r)[\beta_2 a I_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 a K_0(\alpha_1 a)I_1(\alpha_2 a)]\}$ $\div\ \{[\alpha b K_0(\alpha b)K_1(\alpha_1 b)-\beta_1 b K_0(\alpha_1 b)K_1(\alpha b)]$ $[\beta_1 a I_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 a I_0(\alpha_2 a)I_1(\alpha_1 a)]\ +$ $[\beta_2 a I_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 a K_0(\alpha_1 a)I_1(\alpha_2 a)]$ $[\beta_1 b I_0(\alpha_1 b)K_1(\alpha b)+\alpha b K_0(\alpha b)I_1(\alpha_1 b)]\}$
DI1	$\mathrm{Im}[\alpha b K_0(\alpha b)K_1(\alpha_1 b)-\beta_1 b K_0(\alpha_1 b)K_1(\alpha b)]$

DI2	$\mathrm{Im}[\beta_1 a I_0(\alpha_1 a) I_1(\alpha_2 a) - \beta_2 a I_0(\alpha_2 a) I_1(\alpha_1 a)]$
DI3	$\mathrm{Im}[\beta_2 a I_0(\alpha_2 a) K_1(\alpha_1 a) + \beta_1 a K_0(\alpha_1 a) I_1(\alpha_2 a)]$
DI4	$\mathrm{Im}[\beta_1 b I_0(\alpha_1 b) K_1(\alpha b) + \alpha b K_0(\alpha b) I_1(\alpha_1 b)]$
DNI	$\mathrm{Im}\{ K_1(\alpha_1 r)[\beta_1 a I_0(\alpha_1 a) I_1(\alpha_2 a) - \beta_2 a I_0(\alpha_2 a) I_1(\alpha_1 a)] + I_1(\alpha_1 r)[\beta_2 a I_0(\alpha_2 a) K_1(\alpha_1 a) + \beta_1 a K_0(\alpha_1 a) I_1(\alpha_2 a)]\}$
DNR	$\mathrm{Re}\{ K_1(\alpha_1 r)[\beta_1 a I_0(\alpha_1 a) I_1(\alpha_2 a) - \beta_2 a I_0(\alpha_2 a) I_1(\alpha_1 a)] + I_1(\alpha_1 r)[\beta_2 a I_0(\alpha_2 a) K_1(\alpha_1 a) + \beta_1 a K_0(\alpha_1 a) I_1(\alpha_2 a)]\}$
DR1	$\mathrm{Re}[\alpha b K_0(\alpha b) K_1(\alpha_1 b) - \beta_1 b K_0(\alpha_1 b) K_1(\alpha b)]$
DR2	$\mathrm{Re}[\beta_1 a I_0(\alpha_1 a) I_1(\alpha_2 a) - \beta_2 a I_0(\alpha_2 a) I_1(\alpha_1 a)]$
DR3	$\mathrm{Re}[\beta_2 a I_0(\alpha_2 a) K_1(\alpha_1 a) + \beta_1 a K_0(\alpha_1 a) I_1(\alpha_2 a)]$
DR4	$\mathrm{Re}[\beta_1 b I_0(\alpha_1 b) K_1(\alpha b) + \alpha b K_0(\alpha b) I_1(\alpha_1 b)]$
S1	$d\alpha$
SNI	$\mathrm{Im}\{K_1(\alpha_1 b)[\beta_1 a I_0(\alpha_1 a) I_1(\alpha_2 a) - \beta_2 a I_0(\alpha_2 a) I_1(\alpha_1 a)] + I_1(\alpha_1 b)[\beta_2 a I_0(\alpha_2 a) K_1(\alpha_1 a) + \beta_1 a K_0(\alpha_1 a) I_1(\alpha_2 a)]\}$
SNR	$\mathrm{Re}[K_1(\alpha_1 b)[\beta_1 a I_0(\alpha_1 a) I_1(\alpha_2 a) - \beta_2 a I_0(\alpha_2 a) I_1(\alpha_1 a)] + I_1(\alpha_1 b)[\beta_2 a I_0(\alpha_2 a) K_1(\alpha_1 a) + \beta_1 a K_0(\alpha_1 a) I_1(\alpha_2 a)]\}$
SSR	$\mathrm{Re}\ \frac{1}{K_1(\alpha b)}\left[K_1(\alpha_1 b)[\beta_1 a I_0(\alpha_1 a) I_1(\alpha_2 a) - \beta_2 a I_0(\alpha_2 a) I_1(\alpha_1 a)] + I_1(\alpha_1 b)[\beta_2 a I_0(\alpha_2 a) K_1(\alpha_1 a) + \beta_1 a K_0(\alpha_1 a) I_1(\alpha_2 a)]\right] \left[[\alpha b K_0(\alpha b) K_1(\alpha_1 b) - \beta_1 b K_0(\alpha_1 b) K_1(\alpha b)]\ [\beta_1 a I_0(\alpha_1 a) I_1(\alpha_2 a) - \beta_2 a I_0(\alpha_2 a) I_1(\alpha_1 a)] + [\beta_2 a I_0(\alpha_2 a) K_1(\alpha_1 a) + \beta_1 a K_0(\alpha_1 a) I_1(\alpha_2 a)]\ [[\beta_1 b I_0(\alpha_1 b) K_1(\alpha b) + \alpha b K_0(\alpha b) I_1(\alpha_1 b)] \right]^{-1} - \frac{I_1(\alpha b)}{K_1(\alpha b)}$
X	α
X1	$\mathrm{Re}[\alpha_1]$
X1A	$\mathrm{Re}[\alpha_1 a]$
X1B	$\mathrm{Re}[\alpha_1 b]$
X1R	$\mathrm{Re}[\alpha_1 r]$

XA	αa
XB	αb
XF2	$\frac{2}{\pi}\frac{1}{\alpha^3} K(r_2,r_1)\ \sin(\alpha\ell/2)\ d\alpha$
XFACT	$\frac{1}{\alpha^3} K(r_2,r_1)\sin(\alpha\ell/2)$
XFACT2	$\frac{2}{\pi}\frac{1}{\alpha^3} K(r_2,r_1)\ \sin(\alpha\ell/2)\ \cos(\alpha z)d\alpha$
XJR21	$\frac{1}{\alpha^3} J(r_2,r_1)$
XKR21	$\frac{1}{\alpha^3} K(r_2,r_1)$
XL	$\alpha\ell$
XX	α^2
XXXX	α^4
Y1	$\mathrm{Im}[\alpha_1]$
Y1A	$\mathrm{Im}[\alpha_1 a]$
Y1B	$\mathrm{Im}[\alpha_1 b]$
Y1R	$\mathrm{Im}[\alpha_1 r]$
ZI0IA	$\mathrm{Im}[\alpha_1 a I_0(\alpha_1 a)]$
ZI0IB	$\mathrm{Im}[\alpha_1 b I_0(\alpha_1 b)]$
ZI0IR	$\mathrm{Im}[\alpha_1 r I_0(\alpha_1 r)]$
ZI0RA	$\mathrm{Re}[\alpha_1 a I_0(\alpha_1 a)]$
ZI0RB	$\mathrm{Re}[\alpha_1 b I_0(\alpha_1 b)]$
ZI0RR	$\mathrm{Re}[\alpha_1 r I_0(\alpha_1 r)]$
ZK0IA	$\mathrm{Im}[\alpha_1 a K_0(\alpha_1 a)]$
ZK0IB	$\mathrm{Im}[\alpha_1 b K_0(\alpha_1 b)]$
ZK0IR	$\mathrm{Im}[\alpha_1 r K_0(\alpha_1 r)]$
ZK0RA	$\mathrm{Re}[\alpha_1 a K_0(\alpha_1 a)]$
ZK0RB	$\mathrm{Re}[\alpha_1 b K_0(\alpha_1 b)]$
ZK0RR	$\mathrm{Re}[\alpha_1 r K_0(\alpha_1 r)]$

Sample output

Output sent to printer:

```
 ABENCAR   TIME  6:34:51  DATE  8/18/89
     IN RAD  OT RAD  LENGTH  RAD CLR WALLTH  % WALL  TUB IR  TUB OR
ACT  1.7750  2.0670  0.2650  0.0075  0.2200   45.45  1.5475  1.7675
NOR  0.9240  1.0760  0.1379  0.0039  0.1145   45.45  0.8056  0.9201
RBAR 1.9210 FREQ= 4.000000E+02 RHO=  3.8400 PERM= 1.000 WUSRR= 195.8126
 NORM IMPD:RL  0.135056 IM  0.494500 AIR IND 1.826299E-02
 NORM DSF:RL 2.8462E-04 IM-1.0838E-04 VOLN 1.1079E-04
 MAXIMUM MAG  0.1845D-03   PHA AT MAX MAG  87.61   OD DEFECT
```

Partial listing of output defect axial position (ZD), defect magnitude (DFM) and defect phase (DFP) that is sent to LOD file (either FORT39 or ENCSCN.DAT):

0.000	0.1845D-03	87.61
0.010	0.1832D-03	87.43
0.020	0.1793D-03	86.87
0.030	0.1729D-03	85.89
0.040	0.1642D-03	84.46
0.050	0.1537D-03	82.54
0.060	0.1420D-03	80.14
0.070	0.1296D-03	77.30
0.080	0.1172D-03	74.09
0.090	0.1052D-03	70.61
0.100	0.9405D-04	66.98

The program also plots the output on the CRT, and a plot similar to Fig. 17 can be obtained. The phases and magnitudes given in this report are referenced to the X axis being zero phase and measured counterclockwise from X axis, which is standard for mathematics and electrical engineering. Standard eddy-current practice is to measure the phase shift in a clockwise direction from the -X axis. Furthermore, the phase is rotated until the probe wobble/fill factor/lift-off variations lie in a horizontal direction.

Listing

```
      PROGRAM ABENCAR
C     VERSION August 17, 1989
C     Program to calculate the normalized impedance change in
C      an absolute encircling coil due to a defect
C      in a single tube as the coil scans past the defect.  The
C      program averages the effect of the defect over the depth
C      of the defect.
C
C     Z=0.0 AT CENTER OF COIL.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 L
      CHARACTER*1 FF
      DIMENSION S1(6),S2(6),ERR(6)
      DIMENSION CX(0:200),CY(0:200),XFACT2(0:200)
      DIMENSION SMZDFRA(0:200,30),SMZDFIA(0:200,30)
      DATA LOU/8/,PI/3.141592653/,LOD/39/,LOE/38/
      DATA S1/.005,.02,.05,.1,.1,.5/
      DATA S2/1.0,2.0,5.0,10.0,50.0,200.0/
      DATA ERR/.1,.01,.001,1.E-4,1.E-5,1.E-10/
      DATA FREQ/ 4.0E2/,RHO1/3.84/,U1/1.0/
      DATA TRN/325./,ISIDE/1/,DELTAZ/0.01/
      DATA MODE/16/,NZT/100/,NRT/20/
      DATA OIM/50/,ORL/330/
      FF=CHAR(12)
      OPEN(LOD,FILE='ENCSCN.DAT',STATUS='NEW')
c      OPEN(LOE,FILE='ENCDAT.DAT',STATUS='OLD')
   11 XMAX=0.
      YMAX=0.
      DFMMAX=0.
      DO 14 NZ=0,NZT
      DO 12 NR=1,NRT
      SMZDFRA(NZ,NR)=0.
      SMZDFIA(NZ,NR)=0.
   12 CONTINUE
   14 CONTINUE
c      READ(LOE,*,END=1001)DFDIAM,DFDEP
      dfdiam=0.1
      dfdep=0.1
C     TIME AND DATE ARE PRINTED
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT(' ABENCAR   TIME ',I2,':',I2,':',I2
     *,'  DATE ',I2,'/',I2,'/',I2)
      WRITE(LOU,5)
    5 FORMAT(5X,'IN RAD',2X,'OT RAD',2X,'LENGTH',2X,'RAD CLR'
     *,1X,'WALLTH',2X,'% WALL',2X,'TUB IR',2X,'TUB OR')
      R1=1.775
```

```
      R2=2.067
      L=0.265
      RCL=0.0075
      T1=0.22
      POW=(DFDEP/T1)*100.
      B=R1-RCL
      A=B-T1
      RDT=-DFDEP
      R3=0.5*(R1+R2)
      WRITE(LOU,10)R1,R2,L,RCL,T1,POW,A,B
      R1=R1/R3
      R2=2.0-R1
      L=L/R3
      RCL=RCL/R3
      RDT=RDT/R3
      RD=RD/R3
      T1=T1/R3
      A=A/R3
      B=B/R3
C      VOLN=0.1666667*PI*(DFDIAM/R3)*(DFDIAM/R3)*(DFDIAM/R3)
      VOLN=PI*DFDIAM*DFDIAM*DFDEP/(4.*R3*R3*R3)
      WUSRR=0.5093979*U1*R3*R3*FREQ/RHO1
      WRITE(LOU,15)R1,R2,L,RCL,T1,POW,A,B
   10 FORMAT('ACT ',5(F7.4,1X),F7.2,1X,3(F7.4,1X))
   15 FORMAT('NOR ',5(F7.4,1X),F7.2,1X,3(F7.4,1X))
      WRITE(LOU,20)R3,FREQ,RHO1,U1,WUSRR
   20 FORMAT(' RBAR',F7.4,' FREQ= ',1PE13.6,' RHO=',0PF9.4,
     *' PERM=',F7.3,' WUSRR=',F11.4)
      CALL QSMODE(MODE)
      CALL GRID
      SMAIR=0.0
      SMIMPR=0.0
      SMIMPI=0.0
      SMZDFR=0.0
      SMZDFI=0.0
C     AIR=0.
C     AII=0.
C     AUR=0.
C     AUI=0.
      B1=0.0
      B2=S2(1)
      DO 100 JKL=1,6
   30 RI9=SMAIR
      X=B1-0.5*S1(JKL)
C     DETERMINE NUMBER OF STEPS,ISTEPS,BASED ON STEP SIZE&INTERVAL
      ISTEPS=DNINT((B2-B1)/S1(JKL))
      DO 90 I=1,ISTEPS
      X=X+S1(JKL)
      CALL KJBSSL(X,R1,R2,XKR21,XJR21)
      XL=X*L
      IF(XL.GT.5.0E-3) GO TO 60
```

```
      A1=XL*XL*(0.5-XL/6.0)
      GO TO 80
   60 IF(XL.GT.75.0) GO TO 70
      A1=XL+DEXP(-XL)-1.0
      GO TO 80
   70 A1=XL-1.0
   80 SMAIR=SMAIR+2.*XJR21*XJR21*A1*S1(JKL)
      IF(X.GT.160.)GO TO 90
      XX=X*X
      XXXX=XX*XX
      X1=DSQRT(0.5*(XX+DSQRT(XXXX+WUSRR*WUSRR)))/U1
      Y1=WUSRR/(2*X1*U1*U1)
      XA=X*A
      XB=X*B
      X1A=X1*A
      Y1A=Y1*A
      X1B=X1*B
      Y1B=Y1*B
      CALL CMDBES(X1A,Y1A,ZIORA,ZIOIA,ZKORA,ZKOIA,BI1RA,BI1IA
     *,BK1RA,BK1IA)
      CALL CMDBES(X1B,Y1B,ZIORB,ZIOIB,ZKORB,ZKOIB,BI1RB,BI1IB
     *,BK1RB,BK1IB)
      CALL BESI(XA,BIOA,BI1A)
      CALL BESK(XA,BKOA,BK1A)
      CALL BESI(XB,BIOB,BI1B)
      CALL BESK(XB,BKOB,BK1B)
      DR1=XB*BKOB*BK1RB-ZKORB*BK1B/U1
      DI1=XB*BKOB*BK1IB-ZKOIB*BK1B/U1
      DR2=ZIORA*BI1A/U1-XA*BIOA*BI1RA
      DI2=ZIOIA*BI1A/U1-XA*BIOA*BI1IA
      DR3=XA*BIOA*BK1RA+ZKORA*BI1A/U1
      DI3=XA*BIOA*BK1IA+ZKOIA*BI1A/U1
      DR4=ZIORB*BK1B/U1+XB*BKOB*BI1RB
      DI4=ZIOIB*BK1B/U1+XB*BKOB*BI1IB
      DDR=DR1*DR2-DI1*DI2+DR3*DR4-DI3*DI4
      DDI=DR1*DI2+DR2*DI1+DR3*DI4+DR4*DI3
      SNR=BK1RB*DR2-BK1IB*DI2+BI1RB*DR3-BI1IB*DI3
      SNI=BK1RB*DI2+BK1IB*DR2+BI1RB*DI3+BI1IB*DR3
      DEN=DDR*DDR+DDI*DDI
      SSR=((SNR*DDR+SNI*DDI)/DEN-BI1B)/BK1B
      SSI=(SNI*DDR-SNR*DDI)/(BK1B*DEN)
      XFACT=XKR21*DSIN(XL/2.)
      SMIMPR=SMIMPR+8.*XFACT*XFACT*S1(JKL)*SSR/PI
      SMIMPI=SMIMPI+8.*XFACT*XFACT*S1(JKL)*SSI/PI
      XF2=2.*XFACT*S1(JKL)/PI

      DO 89 NR=1,NRT
      RD=(REAL(NR)-0.5)*(RDT/REAL(NRT))
      IF(ISIDE.EQ.1) THEN
      RD=B+RD
      ELSE
```

```
      RD=A-RD
      END IF
      X1R=X1*RD
      Y1R=Y1*RD

      CALL CMDBES(X1R,Y1R,ZIORR,ZIOIR,ZKORR,ZKOIR,BI1RR,BI1IR
     *,BK1RR,BK1IR)

      DNR=BK1RR*DR2-BK1IR*DI2+BI1RR*DR3-BI1IR*DI3
      DNI=BK1IR*DR2+BK1RR*DI2+BI1IR*DR3+BI1RR*DI3
      DFR=(DNR*DDR+DNI*DDI)/DEN
      DFI=(DNI*DDR-DNR*DDI)/DEN

      DO 88 NZ=0,NZT
      IF(NR.GT.1)GO TO 87
      ZD=DELTAZ*NZ
      XFACT2(NZ)=XF2*DCOS(X*ZD)
   87 SMZDFRA(NZ,NR)=SMZDFRA(NZ,NR)+XFACT2(NZ)*DFR
      SMZDFIA(NZ,NR)=SMZDFIA(NZ,NR)+XFACT2(NZ)*DFI
   88 CONTINUE
   89 CONTINUE
   90 CONTINUE
      B1=B2
      B2=B2+S2(JKL)
      CHECK=(SMAIR-RI9)/SMAIR
      IF(ABS(CHECK).GT.ERR(JKL)) GO TO 30
  100 CONTINUE
      DO 990 NZ=0,NZT
      ZD=DELTAZ*NZ
      SMZDFR=0.
      SMZDFI=0.
      DO 120 NR=1,NRT
      SMZDFR=SMZDFR+SMZDFRA(NZ,NR)
      SMZDFI=SMZDFI+SMZDFIA(NZ,NR)
  120 CONTINUE
      SMZDFR=SMZDFR/NRT
      SMZDFI=SMZDFI/NRT
C       WRITE(LOU,*)NZ,SMZDFR,SMZDFI
      DSFR=-1.5*WUSRR*(SMZDFR*SMZDFR-SMZDFI*SMZDFI)/(SMAIR*PI)
      DSFI=-1.5*WUSRR*2.0*SMZDFR*SMZDFI/(SMAIR*PI)
  135 Q6=0.0254*4.0E-7*TRN*TRN*R3*PI*PI*SMAIR/(L*(R2-R1))**2
      ZNIM=SMIMPR/SMAIR+1.0
      ZNRL=-SMIMPI/SMAIR
      ZNDFR=VOLN*DSFR
      ZNDFI=VOLN*DSFI
      IF(NZ.EQ.0) WRITE(LOU,140)ZNRL,ZNIM,Q6
      DFM=VOLN*SQRT(DSFR*DSFR+DSFI*DSFI)
      DFP=ATAN2(DSFI,DSFR)
      CX(NZ)=DFM*COS(DFP)
      CY(NZ)=DFM*SIN(DFP)
      DFP=DFP*(180./PI)
```

```
      IF(ABS(CX(NZ)).GT.XMAX) XMAX=ABS(CX(NZ))
      IF(ABS(CY(NZ)).GT.YMAX) YMAX=ABS(CY(NZ))
      IF(DFM.GT.DFMMAX) THEN
      DFMMAX=DFM
      DFPMMAX=DFP
      END IF
      WRITE(LOD,162)ZD,DFM,DFP
  990 CONTINUE
      GIM=300./YMAX
      GRL=300./XMAX
      IF(GIM.GT.GRL) THEN
      GIM=GRL
      ELSE
      GRL=GIM
      END IF
      IM1=GIM*CY(0)+OIM
      IR1=GRL*CX(0)+ORL
C       WRITE(LOU,150)ZNDFR,ZNDFI,DFM,DFP
      WRITE(LOU,160)DSFR,DSFI,VOLN
      WRITE(LOU,164)DFMMAX,DFPMMAX
      IF(ISIDE.EQ.1) THEN
      WRITE(LOU,*)'   OD DEFECT'
      ELSE
      WRITE(LOU,*)'   ID DEFECT'
      END IF

      DO 1000 NZ=1,NZT
      IM2=GIM*CY(NZ)+OIM
      IR2=GRL*CX(NZ)+ORL
      CALL QLINE(IR1,IM1,IR2,IM2,15)
      IR1=IR2
      IM1=IM2
 1000 CONTINUE
      WRITE(LOU,*)
      WRITE(LOU,*)
      CALL PRTSC
      WRITE(LOU,*)FF
c       GO TO 11

  140 FORMAT(' NORM IMPD:RL',F10.6,' IM',F10.6,
     *' AIR IND',1PE13.6)
C  150 FORMAT(' NORM CHG:RL ',0PF10.6,' IM',0PF10.6,
C     *' MAG',0PF10.6,' PHA ',0PF7.2)
  160 FORMAT(' NORM DSF:RL',1PE11.4,' IM',1PE11.4,' VOLN',1PE11.4)
  161 FORMAT('    ZD        MAG        PHA')
  162 FORMAT(F6.3,5X,D11.4,5X,F7.2)
  164 FORMAT(' MAXIMUM MAG ',D11.4,'    PHA AT MAX MAG ',F7.2,\)
 1001 END
```

DFENCAR calculates defect impedance change, average over depth

Program DFENCAR calculates the change in the normalized impedance of a differential encircling coil due to the presence of a defect in a single conducting tube, as shown in Fig. 20.

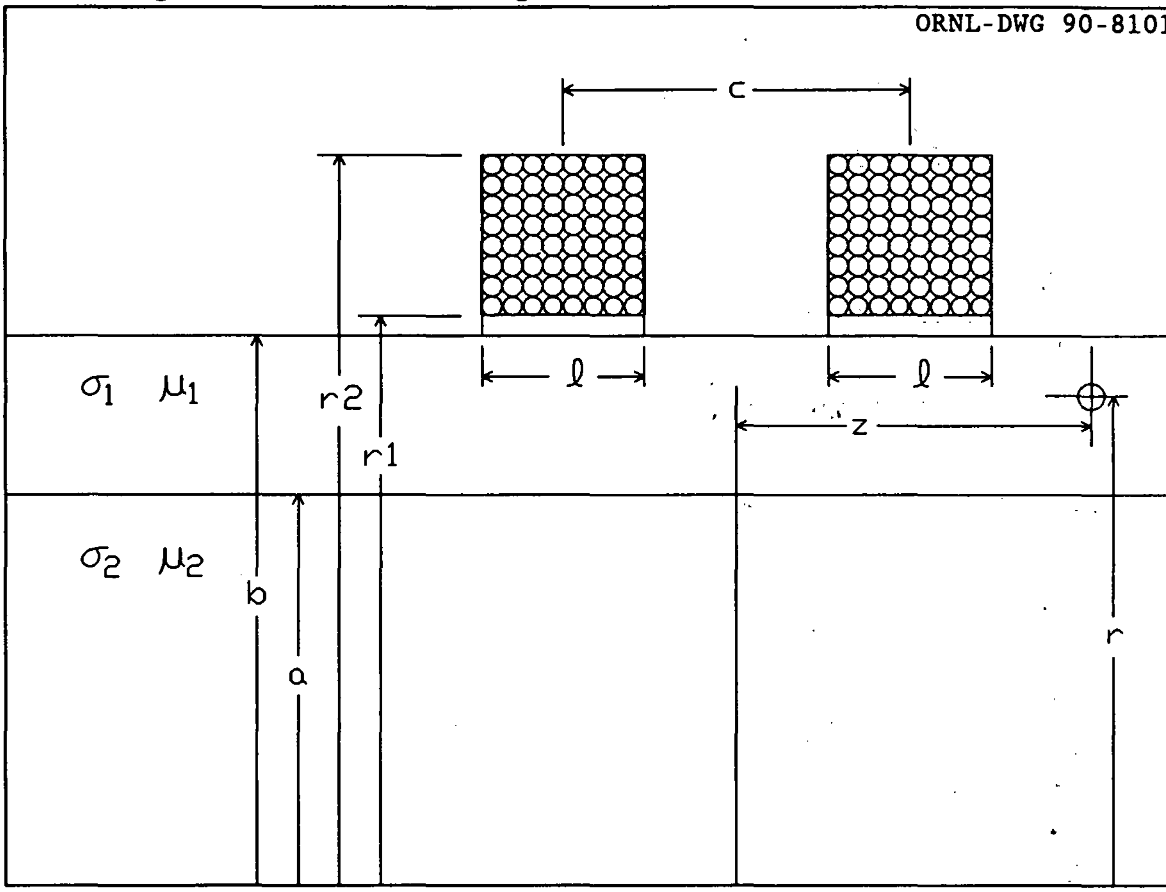

Fig. 20. Cross sectional view of a differential encircling probe.

The distance to the defect is measured from the center of the coil assembly, and the center-to-center distance of the coils is denoted *c*. The impedance difference between the two matched coils is:

$$Z_{nd} = Z_{1d} - Z_{2d} \quad , \tag{54}$$

Substituting in from Eq. (29) for the impedance change of each coil and using some trigonometric identities gives for the normalized impedance difference:

$$Z_{nd}(r,z) = \frac{-3(\omega\mu\sigma_1\bar{r}^2)Vol_n}{2\pi I_{air}} \tag{55}$$

$$\times \left[\int_0^\infty \frac{K(r_2,r_1)}{\pi\alpha^3} \left\{ \frac{K_1(\alpha_1 r)D_2 + I_1(\alpha_1 r)D_3}{(D_1D_2+D_3D_4)} \right\} sin(\frac{\alpha\ell}{2}) sin(\frac{\alpha c}{2}) sin(\alpha z) 4d\alpha \right]$$

$$\times \left[\int_0^\infty \frac{K(r_2,r_1)}{\pi\alpha^3} \left\{ \frac{K_1(\alpha_1 r)D_2 + I_1(\alpha_1 r)D_3}{(D_1D_2+D_3D_4)} \right\} sin(\frac{\alpha\ell}{2}) cos(\frac{\alpha c}{2}) cos(\alpha z) 4d\alpha \right]$$

The definitions of the various terms in Eq. (55) are given in Eqs. (45) through (53) in the discussion of the absolute encircling coil. The program performs the calculations for a number of different axial distances between the center of the coil and the defect, and it can divide the defect into a number of parts and calculate the effect of each part separately.

Summary

1. Dimension arrays and declare variable types.
2. Initialize variables.
3. Begin the integration. Calculate the expressions that are independent of the position of the defect.
4. Select a value for the radial distance between the defect and the center of the coil. Do the calculations which depend only upon this component of the position.
5. Select a value for the axial distance between the defect and the center of the coil. Complete the calculations.
6. Loop to 5 until done.
7. Loop to 4 until done.
8. Output the results.

Variables

A number of the variables which occur mainly inside the integration loops do not correspond to anything physical. These variables are described in a section at the end. Starred variables must be assigned by the user.

A	The normalized inner radius of the tube.
B	The normalized outer radius of the tube.
C*	The axial distance between the centers of the two coils. The value is input in inches and normalized by the program.
DELTAZ*	The normalized axial distance between the points at which the calculations are done.
DFDEP*	The depth of the defect in the tube in inches.
DFDIAM*	The diameter of the defect in the tube in inches.

DFM	The magnitude of the normalized impedance change in the coil due to the defect.
DFP	The phase of the normalized impedance change in the coil due to the defect.
DSFI	The imaginary part of the defect sensitivity factor of the coil.
DSFR	The real part of the defect sensitivity factor of the coil.
FREQ*	The operating frequency in hertz.
ISIDE*	The side of the tube where the defect is located. If ISIDE = 1, the defect is on the outside of the tube; if ISIDE = 2, the defect is on the inside of the tube.
L*	The length of each coil. The value is input in inches and normalized by the program.
LOD*	The number of the I/O unit connected to the output data file.
LOE*	The number of the I/O unit connected to a file which contains the diameters and depths of the defects for which calculations are to be performed.
LOU	The number of the I/O unit connected to the printer.
NRT*	The total number of parts centered along the axis of the defect into which the defect is divided to perform the calculations.
NZT*	The total number of different values of the axial distance between the center of the coil and the defect for which calculations are performed.
POW	The depth of the defect expressed as a percentage of wall thickness.
Q6	The inductance in henries of the coil in air.
R1*	The inner radius of each coil. The value is input in inches and normalized by the program.
R2*	The outer radius of each coil. The value is input in inches and normalized by the program.
R3	The mean radius of each coil in inches.
RCL*	The distance between the inside of the coil and the outside wall of the tube. The value is input in inches and normalized by the program.
RD	The radial distance between the center of the coil and the defect (see note 1).
RDT	The normalized depth to the bottom of the defect. A negative number.
RHO1*	The electrical resistivity of the tube in μΩ-cm.
T1*	The thickness of the tube wall. The value is input in inches and normalized by the program.
TRN*	The number of turns in each coil.
U1*	The relative magnetic permeability of the tube.
VOLN	The normalized volume of the defect.
WUSRR	The product of the angular operating frequency,

	the magnetic permeability of the tube, the electrical conductivity of the tube, and the square of the mean radius of the coil.
ZD	The axial distance between the center of the coil and the defect.
ZNDFI	The imaginary part of the normalized impedance change in the coil due to the defect.
ZNDFR	The real part of the normalized impedance change in the coil due to the defect.
ZNIM	The imaginary part of the normalized impedance of the coil when no defects are present.
ZNRL	The real part of the normalized impedance of the coil when no defects are present.

Integration section of Program DFENCAR

Symbol definitions

The following are definitions of the symbols used to describe the program variables which appear in the integration section of the program. All lengths are normalized unless otherwise noted.

α	Integration variable
a	Inner radius of the tube
β_1	$(\alpha^2+j\omega\mu\sigma_1\overline{r}^2)^{1/2}/\mu$
b	Outer radius of the tube
c	Distance between the centers of the coils
$I_0(x)$	Modified Bessel function of the first kind of order 0
$I_1(x)$	Modified Bessel function of the first kind of order 1
$J(x_2,x_1)$	Integral of $xJ_1(x)$ with respect to x from αx_1 to αx_2
$J_1(x)$	Bessel function of the first kind of order 1
$K(x_2,x_1)$	Integral of $xK_1(x)$ with respect to x from αx_1 to αx_2
$K_0(x)$	Modified Bessel function of the second kind of order 0
$K_1(x)$	Modified Bessel function of the second kind of order 1
ℓ	Length of the coil
μ	Relative magnetic permeability of the tube
r	Radial distance between center of coil and defect
$\overline{r}$	Coil mean radius in inches
r_1	Inner radius of coil
r_2	Outer radius of coil
σ_1	Electrical conductivity of the tube
z	Axial distance between defect and probe center
ω	Angular operating frequency

Variables appearing in the integration section

Program variable	Symbolic equivalent
A1	$\alpha\ell + \exp(-\alpha\ell) - 1$
BI0A	$I_0(\alpha a)$
BI0B	$I_0(\alpha b)$
BI1A	$I_1(\alpha a)$
BI1B	$I_1(\alpha b)$
BI1IA	$\mathrm{Im}[I_1(\alpha_1 a)]$
BI1IB	$\mathrm{Im}[I_1(\alpha_1 b)]$
BI1IR	$\mathrm{Im}[I_1(\alpha_1 r)]$
BI1RA	$\mathrm{Re}[I_1(\alpha_1 a)]$
BI1RB	$\mathrm{Re}[I_1(\alpha_1 b)]$

BI1RR	$\mathrm{Re}[I_1(\alpha_1 r)]$
BK0A	$K_0(\alpha a)$
BK0B	$K_0(\alpha b)$
BK1A	$K_1(\alpha a)$
BK1B	$K_1(\alpha b)$
BK1IA	$\mathrm{Im}[K_1(\alpha_1 a)]$
BK1IB	$\mathrm{Im}[K_1(\alpha_1 b)]$
BK1IR	$\mathrm{Im}[K_1(\alpha_1 r)]$
BK1RA	$\mathrm{Re}[K_1(\alpha_1 a)]$
BK1RB	$\mathrm{Re}[K_1(\alpha_1 b)]$
BK1RR	$\mathrm{Re}[K_1(\alpha_1 r)]$

DDI

$$\mathrm{Im}\{[\alpha b K_0(\alpha b)K_1(\alpha_1 b)-\beta_1 b K_0(\alpha_1 b)K_1(\alpha b)][\beta_1 a I_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 a I_0(\alpha_2 a)I_1(\alpha_1 a)] + [\beta_2 a I_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 a K_0(\alpha_1 a)I_1(\alpha_2 a)][\beta_1 b I_0(\alpha_1 b)K_1(\alpha b)+\alpha b K_0(\alpha b)I_1(\alpha_1 b)]\}$$

DDR

$$\mathrm{Re}\{[\alpha b K_0(\alpha b)K_1(\alpha_1 b)-\beta_1 b K_0(\alpha_1 b)K_1(\alpha b)][\beta_1 a I_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 a I_0(\alpha_2 a)I_1(\alpha_1 a)] + [\beta_2 a I_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 a K_0(\alpha_1 a)I_1(\alpha_2 a)][\beta_1 b I_0(\alpha_1 b)K_1(\alpha b)+\alpha b K_0(\alpha b)I_1(\alpha_1 b)]\}$$

DFR

$$\mathrm{Re}\ \{ K_1(\alpha_1 r)[\beta_1 a I_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 a I_0(\alpha_2 a)I_1(\alpha_1 a)] + I_1(\alpha_1 r)[\beta_2 a I_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 a K_0(\alpha_1 a)I_1(\alpha_2 a)]\} \div \{[\alpha b K_0(\alpha b)K_1(\alpha_1 b)-\beta_1 b K_0(\alpha_1 b)K_1(\alpha b)][\beta_1 a I_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 a I_0(\alpha_2 a)I_1(\alpha_1 a)] + [\beta_2 a I_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 a K_0(\alpha_1 a)I_1(\alpha_2 a)][\beta_1 b I_0(\alpha_1 b)K_1(\alpha b)+\alpha b K_0(\alpha b)I_1(\alpha_1 b)]\}$$

DI1	$\mathrm{Im}[\alpha b K_0(\alpha b)K_1(\alpha_1 b)-\beta_1 b K_0(\alpha_1 b)K_1(\alpha b)]$
DI2	$\mathrm{Im}[\beta_1 a I_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 a I_0(\alpha_2 a)I_1(\alpha_1 a)]$
DI3	$\mathrm{Im}[\beta_2 a I_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 a K_0(\alpha_1 a)I_1(\alpha_2 a)]$
DI4	$\mathrm{Im}[\beta_1 b I_0(\alpha_1 b)K_1(\alpha b)+\alpha b K_0(\alpha b)I_1(\alpha_1 b)]$

DNI	$\mathrm{Im}\ \{\ K_1(\alpha_1 r)\,[\beta_1 aI_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 aI_0(\alpha_2 a)I_1(\alpha_1 a)]\ +\ I_1(\alpha_1 r)\,[\beta_2 aI_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 aK_0(\alpha_1 a)I_1(\alpha_2 a)]\,\}$
DNR	$\mathrm{Re}\ \{\ K_1(\alpha_1 r)\,[\beta_1 aI_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 aI_0(\alpha_2 a)I_1(\alpha_1 a)]\ +\ I_1(\alpha_1 r)\,[\beta_2 aI_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 aK_0(\alpha_1 a)I_1(\alpha_2 a)]\,\}$
DR1	$\mathrm{Re}[\alpha bK_0(\alpha b)K_1(\alpha_1 b)-\beta_1 bK_0(\alpha_1 b)K_1(\alpha b)]$
DR2	$\mathrm{Re}[\beta_1 aI_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 aI_0(\alpha_2 a)I_1(\alpha_1 a)]$
DR3	$\mathrm{Re}[\beta_2 aI_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 aK_0(\alpha_1 a)I_1(\alpha_2 a)]$
DR4	$\mathrm{Re}[\beta_1 bI_0(\alpha_1 b)K_1(\alpha b)+\alpha bK_0(\alpha b)I_1(\alpha_1 b)]$
S1	$d\alpha$
SNI	$\mathrm{Im}\{K_1(\alpha_1 b)\,[\beta_1 aI_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 aI_0(\alpha_2 a)I_1(\alpha_1 a)]\ +\ I_1(\alpha_1 b)\,[\beta_2 aI_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 aK_0(\alpha_1 a)I_1(\alpha_2 a)]\,\}$
SNR	$\mathrm{Re}[K_1(\alpha_1 b)\,[\beta_1 aI_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 aI_0(\alpha_2 a)I_1(\alpha_1 a)]\ +\ I_1(\alpha_1 b)\,[\beta_2 aI_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 aK_0(\alpha_1 a)I_1(\alpha_2 a)]\,\}$
SSR	$\mathrm{Re}\ \frac{1}{K_1(\alpha b)}\Big[K_1(\alpha_1 b)\,[\beta_1 aI_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 aI_0(\alpha_2 a)I_1(\alpha_1 a)]\ +\ I_1(\alpha_1 b)\,[\beta_2 aI_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 aK_0(\alpha_1 a)I_1(\alpha_2 a)]\Big]\ \Big[\ [\alpha bK_0(\alpha b)K_1(\alpha_1 b)-\beta_1 bK_0(\alpha_1 b)K_1(\alpha b)]\ [\beta_1 aI_0(\alpha_1 a)I_1(\alpha_2 a)-\beta_2 aI_0(\alpha_2 a)I_1(\alpha_1 a)]\ +\ [\beta_2 aI_0(\alpha_2 a)K_1(\alpha_1 a)+\beta_1 aK_0(\alpha_1 a)I_1(\alpha_2 a)]\ [\,[\beta_1 bI_0(\alpha_1 b)K_1(\alpha b)+\alpha bK_0(\alpha b)I_1(\alpha_1 b)]\ \Big]^{-1}-\frac{I_1(\alpha b)}{K_1(\alpha b)}$
X	α
X1	$\mathrm{Re}[\alpha_1]$
X1A	$\mathrm{Re}[\alpha_1 a]$
X1B	$\mathrm{Re}[\alpha_1 b]$
X1R	$\mathrm{Re}[\alpha_1 r]$
XA	αa
XB	αb
XFACT	$\frac{1}{\alpha^3}\ K(r_2,r_1)\sin(\alpha\ell/2)$

XFACT1	$\frac{4}{\pi}\frac{1}{\alpha^3} K(r_2,r_1)\ \sin(\alpha\ell/2)\ \sin(\alpha c/2)\ \sin(\alpha z)d\alpha$
XFACT2	$\frac{4}{\pi}\frac{1}{\alpha^3} K(r_2,r_1)\ \sin(\alpha\ell/2)\ \cos(\alpha c/2)\ \cos(\alpha z)d\alpha$
XJR21	$\frac{1}{\alpha^3} J(r_2,r_1)$
XKR21	$\frac{1}{\alpha^3} K(r_2,r_1)$
XL	$\alpha\ell$
XX	α^2
XXXX	α^4
Y1	$\mathrm{Im}[\alpha_1]$
Y1A	$\mathrm{Im}[\alpha_1 a]$
Y1B	$\mathrm{Im}[\alpha_1 b]$
Y1R	$\mathrm{Im}[\alpha_1 r]$
ZI0IA	$\mathrm{Im}[\alpha_1 a I_0(\alpha_1 a)]$
ZI0IB	$\mathrm{Im}[\alpha_1 b I_0(\alpha_1 b)]$
ZI0IR	$\mathrm{Im}[\alpha_1 r I_0(\alpha_1 r)]$
ZI0RA	$\mathrm{Re}[\alpha_1 a I_0(\alpha_1 a)]$
ZI0RB	$\mathrm{Re}[\alpha_1 b I_0(\alpha_1 b)]$
ZI0RR	$\mathrm{Re}[\alpha_1 r I_0(\alpha_1 r)]$
ZK0IA	$\mathrm{Im}[\alpha_1 a K_0(\alpha_1 a)]$
ZK0IB	$\mathrm{Im}[\alpha_1 b K_0(\alpha_1 b)]$
ZK0IR	$\mathrm{Im}[\alpha_1 r K_0(\alpha_1 r)]$
ZK0RA	$\mathrm{Re}[\alpha_1 a K_0(\alpha_1 a)]$
ZK0RB	$\mathrm{Re}[\alpha_1 b K_0(\alpha_1 b)]$
ZK0RR	$\mathrm{Re}[\alpha_1 r K_0(\alpha_1 r)]$

<u>Sample output</u>

A sample of the printer output is given below:

```
 DFENCAR   TIME 11:32: 0  DATE  8/18/89
    IN RAD OT RAD  LENGTH  RAD CLR WALLTH  % WALL  C TO C  TUB IR  TUB OR
ACT 1.7750 2.0670  0.2650  0.0075  0.2200   45.45  0.5150  1.5475  1.7675
NOR 0.9240 1.0760  0.1379  0.0039  0.1145   45.45  0.2681  0.8056  0.9201
RBAR 1.9210 FREQ=  4.000000E+02 RHO=   3.8400 PERM=  1.000 WUSRR= 195.8126
 NORM IMPD:RL  0.135056 IM  0.494500 AIR IND 1.826299E-02
```

```
NORM DSF:RL 3.3340E-04 IM-2.1253E-04 VOLN 1.1079E-04
MAXIMUM MAG  0.1826D-03   PHA AT MAX MAG   91.17   OD DEFECT
```

Partial listing of output defect axial position (ZD), defect magnitude (DFM) and defect phase (DFP) that is sent to LOD file (either FORT39 or ENCSCN.DAT):

```
0.010      0.1723D-04      0.8304D+02
0.020      0.3455D-04      0.8350D+02
0.030      0.5201D-04      0.8425D+02
0.040      0.6962D-04      0.8524D+02
0.050      0.8728D-04      0.8641D+02
0.060      0.1047D-03      0.8766D+02
0.070      0.1216D-03      0.8888D+02
0.080      0.1373D-03      0.8996D+02
0.090      0.1513D-03      0.9083D+02
0.100      0.1631D-03      0.9144D+02
.......
```

If the proper plotting software has been installed, a plot of these data is made on the CRT and a hard copy can be obtained, as shown in Fig. 21. The plot forms one-half the normal Lissajous pattern one gets in an eddy-current test with a differential bobbin coil. The second half can be obtained by reflecting the signal in the -*z* direction, since the signal is anti-symmetric. The phases and magnitudes given in this report are referenced to the X axis being zero phase and measured counterclockwise from X axis, which is standard for mathematics and electrical engineering. Standard eddy-current practice is to measure the phase shift in a clockwise direction from the -X axis. Furthermore, the phase is rotated until the probe wobble/fill factor/lift-off variations lie in a horizontal direction.

ORNL-DWG 90-8102

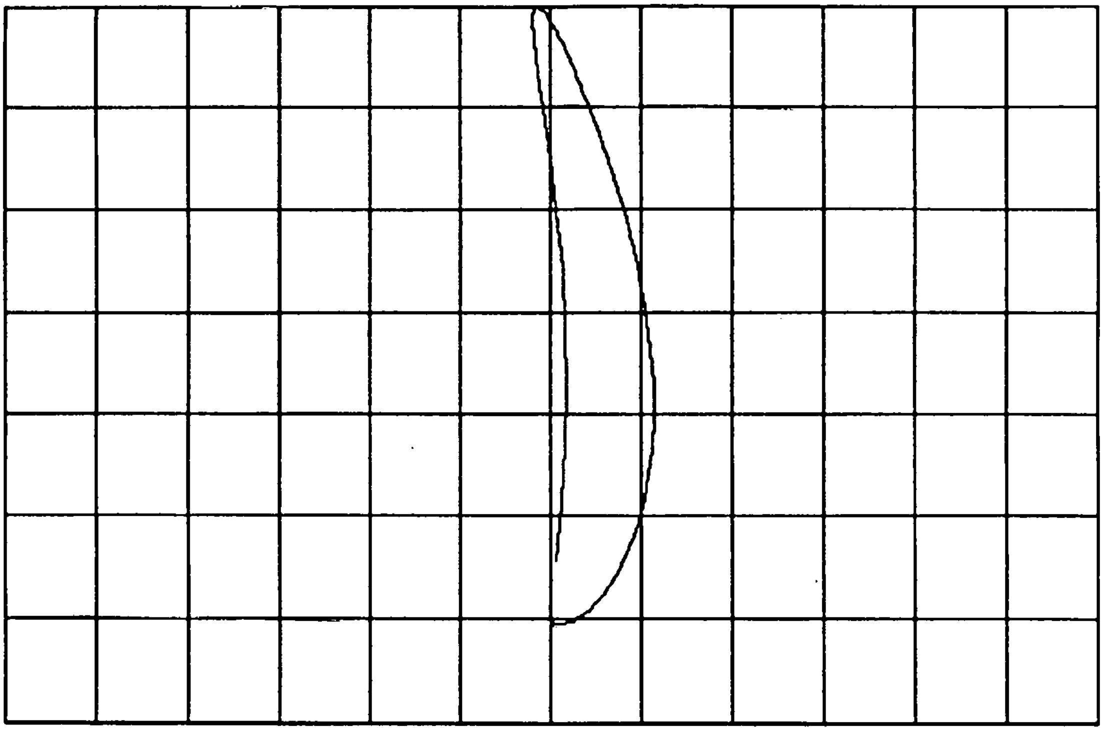

Fig. 21. Plot of defect signal on complex impedance plane as the defect is moved from the probe center in the plus *z* direction.

Listing

```
      PROGRAM DFENCAR
C     VERSION July 11, 1988
C     PROGRAM TO CALCULATE THE NORMALIZED IMPEDANCE CHANGE
C     FOR A DEFECT IN A SINGLE TUBE WITH AN ENCIRCLING COIL
C     Z=0.0 AT CENTER OF COIL.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 L
      CHARACTER*1 FF
      DIMENSION S1(6),S2(6),ERR(6)
      DIMENSION CX(200),CY(200)
      DIMENSION SMZDFR1A(200,30),SMZDFI1A(200,30)
      DIMENSION SMZDFR2A(200,30),SMZDFI2A(200,30)
      DATA LOU/8/,PI/3.141592653/,LOD/39/,LOE/38/
      DATA S1/.005,.02,.05,.1,.1,.5/
      DATA S2/1.0,2.0,5.0,10.0,50.0,200.0/
      DATA ERR/.1,.01,.001,1.E-4,1.E-5,1.E-10/
      DATA FREQ/ 4.0E2/,RHO1/3.84/,U1/1.0/
      DATA TRN/325./,ISIDE/1/,DELTAZ/0.01/
      DATA MODE/16/,NZT/100/,NRT/20/
      DATA OIM/50/,ORL/330/
      FF=CHAR(12)
C      OPEN(LOD,FILE='ENCSCN.DAT',STATUS='NEW')
      OPEN(LOE,FILE='ENCDAT.DAT',STATUS='OLD')
   11 XMAX=0.
      YMAX=0.
      DFMMAX=0.
      DO 14 NZ=1,NZT
      DO 12 NR=1,NRT
      SMZDFR1A(NZ,NR)=0.
      SMZDFI1A(NZ,NR)=0.
      SMZDFR2A(NZ,NR)=0.
      SMZDFI2A(NZ,NR)=0.
   12 CONTINUE
   14 CONTINUE
      READ(LOE,*,END=1001)DFDIAM,DFDEP
C     TIME AND DATE ARE PRINTED
      CALL GETTIM(IHR,IMN,ISE,IFR)
      CALL GETDAT(IYR,IMO,IDA)
      IYR=IYR-1900
      WRITE(LOU,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT(' DFENCAR   TIME ',I2,':',I2,':',I2
     *,'  DATE ',I2,'/',I2,'/',I2)
      WRITE(LOU,5)
    5 FORMAT(5X,'IN RAD',2X,'OT RAD',2X,'LENGTH',2X,'RAD CLR'
     *,1X,'WALLTH',2X,'% WALL',2X,'C TO C',2X,'TUB IR',2X,'TUB OR')
      R1=1.775
      R2=2.067
      L=0.265
      C=0.515
```

```
      RCL=0.0075
      T1=0.22
      POW=(DFDEP/T1)*100.
      B=R1-RCL
      A=B-T1
      RDT=-DFDEP
      R3=0.5*(R1+R2)
      WRITE(LOU,10)R1,R2,L,RCL,T1,POW,C,A,B
      R1=R1/R3
      R2=2.0-R1
      L=L/R3
      C=C/R3
      RCL=RCL/R3
      RDT=RDT/R3
      RD=RD/R3
      T1=T1/R3
      A=A/R3
      B=B/R3
C      VOLN=0.1666667*PI*(DFDIAM/R3)*(DFDIAM/R3)*(DFDIAM/R3)
      VOLN=PI*DFDIAM*DFDIAM*DFDEP/(4.*R3*R3*R3)
      WUSRR=0.5093979*U1*R3*R3*FREQ/RHO1
      WRITE(LOU,15)R1,R2,L,RCL,T1,POW,C,A,B
   10 FORMAT('ACT ',5(F7.4,1X),F7.2,1X,3(F7.4,1X))
   15 FORMAT('NOR ',5(F7.4,1X),F7.2,1X,3(F7.4,1X))
      WRITE(LOU,20)R3,FREQ,RHO1,U1,WUSRR
   20 FORMAT(' RBAR',F7.4,' FREQ= ',1PE13.6,' RHO=',0PF9.4,
     *' PERM=',F7.3,' WUSRR=',F11.4)
      CALL QSMODE(MODE)
      CALL GRID
C      DO 1000 NZ=1,50
C      ZD=0.04*NZ
      SMAIR=0.0
      SMIMPR=0.0
      SMIMPI=0.0
      SMZDFR1=0.0
      SMZDFI1=0.0
      SMZDFR2=0.0
      SMZDFI2=0.0
C     AIR=0.
C     AII=0.
C     AUR=0.
C     AUI=0.
      B1=0.0
      B2=S2(1)
      DO 100 JKL=1,6
   30 RI9=SMAIR
      X=B1-0.5*S1(JKL)
C     DETERMINE NUMBER OF STEPS,ISTEPS,BASED ON STEP SIZE&INTERVAL
      ISTEPS=DNINT((B2-B1)/S1(JKL))
      DO 90 I=1,ISTEPS
      X=X+S1(JKL)
```

```
      CALL KJBSSL(X,R1,R2,XKR21,XJR21)
      XL=X*L
      IF(XL.GT.5.0E-3) GO TO 60
      A1=XL*XL*(0.5-XL/6.0)
      GO TO 80
   60 IF(XL.GT.75.0) GO TO 70
      A1=XL+DEXP(-XL)-1.0
      GO TO 80
   70 A1=XL-1.0
   80 SMAIR=SMAIR+2.*XJR21*XJR21*A1*S1(JKL)
      IF(X.GT.160.)GO TO 90
      XX=X*X
      XXXX=XX*XX
      X1=DSQRT(0.5*(XX+DSQRT(XXXX+WUSRR*WUSRR)))/U1
      Y1=WUSRR/(2*X1*U1*U1)
      XA=X*A
      XB=X*B
      X1A=X1*A
      Y1A=Y1*A
      X1B=X1*B
      Y1B=Y1*B
      CALL CMDBES(X1A,Y1A,ZIORA,ZIOIA,ZKORA,ZKOIA,BI1RA,BI1IA
     *,BK1RA,BK1IA)
      CALL CMDBES(X1B,Y1B,ZIORB,ZIOIB,ZKORB,ZKOIB,BI1RB,BI1IB
     *,BK1RB,BK1IB)
      CALL BESI(XA,BIOA,BI1A)
      CALL BESK(XA,BKOA,BK1A)
      CALL BESI(XB,BIOB,BI1B)
      CALL BESK(XB,BKOB,BK1B)
      DR1=XB*BKOB*BK1RB-ZKORB*BK1B/U1
      DI1=XB*BKOB*BK1IB-ZKOIB*BK1B/U1
      DR2=ZIORA*BI1A/U1-XA*BIOA*BI1RA
      DI2=ZIOIA*BI1A/U1-XA*BIOA*BI1IA
      DR3=XA*BIOA*BK1RA+ZKORA*BI1A/U1
      DI3=XA*BIOA*BK1IA+ZKOIA*BI1A/U1
      DR4=ZIORB*BK1B/U1+XB*BKOB*BI1RB
      DI4=ZIOIB*BK1B/U1+XB*BKOB*BI1IB
      DDR=DR1*DR2-DI1*DI2+DR3*DR4-DI3*DI4
      DDI=DR1*DI2+DR2*DI1+DR3*DI4+DR4*DI3
      SNR=BK1RB*DR2-BK1IB*DI2+BI1RB*DR3-BI1IB*DI3
      SNI=BK1RB*DI2+BK1IB*DR2+BI1RB*DI3+BI1IB*DR3
      DEN=DDR*DDR+DDI*DDI
      SSR=((SNR*DDR+SNI*DDI)/DEN-BI1B)/BK1B
      SSI=(SNI*DDR-SNR*DDI)/(BK1B*DEN)
      XFACT=XKR21*DSIN(XL/2.)
      SMIMPR=SMIMPR+8.*XFACT*XFACT*S1(JKL)*SSR/PI
      SMIMPI=SMIMPI+8.*XFACT*XFACT*S1(JKL)*SSI/PI

      DO 89 NR=1,NRT
      RD=(REAL(NR)-0.5)*(RDT/REAL(NRT))
      IF(ISIDE.EQ.1) THEN
```

```
      RD=B+RD
      ELSE
      RD=A-RD
      END IF
      X1R=X1*RD
      Y1R=Y1*RD

      CALL CMDBES(X1R,Y1R,ZIORR,ZIOIR,ZKORR,ZKOIR,BI1RR,BI1IR
     *,BK1RR,BK1IR)

      DNR=BK1RR*DR2-BK1IR*DI2+BI1RR*DR3-BI1IR*DI3
      DNI=BK1IR*DR2+BK1RR*DI2+BI1IR*DR3+BI1RR*DI3
      DFR=(DNR*DDR+DNI*DDI)/DEN
      DFI=(DNI*DDR-DNR*DDI)/DEN
      DO 88 NZ=1,NZT
      ZD=DELTAZ*NZ
      XFACT1=4.*XFACT*DSIN(X*ZD)*DSIN(0.5*X*C)*S1(JKL)/PI
      XFACT2=4.*XFACT*DCOS(X*ZD)*DCOS(0.5*X*C)*S1(JKL)/PI
      SMZDFR1A(NZ,NR)=SMZDFR1A(NZ,NR)+XFACT1*DFR
      SMZDFI1A(NZ,NR)=SMZDFI1A(NZ,NR)+XFACT1*DFI
      SMZDFR2A(NZ,NR)=SMZDFR2A(NZ,NR)+XFACT2*DFR
      SMZDFI2A(NZ,NR)=SMZDFI2A(NZ,NR)+XFACT2*DFI
   88 CONTINUE
   89 CONTINUE
   90 CONTINUE
      B1=B2
      B2=B2+S2(JKL)
      CHECK=(SMAIR-RI9)/SMAIR
      IF(ABS(CHECK).GT.ERR(JKL)) GO TO 30
  100 CONTINUE
      DO 990 NZ=1,NZT
      ZD=DELTAZ*NZ
      SMZDFR1=0.
      SMZDFI1=0.
      SMZDFR2=0.
      SMZDFI2=0.
      DO 120 NR=1,NRT
      SMZDFR1=SMZDFR1+SMZDFR1A(NZ,NR)
      SMZDFI1=SMZDFI1+SMZDFI1A(NZ,NR)
      SMZDFR2=SMZDFR2+SMZDFR2A(NZ,NR)
      SMZDFI2=SMZDFI2+SMZDFI2A(NZ,NR)
  120 CONTINUE
      SMZDFR1=SMZDFR1/NRT
      SMZDFI1=SMZDFI1/NRT
      SMZDFR2=SMZDFR2/NRT
      SMZDFI2=SMZDFI2/NRT
C       WRITE(LOU,*)NZ,SMZDFR1,SMZDFI1
C       WRITE(LOU,*)'          ',SMZDFR2,SMZDFI2
      DSFR=-1.5*WUSRR*(SMZDFR1*SMZDFR2-SMZDFI1*SMZDFI2)/(SMAIR*PI)
      DSFI=-1.5*WUSRR*(SMZDFR1*SMZDFI2+SMZDFR2*SMZDFI1)/(SMAIR*PI)
  135 Q6=0.0254*4.0E-7*TRN*TRN*R3*PI*PI*SMAIR/(L*(R2-R1))**2
```

```
      ZNIM=SMIMPR/SMAIR+1.0
      ZNRL=-SMIMPI/SMAIR
      ZNDFR=VOLN*DSFR
      ZNDFI=VOLN*DSFI
      IF(NZ.EQ.1) WRITE(LOU,140)ZNRL,ZNIM,Q6
      DFM=VOLN*SQRT(DSFR*DSFR+DSFI*DSFI)
      IF(ZD.EQ.0.0) THEN
      DFP=0.
      ELSE
      DFP=ATAN2(DSFI,DSFR)
      END IF
      CX(NZ)=DFM*COS(DFP)
      CY(NZ)=DFM*SIN(DFP)
      DFP=DFP*(180./PI)
      IF(ABS(CX(NZ)).GT.XMAX) XMAX=ABS(CX(NZ))
      IF(ABS(CY(NZ)).GT.YMAX) YMAX=ABS(CY(NZ))
      IF(DFM.GT.DFMMAX) THEN
      DFMMAX=DFM
      DFPMMAX=DFP
      END IF
C      WRITE(LOD,162)ZD,DFM,DFP
  990 CONTINUE
      GIM=300./YMAX
      GRL=300./XMAX
      IF(GIM.GT.GRL) THEN
      GIM=GRL
      ELSE
      GRL=GIM
      END IF
      IM1=GIM*CY(1)+OIM
      IR1=GRL*CX(1)+ORL
C      WRITE(LOU,150)ZNDFR,ZNDFI,DFM,DFP
      WRITE(LOU,160)DSFR,DSFI,VOLN
      WRITE(LOU,164)DFMMAX,DFPMMAX
      IF(ISIDE.EQ.1) THEN
      WRITE(LOU,*)'  OD DEFECT'
      ELSE
      WRITE(LOU,*)'  ID DEFECT'
      END IF

      DO 1000 NZ=2,NZT
      IM2=GIM*CY(NZ)+OIM
      IR2=GRL*CX(NZ)+ORL
      CALL QLINE(IR1,IM1,IR2,IM2,15)
      IR1=IR2
      IM1=IM2
 1000 CONTINUE
      WRITE(LOU,*)
      WRITE(LOU,*)
      CALL PRTSC
      WRITE(LOU,*)FF
```

```
      GO TO 11

  140 FORMAT(' NORM IMPD:RL',F10.6,' IM',F10.6,
     *' AIR IND',1PE13.6)
C  150 FORMAT(' NORM CHG:RL ',0PF10.6,' IM',0PF10.6,
C     *' MAG',0PF10.6,' PHA ',0PF7.2)
  160 FORMAT(' NORM DSF:RL',1PE11.4,' IM',1PE11.4,' VOLN',1PE11.4)
  161 FORMAT('    ZD              MAG               PHA')
  162 FORMAT(F6.3,5X,D11.4,5X,D11.4)
  164 FORMAT(' MAXIMUM MAG ',D11.4,'   PHA AT MAX MAG ',F7.2,\)

 1001 END
```

SUBROUTINES

The following is a collection of subroutines that are used by many of the programs. Rather than list them every time with each program, they are collected here. A more detailed description of these routines is given in other reports.[4-6]

Subroutine BESO(XJOR,R)

Subroutine BESO calculates $J_0(r)$, the Bessel function of the first kind of order 0. It is called by program PCBLDF.

Input

R The argument of the function.

Output

XJOR The function $J_0(r)$

Listing

```
      SUBROUTINE BESO(XJOR,R)
C      PROGRAM TO CALCULATE JO(R)
      IMPLICIT REAL*8 (A-H,O-Z)
      IF(R.GT.3.0)GO TO 50
      Y=R*R/9.0
      XJOR=(((((.00021*Y-.0039444)*Y+.0444479)*Y-.3163866)
     1*Y+1.2656208)*Y-2.2499997)*Y+1.0
      GO TO 100
   50 Y=3/R
      FO=(((((.00014476*Y-.00072805)*Y+.00137237)*Y-.00009512)
     1*Y-.00552740)*Y-.00000077)*Y+.79788456
      ANG=(((((.00013558*Y-.00029333)*Y-.00054125)*Y+.00262573)
     1*Y-.00003954)*Y-.04166397)*Y-.78539816+R
      XJOR=FO*COS(ANG)/SQRT(R)
  100 RETURN
      END
```

Subroutine BESEL1(Q1,RJ1)

Subroutine BESEL1 calculates $J_1(r)$, the Bessel function of the first kind of order 1. It is called by programs PCDSF and PCAVZSCN.

<u>Input</u>

Q1 The argument of the function

<u>Output</u>

RJ1 The function $J_1(r)$

<u>Listing</u>

```
      SUBROUTINE BESEL1(Q1,RJ1)
C
C     VERSION 7 DEC 1982 CALCULATES J1(Q1) TO WITHIN 4E-8
C
      IMPLICIT REAL*8 (A-H,O-Z)
      IF(Q1.GT.3) GO TO 20
      Q1S=Q1*Q1
      Q2S=((2.1E-11*Q1S-5.38E-9)*Q1S+6.757E-7)*Q1S-5.42443E-5
      Q2S=((Q2S*Q1S+2.60415E-3)*Q1S-6.25E-2)*Q1S+.5
      RJ1=Q1*Q2S
      RETURN
   20 Q3S=(((-.14604057/Q1+.27617679)/Q1-.20210391)/Q1+4.61835E-3)/Q1
      Q3S=((Q3S+.14937)/Q1+4.68E-6)/Q1+.79788456
      Q4S=(((-.21262014/Q1+.19397232)/Q1+6.022188E-2)/Q1-.17222733)/Q1
      Q4S=((Q4S+5.085E-4)/Q1+.37498836)/Q1-2.35619449+Q1
      RJ1=Q3S*DCOS(Q4S)/DSQRT(Q1)
      RETURN
      END
```

Subroutine BESI(X,XIZRO,XIONE)

Subroutine BESI calculates $I_0(x)$ and $I_1(x)$ where I_0 is the modified Bessel function of the first kind of order 0, and I_1 is the modified Bessel function of the first kind of order 1.

<u>Input</u>

X The argument of the functions

<u>Output</u>

XKZRO $I_0(x)$
XKONE $I_1(x)$

<u>Listing</u>

```
      SUBROUTINE BESI (X, XIZRO, XIONE)
C  ************************************
C
C          (J. M. BLAIR, AECL-4928, OCT. 1974)
C          TABLE 13, P.12 (IO(X), X .LE. 15)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DATA PO, P1, P2, P3, P4, P5, P6, P7, P8, P9/
     1 .137394871E12, .335834006E11, .195725233E10, .481197766E08,
     2 .627000932E06, .487113418E04, .240359483E02, .773737707E-1,
     3 .157463530E-3, .193496966E-6/
        DATA Q0, Q1, Q2, Q3/
     1 .346485713E11,-.229970736E09, .693700416E06,-.116950647E04/
C
C          TABLE 29, P. 22 (I1(X), X .LE. 15)
C
      DATA RO, R1, R2, R3, R4, R5, R6, R7, R8, R9/
     1-.130304539E09,-.157394690E08,-.610953810E06,-.113886320E05,
     2-.121620389E03,-.819523728E00,-.367341959E-2,-.113297426E-4,
     3-.226143979E-7,-.340950565E-10/
      DATA SO, S1, S2/
     1-.893665937E08, .475582653E06,-.104386692E04/
C
C          TABLE 43, P.32 (IO(X), X .GT. 15)
C
      DATA TO, T1, T2/
     1 .468486946E00,-.117012345E01, .278005347E00/
      DATA UO, U1/
     1 .116422863E01,-.306426483E01/
C
C          TABLE 61, P. 43 (I1(X), X .GT. 15)
C
      DATA VO, V1, V2/
```

```
     1 .106280324E01,-.267327163E01, .113698557E01/
      DATA W0, W1/
     1 .273391652E01,-.577627341E01/
      IF (X .GT. 15.) GO TO 100
      Z = X**2
      Q = Z - 225.
      XIZRO = (P0 + Z*(P1 + Z*(P2 + Z*(P3 + Z*(P4 + Z*(P5 + Z*(P6 +
     1 Z*(P7 + Z*(P8 + Z*P9)))))))))/(Q0 + Q*(Q1 + Q*(Q2 + Q*(Q3 + Q))))
      XIONE = X*(R0 + Z*(R1 + Z*(R2 + Z*(R3 + Z*(R4 + Z*(R5 + Z*(R6 +
     1 Z*(R7 + Z*(R8 + Z*R9)))))))))/(S0 + Q*(S1 + Q*(S2 + Q)))
C
      RETURN
100   Z = 1./X
      Q = DSQRT(Z)*DEXP(X)
      Z = Z - .0666666667
      XIZRO = Q*(T0 + Z*(T1 + Z*T2))/(U0 + Z*(U1 + Z))
      XIONE = Q*(V0 + Z*(V1 + Z*V2))/(W0 + Z*(W1 + Z))
      RETURN
      END
```

Subroutine BESK(X,XKZRO,XKONE)

Subroutine BESK calculates $K_0(x)$ and $K_1(x)$ where K_0 is the modified Bessel function of the second kind of order 0, and K_1 is the modified Bessel function of the second kind of order 1.

<u>Input</u>

X	The argument of the functions

<u>Output</u>

XKZRO	$K_0(x)$
XKONE	$K_1(x)$

<u>Listing</u>

```
      SUBROUTINE BESK (X, XKZRO, XKONE)
C   ************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      DATA Q0 /-4.21100684E1/
      DATA P0, P1, P2, P3, P4 /1.15931516E-1,  2.78982863E-1,
     1          2.52490595E-2, 8.45673143E-4,  1.53265946E-5/
      DATA R0, R1, R2         /1.29684595E+1,  3.28698873E00,
     1          1.10173127E-1/
      DATA V0, V1, V2, V3     /1.16185714E+2,  3.92399581E+2,
     1          3.09123840E+2, 4.78236536E+1/
      DATA S0, S1, S2, S3     /9.27027874E+1,  3.24677382E+2,
     1          2.80713014E+2, 5.71852878E+1/
      DATA T0, T1, T2, T3, T4 /4.35972688E00,  1.50242580E+1,
     1          1.38870631E+1, 3.64579096E00,  1.31176117E-1/
      DATA U0, U1, U2         /3.47855876E00,  1.06831663E+1,
     1          7.48163646E00/
      IF (X .GT. 1.) GO TO 100
      CALL BESI (X, XIZRO, XIONE)
      ZLOC = DLOC (X)
      Z = X*X
      XKZRO = P0 + Z*(P1 + Z*(P2 + Z*(P3 + Z*P4))) - ZLOG*XIZRO
      XKONE = ((R0 + Z*(R1 + Z*R2))/(Q0 + Z))*X + ZLOG*XIONE + 1./X
      RETURN
  100 Z = 1./X
      SXEX = DSQRT(Z)*DEXP(-X)
      XKZRO = SXEX*(V0 + Z*(V1 + Z*(V2 + Z*V3)))/
     1 (S0 + Z*(S1 + Z*(S2 + Z*(S3 + Z))))
      XKONE = SXEX*(T0 + Z*(T1 + Z*(T2 + Z*(T3 + Z*T4))))/
     1 (U0 + Z*(U1 + Z*(U2 + Z)))
      RETURN
      END
```

Subroutine BESSEL(XJ1,X,R)

Subroutine BESSEL calculates

$$\frac{1}{\alpha^3} J(r_1,0) = \frac{1}{\alpha^3} \int_0^{r_1} \alpha r J_1(\alpha r)\, dr$$

where J_1 is the Bessel function of the first kind of order 1. It is called by programs PCBLDF, PCDSF, and PCAVZSCN.

Input

X	α in the above equation
R	r_1 in the above equation

Output

XJ1	The right side of the above equation

Listing

```
      SUBROUTINE BESSEL(XJ1,X,R)
C     CALCULATES INT(XR*XJ1(XR)/(X*X*X) FROM 0 TO XR
      IMPLICIT REAL*8 (A-H,O-Z)
      DATA PI04/.785398163/
      Z=X*R
      IF(Z.GT.5.0) GO TO 1090
      L5=2.0*Z+3.0
      F1=0.5*R*R*R
      XJ1=F1/3.0
      DO 1070 N=1,L5
      RN=N
      F1=-F1*.25*Z*Z/(RN*RN+RN)
 1070 XJ1=XJ1+F1/(2.0*RN+3.0)
      GO TO 1210
 1090 IF(Z.GT.30.0) GO TO 1160
      Q1=(((-188.1357/Z+109.1142)/Z-23.79333)/Z+2.050931)/Z
      Q1=((Q1-.1730503)/Z+.7034845)/Z-.064109E-3
      Q2=(((-5.817517/Z+2.105874)/Z-.6896196)/Z+.4952024)/Z
      Q2=(Q2-.187344E-2)/Z+.7979095
      XJ1=(1.0-DSQRT(Z)*(Q2*DCOS(Z-PI04)-Q1*DSIN(Z-PI04)))/(X*X*X)
      GO TO 1210
 1160 P3=1.0/(Z*Z)
      P1=Z*(-1.0+P3*(-.5546875+2.48062114*P3))
```

```
      P2=.875+P3*(-.93457031+8.98975114*P3)
      XJ1=1.0+.79788456*(P1*DCOS(Z-.78539816)
     8 +P2*DSIN(Z-.78539816))/DSQRT(Z)
      XJ1=XJ1/(X*X*X)
 1210 RETURN
      END
```

Subroutine CMDBES(X,Y,ZIOR,ZIOI,ZKOR,ZKOI,BI1R,BI1I,BK1R,BK1I)

Subroutine CMDBES calculates the following four quantities: $zI_0(z)$, $zK_0(z)$, $I_1(z)$, and $K_1(z)$, where z is a complex number and where I_0 is the modified Bessel function of the first kind of order 0, K_0 is the modified Bessel function of the second kind of order 0, I_1 is the modified Bessel function of the first kind of order 1, and K_0 is the modified Bessel function of the second kind of order 1.

<u>Input</u>

X	Re(z)
Y	Im(z)

<u>Output</u>

ZIOI	Im[$zI_0(z)$]
ZIOR	Re[$zI_0(z)$]
ZKOI	Im[$zK_0(z)$]
ZKOR	Re[$zK_0(z)$]
BI1I	Im[$I_1(z)$]
BI1R	Re[$I_1(z)$]
BK1I	Im[$K_1(z)$]
BK1R	Re[$K_1(z)$]

<u>Listing</u>

```
      SUBROUTINE CMDBES(X,Y,ZIOR,ZIOI,ZKOR,ZKOI,BI1R,BI1I,BK1R,BK1I)
C  ***************************
C
C         COMPUTES F(1) + J * F(2) = Z * I0(Z)
C                  F(3) + J * F(4) = Z * K0(Z)
C                  F(5) + J * F(6) = I1(Z)
C                  F(7) + J * F(8) = K1(Z)
C         OF THE COMPLEX ARGUMENT Z = X + J * Y
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION F(8)
      R=DSQRT(X*X+Y*Y)
      C1=X/R
      S1=Y/R
      PHI=DATAN(S1/C1)
      IF(R.GT.8.0) GO TO 100
C
C        FOR R.LE.8 USE RATIONAL APPROXIMATION FOR K0(Z) AND K1(Z)
C        AND BACKWARD RECURRENCE  FOR I0(Z) AND I1(Z)
C
      CALL COMKB(X,Y,ZKOR,ZKOI,BK1R,BK1I)
      CALL CMI(X,Y,ZIOR,ZIOI,BI1R,BI1I)
      GO TO 125
```

```
C
C         ASYMPTOTIC SERIES FOR R.GT.8
C
 100  F(2) = 0.
      F(4) = 0.
      F(6) = 0.
      F(8) = 0.
      F(1) = 1.0
      F(3) = 1.0
      F(5) = 1.0
      F(7) = 1.0
      ODD=1.0
      T=1.0
      U=1.0
      P=1.0
      SIGN=-1.0
      V=.1250/R
      C=C1
      S=S1
      L5=3+20./R
      DO 110 N=1,L5
      S6=-ODD*ODD
      S7=V/P
      T=S7*S6*T
      U=U*(4.0+S6)*S7
      T1=C*T
      F(1)=T1*SIGN+F(1)
      F(3)=T1+F(3)
      T1=-S*T
      F(2)=T1*SIGN+F(2)
      F(4)=T1+F(4)
      T1=C*U
      F(5)=T1*SIGN+F(5)
      F(7)=T1+F(7)
      T1=-S*U
      F(6)=T1*SIGN+F(6)
      F(8)=T1+F(8)
      SIGN=-SIGN
      P=P+1.0
      ODD=ODD+2.0
      C2=C*C1-S*S1
      S=S*C1+C*S1
      C=C2
 110  CONTINUE
      XLIM=175.
      IF(DABS(X).GT.XLIM) GO TO 140
      S6=DEXP(X)
      S7=1.D0/S6
      T1=DSQRT(R)
      S6=.39894228*S6/T1
      S7=1.25331413*S7/T1
```

```
C
C         THE ABOVE FACTORS ARE 1/DSQRT(2*PI) AND DSQRT(PI/2)
C
      ARG=.50*PHI
      C=DCOS(ARG)
      S=DSIN(ARG)
C
C      MULTIPLY BY COS(PHI/2) - J * SIN(PHI/2) FROM 1/DSQRT(Z)
C
      DO 115 N=2,8,2
      T=F(N-1)*C+F(N)*S
      F(N)=F(N)*C-F(N-1)*S
      F(N-1)=T
 115  CONTINUE
      C=DCOS(Y)
      S=DSIN(Y)
C
C          MULTIPLY I FNS. BY S6*DEXP(J*Y) AND K FNS. BY S7*DEXP(-J*Y)
C
      DO 120 N=1,5,4
      T=F(N)*C-F(N+1)*S
      F(N+1)=S6*(F(N+1)*C+F(N)*S)
      F(N)=T*S6
      T=F(N+2)*C+F(N+3)*S
      F(N+3)=S7*(F(N+3)*C-F(N+2)*S)
      F(N+2)=T*S7
 120  CONTINUE
      ZIOR=F(1)
      ZIOI=F(2)
      ZKOR=F(3)
      ZKOI=F(4)
      BI1R=F(5)
      BI1I=F(6)
      BK1R=F(7)
      BK1I=F(8)
C
C          MULTIPLY ZERO-ORDER FUNCTIONS BY Z
C
 125  T=X*ZIOR-Y*ZIOI
      ZIOI=ZIOR*Y+ZIOI*X
      ZIOR=T
      T=X*ZKOR-Y*ZKOI
      ZKOI=ZKOR*Y+ZKOI*X
      ZKOR=T
 140  RETURN
      END
C
C   ******************************************
      SUBROUTINE CMI(X,Y,BIZR,BIZI,BIOR,BIOI)
C   ******************************************
C
```

```
C         COMPUTES THE REAL AND IMAGINARY PARTS OF THE MODIFIED BESSEL
C         FUNCTIONS OF THE FIRST KIND, I0(Z) AND I1(Z), Z = X + I*Y, BY
C         BACKWARD RECURRENCE
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION FRE(52),FIM(52)
      DATA A,B/1.E-18,1.E-18/
C
C         CALCULATE N,THE NUMBER OF TERMS
C
      R=DSQRT(X**2+Y**2)
        N = 1.3*R + 9.
C
C         COMPUTE U AND V, REAL AND IMAG PARTS OF 2/Z
C
      T=Y/X
      U=2./(T*Y+X)
      V=-U*T
C
C         INITIALIZE
C
      FRE(N+2)=0.
      FIM(N+2)=0.
      SRE=A
      FRE(N+1)=A
      SIM=B
      FIM(N+1)=B
      XN=FLOAT(N)
      TNUOZR=XN*U
      TNUOZI=XN*V
      NU=N+1
C
C         BACKWARD RECURRENCE
C
      DO 135 K=1,N
      FRE(NU-1)=FRE(NU+1)+TNUOZR*FRE(NU)-TNUOZI*FIM(NU)
      SRE=FRE(NU-1)+SRE
      FIM(NU-1)=FIM(NU+1)+TNUOZR*FIM(NU)+TNUOZI*FRE(NU)
      SIM=FIM(NU-1)+SIM
      NU=NU-1
      TNUOZR=TNUOZR-U
      TNUOZI=TNUOZI-V
 135  CONTINUE
C
C         ADJUST SUM
C
      SRE=SRE+SRE-FRE(1)
      SIM=SIM+SIM-FIM(1)
      T1=SIM/SRE
      U1=1./(T1*SIM+SRE)
      V1=-U1*T1
```

```
C
C        NORMALIZE
C
      EX=DEXP(X)
      SY=DSIN(Y)
      CY=DCOS(Y)
      FANRE=EX*(CY*U1-SY*V1)
      FANIM=EX*(CY*V1+SY*U1)
      BIZR=FANRE*FRE(1)-FANIM*FIM(1)
      BIZI=FANRE*FIM(1)+FANIM*FRE(1)
      BIOR=FANRE*FRE(2)-FANIM*FIM(2)
      BIOI=FANRE*FIM(2)+FANIM*FRE(2)
  337 RETURN
      END
C   ***************************************************
      SUBROUTINE COMKB (X, Y, BKZR, BKZI, BKOR, BKOI)
C   ***************************************************
C
C        COMPUTES THE REAL & IMAGINARY PARTS OF THE MODIFIED
C        BESSEL FUNCTIONS K0(Z) & K1(Z), WHERE Z = X + I*Y,
C        BY THE METHOD OF Y.L. LUKE, THE SPECIAL FUNCTIONS
C        AND THEIR APPROXIMATIONS, VOL.2, PG. 229.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION P1ZE(16), P2ZE(16), P3ZE(16), Q1ZE(16)
      DIMENSION P1ON(16), P2ON(16), P3ON(16), Q1ON(16)
        DIMENSION RTEST(5), NTEST(5)
      DATA P1ZE /3*0.0
     1          ,-1.59863946,-1.91851852,-2.11452184
     2          ,-2.24917817,-2.34787879,-2.42347618
     3          ,-2.48328717,-2.53181273,-2.57198289
     4          ,-2.60579048,-2.63463947,-2.65954816
     5          ,-2.68127340/
      DATA P2ZE /3*0.0
     1          , .632653061, 1.07407407, 1.38016529
     2          , 1.6035503, 1.77333333, 1.90657439
     3          , 2.01385042, 2.10204082, 2.17580340
     4          , 2.2384, 2.29218107, 2.33888228
     5          , 2.3798127/
      DATA P3ZE /3*0.0
     1          ,-3.40136054E-02,-1.55555556E-01,-2.65643447E-01
     2          ,-3.54372124E-01,-4.25454545E-01,-4.83098217E-01
     3          ,-5.3056325E-01,-5.70228091E-01,-6.03820515E-01
     4          ,-6.32609524E-01,-6.575416E-01,-6.79334126E-01
     5          ,-6.9853933E-01/
      DATA Q1ZE /3*0.0
     1          , 1.63265306, 1.38271605, 1.19008264
     2          , 1.04142012, 9.24444444E-01, 8.30449827E-01
     3          , 7.53462604E-01, 6.89342404E-01, 6.35160681E-01
     4          , 5.888E-01, 5.48696845E-01, 5.13674197E-01
     5          , 4.82830385E-01/
```

```
      DATA P1ON /3*0.0
     1          ,-1.888888889,-2.09090909,-2.23076923
     2          ,-2.33333333,-2.41176471,-2.47368421
     3          ,-2.52380952,-2.56521739,-2.6
     4          ,-2.62962963,-2.65517241,-2.67741935
     5          ,-2.6969697/
      DATA P2ON /3*0.0
     1          , 7.77777778E-01, 1.18181818, 1.46153846
     2          , 1.66666667, 1.82352941, 1.94736842
     3          , 2.04761905, 2.13043478, 2.2
     4          , 2.25925926, 2.31034483, 2.35483871
     5          , 2.39393939/
      DATA P3ON /3*0.0
     1          , 1.11111111E-01,-9.09090909E-02,-2.30769231E-01
     2          ,-3.33333333E-01,-4.11764706E-01,-4.73684211E-01
     3          ,-5.23809524E-01,-5.65217391E-01,-.6
     4          ,-6.2962963E-01,-6.55172414E-01,-6.77419355E-01
     5          ,-6.96969697E-01/
      DATA Q1ON /3*0.0
     1          , 1.77777778, 1.45454545, 1.23076923
     2          , 1.06666667E 00, 9.41176471E-01, 8.42105263E-01
     3          , 7.61904762E-01, 6.95652174E-01, .64
     4          , 5.92592593E-01, 5.51724138E-01, 5.16129032E-01
     5          , 4.84848485E-01/
      DATA C/1.25331414/
      DATA RTEST/ 1., 4., 16., 36., 64./
      DATA NTEST/ 15, 10, 10, 6, 6/
C
C        FIND NTERM, THE NUMBER OF TERMS
C        WE ASSUME THE ARGUMENT HAS BEEN CHECKED TO INSURE THAT R.LE.8
C
      RSQ = X**2 + Y**2
      DO 100 I = 2, 5
      IT = I - 1
      IF(RSQ .LT. RTEST(I) ) GO TO 105
 100  CONTINUE
 105  NTERM = NTEST(IT)
C
C        INITIALIZE FK-1, FK-2, AND FK-3 FOR N=0 AND N=1
C
      FKM3RZ = 1.0
      FKM3RO = 1.0
      PKM3RZ = 1.0
      PKM3RO = 1.0
      FKM3IZ = 0.0
      FKM3IO = 0.0
      PKM3IZ = 0.0
      PKM3IO = 0.0
      HX = X*16.0
      FKM2RZ = (HX + 9.0)/9.0
      PKM2RZ = (HX + 7.0)/9.0
```

```
      FX = X* 3.20
      FKM2RO = FX + 1.0
      PKM2RO = FKM2RO + 1.20
      FKM2IO = Y*3.20
      PKM2IO = FKM2IO
      HY = Y* 16.0
      FKM2IZ = HY/9.0
      PKM2IZ = FKM2IZ
      HYS = HY**2
      T = HX + 25.0
      FKM1RZ = (HX*T + 75.0 - HYS)/75.0
      FKM1IZ = HY*(HX + T)/75.0
      T = HX + 23.0
      PKM1RZ = (HX*T + 43.0 - HYS)/75.0
      PKM1IZ = HY*(HX + T)/75.0
      T = HX + 21.0
      FKM1RO = (HX*T + 35.0 - HYS)/35.0
      FKM1IO = HY*(HX + T)/35.0
      T = HX + 27.0
      PKM1RO = (HX*T + 131.0 - HYS)/35.0
      PKM1IO = HY*(HX + T)/35.0
C
C        BEGIN RECURRENCE
C
      DO 110 K = 3, NTERM
      KP1 = K + 1
C
C        CALCULATIONS OF FKRZ, FKIZ, PKRZ, AND PKIZ FOR N = 0
C
      P1 = P1ZE(KP1)
      P2 = P2ZE(KP1)
      P3 = P3ZE(KP1)
      Q1 = Q1ZE(KP1)
      HX = Q1*X
      HY = Q1*Y
      T1 = FKM1RZ + FKM2RZ
      T2 = FKM1IZ + FKM2IZ
      FKRZ = HX*T1 - P1*FKM1RZ - P2*FKM2RZ - HY*T2 - P3*FKM3RZ
      FKIZ = HX*T2 - P1*FKM1IZ - P2*FKM2IZ + HY*T1 - P3*FKM3IZ
      FKM3RZ = FKM2RZ
      FKM2RZ = FKM1RZ
      FKM1RZ = FKRZ
      FKM3IZ = FKM2IZ
      FKM2IZ = FKM1IZ
      FKM1IZ = FKIZ
      T1 = PKM1RZ + PKM2RZ
      T2 = PKM1IZ + PKM2IZ
      PKRZ = HX*T1 - P1*PKM1RZ - P2*PKM2RZ - HY*T2 - P3*PKM3RZ
      PKIZ = HX*T2 - P1*PKM1IZ - P2*PKM2IZ + HY*T1 - P3*PKM3IZ
      PKM3RZ = PKM2RZ
      PKM2RZ = PKM1RZ
```

```
      PKM1RZ = PKRZ
      PKM3IZ = PKM2IZ
      PKM2IZ = PKM1IZ
      PKM1IZ = PKIZ
C
C        CALCULATIONS OF FKRO, FKIO, PKRO, AND PKIO FOR N = 1
C
      P1 = P1ON(KP1)
      P2 = P2ON(KP1)
      P3 = P3ON(KP1)
      Q1 = Q1ON(KP1)
      HX = Q1*X
      HY = Q1*Y
      T1 = FKM1RO + FKM2RO
      T2 = FKM1IO + FKM2IO
      FKRO = HX*T1 - P1*FKM1RO - P2*FKM2RO - HY*T2 - P3*FKM3RO
      FKIO = HX*T2 - P1*FKM1IO - P2*FKM2IO + HY*T1 - P3*FKM3IO
      FKM3RO = FKM2RO
      FKM2RO = FKM1RO
      FKM1RO = FKRO
      FKM3IO = FKM2IO
      FKM2IO = FKM1IO
      FKM1IO = FKIO
      T1 = PKM1RO + PKM2RO
      T2 = PKM1IO + PKM2IO
      PKRO = HX*T1 - P1*PKM1RO - P2*PKM2RO - HY*T2 - P3*PKM3RO
      PKIO = HX*T2 - P1*PKM1IO - P2*PKM2IO + HY*T1 - P3*PKM3IO
      PKM3RO = PKM2RO
      PKM2RO = PKM1RO
      PKM1RO = PKRO
      PKM3IO = PKM2IO
      PKM2IO = PKM1IO
      PKM1IO = PKIO
 110  CONTINUE
C
C        EVALUATE CONSTANT TERM FOR K0(Z) AND K1(Z)
C        C IS SQUARE ROOT OF PI/2
C
      X2 = -X
      EMX = DEXP(X2)
      RATYX = DABS (Y/X)
      IF (RATYX .GT. 1.E-3) D = DSQRT(RSQ)
      IF ((1.E-8 .LT. RATYX) .AND. (RATYX .LT. 1.E-3)) D =
     *  DABS (X) * (1. + 1./2. *RATYX*RATYX)
      IF (1.E-8.GE.RATYX) D=DABS(X)
      C2 = EMX*C/D
      SR = DCOS(Y)
      TI = - DSIN(Y)
      IF(Y.NE.0.0) GO TO 120
      IF(X.GE.0.0) GO TO 115
      HI = DSQRT(X2)
```

```
      GR = 0.0
      GO TO 125
 115  GR = DSQRT(X)
      HI = 0.D0
      GO TO 125
 120  GR = DSQRT((X + D)*.5)
      HI = DSQRT((X2 + D)*.5)
      IF(Y.LT.0.0)GO TO 125
      HI=-HI
 125  AR = C2*(GR*SR - HI*TI)
      BI = C2*(HI*SR + GR*TI)
C
C        CALCULATE K0(Z) = BKZR + BKZI*I
C
      DEN = FKRZ**2 + FKIZ**2
      UR = (PKRZ*FKRZ + PKIZ*FKIZ)/DEN
      VI = (PKIZ*FKRZ - PKRZ*FKIZ)/DEN
      BKZR = AR*UR - BI*VI
      BKZI = BI*UR + AR*VI
C
C        CALCULATE K1(Z) = BKOR + BKOI*I
C
      DEN = FKRO**2 + FKIO**2
      UR = (PKRO*FKRO + PKIO*FKIO)/DEN
      VI = (PKIO*FKRO - PKRO*FKIO)/DEN
      BKOR = AR*UR - BI*VI
      BKOI = BI*UR + AR*VI
  556 RETURN
      END
```

Subroutine GRID

Subroutine GRID draws a grid on the screen.

Listing

```
      SUBROUTINE GRID
      CALL QLINE(30,0,630,0,7)
      CALL QLINE(30,50,630,50,7)
      CALL QLINE(30,100,630,100,7)
      CALL QLINE(30,150,630,150,7)
      CALL QLINE(30,200,630,200,7)
      CALL QLINE(30,250,630,250,7)
      CALL QLINE(30,300,630,300,7)
      CALL QLINE(30,349,630,349,7)
      CALL QLINE(30,0,30,349,7)
      CALL QLINE(80,0,80,349,7)
      CALL QLINE(130,0,130,349,7)
      CALL QLINE(180,0,180,349,7)
      CALL QLINE(230,0,230,349,7)
      CALL QLINE(280,0,280,349,7)
      CALL QLINE(330,0,330,349,7)
      CALL QLINE(380,0,380,349,7)
      CALL QLINE(430,0,430,349,7)
      CALL QLINE(480,0,480,349,7)
      CALL QLINE(530,0,530,349,7)
      CALL QLINE(580,0,580,349,7)
      CALL QLINE(630,0,630,349,7)
      RETURN
      END
```

Subroutine IJBSSL(A,R1,R2,I,J)

Subroutine IJBSSL calculates

$$\frac{1}{\alpha^3} I(r_2,r_1) = \frac{1}{\alpha^3} \int_{r_1}^{r_2} \alpha r I_1(\alpha r)\, dr$$

and

$$\frac{1}{\alpha^3} J(r_2,r_1) = \frac{1}{\alpha^3} \int_{r_1}^{r_2} \alpha r J_1(\alpha r)\, dr$$

where I_1 is the modified Bessel function of the first kind of order 1, and J_1 is the Bessel function of the first kind of order 1.

Input

A	α in the above equations
R1	r_1 in the above equations
R2	r_2 in the above equations

Output

I	The right side of the first equation above
J	The right side of the second equation above

Listing

```
      SUBROUTINE IJBSSL (A, R1, R2, I, J)
C
C     THIS SUBROUTINE EVALUATES I(X2,X1)/A**3 AND
C     J(X2,X1)/A**3 WHERE X1 = A*R1 AND X2 = A*R2
C     IT USES THE METHODS DESCRIBED ON PAGE 259 AND
C     ON PAGE 261 OF "COMPUTER PROGRAMS FOR SOME EDDY-
C     CURRENT PROBLEMS - 1970"
C
C     DICTIONARY
C     X      - VARIABLE USED TO EVALUATE INTEGRAL OF
C                 X*I1(X) AND X*J1(X)
C     A      - COMMON FACTOR IN X
C     R(K)   - DIFFERENT FACTOR FOR X
C     I      - VALUE OF I(X2,X1)/A**3
```

```
C     J      - VALUE OF J(X2,X1)/A**3
C     II(K)  - INTEGRAL OF (X*I(X))/A**3
C     JJ(K)  - INTEGRAL OF (X*J(X))/A**3
C     I1     - WORKING VALUE OF II(K)
C     J1     - WORKING VALUE OF JJ(K)
C     K      - SUBSCRIPT USED TO KEEP TRACK OF X1 OR X2
C     N      - VARIABLE OF SUMMATION
C     FN     - FLOATING POINT VALUE OF N
C     LIMIT  - LIMIT OF SUMMATION
C     T1     - ((R(K)**3) * X**(2N)) / ((2**(2N+1))*N!*(N+1)!)
C            - FIRST CONSTANT FOR INTG OF X*J1(X)
C     T2     - (-1)**N
C            - SECOND CONSTANT FOR INTG OF X*J1(X)
C     T3     - CONSTANT FOR INTG OF X*I1(X)
C     T4     - T1/(2N+3)
C     T5     - X**2 OR X - PI/4
C     SQR    - -DSQRT(X) OR DSQRT(2/(PI*X))
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 R(2), II(2), JJ(2), I, J, I1, J1
      R(1) = R1
      R(2) = R2
      AAA=A*A*A
C
C     GET BOTH VALUES FOR II(K) AND JJ(K)
C
        DO 50 K=1,2
        X = A*R(K)
        T5 = X*X
C
C     DECIDE WHICH METHOD TO USE
C
        IF (X .GT. 10.) GO TO 20
C
        LIMIT=INT(2.*X+8.)
        T1 = R(K)*R(K)*R(K)/2.
        T2 = 1.
        J1 = T1/3.
        I1 = T1/3.
C
C       EVALUATE SUMMATIONS
C
          DO 10 N=1,LIMIT
          FN = FLOAT(N)
          T1 = T1*T5 / (4.*FN*(FN+1.))
          T2 = -T2
          T4 = T1 / (2.*FN + 3.)
          J1 = J1 + T2*T4
          I1 = I1 + T4
   10     CONTINUE
        IF(X.LT.1.E-10)J1=0.
```

```
      IF(X.LT.1.E-10)I1=0.
      II(K) = I1
      JJ(K) = J1
      GO TO 50
C
C     FOLLOWING USED WHEN X>10
C     IF X>30 WE USE DIFFERENT CONSTANTS
C
   20 IF (X .GT. 30.) GO TO 30
C
      T1 = -((((((-188.1357/X + 109.1142)/X - 23.79333)/X
     2           + 2.050931)/X - 0.1730503)/X + 0.7034845)/X
     3           - 0.064109E-3)
C
      T2 = ((((-5.81751/X + 2.105874)/X - .6896196)/X
     2           + .4952024)/X - (.187344E-2))/X + .7979095
C
      SQR = -DSQRT(X)
      GO TO 40
C
   30 T1 = (8.98975114/T5 - .93457031)/T5 + .875
      T2 = (2.48062114/T5 - .5546875)/X - X
      SQR = .7978845608 * DSQRT(1./X)
C
   40 T5 = X - .7853981635
      JJ(K) = (1. + SQR*(T2*DCOS(T5) + T1*DSIN(T5))) / AAA
C
      IF (X.GT.140.0) GO TO 50
      T3=(((((((1660.794/X-1737.556)/X+543.6694)/X+11.81804)/X
     2           - 33.78366)/X + 5.108402)/X - .6130935)/X
     3           - .3360836)/X + .3987795
C
      II(K) = (T3*DSQRT(X)*DEXP(X)) / AAA
   50 CONTINUE
C
C     GET DIFFERENCE
      I = II(2) - II(1)
      J = JJ(2) - JJ(1)
  668 RETURN
      END
```

Subroutine KJBSSL(A,R1,R2,K,J)

Subroutine KJBSSL calculates

$$\frac{1}{\alpha^3} K(r_2,r_1) = \frac{1}{\alpha^3} \int_{r_1}^{r_2} \alpha r K_1(\alpha r)\, dr$$

and

$$\frac{1}{\alpha^3} J(r_2,r_1) = \frac{1}{\alpha^3} \int_{r_1}^{r_2} \alpha r J_1(\alpha r)\, dr$$

where K_1 is the modified Bessel function of the second kind of order 1, and J_1 is the Bessel function of the first kind of order 1.

Input

A	α in the above equations
R1	r_1 in the above equations
R2	r_2 in the above equations

Output

K	The right side of the first equation above
J	The right side of the second equation above

Listing

```
      SUBROUTINE KJBSSL (A, R1, R2, K, J)
C
C       THIS SUBROUTINE EVALUATES K(X2,X1)/(A**3) AND
C       J(X2,X1)/(A**3) WHERE X1 = A*R(1) AND X2 = A*R(2)
C       IT USES THE METHODS DESCRIBED ON PAGE 259 AND ON
C       PAGE 263 OF "COMPUTER PROGRAMS FOR SOME EDDY-
C       CURRENT PROBLEMS - 1970"
C
C       DICTIONARY
C      X       - VARIABLE USED TO EVALUATE INTEGRAL
C                   X*K1(X) AND X*J1(X)
C       A      - COMMON FACTOR IN X
C       R(I)   - FACTOR FOR DIFFERENT VALUES OF X
C       K      - VALUE OF K(X2,X1)/(A**3)
```

```
C        J      - VALUE OF J(X2,X1)/(A**3)
C      JJ(I)  - INTEGRAL OF (X*J1(X)) - PI/2
C      KK(I)  - INTEGRAL OF (X*I1(X)) - PI/2
C        K1     - WORKING VALUE OF KK(I)
C        J1     - WORKING VALUE OF JJ(I)
C        I      - SUBSCRIPT USED TO KEEP TRACK OF X1, X2
C        N      - VARIABLE OF SUMMATION
C        FN     - FLOATING POINT VALUE OF N
C        LLIMIT- VARIABLE OF SUMMATION
C        ZLOG   - LOG(X/2) - .577215665
C        T1     - ((R(I)**3) * X**(2N)) / ((2**(2N+1))*N!*(N+1)!)
C                 FIRST CONSTANT FOR INTG OF X*J1(X)
C        T2     - (-1)**N
C                 SECOND CONSTANT FOR INTG OF X*J1(X)
C        T3     - 1/1 + 1/2 + ... + 1/N
C                 CONSTANT FOR INTG OF X *K1(X)
C        T4     - 1/(2N+3)
C        T5     - X**2 OR X - PI/4
C        SQR    - -DSQRT(X) OR DSQRT(2/(PI*X))
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8  KK(2), JJ(2), K, J, K1, J1
      DIMENSION R(2)
      DATA C1/.577215665/,C2/.8333333333/,PIO2/1.5707963268/
     *    ,C3/.7978845608/,PIO4/.7853981634/
      R(1) = R1
      R(2) = R2
      AA=A*A
      AAA=AA*A
C
C        GET BOTH VALUES FOR KK(I) AND JJ(I)
C
        DO 50 I=1,2
        X = A*R(I)
        IF(X.GT.1.E-9)GO TO 5
        JJ(I)=0.
        KK(I)=-PIO2/AAA
        GO TO 50
    5   T5 = X*X
C
C        DECIDE WHICH METHOD TO USE
C
        IF (X .GT. 5.) GO TO 20
C
        LIMIT=INT(2.*X+8.)
        T1=R(I)*R(I)*R(I)/2.
        T2 = 1.
        T3 = 0.
        ZLOG=DLOG(X/2.)+C1
        J1 = T1/3.
        K1 = J1*(ZLOG-C2) + R(I)/(AA)
```

```
C
C        EVALUATE SUMMATION
C
          DO 10 N=1,LIMIT
          FN = FLOAT(N)
          T1 = T1*T5 / (4.*FN*(FN+1.))
          T2 = -T2
          T3 = T3 + 1./FN
          T4 = 1./(2.*FN+3.)
          J1 = J1 + T2*T1*T4
          K1 = K1 + T1*T4*(ZLOG-T4-T3-1./(2*(FN+1.)))
   10     CONTINUE
        KK(I)=K1-PIO2/AAA
        JJ(I) = J1
        GO TO 50
C
C       FOLLOWING IS USED WHEN X>5
C       IF X>30 WE USE DIFFERENT CONSTANTS FOR X*J1(X)
C
   20   IF (X .GT. 30.) GO TO 30
C
        T1 = -((((((-188.1357/X + 109.1142)/X - 23.79333)/X
     2            + 2.050931)/X - 0.1730503)/X + 0.7034845)/X
     3             -0.064109E-3)
C
        T2 = ((((-5.817517/X + 2.105874)/X - .6896196)/X
     2           + .4952024)/X - (.187344E-2))/X + .7979095
C
        SQR = -DSQRT(X)
        GO TO 40
C
   30   T1 = (8.98975114/T5 - .93457031)/T5 + .875
        T2 = (2.48062114/T5 - .5546875)/X - X
        SQR=C3*DSQRT(1./X)
C
   40   T5=X-PIO4
        JJ(I) = (1.+SQR*(T2*DCOS(T5)+T1*DSIN(T5)))/AAA
C
C       IF X IS GREATER THAN 77, WE EXPERIENCE UNDERFLOW
C       IN CALCULATING KK(I) AND SO SET KK(I) TO 0.0
C
        IF (X .GT. 77.0) GO TO 45
        T3 = ((((.79898397/X - 1.1768576)/X +0.91571421)/X
     2           - .67491295)/X + 1.0958276)/X + 1.2533263
        KK(I)=-DSQRT(X)*DEXP(-X)*T3/AAA
        GO TO 50
   45   KK(I) = 0.0
   50   CONTINUE
C
      K=KK(2)-KK(1)
      J=JJ(2)-JJ(1)
```

```
RETURN
END
```

Subroutine PCLKUP(DEPTH,RHSMAG,LHSPHA)

Subroutine PCLKUP searches through a lookup table created by program PCBLDF to find the depth and magnitude of an integral corresponding to the phase which it is given as input. It is called by programs PCINV and PCRTSCAN.

Input

LHSPHA	The phase of the integral

Output

DEPTH	The depth at which the integral has this phase
RHSMAG	The magnitude of the integral when it has this phase

Listing

```
      SUBROUTINE PCLKUP(DEPTH,RHSMAG,LHSPHA)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LHSPHA
      DATA PI/3.141592653/,LOU/8/,LOE/40/
      OPEN(LOE,FILE='ASPHAJO.DAT',STATUS='OLD')
C     OPEN(LOE,FILE='ADPHAF.DAT',STATUS='OLD')
      RHSPHA0=0.
 1120 READ(LOE,*,END=1380)Z,RHSMAG,RHSPHA
      DPH=ABS(ABS(RHSPHA)-ABS(RHSPHA0))
      DMG=RHSMAG-RHSMAG0
      RLMR=ABS(ABS(RHSPHA)-ABS(LHSPHA))
      DZ=Z-Z0
      IF(RHSPHA0.EQ.0.)GO TO 1180
      IF(RHSPHA.GE.LHSPHA) THEN
      IF(RHSPHA0.LT.LHSPHA) THEN
      AF=RLMR/DPH
      DEPTH=Z-DZ*AF
      RHSMAG=RHSMAG-DMG*AF
      GO TO 1400
      END IF
      ELSE
      IF(RHSPHA0.GE.LHSPHA) THEN
      AF=RLMR/DPH
      DEPTH=Z-DZ*AF
      RHSMAG=RHSMAG-DMG*AF
      GO TO 1400
      END IF
      END IF
 1180 Z0=Z
      RHSMAG0=RHSMAG
      RHSPHA0=RHSPHA
```

```
      PHDO=RLMR
C      WRITE(LOU,*)Z,RHSPHA
      WRITE(0,*)Z,RHSPHA
 1200 GO TO 1120
 1380      DEPTH=0.
      RHSMAG=0..
 1400 CLOSE(LOE)
      RETURN
      END
```

Subroutine RFLKUP(DEPTH,RHSMAG,LHSPHA)

Subroutine RFLKUP searches through a lookup table created by program RFBLDF to find the depth and magnitude of an integral corresponding to the phase which it is given as input. It is called by program RFINV.

Input

LHSPHA	The phase of the integral

Output

DEPTH	The depth at which the integral has phase LHSPHA
RHSMAG	The magnitude of the integral when it has phase LHSPHA

Listing

```
      SUBROUTINE RFLKUP(DEPTH,RHSMAG,LHSPHA)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LHSPHA
      DATA PI/3.141592653/,LOU/8/,LOE/40/
      OPEN(LOE,FILE='RF25P.500',STATUS='OLD')
      RHSPHA0=0.
 1140 READ(LOE,*,END=1280)Z,RHSMAG,RHSPHA
      DPH=ABS(ABS(RHSPHA)-ABS(RHSPHA0))
      DMG=RHSMAG-RHSMAGO
      RLMR=ABS(ABS(RHSPHA)-ABS(LHSPHA))
      DZ=Z-Z0
      IF(RHSPHA0.EQ.0.)GO TO 1180
      IF(RHSPHA.GE.LHSPHA) THEN
      IF(RHSPHA0.LT.LHSPHA) THEN
      AF=RLMR/DPH
      DEPTH=Z-DZ*AF
      RHSMAG=RHSMAG-DMG*AF
      GO TO 1400
      END IF
      ELSE
      IF(RHSPHA0.GE.LHSPHA) THEN
      AF=RLMR/DPH
      DEPTH=Z-DZ*AF
      RHSMAG=RHSMAG-DMG*AF
      GO TO 1400
      END IF
      END IF
 1180 Z0=Z
      RHSMAG0=RHSMAG
      RHSPHA0=RHSPHA
      PHD0=RLMR
      WRITE(0,*)Z,RHSPHA
```

```
1200 GO TO 1140
1280 DEPTH=0.
     RHSMAG=0.
1400 CLOSE(LOE)
     RETURN
     END
```

Data File REF.DAT

The following data file is required to run some of the reflection coil programs in this report.

20-A	0.0200	0.7500	1.2500	1.5000	0.3500	0.7000	0.5000	0.0000	0.3000	3.7600	13.8000	85.0	56.0
20-B	0.0200	0.7500	1.2500	0.9000	0.3500	0.7000	0.3000	0.0000	0.3000	6.9737	128.4000	84.0	160.0
20-C	0.0200	0.7500	1.2500	0.9000	0.3500	0.7000	0.3000	0.0000	0.3000	2.3424	24.3053	53.0	65.0
20-D	0.0200	0.7500	1.2500	0.9000	0.3600	0.6960	0.3000	0.0000	0.3000	2.9200	23.1000	50.0	72.0
20-F	0.0200	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.3000	0.2100	2.1000	20.0	25.0
R20F	0.0200	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.3000	0.2100	2.1000	20.0	25.0
20-G	0.0200	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.2500	3.9351	24.6792	56.0	66.0
R20G	0.0200	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.2500	3.7000	26.6000	50.0	54.0
20-N	0.0200	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.1500	0.0660	0.3720	6.0	8.0
30-B	0.0300	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.1800	34.8600	74.2300	208.0	144.0
30-D	0.0300	0.7500	1.2500	0.5000	0.3500	0.7000	0.1670	0.0000	0.1800	10.5400	62.8000	100.0	112.0
R30D	0.0300	0.7500	1.2500	0.5000	0.3500	0.7000	0.1670	0.0000	0.1800	10.5400	62.8000	100.0	112.0
30-E	0.0300	0.8000	1.2000	0.5000	0.3500	0.7000	0.1670	0.0000	0.2000	0.8000	2.5780	24.0	21.0
R30F	0.0300	0.7500	1.2500	0.5000	0.3500	0.7000	0.1667	0.0000	0.2333	4.2000	9.1000	64.0	108.0
40-A	0.0400	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.1000	5.4700	205.2000	90.0	264.0
40-B	0.0400	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.1000	15.7400	79.2100	112.0	240.0
40-C	0.0400	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.2000	84.8500	659.5000	378.0	510.0
40-D	0.0400	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.1000	9.7440	114.5000	99.0	154.0
R40E	0.0400	0.7500	1.2500	0.4000	0.3500	0.7000	0.1250	0.0000	0.1500	1.3720	11.2150	35.0	46.0
R40F	0.0400	0.7500	1.2500	0.4000	0.3500	0.7000	0.1250	0.0000	0.3250	3.5000	140.0000	63.0	75.0
60-A	0.0600	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.1200	59.0600	1067.3999	249.0	450.0
60-B	0.0600	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.0500	52.7700	865.6001	252.0	427.5
60-D	0.0600	0.7500	1.2500	0.4000	0.3530	0.7050	0.1000	0.0000	0.0400	0.1970	1.0490	12.0	13.5
60-E	0.0600	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.0400	23.6050	190.7000	170.0	223.0
60-F	0.0600	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.1000	3.1470	31.6700	60.0	90.0
60-G	0.0600	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.1000	3.2565	30.3660	42.0	50.0
R60H	0.0600	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.0400	8.6700	77.2000	104.0	138.0
R60I	0.0600	0.7500	1.2500	0.3000	0.3500	0.7000	0.1000	0.0000	0.5000	8.6700	77.2000	104.0	138.0
83-A	0.0832	0.7508	1.2490	0.3604	0.3604	0.6967	0.0961	0.0000	0.1000	27.2000	728.8999	234.0	528.0
83-B	0.0832	0.7500	1.2500	0.3600	0.3600	0.6960	0.0960	0.0000	0.1000	79.3600	530.5000	360.0	410.0
R83C	0.0830	0.7530	1.2530	0.3614	0.3614	0.6990	0.0964	0.0000	0.0360	10.9000	85.9000	150.0	175.0
100A	0.1000	0.7900	1.2100	0.4200	0.4700	0.7000	0.1000	0.0000	0.0800	0.0310	0.4050	8.8	10.0
120A	0.1200	0.7500	1.2500	0.3000	0.3500	0.7000	0.0667	0.0000	0.0500	13.0600	182.3000	185.0	245.0
140A	0.1400	0.7500	1.2500	0.3930	0.3140	0.7000	0.0643	0.0000	0.0643	0.0950	1.1800	20.0	24.0
150A	0.1500	0.7500	1.2500	0.4000	0.3600	0.7000	0.1000	0.0000	0.0750	182.2000	3335.0000	900.0	1400.0
150B	0.1500	0.7500	1.2500	0.3600	0.3600	0.7000	0.1000	0.0000	0.0750	67.3000	531.0000	513.0	590.0
R150B	0.1500	0.7500	1.2500	0.3600	0.3600	0.7000	0.1000	0.0000	0.1000	67.3000	531.0000	513.0	590.0
200A	0.2000	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.0450	100.1600	7023.1406	860.0	3275.0
200B	0.2000	0.7500	1.2500	0.4000	0.3500	0.7000	0.1250	0.0000	0.0450	174.6000	1656.0000	1008.0	1375.0
250A	0.2500	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.0200	531.0000	5147.0000	2350.0	3450.0
300A	0.3000	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.1000	57.5550	3443.0098	810.0	2925.0
300B	0.3000	0.7500	1.2500	0.4000	0.3500	0.7000	0.0670	0.0000	0.2500	35.0950	472.5500	616.0	660.0
400A	0.4000	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.0400	3.2378	99.0500	238.0	858.0
400B	0.4000	0.7500	1.2500	0.6000	0.3500	0.7000	0.2000	0.0000	0.0125	3.4130	190.3800	238.0	825.0
400C	0.4000	0.7500	1.2500	0.4000	0.3500	0.7000	0.0650	0.0000	0.0400	13.0070	161.2700	414.0	450.0
500C	0.5000	0.7500	1.2500	0.4000	0.3500	0.7000	0.0650	0.0000	0.0400	25.3000	257.4000	585.0	693.0
183B	0.1830	0.9618	1.0382	0.3825	0.7650	0.9180	0.1530	0.0000	0.0546	18.9840	25.7200	116.0	93.0
200W-A	0.2000	0.7400	1.2600	0.5000	0.0000	0.0000	0.0000	0.0000	0.0270	5.8000	0.0000	194.0	0.0
200W-B	0.2000	0.7400	1.2600	0.5000	0.0000	0.0000	0.0000	0.0000	0.0270	5.8000	0.0000	194.0	0.0
65-P	0.0650	0.4615	1.5385	0.1538	0.0000	0.0000	0.0000	0.0000	0.0000	4.1500	0.0000	73.0	0.0
END	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

REFERENCES

1. M. L. Burrows, *A Theory of Eddy Current Flaw Detection*, University Microfilms, Inc., Ann Arbor, Michigan, 1964.

2. C. V. Dodd, W. E. Deeds, J. W. Luquire, and W. G. Spoeri, Union Carbide Corporation Nuclear Division, Oak Ridge National Laboratory, *Some Eddy-Current Problems and Their Integral Solutions*, ORNL-4384, April 1969.*

3. C. V. Dodd, C. C. Cheng, and W. E. Deeds, "Induction Coils Coaxial with an Arbitrary Number of Cylindrical Conductors," *J. Appl. Phys.* 45(2), 638-47.†

4. C. V. Dodd, C. C. Cheng, C. W. Nestor, and R. B. Hofstra, Union Carbide Corporation Nuclear Division, Oak Ridge National Laboratory, *Design of Induction Probes for Measurement of Level of Liquid Metals*, ORNL-TM-4175, May 1973.*

5. W. A. Simpson, C. V. Dodd, J. W. Luquire, and W. G. Spoeri, Union Carbide Corporation Nuclear Division, Oak Ridge National Laboratory, *Computer Programs for Some Eddy-Current Problems - 1970*, ORNL-TM-3295, June 1971.*

6. C. W. Nestor Jr., C. V. Dodd, and W. E. Deeds, Union Carbide Corporation Nuclear Division, Oak Ridge National Laboratory, *Analysis and Computer Programs for Eddy Current Coils Concentric with Multiple Cylindrical Conductors*, ORNL-5220, July 1979.*

---

*Available for purchase from National Technical Information Service, Springfield, Virginia 22161.

†Available in public technical libraries.

NUREG/CR-5553
ORNL/TM-11505
Dist. Category RF

INTERNAL DISTRIBUTION

1. W. R. Corwin
2-6. L. D. Chitwood
7-11. W. E. Deeds
12-16. C. V. Dodd
17. R. W. McClung
18. D. J. McGuire
19. R. K. Nanstad
20-24. J. R. Pate
25. J. G. Purett
26. C. E. Pugh
27. W. A. Simpson
28. G. M. Slaughter
29. J. R. Weir
30-32. M&C Records Office
33. ORNL Patent Section
34. Central Research Library
35. Document Reference Section
36-37. Laboratory Records Department
38. Laboratory Records (RC)
39. Nuclear Safety Information Center

EXTERNAL DISTRIBUTION

40. NRC, OFFICE OF NUCLEAR REGULATORY RESEARCH, Washington, DC 20555
    G. A. Arlotto
    J. Muscara
    C. Z. Serpan

41. ELECTRIC POWER RESEARCH INSTITUTE, P.O. Box 217097, Charlotte, NC 28221
    Mike Elmo
    Kenji Kryzwosz

42. BATTELLE PACIFIC NORTHWEST LABORATORIES, P.O. Box 999, Richland, WA 99352
    Bob Ferris

43. ROCKY FLATS PLANT, P.O. Box 464, Golden, CO 80402
    Bruce Allen

44. 23 Greensburg Street, Delmont Borough, PA 15626
    Steve D. Brown

45. DOE, Oak Ridge Operations Office, Oak Ridge, TN 37831-8600
Office of Assistant Manager for Energy Research
and Development

46-47. DOE, Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831

48-297. Given distribution as shown in category RF (NTIS-10)

NRC FORM 335
(2-89)
NRCM 1102,
3201, 3202

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

(See instructions on the reverse)

1. REPORT NUMBER
(Assigned by NRC. Add Vol., Supp., Rev., and Addendum Numbers, If any.)

NUREG/CR-5553
ORNL/TM-11505

2. TITLE AND SUBTITLE

Computer Programs for Eddy-Current Defect Studies

3. DATE REPORT PUBLISHED

MONTH	YEAR
June	1990

4. FIN OR GRANT NUMBER

B0417

5. AUTHOR(S)

J. R. Pate, C. V. Dodd

6. TYPE OF REPORT

Technical

7. PERIOD COVERED *(Inclusive Dates)*

8. PERFORMING ORGANIZATION – NAME AND ADDRESS *(If NRC, provide Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address; if contractor, provide name and mailing address.)*

Oak Ridge National Laboratory
Oak Ridge, TN 37831

9. SPONSORING ORGANIZATION – NAME AND ADDRESS *(If NRC, type "Same as above"; if contractor, provide NRC Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address.)*

Division of Engineering
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, D. C. 20555

10. SUPPLEMENTARY NOTES

11. ABSTRACT *(200 words or less)*

Several computer programs to aid in the design of eddy-current tests and probes have been written. The programs, written in Fortran, deal in various ways with the response to defects exhibited by four types of probes: the pancake probe, the reflection probe, the circumferential boreside probe, and the circumferential encircling probe. Programs are included which calculate the impedance or voltage change in a coil due to a defect, which calculate and plot the defect sensitivity factor of a coil, and which invert calculated or experimental readings to obtain the size of a defect. The theory upon which the programs are based is the Burrows point defect theory, and thus the calculations of the programs will be more accurate for small defects.

12. KEY WORDS/DESCRIPTORS *(List words or phrases that will assist researchers in locating the report.)*

Eddy-Current
Fortran
Burrows point defect theory
probes

13. AVAILABILITY STATEMENT

14. SECURITY CLASSIFICATION

(This Page)
Unclassified

(This Report)
Unclassified

15. NUMBER OF PAGES

16. PRICE

NRC FORM 335 (2-89)

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

SPECIAL FOURTH-CLASS RATE
POSTAGE & FEES PAID
USNRC

PERMIT No. G-67