FORTTRAN Programs for Transient Eddy Current Calculations Using a Perturbation-Polynomial Expansion Technique

K. H. Carpenter

NOTICE

COPIES OF THIS REPORT ARE AVAILABLE. It
has been reproduced in the best possible
copy to permit the broadest possible avail-
ability.

MASTER

OAK RIDGE NATIONAL LABORATORY

OPERATED BY OAK RIDGE CORPORATION FOR THE UNITED STATES DEPARTMENT OF ENERGY

BLANK PAGE

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
Price: Printed Copy \$5.00; Microfiche \$2.25

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the Energy Research and Development Administration, United States Nuclear Regulatory Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Contract No. W-7405 eng 26

COMPUTER SCIENCES DIVISION

FORTRAN PROGRAMS FOR TRANSIENT EDDY CURRENT CALCULATIONS
USING A PERTURBATION-POLYNOMIAL EXPANSION TECHNIQUE

K. H. Carpenter

Date Published: November 1976

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration nor any of their employees nor any of their contractors, subcontractors, or other employees makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

NOTICE

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

UNION CARBIDE CORPORATION, NUCLEAR DIVISION
operating the
Oak Ridge Gaseous Diffusion Plant Oak Ridge National Laboratory
Oak Ridge Y-12 Plant Paducah Gaseous Diffusion Plant
for the
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

MASTER

ACKNOWLEDGEMENT

The author wishes to thank Dr. H. T. Yeh of the Magnetism and Superconductivity Section of the Thermonuclear Division of ORNL for his helpful advice and encouragement in development of the programs described here. This work was supported by the Magnetism and Superconductivity Section of the Thermonuclear Division.

TABLE OF CONTENTS

ABSTRACT	1
I. INTRODUCTION	1
II. DESCRIPTION OF PROGRAMS	2
1. AMEDY	2
2. PLEQ	6
3. DEQLP	6
4. Test Programs and Miscellaneous Routines	9
III. SAMPLE INPUTS AND OUTPUTS	9
1. AMEDY	9
2. PLEQ	10
3. DEQLP	13
IV. SOURCE LISTINGS	21
V. REFERENCES	63

BLANK PAGE

FORTRAN PROGRAMS FOR TRANSIENT EDDY CURRENT CALCULATIONS USING A PERTURBATION-POLYNOMIAL EXPANSION TECHNIQUE

K. H. Carpenter

ABSTRACT

A description is given of FORTRAN programs for transient eddy current calculations in thin, non-magnetic conductors using a perturbation-polynomial expansion technique. Basic equations are presented as well as flow charts for the programs implementing them. The implementation is in two steps - a batch program to produce an intermediate data file and interactive programs to produce graphical output. FORTRAN source listings are included for all program elements, and sample inputs and outputs are given for the major programs.

I. INTRODUCTION

We have developed an approach for calculation of eddy currents in thin, non-magnetic conductors due to pulsed fields which is based on a perturbation expansion of the integral form of the eddy current equations coupled with a polynomial expansion of the time dependence of the driving fields. The theory of this approach is developed, for general diffusion equations, and applied to one dimensional sample problems in another paper.¹ The technique has been applied to thin conducting plates² and thin conducting cylinders³ where the fields are three dimensional but the region of interest is two dimensional. In the following, brief descriptions and source listings will be presented for the FORTRAN programs used in obtaining the results given in the papers referenced.

The FORT-4 of the routines that follow has a few departures from ANSI standard but should be compatible with most large machines. The primary exceptions are the TYPE and ACCEPT statements used for interaction with PDP-10 terminals. All other I-O operations should be checked for compatibility as well. Calls to graphics subroutines all refer to the TEKTRONIX storage tube terminal routines. These calls would need to be changed if other graphics software or hardware were used.

II. DESCRIPTION OF PROGRAMS

1. AMEDY

The basic equations for eddy current calculation by our method are²

$$\vec{J} = \frac{-\sigma}{t_0} \sum_{m=0}^{\infty} (-\Gamma)^m \left[\vec{A}_m(\vec{x}) + \nabla_x Z_m(\vec{x}) \right] \frac{d^{m+1} f}{dt^{m+1}} \quad (1)$$

where

$$\Gamma = \frac{\mu_0 \sigma}{4\pi} \frac{L^2}{t_0} \quad (2)$$

is the expansion parameter,

$$\sigma \text{ is the conductivity, } \mu_0 = 4\pi \times 10^{-7} \text{ H/m} \quad (3)$$

and L and t_0 are normalization constants between physical space and time coordinates $\vec{r}$, t, and calculational coordinates $\vec{x}$, τ :

$$\vec{r} = L \cdot \vec{x} \quad , \quad t = t_0 \tau \quad (4)$$

The magnetic vector potential inside the conductor due to all currents outside is

$$\vec{A}(\vec{r}, t) = \vec{A}_0(\vec{x}) f(\tau) \quad (5)$$

The A_m , $m > 0$, and Z_m are then obtained from recursively solving

$$\nabla^2 Z_m = 0 \quad \text{with B.C.} \quad \frac{\partial Z_m}{\partial x_n} = -(\vec{A}_m)_n \quad (6)$$

and

$$\vec{A}_{m+1}(\vec{x}) = \int \frac{\vec{A}_m(\vec{x}') + \nabla_{x'} Z_m(\vec{x}')}{|\vec{x} - \vec{x}'|} d^3x' \quad (7)$$

where integration is over the volume of the conductor. Program AMEDY calculates the sequence of functions $A_m + \nabla Z_m$ and stores them on output unit 20 for use by the graphics programs. (The version given in the listing is specifically written for the case where the conducting region is a thin cylinder of finite length with the cylinder radius taken as the normalizing length L ; however, it can be modified easily to correspond to other geometries.) A flow chart for program AMEDY is given in Fig. 1.

Input to AMEDY consists of

1. The spatial dimensions of the conducting region.
2. The number of finite difference mesh points in each dimension.
3. The largest value of m for which $A_m + \nabla Z_m$ is to be calculated. (See listing FORMAT statements for details.)

AMEDY first calls subroutine COSIM to place often used trigonometric data in COMMON storage.

A call to subroutine AOIN then calculates $\vec{A}_0(\vec{x})$ in the conductor corresponding to the source geometry specified in the routine.

(Examples in the listing are for dipoles at the center of the cylinder.)

Fig. 1. Flowchart for AMEDY.

A call to subroutine PMSCRT then solves Eq. (6) for Z_0 . (In the listing, PMSCRT is a double precision version of a Poisson solver developed at NCAR⁴ used with a modification of Eq. (6) corresponding to the case where the volume of the conductor approaches a surface -- a thin cylinder. In this modification the equation solved is

$$\frac{\partial^2 Z}{\partial \theta^2} + \frac{\partial^2 Z}{\partial z^2} = -\frac{1}{R} \frac{\partial Z}{\partial r} \Big|_{r=R} \quad (8)$$

where z is axial coordinate divided by radius R of cylinder and r is radial coordinate and θ angular coordinate. The boundary conditions are $\frac{\partial Z}{\partial z} = -A_z$ at the cylinder ends. It is this reduction to a two dimensional Poisson problem that enables use of the standard Poisson solver. Similar modifications can be made for other geometries where the conductor is essentially two dimensional even though the fields are three dimensional.)

Next subroutine GRADZ is called to calculate $A_0 + \nabla Z_0$ and output the values on the mesh points to the intermediate file. (In the listing a two point central difference is used for the derivatives.)

If higher m values are required, then subroutine AINTG is called to perform the integration of Eq. (7) and control is returned to the point where the call to PMSCRT is made. (In the listing AINTG performs the integration by a two dimensional trapezoidal rule with the point of singularity handled separately. The contribution at the point of singularity ($\vec{x} = \vec{x}'$) is taken as the exact integral of $|\vec{x} - \vec{x}'|^{-1}$ over the mesh volume containing $\vec{x}$, times the numerator of the integrand evaluated at the mesh point.)

(A large diagnostic output is written on unit 6 by the version of AMEDY listed. The corresponding WRITE statements can be removed without affecting the program otherwise.)

If one desires to obtain the components of the external field $\vec{A}_0(\vec{x})$ as output, this can be accomplished by replacing subroutine PWSCRT with a dummy routine which returns $Z_0 \equiv 0$. Then the output for $A_0 + \nabla Z_0$ will be the desired A_0 . (Such a dummy routine is included in the listing below.)

2. PLEQ

Program PLEQ uses the data for $\vec{A}_0 + \nabla Z_0$ produced by AMEDY and outputs a plot of it. For a ramp time dependence, this is a plot of the equilibrium eddy current. The flow chart is given in Fig. 2. (The version in the listing represents the current density vector at a point as an arrow of length proportional to the vector's magnitude and directed the same as the vector. An arrow is drawn at each mesh point or every second or third mesh point depending on an input from the user. Special PDP-10 routines and Tektronix plot routines are called.) If AMEDY has been run with PWSCRT replaced to output $\vec{A}_0$ only, then PLEQ plots the $\vec{A}_0$ component in the surface.

3. DEQLP

Program DEQLP is the one that makes the polynomial fit to $f(\tau)$ so that a transient solution can be simulated by the perturbation expansion of Eq. (1) when $f(\tau)$ is piecewise continuous. A flow chart of DEQLP is given in Fig. 3. The time interval for the fit and the number of terms in the polynomial are input by the user at the plotting terminal.

Fig. 2. Flowchart for PLEQ.

Fig. 3. Flowchart of DEQLP.

The function $f(t)$ is defined by routine FITFU, and the fit to the polynomial is made by the method of unweighted least squares by means of integration of $f(t)$ times the appropriate Legendre polynomial.

Subroutine GAULP performs the integration by a Gaussian method.

Subroutine PLFIT is then called if the user wishes a plot of the fit.

Then subroutine DEQPT is called to calculate the series sum of Eq. (1) and plot it versus time for a specified mesh point. The data for the coefficients A_m at the mesh point is read from the intermediate file produced by AMEDY.

Several alternate versions of DEQPT are given in the listing. These are used for special purposes. Two serve to plot the special one-dimensional diffusion equation solutions used to test the method.¹ One plots the derivative of the fit plotted by PLFIT. Each routine has an interactive dialog with the user at a Tektronix plotting terminal, and the FORTRAN is specialized to the PDP-10.

4. Test Programs and Miscellaneous Routines

For completeness the listing below includes the test programs used to check individual program modules. Also other auxiliary routines are listed. The uses of each are given in comments therein.

III. SAMPLE INPUTS AND OUTPUTS

1. AMEDY

The following is a sample control file for PDP-10 batch operation of AMEDY. The same procedure would be followed interactively, but run times tend to be long. The output is a binary intermediate file and is omitted here.


```
.ASSIGN NUL 6  
.ASSIGN DSK 20  
.RUN AMEDY[200,21570]  
2  
10  
30  
18  
.COPY AMEDY1.DAT<355>[200,21570]=FOR20.DAT  
.DEL FOR20.DAT  
.Q AMEDY.LOG/DISPOSE:RENAME  
.K/P
```

2. PLEQ

The following is typical input and output of PLEQ taken from the terminal CRT.

RUN PLEG

%FASOPN FILE WAS NOT FOUND

UNIT=20 (ISK FOR20 DATE200.215703/ACCESS=SEQIN/MODE=BINARY

ENTER NEW FILE SPECS. END WITH AN #CALD

TAMEDY18

ENTER PLOT INC. (ID) FOR PLOT WITH N= 10 M= 30 AL= 2.00000E+00 DR= 0.0000
0E-01

*2

MAGNIFY(Y OR N) N

EDDY CURRENT EQUILIBRIUM FOR RAMP EXCITATION.
 ORBITING FIELD CENTERED DIPOLE PER TO AXIS DATE RUN: 26-FEB-76
 L R = 2 00000E+00 VEP INC = 10 HOR INC = 30

3. DEQLP

The following are typical sequential CRT pages on running DEQLP with AMEDY data.

RUN DEQLP

ENTER AL, AU, ID - FORMAT(G)

-1

1

1

COEFFICIENTS OF POLYNOMIAL FIT TO RAMP

LOWER LIMIT = -1.00000000D+00

UPPER LIMIT =

1.00000000D+00

N	B(N)
0	4.27246094D-02
1	5.00000000D-01
2	1.15356445D+00
3	2.24869777D-16
4	-1.40991211D+00
5	-2.96248450D-16
6	7.33154297D-01

DO YOU WANT A PLOT OF FIT? (Y OR N)

Y

N = 6
 M = -1.0000+00
 P = 1.0000+00
 C(1), I=0, N
 0.4272461D-01
 0.5000000D+00
 0.1153564D+01
 0.2248698D-15
 -0.1409912D+01
 -0.2962494D-15
 0.7331543D+00

DO YOU WANT A PLOT OF SOLUTION? (Y OR N)

Y

ENTER MESH PT. AT WHICH PLOTS WANTED - I,J - FORMAT(G)

5

8

%FRSOPH FILE WAS NOT FOUND

UNIT=20 DSK FOR20 DATE200.215703 ACCESS=SEQIN MODE=ASCII

ENTER NEW FILE SPECS. END WITH AN \$(ALT)

*AMEDY18

ENTER GAMMA, AL, AU - FORMAT(G)

.02

-1

1

%FRSAPR FLOATING UNDERFLOW PC= 16703

DO YOU WANT A PLOT OF JTHETA? Y OR N.Y

$N = 6$
 $U_1 = 2.000E-02$
 $U_2 = -1.000E+00$
 $U_3 = 1.000E+00$
 $U_4 = 5$
 $U_5 = 8$
 $U_6 = 10$
 $U_7 = 30$
 $U_8 = 2.000E+00$
 $L_F = 2$
 $\times 10^{-2}$

CENTERED DIPOLE PER TO AXIS RUN 26-FEB-76 THETA COMPONENT

DO YOU WANT A PLOT OF JZ? Y OR N: N
DO YOU WANT A PLOT OF MAGNITUDE? Y OR N: N
DO YOU WANT A PLOT OF ANGLE? Y OR N: Y

$N = 5$
 $GMMH = 2.000E-02$
 $HL = -1.000E+00$
 $HU = 1.000E+00$
 $NI = 0.5$
 $NI = 0$
 $NI = 10$
 $NI = 30$
 $L R = 2.000E+00$

CENTERED DIPOLE PER. TO AXIS

RUN 26-FEB-76

ANGLE (= ATANK JZ/JTH)

DO YOU WANT NEW PLOTS FOR SAME FIT & MESH POINT? :N
DO YOU WANT PLOTS FOR ANOTHER MESH POINT WITH SAME FIT? :N
DO YOU WANT A NEW FIT? (Y OR N)

N
STOP

END OF EXECUTION
CPU TIME: 5.05 ELAPSED TIME: 2:54.95
EXIT

IV. SOURCE LISTINGS

The following output of PDP-10 compressed file EDY.CPR lists the files described above. The files are separated by lines with an exclamation point in the first column followed by the file name. Sub-files within files are separated by lines beginning with an asterisk followed by the sub-file name. The PDP-10 command files for the various executable combinations of these files are included. The first file is an index of those that follow. The graphics routines and assembly language routines called are not included.

ILIST.EDY 171
 /AM1CHD.EDY
 /AM2CHD.EDY
 /AM3CHD.EDY
 /PEOCHD.EDY
 /ETPCHD.EDY
 /ID1CHD.EDY
 /ID2CHD.EDY
 /AMPA1.EDY
 /PEOHA1.EDY
 /PTTHA1.EDY
 /GREHA1.EDY
 /EDYTH.EDY
 /IDEX1.EDY
 /IDEX2.EDY
 /FINDER.EDY
 /ANF.EDY
 /FITFU1.EDY
 /FITFU2.EDY
 /AGINI1.EDY
 /AGIN2.EDY
 /PUSDUM.EDY
 /DPUSCR.EDY
 /TAICO.EDY
 /TGRADZ.EDY

This file lists all files used in eddy current programs and contained in compressed file EDY.CPR. Only the files in EDY.CPR plus TTYCR.MAC and the Tektronics graphics routines are needed to execute the eddy current programs on the PDP-10.

Files named ???CHD.EDY are PDP-10 command files.
 Files named ???THA1.EDY are main programs (and may also contain subprogram elements not shared with other programs).
 Files named T?????.EDY are test programs for subprograms.
 All other files contain subprograms.

K.H. Carpenter 21 May 1976

IA1CHD.EDY 44
 AMPA1.EDY
 PUSDUM.EDY
 AGINI1.EDY

:This is the PDP-10 command file for program AMEDY
 :to calculate the perturbation series for a centered
 :dipole directed along the cylinder axis.
 :See comments in AM2CHD.EDY.

IA2CHD.EDY 72
 AMPA1.EDY
 DPUSCR.EDY
 AGIN2.EDY

:This is the PDP-10 command file for program AMEDY
 :to calculate the perturbation series for a centered
 :dipole perpendicular to the cylinder axis.
 :A large diagnostic output file will be produced on unit 6.
 :To kill it, .ASSIGN NUL 6.
 :Run times of 30 min. or so may be required for large MAXM
 :and fine meshes.

!A82CMD.EDY 45

ATTM1.EDY

PUSDM.EDY

ADIN2.EDY

:This is the PDP-10 command file for using program ANEDY
:to calculate data to input to PLED to plot the
:driving field components in the cylinder surface.
:Set MACT=0 for this case.

IPEOCMD.EDY 25

PEJMA1.EDY

TTYCR.MAC

SYS:CALTEK/SEA

:This is the PDP-10 command file for arrow plots
:of equilibrium eddy currents.

IETPCMD.EDY 44

PTTHA1.EDY

EDYTH.EDY

FITFU2.EDY

ANF.EDY

TTYCR.MAC

SYS:TEKADV/SEA

SYS:CALTEK/SEA

:This is the PDP-10 command file for plotting eddy current components
:versus time using the data generated by program ANEDY.

!ID1CMD.EDY 63

PTTHA1.EDY

IDEX1.EDY

FITFU1.EDY

ANF.EDY

TTYCR.MAC

SYS:TEKADV/SEA

SYS:CALTEK/SEA

:This is the PDP-10 command file for the 1st example of 1-Dim.
:diffusion eq. solution of K.H.Carpenter, "Solution of the Diffusion
:Equation by a Perturbation Approach with Polynomial Approximations to the
:Driving term."

!ID2CMD.EDY 71

PTTHA1.EDY

IDEX2.EDY

FITFU2.EDY

ANF.EDY

TTYCR.MAC

SYS:TEKADV/SEA

SYS:CALTEK/SEA

:This is the PDP-10 command file for the 2nd example of K.H.Carpenter,
:"Solution of the Diffusion Equation by a Perturbation Approach with
:Polynomial Approximations to the Driving term."
:(solution of diffusion eq. in 1 dim. with discontinuous conductivity.)

IATMA1.EDY 1262

PROGRAM ANEDY

IK. H. Carpenter, 24 May 1976.

C

C

This is the main program for calculation of perturbation series

C

$A_m(I,J)$ for eddy currents in a cylindrical shell of radius R,

C

length L, and subdivided into N sections on L and M sections on theta.

C

Length and thickness are normalized to radius.

```

C      The maximum value of N & M is 30.
C      MMAX1 is highest term in series  $A_n(I,J)$ , e.g.,  $A_3(I,J)$  for MMAX1=3.
C
C      -----
C      Prompts for input are given on TTY; input is read from UNIT=5.
C      Output is written in binary mode on UNIT=20.
C      Diagnostic output is on UNIT=6.
C      Subroutines called are COSIM, ABIN, PUSCRT (double precision version),
C      GRADZ, AINTG.
C      The version of ABIN used determines the geometrical distribution
C      of the driving field.
C
C      -----
C      Theory is summarized in: K.H.Carpenter & H.T.Yeh,
C      "Eddy Current Calculations for Thin Cylinders of Finite Length with
C      Driving Fields of Ramp Time Dependence."
C      *****
C      IMPLICIT REAL*8(A-H,O-Z)
C      COMMON/ACSN/CHK(31),SH(31),AOUTH(31,31),AOUR(31,31)
1      ,AOUZ(31,31),AINTH(31,31),AINTZ(31,31),AL,AR,BR,PI,ML,N
2      ,I01,I02,I03,M1,M1
C      DIMENSION W(434),BDD(31),BDL(31),BDC(1),BDD(1)
C      NTEST = 0
C      NEW = 0
C      I01 = 6
C      I02 = 5
C      I03 = 20
C      TYPE 10, I02
10      FORMAT(' ENTER AL=LENGTH/RADIUS, '
1      ' N, M, MMAX1:/' FORMAT(G); UNIT',15)
C      READ(I02,20)AL,N,M,MMAX1
20      FORMAT(G)
C      N1=N+1
C      M1=M+1
C      CALL COSIM
C      REWIND I03
C      WRITE(I03)N, M, MMAX1, AL
C      *****
C      WRITE(I01,25) M,N,MMAX1,AL
25      FORMAT('M=',I3,' N=',I3,' MMAX1=',I3,' AL=',1PE20.0)
C      *****
C      CALL ABIN
C      *****
26      WRITE(I01,27)((I,J,AOUR(I,J),AOUZ(I,J),AOUTH(I,J),J=1,M1),I=1,M1)
27      FORMAT('RESULTS OF CALL TO ABIN'/' I',I3,' J',I3,
1      ' AOUR(I,J)',11X,' AOUZ(I,J)',10X,' AOUTH(I,J)'/
2      '(1X,2I4,1P3E20.0)')
C      *****
28      DO 30 L = 1,M1
C      AOUTH(L,M1)=AOUTH(L,1)
C      AOUR(L,M1) =AOUR(L,1)
C      AOUZ(L,M1) =AOUZ(L,1)
C      AINTH(L,M1)=AINTH(L,1)
30      AINTZ(L,M1) =AINTZ(L,1)
C      DO 40 K = 1,M1
C      BDD(K) = -AOUZ(1,K)
40      BDL(K) = -AOUZ(M1,K)
C      APR = PI*2.
C
C      CALL PUSCRT(NEW,0,DB,AL,N,3,BDD,BDL,0,DB,APR,M,0
1      ,BDC,BDD,0,DB,AOUR,31,PT,IER,U)
C      WRITE(I01,50) IER, PT

```

```

50  FORMAT(' IER=', IS, '   PERTURB=', E12.5)
    IF (IER.NE.0) STOP2
C
    CALL GRADZ
    WRITE(103)((AINTH(1,J),AINZ(1,J),I=1,N1),J=1,M)
CCCCCCCC
    WRITE(101,45)NTEST,((1,J,AINTH(1,J),AINZ(1,J),J=1,M,I=1,N1)
45  FORMAT(' IAP=GRADZL M=', I4, '   I   J'
      1  .10X, 'AINTH(1,J)', 11X, 'AINZ(1,J)' /
      2  (1X, 2I4, 1P2E20.0))
CCCCCCCC
    NTEST = NTEST+1
    IF (NTEST.GT.MAXND STOP
    NEW = 1
    CALL AINTG
    GO TO 20
    END
    SUBROUTINE COSIM
    IMPLICIT REAL*8(A-H,O-Z)
C    THIS ROUTINE SETS UP TRIG TABLE FOR AINTG.
    COMMON/ACSPN/CM(31),SM(31),AOUTH(31,31),AOUR(31,31)
      1  ,AOUZ(31,31),AINTH(31,31),AINZ(31,31),AL,AR,DR,P1,MLN
      2  ,I01,I02,I03,M1,N1
    PI = 4.00*DATAN(1.00)
    PM = 2.*PI/M
    WRITE(101,77) PI,PM
77  FORMAT(' PI=', 1P03E20,'   PM=', 030.20)
    DO 10 KS = 1,M
    CM(KS) = DCOS((KS-1)*PM)
    SM(KS) = DSIN((KS-1)*PM)
10
CCCCCCCC
    WRITE(101,20)(CM(J),SM(J),J=1,M)
20  FORMAT(1P2E20.0)
CCCCCCCC
    RETURN
    END
    SUBROUTINE GRADZ
    IMPLICIT REAL*8(A-H,O-Z)
C    THIS SUBROUTINE CALCULATES A + GRAD Z TERM IN EDDY CURRENT
C    PROGRAM.  SEE KHC 1/16/76 FOR DETAILS.
C
    COMMON/ACSPN/CM(31),SM(31),AOUTH(31,31),AOUR(31,31)
      1  ,AOUZ(31,31),AINTH(31,31),AINZ(31,31),AL,AR,DR,P1,MLN
      2  ,I01,I02,I03,M1,N1
    D1 = M/(4.*PI)
    D2 = M/(2.*AL)
    N1 = M+1
    DO 20 J = 1,M
    AINZ(1,J) = 0.
    AINZ(N1,J) = 0.
    DO 10 I = 2,N
    AINZ(1,J) = AOUZ(1,J) + D2*(AOUR((I+1),J)-AOUR((I-1),J))
10  CONTINUE
20  CONTINUE
    DO 40 I = 1,N1
    AINTH(1,I) = AOUTH(1,I) + D1*(AOUR(1,2)-AOUR(1,M))
    DO 30 J = 2,M
    AINTH(1,J) = AOUTH(1,J) + D1*(AOUR(1,J+1)-AOUR(1,J-1))
30  CONTINUE
40  CONTINUE

```



```

      RETURN
      END
      SUBROUTINE AINTG
      IMPLICIT REAL*8(A-H,O-Z)
C     THIS SUBROUTINE DOES THE INTEGRAL OVER R & THETA REQUIRED
C     FOR THE PERTURBATION SOLUTION OF EDDY CURRENT PROBLEM
C     FOR A ON A RT. CIRCULAR CYLINDER.
C     RADIUS IS AR. LENGTH IS AL. N IS # OF DIVISIONS OF LENGTH
C     M # OF DIVISIONS OF ANGLE THETA.
C     OUTPUT ARRAYS ARE NORMALIZED.
C     SEE THEORETICAL DERIVATION KNC 12/29/75.
C
      COMMON/ACSPN/CM(31),SM(31),ADJTH(31,31),AOUR(31,31)
      1  ,AOUZ(31,31),AINTH(31,31),AINZ(31,31),AL,AR,DR,PI,M,N
      2  ,I01,I02,I03,M1,M1
C
      SQR2 = DSORT(2.D0)
      ALRN = AL/(N*SQR2)
      AB = (AL*PI)/(N*2.D0*PI)
      AB2 = DSORT(1.D0 + AB**2)
      AXT = (2.D0*SQR2*N/AL)*(DLOG(AB+AB2)+AB*DLOG((1.D0+AB2)/AB))
      AK = PI *SQR2*AL/(M*N)
      M1 = N
C
CCCCCCCC
      WRITE(I01,9) N,M,AL,AR
      9  FORMAT(/'RESULTS OF CALL TO AINTG'/'ON =',I4,' M=',I4
      1  ,', AL =',IPE16.5,' AR =',E16.5/' J' K ',
      2  ,4X,'AINTH(J,K)', 3X,'AOUTH(J,K)', 5X,'AOUTH(J,K)'
      3  ,6X,'AOUR(J,K)', 6X
      4  ,',AINZ(J,K)',4X,'AOUZ(J,K)',6X,'AOUZ(J,K)')
CCCCCCCC
      DO 30 J = 1,M1
      DO 40 K = 1,M
      AOUR(J,K) = 0.
      AOUTH(J,K) = AINTH(J,K)*AXT
      AOUZ(J,K) = AINZ(J,K)*AXT
      IF((J.EQ.1).OR.(J.EQ.N1)) AOUTH(J,K)=0.5D0*AOUTH(J,K)
      IF((J.EQ.1).OR.(J.EQ.N1)) AOUZ(J,K)=0.5D0*AOUZ(J,K)
C
CCCCCCCC
      WRITE(I01,5) J,K,AINTH(J,K),AOUTH(J,K),AINZ(J,K),AOUZ(J,K)
      5  FORMAT(1X,I3,I4,1P2E15.5,30X,2E15.5)
CCCCCCCC
C
      DO 20 KK = 1,M
      IF(K-KK) 1,2,3
      1  KM = KK-K+1
      ASM = -1.
      GO TO 4
      2  KM = 1
      ASM = 0
      GO TO 4
      3  KM = K-KK+1
      ASM = 1.
      4  IF((KK.EQ.K).AND.(1.EQ.J)) GO TO 23
      DN1 = DSORT(1.-CM(KM)+((J-1)*ALRN)**2)*2.
      AOUTH(J,K)=AOUTH(J,K)+AINTH(1,KK)*CM(KM)/DN1
      AOUR(J,K) =AOUR(J,K) +AINTH(1,KK)*ASM*SM(KM)/DN1
      IF((KK.EQ.K).AND.(N1.EQ.J)) GO TO 25

```

```

23  DN2 = 2.*DSORT(1.-CH(KPD+((J-N1)*ALRN)**2)
    AOUTH(J,K)=AOUTH(J,K)+AINTH(N1,KK)*CH(KPD)/DN2
    AOUR(J,K)=AOUR(J,K)+AINTH(N1,KK)*ASM*SH(KPD)/DN2
25  DO 10 JJ = 2,N
    IF((JJ.EQ.J).AND.(KK.EQ.K)) GO TO 10
    D3 = DSORT(1.-CH(KPD+((J-JJ)*ALRN)**2)
    AOUTH(J,K)=AOUTH(J,K)+AINTH(JJ,KK)*CH(KPD)/D3
    AOUR(J,K)=AOUR(J,K)+AINTH(JJ,KK)*ASM*SH(KPD)/D3
    AOUZ(J,K)=AOUZ(J,K)+AINTZ(JJ,KK)/D3
10  CONTINUE
20  CONTINUE
    AOUTH(J,K) = AK*AOUTH(J,K)
    AOUR(J,K) = AK*AOUR(J,K)
    AOUZ(J,K) = AK*AOUZ(J,K)
CCCCCCCC
27  WRITE(101,27) AOUTH(J,K),AOUR(J,K),AOUZ(J,K)
    FORMAT('+',38X,1P27:5.5,38X,E15.5)
CCCCCCCC
40  CONTINUE
30  CONTINUE
    RETURN
    END

IPEOMAT.EDY      686
PROGRAM PLED      !K. H. CARPENTER, 24 MAY 1976
C THIS PROGRAM MAKES ARROW PLOT OF EQUILIBRIUM EDDY CURRENTS
C IN A THIN CYLINDER FOR RAMP EXCITATION USING DATA FROM FOR20.DAT
C PRODUCED BY PROGRAM AMEDY.
C
C If 'Y' answered to "MAGNIFY?", plot is magnified by two (with
C half off screen).
C Plot increment must be 1,2, or 3 mesh points.
C
DOUBLE PRECISION AL,AR,AINTH(31,31),AINTZ(31,31)
DIMENSION ID(2),INAME(6)
COMMON/CARD/PART,SHARP,IHEAD,XMIN
C
103 = 20
CALL ERRSET(0)
PI = 3.141592654
READ(103)N,M,MAXM,AL,DR
N1=N+1
5  TYPE 10,N,M,AL,DR
10  FORMAT(' ENTER PLOT INC. (1) FOR PLOT WITH N=',13,' M=',13,
1  ' AL=',1PE12.5,' DR=',E12.5/1X,'*')
ACCEPT 15,INP
IF((INP.NE.1).AND.(INP.NE.2).AND.(INP.NE.3))GO TO 5
15  FORMAT(1)
TYPE 20
20  FORMAT(' MAGNIFY?(Y or N):')
ACCEPT 22,MAG
22  FORMAT(A1)
CALL TTYCR(1)
CALL INITT(120)
TYPE 25
25  FORMAT('          EDDY CURRENT EQUILIBRIUM FOR RAMP EXCITATION.')
READ(103) INAME, ID
CUR = 2.*PI
TYPE 20, INAME, ID, AL, N, M
20  FORMAT(' DRIVING FIELD ',6A5,' DATE RUN: '

```

```

1  .205/ L/R ='.1PE12.5.' VER.INC=''.13.' NOR.INC=''.13)
   CALL MOVEA(0.,100.)
   CALL ANMODE
   TYPE 30
30  FORMAT(' L'/' E'/' N'/' G'/' T'/' ' H')
   CALL MOVEA(45.,20.)
   CALL ANMODE
   TYPE 32
32  FORMAT('      CIRCUMFERENCE')
   INEAD = 1
   PART = 0.2
   SHARP = 90.
   PMIN = 0.1
   TOTX = 930.
   TOTY = 600.
   IF(MAG.ED.'Y')TOTX=1000.
   IF(MAG.ED.'Y')TOTY=1300.
   OFFX = 50.
   OFFY = 35.
   RACL = CUR/AL
   RADY = TOTX/TOTY
   IF(RACL.GE.RADY)GO TO 33
   DY = TOTY/N
   DX = TOTY*RACL/N
   GO TO 35
33  DX = TOTX/N
   DY = (TOTY/RACL)/N
C   IF(CUR.GE.ALS) DX=TOTY/(N+4)
C   IF(ALS.GT.CUR)DY=TOTY/(N+4)
C   IF(CUR.GE.ALS)DY=((DX*AL/CUR)*N)/N
C   IF(ALS.GT.CUR)DX=((DY*CUR/AL)*N)/N
35  XM1N = PMIN+AMIN1(DX,DY)
   XM1 = OFFX
   YM1 = OFFY
   YMA = OFFY + DY*N
   XMA = OFFX + DX*XM1
C   DRAW BORDER
C   CALL PLOT(XM1,YM1,3)
C   CALL PLOT(XMA,YM1,2)
C   CALL PLOT(XMA,YMA,2)
C   CALL PLOT(XM1,YMA,2)
C   CALL PLOT(XM1,YM1,2)
   CALL MOVEA(XM1,YM1)
   CALL DASHA(XMA,YM1,30)
   CALL DASHA(XMA,YMA,50)
   CALL DASHA(XM1,YMA,30)
   CALL DASHA(XM1,YM1,30)
   READ(103,END=40)((AINTH(1,J),AINZ(1,J),I=1,N1),J=1,ND)
   GO TO 50
40  CALL ANMODE
   CALL HOME
   TYPE 45
45  FORMAT('//// READ ERROR')
50  SCALE = 0.9 * AMIN1(DX,DY)
   AMAX = 0.
   DO 60 J = 1,M
   DO 60 I = 1,N1
60  AMAX=AMAX1(AMAX,ABS(SNGL(AINTH(1,J))),ABS(SNGL(AINZ(1,J))))
   SCALE = INP*SCALE/AMAX
   DO 70 J = 1,M.INP

```

```

70      DO 70 I = 1,N1,1NP
1      CALL ARO(AINX(I,J)*SCALE,AINZ(I,J)*SCALE,DOO=(J-1)+OFFX
        .BY=(I-1)+OFFY)
        ACCEPT 22, XXXX
        CALL TTYCR(8)
        STOP
        END

```

```

SUBROUTINE ARO(LX,LY,X,Y)
  REAL LX,LY
  COMMON/CARD/PART,SHARP,INHEAD,XMIN
  DATA PI/3.141592654/
  PI4=PI/4.
  IF(LX+LY+LY*LY.LE.XMIN*2)RETURN
  CALL PLOT(X,Y,3)
  CALL PLOT(X+LX,Y+LY,2)
  A = ATAN2(LY,LX)
  A1=A+PI4*(4.+SHARP/98.)
  TL=SQRT(LX+LY+LY*LY)*PART
  CALL PLOT(X+LX+TL*COS(A1),Y+LY+TL*SIN(A1),2)
  IF(INHEAD.LT.1)RETURN
  A2=A+PI4*(4.-SHARP/98.)
  CALL PLOT(X+LX+TL*COS(A2),Y+LY+TL*SIN(A2),2)
  CALL PLOT(X+LX,Y+LY,2)
  RETURN
  END

```

```

IPTTHAI.EDY 4432
*DEOLP.PT

```

```

PROGRAM DEOLP !K.H.Carpenter, 24 May 1976.

```

```

C *****
C DEOLP is the main program for generating plots versus time of
C the solution of a diffusion-like equation (component) at one
C point in space by means of the perturbation expansion method
C with polynomial fit to the time dependence of the driving term.
C (REF: K.H.Carpenter, "Solution of the Diffusion Equation by a
C Perturbation Approach with Polynomial Approximations to the
C Driving Term.")
C Plots are made via a TEKTRONICS terminal using advanced graphics
C programs on the PDP-10.
C First a polynomial fit to the driving function is made.
C AL, AU, are lower, upper time limits, and ID is no.
C of terms in polynomial fit (degree + 1).
C Driving function is specified by FUNCTION FITFU loaded with program.
C Subroutine PLFIT plots the polynomial fit to FITFU.
C The particular equation solved and the type of solution plots
C obtained are determined by subroutine DEOPT.
C -----

```

```

C LOAD PROGRAM DEOLP with GAULP,DLEP,DTLEP,PLFIT(all in file PTTHAI)
C and with desired versions of FITFU,DEOPT and routines called
C thereby (usually ANF) and with plot control programs TTYCR,
C TEKADV/SEA,CALTEK/SEA.
C *****

```

```

5      IMPLICIT REAL*8(A-H,O-Z)
10     DIMENSION ANSW(20), WORK(40), B(20)
C     COMMON/VCOM/ AL, AU, ID, B
      TYPE 10
      FORMAT(' ENTER AL, AU, ID - FORMAT(G)')
      ACCEPT 20, AL, AU, ID

```

```

20  FORMAT(G)
    IF((AU-AL).LE.0.) GO TO 70
    IF(ID.LE.2) GO TO 70
    CALL GAULP(-1.000,1.000,2,ANSU,32,10)
    DO 25 N1 = 1, ID
25  ANSU(N1) = (N1 - 0.500)*ANSU(N1)
    A1 = 2.000/(AU - AL)
    A2 = -(AU + AL)/(AU - AL)
    CALL DTLEP(A1,A2,0,10,ANSU,WORK)
    TYPE 30, AL, AU
30  FORMAT(' COEFFICIENTS OF POLYNOMIAL FIT TO DATA /
1  ' LOWER LIMIT =',1PD10.0,' UPPER LIMIT =',010.0')
    TYPE 31
31  FORMAT('          N          B(N)')
    DO 32 K = 1, ID
    L = K-1
32  TYPE 34, L, B(K)
34  FORMAT(17,1PD20.0)
    TYPE 45
45  FORMAT(' DO YOU WANT A PLOT OF FIT? (Y OR N)')
    ACCEPT 50, 000
50  FORMAT(A1)
    CALL TTYCR(1)
    CALL INITT(120)
    IF(000.EQ.'Y') CALL PLFIT
    TYPE 60
60  FORMAT(' DO YOU WANT A PLOT OF SOLUTION? (Y OR N)')
    ACCEPT 50, 000
    IF(000.EQ.'Y') CALL DEOPT
    CALL TTYCR(0)
    TYPE 65
65  FORMAT(' DO YOU WANT A NEW FIT? (Y OR N)')
    ACCEPT 50, 000
    IF(000.EQ.'Y') GO TO 5
    STOP
70  TYPE 80
80  FORMAT(' INVALID ENTRY. AU.GT.AL & ID.GE.2 REQUIRED')
    GO TO 5
    END

```

*GAULP.PT

```

    SUBROUTINE GAULP(XL,XU,NFOLD2,ANSU,NPT2,10)
    IMPLICIT REAL*8(A-H,O-Z)
    COMMON/COMMON/ AL, AU, ID, 0
    DIMENSION ANSU(20), Y( 20)
C   SUBROUTINE GAULP IS A MODIFICATION OF GAUSSIAN INTEGRATION
C   ROUTINE GAUSS, TO INTEGRATE AN ARRAY OF LEGENDRE POLY. TIMES
C   A FUNCTION TO BE FITTED BY A POLYNOMIAL.
C   FUNCTION FITFU AND SUBROUTINE DTLEP ARE CALLED BY GAULP.
C.....
C   PROGRAM AUTHOR E. B. HARRIS
C   GASEOUS DIFFUSION DEVELOPMENT DIVISION, UNION CARBIDE CORP.,
C   NUCLEAR DIV.,
C   OAK RIDGE, TENN.
C
C   XL=LOWER LIMIT OF INTEGRAL (DOUBLE PRECISION)
C   XU=UPPER LIMIT OF INTEGRAL (DOUBLE PRECISION)
C   NFOLD2=NUMBER OF EQUAL SUBINTERVALS THAT (XL,XU) IS TO BE
C   SUBDIVIDED INTO
C   ANSU=APPROXIMATE VALUE OF THE INTEGRAL AS EVALUATED BY THIS
C   SUBROUTINE (DOUBLE PRECISION)

```

```

C      FUN=THE NAME OF THE DOUBLE PRECISION FUNCTION SUBPROGRAM THAT
C      DEFINES THE INTEGRAND FUNCTION. FUN HAS ONE DOUBLE PRECISION
C      VARIABLE, SAY X, AS AN ARGUMENT AND RETURNS THE VALUE OF THE
C      INTEGRAND AT X (WHERE X IS THE VARIABLE OF INTEGRATION). THE
C      NAME PASSED TO FUN MUST BE DECLARED EXTERNAL IN THE CALLING
C      PROGRAM.
C      NPT2=NUMBER OF POINTS (ORDER) AT WHICH THE INTEGRAND IS TO BE
C      EVALUATED IN EACH SUBINTERVAL, I.E., TOTAL NUMBER OF
C      EVALUATIONS = NPT2*MFOLD2. PERMISSABLE VALUES OF NPT2 ARE
C      2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 20, 24, 32, 40, 48, 64,
C      80, AND 96. IF NPT2 IS .LT. 2, 2 POINTS ARE USED. IF NPT2 IS
C      .GT. 96, 96 POINTS ARE USED. WITHIN THE RANGE OF 2-96, IF NPT2
C      IS NOT ONE OF THE PERMISSABLE VALUES, THE NEXT HIGHER
C      PERMISSABLE VALUE IS USED.
C
C      THE DATA STATEMENTS BEGINNING WITH AN2 AND ENDING WITH AN96D
C      ARE THE ABSCISSAS AND WEIGHT FACTORS FOR THE GAUSSIAN INTEGRATION
C      AND ARE TAKEN FROM THE HANDBOOK OF MATHEMATICAL FUNCTIONS EDITED
C      BY MILTON ABRAMOWITZ AND IRENE A. STEGUN, APPLIED MATHEMATICS
C      SERIES 55, U. S. DEPARTMENT OF COMMERCE, NATIONAL BUREAU OF
C      STANDARDS, 1964, PAGES 916-919.
C.....
      LOGICAL EVEN,READY
      DIMENSION AN2(2),AN3(4),AN4(4),AN5(6),AN6(6),AN7(8),AN8(8),
      > AN9(10),AN10(10),AN12(12),AN16(16),AN20(20),AN24(24), AN32(32),
      > AN40A(20),AN40B(20),AN40C(24),AN40D(24),AN64A(32), AN64B(32),
      > AN80A(20),AN80B(20),AN80C(20),AN80D(20),AN96A(24), AN96B(24),
      > AN96C(24),AN96D(24),CONST(490),JSTART(19),NPTLEG(19)
      EQUIVALENCE (CONST(1),AN2(1)),(CONST(3),AN3(1)), (CONST(7),AN4(1))
      > ,(CONST(11),AN5(1)),(CONST(17),AN6(1)), (CONST(23),AN7(1)),
      > (CONST(31),AN8(1)),(CONST(39),AN9(1)), (CONST(49),AN10(1)),
      > (CONST(59),AN12(1)),(CONST(71),AN16(1)), (CONST(87),AN20(1)),
      > (CONST(107),AN24(1)),(CONST(131),AN32(1)), (CONST(163),AN40A(1)),
      > (CONST(183),AN40B(1)), (CONST(203),AN40C(1)),(CONST(227),AN40D(1))
      > ), (CONST(251),AN64A(1)),(CONST(283),AN64B(1)), (CONST(315),
      > AN80A(1)),(CONST(335),AN80B(1)), (CONST(355),AN80C(1)),
      > (CONST(375),AN80D(1)), (CONST(395),AN96A(1)),(CONST(419),AN96B(1))
      > ), (CONST(443),AN96C(1)),(CONST(467),AN96D(1))
      DATA NPTLEG/2,3,4,5,6,7,8,9,10,12,16,20,24,32,40,48,64,80,96/,
      > JSTART/4,8,12,16,24,32,40,50,60,72,80,100,132,164,204,252, 316,
      > 396,492/,READY/.FALSE./
      DATA AN2/0.577350269189626D00,1.00000000000000D00/
      DATA AN3/0.00000000230D00,0.00000000000D00,0.77459666924D00,
      > 0.55555555555D00/
      DATA AN4/0.33998104358D00,0.65214515406D00,0.86113631159D00,
      > 0.34785434513D00/
      DATA AN5/0.00000000000D00,0.56888888888D00,0.53046931010D00,
      > 0.47862857049D00,0.90617984593D00,0.23692688505D00/
      DATA AN6/0.23061918608D00,0.46791393457D00,0.66120930646D00,
      > 0.36076157304D00,0.93246951420D00,0.17132449237D00/
      DATA AN7/0.00000000000D00,0.41795918367D00,0.40584515137D00,
      > 0.30183005050D00,0.74153118559D00,0.27970539148D00,
      > 0.94910791234D00,0.12948496616D00/
      DATA AN8/0.18343464249D00,0.36268378337D00,0.52553240991D00,
      > 0.31370664597D00,0.79666647741D00,0.22238103445D00,
      > 0.96020985649D00,0.10122853629D00/
      DATA AN9/0.00000000000D00,0.33023935500D00,0.32425342340D00,
      > 0.31234707704D00,0.61337143270D00,0.26061069640D00,
      > 0.83603110732D00,0.18064016069D00,0.96816023950D00,
      > 0.00127438836D00/

```

DATA AN10/0.14887433898D00.0.29552427471D00.
 > 0.43339539412D00.0.26926671930D00.0.67948956829D00.
 > 0.21908636251D00.0.06506336668D00.0.14945134915D00.
 > 0.97398652851D00.0.06667134430D00/
 DATA AN12/0.12523348851D00.0.24914784581D00.
 > 0.36783149899D00.0.23349253653D00.0.58731795428D00.
 > 0.28316742672D00.0.76998267419D00.0.16887832854D00.
 > 0.98411725637D00.0.10693932599D00.0.98156863424D00.
 > 0.84717533638D00/
 DATA AN16/0.09581258983763744D00.0.18945861845586849D00.
 > 0.28160355877925891D00.0.18268341584492358D00.
 > 0.45881677765722738D00.0.16915651939588253D00.
 > 0.61787624482643740D00.0.14959598881657673D00.
 > 0.75548448835588387D00.0.12462897125553387D00.
 > 0.86563128238783174D00.0.09515851168249278D00.
 > 0.94457582387323257D00.0.06225352393864789D00.
 > 0.98948893499164993D00.0.02715245941175489D00/
 DATA AN20/0.07652652113349733D00.0.15275338713872585D00.
 > 0.22778585114164587D00.0.14917298647268374D00.
 > 0.37378688871541956D00.0.14289618931838285D00.
 > 0.51886788195882789D00.0.13168863844917662D00.
 > 0.63685368872651582D00.0.11819453196151841D00.
 > 0.74633198646815879D00.0.18193811981724843D00.
 > 0.83911697182221882D00.0.08327674157678474D00.
 > 0.91223442825132598D00.0.06267284833418986D00.
 > 0.96397192727791379D00.0.04868142988838694D00.
 > 0.99312859918589492D00.0.01761488713915211D00/
 DATA AN24/0.06485689286268562D00.0.12793819534675215D00.
 > 0.19111885747361638D00.0.12583745634682829D00.
 > 0.31584267969616337D00.0.12167847292788339D00.
 > 0.43779358762684513D00.0.1155856688572568D00.
 > 0.5482147138883953D00.0.18744427811596563D00.
 > 0.64863365193697556D00.0.09761865218411388D00.
 > 0.74812419157855436D00.0.08619816153195327D00.
 > 0.82888198597398292D00.0.07334648141198338D00.
 > 0.88641552788448183D00.0.05929858491543678D00.
 > 0.93827455288273275D00.0.04427743881741988D00.
 > 0.9747285559738949D00.0.02853138862893366D00.
 > 0.99518721999782136D00.0.01234122979998719D00/
 DATA AN32/0.04838766568773831D00.0.09654888951472788D00.
 > 0.14447196158279649D00.0.09563872887927485D00.
 > 0.23928736225213787D00.0.09384439988888456D00.
 > 0.33186868228212764D00.0.09117387869576388D00.
 > 0.42135127613853534D00.0.08765289388448381D00.
 > 0.50689998893222939D00.0.08331192422694675D00.
 > 0.58771575724876232D00.0.07819389578787838D00.
 > 0.66384426693821528D00.0.07234579418884852D00.
 > 0.73218211874828968D00.0.0658222227636184D00.
 > 0.79448379596794248D00.0.05868489347853554D00.
 > 0.84936761373256997D00.0.05639885926237617D00.
 > 0.89632115576685212D00.0.04283589882222668D00.
 > 0.93498687593773968D00.0.03427386231382143D00.
 > 0.96476225558758643D00.0.02539286538926285D00.
 > 0.98561151154526833D00.0.01527439473899567D00.
 > 0.99726336134948155D00.0.00781961888947089D00/
 DATA AN40/0.03877241759685882D00.0.07758594797842481D00.
 > 0.11688487867525528D00.0.07783981316424736D00.
 > 0.19269758878137189D00.0.07611836198862624D00.
 > 0.26815218588725368D00.0.07472316985796826D00.
 > 0.34195489882575847D00.0.07288658239588485D00.

> 0.41377920437160500D00.0.07061164739120677D00.
 > 0.40307500168617071D00.0.06791204581523390D00.
 > 0.54946712509512020D00.0.06400431345660103D00.
 > 0.61255308966798023D00.0.06130624249292093D00.
 > 0.67195660461417954D00.0.0574397690939155D00/
 DATA AN400/0.72731025510992710D00.0.05322704690393682D00.
 > 0.77030565142651930D00.0.04069590763507223D00.
 > 0.82461223083331166D00.0.04307090018567327D00.
 > 0.86595950321225959D00.0.03870216797447201D00.
 > 0.90209080696087429D00.0.033460.95202547040D00.
 > 0.93201200827067653D00.0.02793700090002340D00.
 > 0.95791601921379165D00.0.02224504919416695D00.
 > 0.97725994990377426D00.0.01642105030190700D00.
 > 0.99072623869945700D00.0.01049020453115201D00.
 > 0.99023770971055920100.0.00452127709053319D00/
 DATA AN400/0.03230017096206936D00.0.06473769601268392D00.
 > 0.09700469920946269D00.0.06446616443595000D00.
 > 0.16122235606089171D00.0.06392423050404810D00.
 > 0.22476379039460906100.0.06311419220625402100.
 > 0.20736240735545557D00.0.06203942315909266D00.
 > 0.34075500629216073D00.0.06070443916509300D00.
 > 0.40060640199071672D00.0.05911403969839563D00.
 > 0.46690290475095040D00.0.05727729210040321D00.
 > 0.52316097472223303D00.0.05519950369990416D00.
 > 0.57722472600397270D00.0.05209010940519366D00.
 > 0.62006739677651362D00.0.05035903555305447D00.
 > 0.67707237963266390D00.0.04761665047249047D00/
 DATA AN400/0.72403413092301465D00.0.04467456005669420D00.
 > 0.76715903251574033D00.0.04154500294346474D00.
 > 0.80706620402944262D00.0.03824135106503070D00.
 > 0.84350026162439353D00.0.03477722256477043D00.
 > 0.87657202027424790D00.0.03116722703279000D00.
 > 0.90507913671556967D00.0.02742650970035694D00.
 > 0.93130669070655433D00.0.02357076003932437D00.
 > 0.95290770316043006D00.0.01961616045735552D00.
 > 0.97059159254624725D00.0.01557931572294304D00.
 > 0.98412450372202605D00.0.01147723457923453D00.
 > 0.99353017226635075D00.0.00732755390127626D00.
 > 0.99077100725242611D00.0.00315334605230503D00/
 DATA AN640/0.02435029266342443D00.0.04069095700913972D00.
 > 0.07299312170779903D00.0.04057546744150342D00.
 > 0.12146201929612055D00.0.04034476223400295D00.
 > 0.16964442042399201D00.0.04799930059645630D00.
 > 0.21742364374000700D00.0.04754016571403030D00.
 > 0.26460716220075741D00.0.04696010201621001D00.
 > 0.31132207199321095D00.0.04620479650131441D00.
 > 0.35722015833766011D00.0.04549162792741014D00.
 > 0.40227015796399160D00.0.04459055016375656D00.
 > 0.44636601725346400D00.0.04350372452932345D00.
 > 0.40940314570705295D00.0.04247351512365350D00.
 > 0.53127946401909454D00.0.04126256324262352D00.
 > 0.57109564620263403D00.0.03995374113272034D00.
 > 0.61115535517239725D00.0.03855015317061567D00.
 > 0.640000047125435733D00.0.03705512054024004D00.
 > 0.60522631305423324D00.0.03547221325600230D00/
 DATA AN640/0.71900105017161002D00.0.03300016103714160D00.
 > 0.75201990726053109D00.0.03205792035405155D00.
 > 0.70397235094334140D00.0.03023465707240247D00.
 > 0.81326531512279755D00.0.02033967261425940D00.
 > 0.940629296252500D00.0.02637746971505465D00.

> 0.86599939815489281000.0.02435278256871087000.
 > 0.8893154459951141000.0.02227017388838325000.
 > 0.9105221370785028000.0.0201348231535302000.
 > 0.92956917213193957000.0.01795171577569734000.
 > 0.94641137485848201000.0.01572603047602471000.
 > 0.96180879965205371000.0.01346304789671864000.
 > 0.97332632778991096000.0.01116813946013112000.
 > 0.98333625388452595000.0.00884675982636394000.
 > 0.99101337147674432000.0.00650445796897836100.
 > 0.99634011677195527000.0.00414703326056246000.
 > 0.99938584173577213000.0.00178328872169643000/
 DATA AN80A/0.01951138225679399000.0.03901781365630665000.
 > 0.05850443715242066000.0.03895839596276953000.
 > 0.09749839844158459000.0.03883965105905196000.
 > 0.13616482260914388000.0.03866175977407646000.
 > 0.17471229183264681000.0.03842439300695942500.
 > 0.2129945028576613000.0.03812971131447763000.
 > 0.25095235939227212000.0.03777636436780139000.
 > 0.28852805480451105000.0.03736549023873849000.
 > 0.32566437074770191000.0.03689771463827680000.
 > 0.36230475349948731000.0.03637374990583597000/
 DATA AN80B/0.30939340508196922000.0.03579439395341605000.
 > 0.43387537083175609000.0.03516052904474759000.
 > 0.46869661517054447000.0.03447312045175392000.
 > 0.50280411180879498000.0.03373321498461152000.
 > 0.53614592809713193000.0.03294193939764540000.
 > 0.56867126812270978000.0.03210049867348777000.
 > 0.60033862282975174000.0.03121017410811470000.
 > 0.63107577304687196000.0.03027232175955798000.
 > 0.66085989898611980000.0.02928836958326784000.
 > 0.68963764434202760000.0.02825981605727686000/
 DATA AN80C/0.71736518536289988000.0.02718822750048638000.
 > 0.74480829758359727000.0.02607523576756511000.
 > 0.76950242013504137000.0.02492253576411549000.
 > 0.79393271750460544000.0.02373108205593010000.
 > 0.81695413868146347000.0.02250509324633246000.
 > 0.83803147358025527000.0.02124402611578200000.
 > 0.85943140666311109000.0.01995061087814199000.
 > 0.87872256767821382000.0.01862681420829903000.
 > 0.8966755704387068000.0.01727465205626938000.
 > 0.91326310257175765000.0.01589618358372568000/
 DATA AN80D/0.92945987717244579000.0.01449350884058987000.
 > 0.94224276130987267000.0.01306876159240133000.
 > 0.95459076634363490000.0.01162411412079782000.
 > 0.96548503904379925000.0.01016176604110306000.
 > 0.97430914058572779000.0.00868394526926085000.
 > 0.98204857273852907000.0.00719290476811731000.
 > 0.98929130249975553000.0.00569092245140319000.
 > 0.99122754096560827000.0.00418031312469489000.
 > 0.9933005430825768000.0.00266353558951260000.
 > 0.9953005430825768000.0.00114495000310694000/
 DATA AN80E/0.0102257440496296000.0.03255061449236316000.
 > 0.0407100010501023000.0.03251611071306883000.
 > 0.071071054014555000.0.0324716371406426000.
 > 0.1100003011066592000.0.03234302256057592000.
 > 0.14597371455402694000.0.03220620479403025000.
 > 0.1780968236761866000.0.03203445623199266000.
 > 0.2100313104605072000.0.03181875889441100000.
 > 0.24174315616384001000.0.03158933877072716000.
 > 0.27319881259104914000.0.03131642559686135000.

```

> 0.30436494435449635D00.0.03101033250631303D00.
> 0.33520852209262542D00.0.03067137612366914D00.
> 0.36569686147231363D00.0.03029991542082759D00./
DATA AN96B/0.39579764982090860D00.0.02989634413632838D00.
> 0.42547898840730054D00.0.02946188995816790D00.
> 0.45470942216774300D00.0.02899461415055523D00.
> 0.48345797392059635D00.0.02849741106508538D00.
> 0.51169417715466767D00.0.02797000761684833D00.
> 0.53938810832435743D00.0.02741296272602924D00.
> 0.56651041856139716D00.0.02682686672559176D00.
> 0.59303236477757208D00.0.02621234073567241D00.
> 0.61892584012546857D00.0.02557003600534936D00.
> 0.64416340370496710D00.0.02490063322248361D00.
> 0.66871831004391615D00.0.02420484179236469D00.
> 0.69256453664217156D00.0.02348339908592621D00./
DATA AN96C/0.71567681234896762D00.0.02273706965032937D00.
> 0.73803064374440013D00.0.02196664443874434D00.
> 0.7596023417664749D00.0.02117293989219129D00.
> 0.7803690436743321D00.0.02035679715433332D00.
> 0.80030874413914081D00.0.01951908114014502D00.
> 0.81940031073793167D00.0.01866067962741146D00.
> 0.83762351122810712D00.0.01778250231604526D00.
> 0.85495903343460145D00.0.01688547986424517D00.
> 0.87138850590929650D00.0.01597056290256229D00.
> 0.88683451740242041D00.0.01503872102699493D00.
> 0.90146063531585234D00.0.01409094177231406D00.
> 0.91507142312009807D00.0.01312022956696157D00./
DATA AN96D/0.92771245672230069D00.0.01215168467100031D00.
> 0.93937033975275521D00.0.01116210209983849D00.
> 0.95003271778443763D00.0.01016077053500041D00.
> 0.95968829144874253D00.0.00914067123078338D00.
> 0.96832682846326421D00.0.00812687692569875D00.
> 0.97593917450513646D00.0.0070964707915386D00.
> 0.98251726356301457D00.0.00605054550423596D00.
> 0.98805412632962379D00.0.00501420274292751D00.
> 0.99254390032376262D00.0.00396455433844468D00.
> 0.99598184298720929D00.0.00291073181793494D00.
> 0.99836437506310167D00.0.00185396078894692D00.
> 0.99968950388323076D00.0.00079679206555201D00./

```

C.....

```

FUNT(X) = FITFU(AA*X + AB)
AA = (AU - AL)/2.D00
AB = (AU + AL)/2.D00

```

C.....

C.....

```

C      ON FIRST TIME PROGRAM IS INVOKED, TAKE ONE-HALF OF THE CONSTANTS
C      IN ARRAY CONST.

```

C.....

```

      IF (READY) GO TO 20
      DO 10 I=1,490
        CONST(I)=0.5D+00*CONST(I)
10      CONTINUE
      READY=.TRUE.
20      NPT=IABS(NPT2)
      NFOLDS=IABS(NFOLD2)

```

C.....

```

C      DETERMINE THE PERMISSABLE NUMBER OF POINTS TO USE IN EACH
C      SUBINTERVAL AND THE STARTING POINT IN ARRAY CONST FOR THAT
C      NUMBER OF POINTS.

```

C.....

```

      IF (NPT.LT.2) NPT=2
      IF (NPT.GT.96) NPT=96
      DO 30 I=1,19
        NHOLD=NPTLEG(I)
        IF (NPT.EQ.NHOLD) GO TO 40
        IF (NPT.GT.NHOLD) GO TO 30
        NPT=NHOLD
        GO TO 40
    30  CONTINUE
    40  JHOLD=JSTART(I)
        EVEN=.FALSE.
        IF (MOD(NPT,2).EQ.0) EVEN=.TRUE.
        H=(XJ-XL)/DBLE(FLOAT(NFOLDS))
        A=XJ-0.5D+00*H
        DO 11 NN = 1, ID
    11  ANSW(NN)=0.D+00
        KSTOP=NPT/2
        DO 60 I=1,NFOLDS
          J=JHOLD
          A=A+H
          DO 50 K=1,KSTOP
            J=J-3
            C=CONST(J)*H
            J=J+1
            FUN = FUNT(A+C)
            CALL DLEP(Y,A+C,ID-1)
            DO 22 NN = 1, ID
    22  ANSW(NN) = ANSW(NN) + CONST(J)*Y(NN)*FUN
            FUN = FUNT(A-C)
            CALL DLEP(Y,A-C,ID-1)
            DO 33 NN = 1, ID
    33  ANSW(NN) = ANSW(NN) + CONST(J)*Y(NN)*FUN
          50  CONTINUE
          IF (EVEN) GO TO 60
          FUN = FUNT(A)
          CALL DLEP(Y,A,ID-1)
          DO 44 NN = 1, ID
    44  ANSW(NN) = ANSW(NN) + CONST(J-2)*Y(NN)*FUN
          60  CONTINUE
          DO 55 NN = 1, ID
    55  ANSW(NN) = ANSW(NN)*H
        RETURN
      END
*DLKP.PT
      SUBROUTINE DLEP(Y,X,N)
      IMPLICIT REAL*8 (A-H,O-Z)
C  LEGENDRE POLYNOMIALS P(N,X)
      DIMENSION Y(20)
      Y(1)=1.D0
      IF(N) 1,1,2
    1  RETURN
    2  Y(2)=X
      IF(N-1) 1,1,3
    3  G=2*X*Y(1)
    4  Y(3)=G-Y(1-1)+G-(G-Y(1-1))/DFLOAT(1)
      RETURN
      END
*DTLEP.PT
      SUBROUTINE DTLEP(A,B,POL,N,C,W)

```

```

      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION POL(1),C(1),W(1)
      IF(N-1)2,1,3
1     POL(1) = C(1)
2     RETURN
3     POL(1) = C(1) + B*C(2)
      POL(2) = A*C(2)
      IF(N-2)2,2,4
4     W(1) = 1.D0
      W(2) = B
      W(3) = 0.D0
      W(4) = A
      FI = 1.D0
      DO 6 J = 3, N
      FI = FI + 1.D0
      Q = 1.D0/FI-1.D0
      Q1 = 1.D0 - Q
      P = 0.D0
      DO 5 K = 2, J
      H = (A*P+B*W(2*K-2))*Q1+Q*W(2*K-3)
      P = W(2*K-2)
      W(2*K-2)=H
      W(2*K-3)=P
5     POL(K-1)=POL(K-1)+H*C(J)
      W(2*J-1)=0.D0
      W(2*J)=A*P*Q1
6     POL(J)=C(J)*W(2*J)
      RETURN
      END
*PLFIT.PT
      SUBROUTINE PLFIT
      REAL *4 X, Y
      DOUBLE PRECISION B, AL, AU
      COMMON/VCOM/AL, AU, ID, B
      COMMON/PTCM/X, Y, N, IS
      DIMENSION B(20), X(400), Y(400),N(2),IS(2)
      N(1) = 200
      N(2) = 200
      IS(1) = 'L'
      IS(2) = 'L'
      DELTA = (AU - AL)/200.
      DO 10 I = 1, 200
      T = AL + DELTA*I
      X(I) = T
      X(I+200) = T
      Y(I + 200) = FITFU(T)
      Y(I) = B(1)
      IF(T.EQ.0.) GO TO 10
      DO 5 J = 2, 10
      Y(I) = Y(I) + B(J)*(T**(J-1))
5     WRITE(6,15) X(I), Y(I), Y(I+200)
C 15     FORMAT(1P3E20.6)
10     CONTINUE
      CALL PLOTCH(X, Y, 400, N, IS, 2)
      CALL HOME
      CALL MOVABS(0.700)
      CALL ANMODE
      IDT = ID - 1
      TYPE 25, IDT, AL, AU
25     FORMAT(' N =', I3, ' A =', 1PD10.3, ' B =', D10.3)

```

```

      TYPE 30, (C(K),K=1,ID)
30    FORMAT(/ ' C(1), 1=0,N'/(D15.7))
      ACCEPT 20, DUMMY
      CALL ERASE
20    FORMAT(A5)
      RETURN
      END

```

IGREMAI.EDY 243

C This program calculates the coefficients for the perturbation
 C expansion of the one-dimensional diffusion equation with
 C discontinuous conductivity as used in the IDEX2.EDY
 C version of DEOPT. See that file for further comments.

```

      DIMENSION HI(1001), HO(1001)
C      INITIALIZE
      N = 500
      M = 15
      WRITE(6,25) N
25    FORMAT(' SAVS'/' NO. OF INTERVALS IN INTEGRALS = ', 15)
      N1 = N+1
      AN = N
      D3 = 1./AN
      D2 = D3*D3
      DO 30 I = 1, N1
      A11 = I - 1
30    HI(I) = 384.75 - 20.25*(A11/AN)**2
      L = 1
      WRITE(6,40) L, HI(1)
40    FORMAT('      I      HI(0)' / 16, 1PE20.8)
      DO 110 K = 1, M
      H1 = (HI(1) + HI(N1))/2.
      DO 50 L = 2, N
      H1 = H1 + HI(L)
50    H1 = -10.*H1/AN
      DO 60 J = 1, N1
      J1 = J-1
      AJ1 = J1
      AJN = AJ1/AN
      HO(J) = H1
      IF(J.EQ.1) GO TO 60
      H2 = (HI(1) + HI(J))/2.
      IF(J.EQ.2) GO TO 65
      DO 60 L = 2, J1
      H2 = H2 + HI(L)
60    HO(J) = HO(J) + AJ1 * H2 * D2
      IF(J.EQ.N1) GO TO 85
      H2 = (HI(N1) + HI(J)*AJ1*D3)/2.
68    IF(J.EQ.N) GO TO 80
      J2 = J+1
      DO 70 L = J2, N
      AL1 = L - 1
      H2 = H2 + AL1*HI(L)*D3
70    HO(J) = HO(J) + H2*D3
80    DO 90 L = 1, N1
85    HI(L) = HO(L)
90    K1 = K + 1
      WRITE(6,100) K1, HI(1)
100   FORMAT(16, 1PE20.8)
110   CONTINUE
      STOP

```

END

IEDYTIM.EDY 989

SUBROUTINE DEOPT

This version of DEOPT, when used with DEOLP, plots eddy current components vs. time at a selected mesh point.

The perturbation series coefficient data must be read from file FOR20.DAT in unformatted binary mode as determined by the

READ statements below. The theory is summarized in the papers:

K.H.Carpenter & H.T.Yeh, "Eddy Current Calculations for Thin Cylinders of Finite Length with Driving Fields of Ramp Time Dependence."

K.H.Carpenter, "Solution of the Diffusion Equation by a Perturbation Approach. . ."

This version of DEOPT requires loading of AMF and the time ramp version of FITFU.

DOUBLE PRECISION B,AL,AU,AINTH(31,31),AINZ(31,31)

DIMENSION B(20),X(400),Y(400),N(2),IS(2),HTH(20),HZ(20),

1 HTHA(20),HZA(20),IDD(2),INAM(6),CTTH(20),CTZ(20)

COMMON/VCOM/AL,AU,ID,B

COMMON/PTCM/X,Y,N,IS

DATA NNN/0,MMM/0/

NPP = ID-1

N(1) = 101

N(2) = 101

IS(1) = 'L'

IS(2) = 'L'

5 TYPE 10

10 FORMAT(' ENTER MESH PT. AT WHICH PLOTS WANTED -',

1 ' I,J - FORMAT(G)')

ACCEPT 20, NN, MM

20 FORMAT(G)

IF((NN.LT.1).OR.(MM.LT.1).OR.(NN.GT.31).OR.(MM.GT.31))GO TO 5

IF((NN.EQ.NNN).AND.(MM.EQ.MMM))GO TO 27

CLOSE(UNIT=20)

OPEN(UNIT=20,ACCESS='SEQ IN')

READ(20) NA,MA,MAX1,ALL

READ(20) INAM,IDD

N1 = NA+1

MAX1 = MAX1+1

MAX2 = MAX1+2

DO 25 K = 1, MAX1

READ(20)((AINTH(I,J),AINZ(I,J),I=1,N1),J=1,MA)

HTH(K) = AINTH(NN,MM)

HZ(K) = AINZ(NN,MM)

25 CONTINUE

NNN = NN

MMM = MM

27 IF(ID.GT.MAX2) GO TO 470

30 TYPE 40

40 FORMAT(' ENTER GAMMA, AL, AU - FORMAT(G)')

ACCEPT 20, GA, AL2, AU2

DELTA = (AU2-AL2)/100.0

IF(DELTA.LE.0.0) GO TO 30

HTHA(1) = HTH(1)

HZA(1) = HZ(1)

SI = -1.

DO 50 K = 2, MAX1

GAP = GA**K(K-1)

```

      HTHA(K) = HTH(K)*GAP*SI
      HZA(K) = HZ(K)*GAP*SI
50    SI = -SI
      DO 70 J = 1, ID-2
      CTTH(J) = HTHA(1)*B(J+1)*J
      CTZ(J) = HZA(1)*B(J+1)*J
      DO 60 K = 2, ID - J
      KKK = K+1
      JJJ = J
      ANFF = ANF(KKK, JJJ)
      CTTH(J) = CTTH(J) + HTHA(K)*B(J+K)*ANFF
60    CTZ(J) = CTZ(J) + HZA(K)*B(J+K)*ANFF
70    CONTINUE
      CTTH(NPP) = HTHA(1) * B(ID) * NPP
      CTZ(NPP) = HZA(1) * B(ID) * NPP
      EOTH = HTHA(1)
      EOZ = HZA(1)
      T = AL2
      DO 90 K = 1, 101
      X(K) = T
      X(K+101) = T
      Y(K) = 0.0
      IF(T.GT.0) Y(K) = EOTH
      Y(K+101) = CTTH(1)
      Y(K+202) = CTZ(1)
      TT = T
      DO 80 L = 2, NPP
      Y(K+101) = Y(K+101) + CTTH(L)*TT
      Y(K+202) = Y(K+202) + CTZ(L)*TT
80    TT = TT*T
90    T = T + DELTA
      TYPE 100
100   FORMAT(' DO YOU WANT A PLOT OF JTHETA? Y OR N: '9)
      ACCEPT 110, CD
110   FORMAT(A1)
      IF(CD.NE.'Y') GO TO 140
      IF1 = 1
      GO TO 400
120   TYPE 130
130   FORMAT(' THETA COMPONENT')
      ACCEPT 110, CD
      CALL NEWPAG
140   TYPE 150
150   FORMAT(' DO YOU WANT A PLOT OF JZ? Y OR N: '9)
      ACCEPT 110, CD
      IF(CD.NE.'Y') GO TO 185
      T = AL2
      DO 160 K = 1, 101
      X(K+202) = Y(K+101)
      Y(K+101) = Y(K+202)
      Y(K) = 0.0
      IF(T.GT.0) Y(K) = EOZ
160   T = T + DELTA
      IF1 = 2
      GO TO 400
170   TYPE 180
180   FORMAT(' Z COMPONENT')
      ACCEPT 110, CD
      CALL NEWPAG
185   TYPE 190

```

```

190  FORMAT(' DO YOU WANT A PLOT OF MAGNITUDE? Y OR N: '$)
    ACCEPT 110,CD
    IF(CD.NE.'Y') GO TO 225
    T = AL2
    DO 200 K = 1, 101
        Y(K) = 0.0
        IF(T.GT.0.0) Y(K) = SORT(EQ2**2 + EQTH**2)
        Y(K+101) = SORT(Y(K+101)**2 + X(K+202)**2)
200  T = T + DELTA
    IF1 = 3
    GO TO 400
210  TYPE 220
220  FORMAT(' MAGNITUDE')
    ACCEPT 110,CD
    CALL NEWPAG
225  TYPE 230
230  FORMAT(' DO YOU WANT A PLOT OF ANGLE? Y OR N: '$)
    ACCEPT 110, CD
    IF(CD.NE.'Y') GO TO 455
    ATT = ATAN2(EQ2,EQTH)
    DO 240 K = 1, 101
        Y(K) = ATT
240  Y(K+101) = ATAN2(Y(K+202),X(K+202))
    IF1 = 4
400  CALL PLOTCH(X,Y,202,N,IS,2)
    CALL MOVABS(0,700)
    CALL ANMODE
    TYPE 420, NPP, GA, AL, AU
420  FORMAT(' N = ', I3// ' GAMMA = ', IPE10.3// ' AL = ', E10.3// ' AU = ',
1  E10.3)
    TYPE 430, NN, MM, NA, MA, ALL
430  FORMAT(' NN = ', I3// ' MM = ', I3// ' NA = ', I3// ' MA = ', I3// ' L/R = ',
1  IPE10.3)
    CALL MOVABS(0,50)
    CALL ANMODE
    TYPE 440, INAM, IDD
440  FORMAT(IX,6AS,4X, ' RUN ', 2AS$)
    IF(IF1.EQ.1) GO TO 120
    IF(IF1.EQ.2) GO TO 170
    IF(IF1.EQ.3) GO TO 210
    TYPE 450
450  FORMAT(' ANGLE (= ATAN(JZ/JTH)')
    ACCEPT 110, CD
    CALL NEWPAG
455  TYPE 460
460  FORMAT(' DO YOU WANT NEW PLOTS FOR SAME FIT & MESH POINT? : '$)
    ACCEPT 110,CD
    IF(CD.EQ.'Y') GO TO 30
    TYPE 465
465  FORMAT(' DO YOU WANT PLOTS FOR ANOTHER MESH POINT WITH SAME'
1  ' FIT? '$)
    ACCEPT 110,CD
    IF(CD.EQ.'Y') GO TO 5
    RETURN
470  TYPE 400, ID, MAX2
400  FORMAT(' ID = ', I3, ' MAXIMUM VALUE ALLOWED FOR ID WITH THIS'
1  ' DATA IS', I3)
    RETURN
    END

```


INDEX1.EBY 292

```

SUBROUTINE DEOPT
C   This version of DEOPT is used with DEOLP to calculate & output
C   results for the 1-dimensional diffusion equation that is the
C   first example in: K.H.Carpenter, "Solution of the Diffusion Equation
C   by a Perturbation Approach with Polynomial Approximations to
C   the Driving Term."
C   This version of DEOPT requires loading of ANF and the time step
C   version of FITFU.

DOUBLE PRECISION B, AL, AU
DIMENSION B(20), X(400), Y(400), N(2), IS(2), C(21)
COMMON/VCOM/AL, AU, ID, B
COMMON/PTCM/X, Y, N, IS
NPP = ID-1
N(1) = 101
N(2) = 101
IS(1) = 'L'
IS(2) = 'L'
30  TYPE 40
40  FORMAT(' ENTER GAMMA, AL, AU - FORMAT(G)')
ACCEPT 45, GA, AL2, AU2
45  FORMAT(G)
D = (AU2-AL2)/100.
IF(D.LE.0.)GO TO 30
T = AL2
DO 70 I = 1, ID
GA2 = GA
C(I) = B(I)
IF(1.EQ.ID)GO TO 70
DO 80 J = I+1, ID
C(I) = C(I) - B(J)*ANF(J-I+1,1)*GA2
80  GA2 = -GA2*GA
70  CONTINUE
DO 50 K = 1, 101
X(K) = T
X(K+101) = T
Y(K) = 0.
IF(T.GT.0.) Y(K) = 1. - EXP(-T/GA)
Y(K+101) = C(I)
TT = T
DO 60 L = 2, ID
Y(K+101) = Y(K+101) + C(L)*TT
60  TT = TT*T
50  T = T + D
400 CALL PLOTCH(X, Y, 202, N, IS, 2)
CALL MOVABS(0, 700)
CALL ANFODE
TYPE 420, NPP, GA, AL, AU
420  FORMAT(' N = ', I3, ' GAMMA = ', 1PE10.3, ' AL = ', E10.3, ' AU = ',
1  E10.3)
ACCEPT 430, DDD
430  FORMAT(A1)
CALL HELPPG
TYPE 440
440  FORMAT(' DO YOU WANT ANOTHER PLOT FOR THIS FIT? Y OR N: '$)
ACCEPT 430, DDD
IF(DDD.EQ.'Y')GO TO 30
RETURN
END

```

I1DEX2.EDY 499

SUBROUTINE DEOPT

C This version of DEOPT is used with DEOLP to calculate and output
 C results for the one dimensional diffusion equation with discontinuous
 C conductivity. With a change of scale this is the 2nd example of the
 C paper: K.H.Carpenter, "Solution of the Diffusion Equation
 C by a Perturbation Approach . . ."
 C Data in array HA was calculated by GRENAI.
 C This version of DEOPT requires loading of ANF and time ramp
 C version of FITFU.

DOUBLE PRECISION B, AL, AU

COMMON/COM/AL, AU, ID, B

DIMENSION B(20), X(400), Y(400), N(2), IS(2), H(20), HA(20)

1 .CT(20)

COMMON/PTCM/ X, Y, N, IS

N(1) = 101

N(2) = 101

IS(1) = 'L'

IS(2) = 'L'

DATA HA/-40.5,

1 3.04750000E 02,

2 -3.59247503E 03,

3 3.35361992E 04,

4 -3.13000430E 05,

5 2.92276700E 06,

6 -2.72852160E 07,

7 2.54714000E 08,

8 -2.37777971E 09,

9 2.21977149E 10,

A -2.07227505E 11,

B 1.93455666E 12,

C -1.80595657E 13,

D 1.68580019E 14,

E -1.57383493E 15,

F 1.46926150E 16,

G -1.37161945E 17, 0.,0.,0./

5 TYPE 10

10 FORMAT(' ENTER GAMMA, AL, AU - FORMAT(G)')

ACCEPT 20, GA, AL2, AU2

20 FORMAT(G)

E1 = -.11/GA

DELTA = (AU2 - AL2)/100.

H(1) = HA(1)

DO 30 K = 2, ID

30 H(K) = HA(K)*(GA**(K-1))

DO 40 J=1, ID-1

CT(J) = B(J) * H(1)

KL = ID - J + 1

DO 35 K = 2, KL

KF = K

JF = J

35 CT(J) = CT(J) + H(K)*B(K+J-1)*ANF(KF,JF)

40 CONTINUE

CT(ID) = B(ID)*H(1)

ERR = 0.

T = AL2

XP = -0.11/GA

DO 60 K1 = 1,101

```

      X(K1) = T
      X(K1+101) = T
      Y(K1) = 0.0
      IF(T.GT.0.0) Y(K1) = H(1)*T + H(2)*(1. - EXP(-T))
      Y(K1 + 101) = CT(1)
      DO 50 K2 = 2, 10
50      Y(K1 + 101) = Y(K1+101) + (T**K2-1)*CT(K2)
      ERR = ERR + (Y(K1)-Y(K1+101))**2
60      T = T + DELTA
      ERR = SORT(ERR/101.)
      CALL INITT(120)
      CALL PLOTCH(X,Y,202, N, 15, 2)
      CALL FJPE
      CALL MOVABS(0.700)
      CALL ANMODE
      IDT = 10 - 1
      TYPE 70, IDT, GA
      TYPE 72, AL, AU
      TYPE 75, ERR
      WRITE(6,77) AL2, AU2, GA, ERR, IDT
      ACCEPT 100, DDD
70      FORMAT(' N =', I3, ' GAMMA =', IPE10.3/)
72      FORMAT(' AL =', IPE10.3, ' AU =', E10.3)
75      FORMAT(' RMS ERROR =', IPE15.5/)
77      FORMAT(' AL2 =', IPE10.3, ' AU2 =', E10.3,
1      ' GAMMA =', E9.3, ' RMS ERROR =', E12.5, ' N =', I3)
      CALL ERASE
      TYPE 80
80      FORMAT(' DO YOU WANT NEW PLOT PARAMETERS? (Y OR N)')
      ACCEPT 100, DDD
100     FORMAT(A1)
      IF(DDD.EQ.'Y') GO TO 5
      RETURN
      END

```

IFUNDER.EDY 139

SUBROUTINE DEOPT
 C This version of DEOPT, when used with DEOLP, plots the derivative
 C of FITFU and the derivative of the fit.
 C Requires loading FITFU but not ANF.
 C

```

      DOUBLE PRECISION B,AL,AU
      DIMENSION C(20),X(400),Y(400),H(2),IS(2),
1      CTTH(20)
      COMMON/VCOM/AL,AU,ID,B
      COMMON/PTCH/X,Y,N,IS
      T = AL
      DELTA = (AU-AL)/101
      DO 10 J = 1, 101-1
10      CTTH(J) = B(J+1)*J
      DO 20 J = 1, 101
      X(J) = T
      X(J+101) = T
      Y(J)=0.
      IF(T.GT.0.)Y(J) = 1.
      Y(J+101)=CTTH(1)
      TT=T
      DO 30 L = 2, 101-1
      Y(J+101)=Y(J+101)+CTTH(L)*TT
30      TT = TT*T

```

```

20      T = T+DELTA
        N(1)=101
        N(2)=101
        IS(1)='L'
        IS(2)='L'
        CALL PLOT(MX,Y,202,N,IS,2)
        ACCEPT 40,DDO
40      FORMAT(A1)
        RETURN
      END

```

```

      IANF.EDY      60
      FUNCTION ANF(K,J)
C      THIS FUNCTION CALCULATES (J+K-2)/(J-1)
C
        AK = K
        AJ = J
        IF(K.LT.1) GO TO 30
        IF(J.LT.1) GO TO 30
        ANF = 1.
        IF(K.EQ.1) GO TO 20
        AL = AJ + AK - 2.
10      ANF = ANF*AL
        IF(AL.EQ.AJ) GO TO 20
        AL = AL - 1.
        GO TO 10
20      RETURN
30      TYPE 40
40      FORMAT(' ERROR IN CALL TO ANF')
        ANF = 0.
        RETURN
      END

```

```

      IFITFU1.EDY   39
      FUNCTION FITFU(T)
C      This version of FITFU is for a time step function.
C
        DOUBLE PRECISION FITFU, T
        IF(T) 10, 10, 20
10      FITFU = 0.0
        RETURN
20      FITFU = 1.0D0
        RETURN
      END

```

```

      IFITFU2.EDY   39
      FUNCTION FITFU(T)
C      This version of FITFU is for a ramp time function.
C
        DOUBLE PRECISION FITFU, T
        IF(T) 10, 10, 20
10      FITFU = 0.0
        RETURN
20      FITFU = T
        RETURN
      END

```

```

IA0IN1.EDY      140
SUBROUTINE A0IN
  IMPLICIT REAL*8(A-H,O-Z)
  C      This subroutine generates the input vector field A(r) for
  C      program AREDY for the case of a dipole at the center
  C      of the cylinder and directed along the cylinder axis.
  DIMENSION ID(2), INAME(6)
  COMMON/ACSMH/CM(31),SM(31),AOUTH(31,31),AOUR(31,31)
  1  ,AOUZ(31,31),AINTH(31,31),AINZ(31,31),AL,AR,DR,PI,MLN
  2  ,I01,I02,I03,M1,M1
  DATA INAME/'CEN.DIPOLE PARALLEL TO AXIS'/
  CALL DATE(ID)
  WRITE(I03) INAME,ID
  AL2 = AL**2
  AN = DFLOAT(N)
  DO 20 I = 1,M1
    AI = DFLOAT(I-1)
    X = 1./((DSORT(1.D0+AL2*(AI/AN-.500)**2))**3)
    DO 10 J = 1,M
      AOUR(I,J) = 0.
      AOUZ(I,J) = 0.
      AOUTH(I,J) = X
    10 CONTINUE
  20 CONTINUE
  RETURN
END

IA0IN2.EDY      145
SUBROUTINE A0IN
  IMPLICIT REAL*8(A-H,O-Z)
  C      This subroutine generates the input vector field A(r) for
  C      program AREDY for the case of a dipole at the center of the
  C      cylinder and directed perpendicularly to the cylinder axis.
  DIMENSION ID(2), INAME(6)
  COMMON/ACSMH/CM(31),SM(31),AOUTH(31,31),AOUR(31,31)
  1  ,AOUZ(31,31),AINTH(31,31),AINZ(31,31),AL,AR,DR,PI,MLN
  2  ,I01,I02,I03,M1,M1
  DATA INAME/'CENTERED DIPOLE PER. TO AXIS'/
  CALL DATE(ID)
  WRITE(I03) INAME,ID
  ALN = AL/(2.*N)
  DO 20 I = 1,M1
    ANI = ALN*(2*I - 2 - N)
    DNI = 1./((DSORT(1. + ANI**2))**3)
    DO 10 J = 1,M
      AOUR(I,J) = -ANI*SM(J)*DNI
      AOUTH(I,J) = -ANI*CM(J)*DNI
      AOUZ(I,J) = SM(J)*DNI
    10 CONTINUE
  20 CONTINUE
  RETURN
END

IPWSDUM.EDY      151
SUBROUTINE PWSCRT (INTL,A,B,M,M0DCND,BDA,BDB,C,D,N,M0DCND,BDC,
  1  BDD,ELMBDA,F,IDIMF,PERTRB,IERROR,W)
  C      This is a dummy subroutine to use in place of the

```

```

C      double precision version of PUSCRT with program ANEDY.
C      It is used when the only component of AB is the theta one, or when the
C      effect of edge charges is to be ignored (to generate data for
C      plotting the driving field with PLEQ).
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION F(IDIMF,1)
      DIMENSION BDA(1) ,BDB(1) ,BDC(1) ,BDD(1) ,
1      U(1)
      WRITE(6,99)
99      FORMAT(' THIS VERSION USES BUNNY FOR PUSCRT.')
      NI=N+1
      NI=M+1
      DO 20 K=1,NI
      DO 10 J=1,NI
      F(J,K)=0.
10      CONTINUE
20      CONTINUE
      PERTRB = 0.
      IERROR = 0
      RETURN
      END
IDPUSCR.EDY      4878
      SUBROUTINE PUSCRT (INTL,A,B,M,NBDCND,BDA,BDB,C,D,N,NBDCND,BDC,
1      BDD,ELMBDA,F,IDIMF,PERTRB,IERROR,W)
      IMPLICIT REAL*8 (A-H,O-Z)
C      THIS IS A DOUBLE PRECISION VERSION OF PUSCRT AND THE SUBROUTINES IT CALLS.
C      THE COMMENT STATEMENTS HAVE BEEN REMOVED. SEE THE
C      SINGLE PRECISION VERSION FOR THEM.
      DIMENSION F(IDIMF,1)
      DIMENSION BDA(1) ,BDB(1) ,BDC(1) ,BDD(1) ,
1      U(1)
      IERROR = 0
      IF (A .GE. B) IERROR = 1
      IF (NBDCND.LT.0 .OR. NBDCND.GT.4) IERROR = 2
      IF (C .GE. D) IERROR = 3
      IF (NCHECK(N).GT.1 .OR. N.LE.2) IERROR = 4
      IF (NBDCND.LT.0 .OR. NBDCND.GT.4) IERROR = 5
      IF (IDIMF.LT. M+1) IERROR = 7
      IF (IERROR.NE. 0) RETURN
      NPEROD = NBDCND
      NBY2 = 0
      DELTAX = (B-A)/FLOAT(M)
      TWDELX = 2./DELTAX
      DELXSQ = 1./DELTAX**2
      DELTAY = (D-C)/FLOAT(N)
      TWDELY = 2./DELTAY
      DELYSQ = 1./DELTAY**2
      NP = NBDCND+1
      NP1 = N+1
      MP = NBDCND+1
      MP1 = M+1
      NSTART = 1
      NSTOP = N
      NSKIP = 1
      GO TO ( 20, 5, 10, 15, 20),NP
5      NSTART = 2
      GO TO 20
10      NSTART = 2
15      NSTOP = NP1
      NSKIP = 2

```

```

20 NUNK = NSTOP-NSTART+1
   NSTART = 1
   MSTOP = M
   MSKIP = 1
   GO TO ( 85, 25, 30, 45, 50),MP
25 NSTART = 2
   GO TO 35
30 NSTART = 2
   MSTOP = MP1
   MSKIP = 2
35 DO 40 J=NSTART,NSTOP
   F(2,J) = F(2,J)-F(1,J)*DELXSO
40 CONTINUE
   GO TO 60
45 MSTOP = MP1
   MSKIP = 2
50 DO 55 J=NSTART,NSTOP
   F(1,J) = F(1,J)+8DA(J)*TWDELX
55 CONTINUE
60 GO TO ( 65, 75),MSKIP
65 DO 70 J=NSTART,NSTOP
   F(M,J) = F(M,J)-F(MP1,J)*DELXSO
70 CONTINUE
   GO TO 85
75 DO 80 J=NSTART,NSTOP
   F(MP1,J) = F(MP1,J)-8DB(J)*TWDELX
80 CONTINUE
85 MUNK = MSTOP-MSTART+1
   GO TO (135, 90, 90,100,100),NP
90 DO 95 I=MSTART,MSTOP
   F(1,2) = F(1,2)-F(1,1)*DELYSO
95 CONTINUE
   GO TO 110
100 DO 105 I=MSTART,MSTOP
   F(1,1) = F(1,1)+8DC(I)*TWDELY
105 CONTINUE
110 GO TO (115,125),NSKIP
115 DO 120 I=MSTART,MSTOP
   F(1,N) = F(1,N)-F(1,NP1)*DELYSO
120 CONTINUE
   GO TO 135
125 DO 130 I=MSTART,MSTOP
   F(1,NP1) = F(1,NP1)-8DD(I)*TWDELY
130 CONTINUE
135 DELYSO = DELTAY*DELTAY
   DO 145 I=MSTART,MSTOP
   DO 140 J=NSTART,NSTOP
   F(1,J) = F(1,J)*DELYSO
140 CONTINUE
145 CONTINUE
   ID1 = 6*MUNK+5*MUNK
   ID2 = ID1+MUNK
   ID3 = ID2+MUNK
   ID4 = ID3+MUNK
   S = DELYSO*DELXSO
   ST2 = 2.4S
   DO 150 I=1,MUNK
   J = ID1+I
   W(J) = S
   J = ID2+I

```

```

      W(J) = -ST2+ELMBDA*DELYSO
      J = ID3+1
      W(J) = S
150  CONTINUE
      GO TO (170,170,155,160,165),MP
155  W(ID2) = ST2
      GO TO 170
160  W(ID2) = ST2
165  W(ID3+1) = ST2
170  CONTINUE
      IF (NBDEND .NE. 4) GO TO 190
      IREV = 1
      NPEROD = 2
      NBY2 = N/2
175  DO 185 J=1,NBY2
      MSKIP = N+1-J
      DO 180 I=MSTART,MSTOP
      A1 = F(I,J)
      F(I,J) = F(I,MSKIP)
      F(I,MSKIP) = A1
180  CONTINUE
185  CONTINUE
      GO TO (235,250),IREV
190  CONTINUE
      PERTRB = 0.
      IF (ELMBDA) 235,200,195
195  IERROR = 6
      GO TO 235
200  IF ((NBDEND.EQ.0 .OR. NBDEND.EQ.3) .AND.
1    (NBDEND.EQ.0 .OR. NBDEND.EQ.3)) GO TO 205
      GO TO 235
205  A1 = 1.
      A2 = 1.
      IF (NBDEND .EQ. 3) A2 = 2.
      IF (NBDEND .EQ. 3) A1 = 2.
      S1 = 0.
      MSP1 = MSTART+1
      MSTM1 = MSTOP-1
      NSP1 = NSTART+1
      NSTM1 = NSTOP-1
      DO 215 J=NSP1,NSTM1
      S = 0.
      DO 210 I=MSP1,MSTM1
      S = S+F(I,J)
210  CONTINUE
      S1 = S1+S*A1+F(MSTART,J)+F(MSTOP,J)
215  CONTINUE
      S1 = A2*S1
      S = 0.
      DO 220 I=MSP1,MSTM1
      S = S+F(I,NSTART)+F(I,NSTOP)
220  CONTINUE
      S1 = S1+S*A1+F(MSTART,NSTART)+F(MSTART,NSTOP)+F(MSTOP,NSTART)+
1    F(MSTOP,NSTOP)
      S = (2.+FLOAT(MUNK-2)*A2)*(2.+FLOAT(MUNK-2)*A1)
      PERTRB = S1/S
      DO 230 J=NSTART,NSTOP
      DO 225 I=MSTART,MSTOP
      F(I,J) = F(I,J)-PERTRB
225  CONTINUE

```



```

230 CONTINUE
  PERTB = PERTB/DELYSQ
235 CALL POIS (INTL,NPEROD,NUNK,NBDCND,MUNK,U(ID1+1),U(ID2+1),
1      U(ID3+1),IDIMF,F(MSTART,MSTOP),IERR1,U)
  IF (NBY2.EQ. 0) GO TO 240
  IREV = 2
  GO TO 175
240 CONTINUE
  IF (NBDCND.NE. 0) GO TO 250
  DO 245 I=MSTART,MSTOP
    F(I,NP1) = F(I,1)
245 CONTINUE
250 IF (NBDCND.NE. 0) GO TO 260
  DO 255 J=MSTART,MSTOP
    F(MP1,J) = F(1,J)
255 CONTINUE
  IF (NBDCND.EQ. 0) F(MP1,NP1) = F(1,NP1)
260 CONTINUE
  RETURN
  END
  SUBROUTINE POIS (IFLG,NPEROD,N,M,NPEROD,M,A,B,C,IDIMY,Y,IERROR,U)
    IMPLICIT REAL*8 (A-H,O-Z)
    EXTERNAL      TRID
    DIMENSION     Y(IDIMY,1)
    DIMENSION     U(1)      ,B(1)      ,A(1)      ,C(1)
    IERROR = 0
    IF (M.LE. 2) IERROR = 1
    I = NCHECK(NPEROD-2*((2*NPEROD)/3)+N)
    IF (N.LE.2 .OR. I.GT.1) IERROR = 2
    IF (IDIMY.LT. M) IERROR = 3
    IF (NPEROD.LT.0 .OR. NPEROD.GT.3) IERROR = 4
    IF (IERROR.NE. 0) GO TO 25
    IWDIM1 = 6*N+1
    IWDIM2 = IWDIM1+M
    IWDIM3 = IWDIM2+M
    IWDIM4 = IWDIM3+M
    IWDIM5 = IWDIM4+M
    DO 5 I=1,M
      A(I) = -A(I)
      C(I) = -C(I)
      K = IWDIM5+I-1
      U(K) = -B(I)+2.
5 CONTINUE
    IF (NPEROD.EQ. 0) GO TO 10
    CALL POISGN (NPEROD,N,IFLG,M,A,U(IWDIM5),C,IDIMY,Y,U(1),
1      U(IWDIM1),U(IWDIM2),U(IWDIM3),U(IWDIM4),TRID)
    GO TO 15
10 CALL POISGN (NPEROD,N,IFLG,M,A,U(IWDIM5),C,IDIMY,Y,U(1),
1      U(IWDIM1),U(IWDIM2),U(IWDIM3),U(IWDIM4),TRIDP)
15 DO 20 I=1,M
  A(I) = -A(I)
  C(I) = -C(I)
20 CONTINUE
25 RETURN
  END
  SUBROUTINE POISGN (NPEROD,N,IFLG,M,BA,BB,BC,IDIMD,Q,TWOCOS,B,T,D,
1      U,TRI)
    IMPLICIT REAL*8 (A-H,O-Z)
    DIMENSION     Q(IDIMD,1)
    DIMENSION     BA(1)      ,BB(1)      ,BC(1)      ,TWOCOS(1) ,

```

```

1      B(I)      .T(I)      .D(I)      .U(I)
IF (IFLG .NE. 0) GO TO 5
CALL POINT (NPEROD,N, IEX2, IEX3, IEX5, N2PW, N2P3P, N2P3P5, KS3, KS5,
1      NPSTOP, TWOCOS)
5  MM1 = M-1
DO 20 J=1, N
DO 10 I=1, M
B(I) = -Q(I, J)
10  CONTINUE
CALL TRI (1, 1, 1, M, MM1, BA, BB, BC, B, TWOCOS, D, W)
DO 15 I=1, M
Q(I, J) = B(I)
15  CONTINUE
20  CONTINUE
NP = 1
IF (IEX2 .EQ. 0) GO TO 60
L = IEX2
IF (IEX3+IEX5 .NE. 0) GO TO 25
IF (NPEROD .EQ. 1) L = L-1
IF (L .EQ. 0) GO TO 190
25  DO 55 KOUNT=1, L
K = NP
NP = 2*NP
K2 = NP-1
K3 = NP
K4 = 2*NP-1
JSTART = NP
IF (NPEROD .EQ. 3) JSTART = 1
DO 50 J=JSTART, N, NP
JM1 = J-K
JP1 = J+K
IF (J .EQ. 1) JM1 = JP1
IF (J .NE. N) GO TO 30
JP1 = JM1
IF (NPEROD .EQ. 0) JP1 = K
30  DO 35 I=1, M
B(I) = Q(I, JM1)+Q(I, JP1)
35  CONTINUE
CALL TRI (K, K2, 1, M, MM1, BA, BB, BC, B, TWOCOS, D, W)
DO 40 I=1, M
T(I) = Q(I, J)+B(I)
B(I) = T(I)
40  CONTINUE
CALL TRI (K3, K4, 1, M, MM1, BA, BB, BC, B, TWOCOS, D, W)
DO 45 I=1, M
Q(I, J) = T(I)+2.*B(I)
45  CONTINUE
50  CONTINUE
55  CONTINUE
60  IF (IEX3 .EQ. 0) GO TO 120
L = IEX3
IF (IEX5 .NE. 0) GO TO 65
IF (NPEROD .EQ. 1) L = L-1
IF (L .EQ. 0) GO TO 190
65  K2 = NP-1
DO 115 KOUNT=1, L
K = NP
NP = 3*NP
K1 = K2+1
K2 = K2+K

```

```

K3 = K2+1
K4 = K2+NP
JSTART = NP
IF (NPEROD .EQ. 3) JSTART = 1
DO 110 J=JSTART,N,NP
  IF (J .NE. 1) GO TO 70
  JM1 = J+K
  JM2 = JM1+K
  GO TO 80
70  JM1 = J-K
  JM2 = JM1-K
  IF (J .NE. N) GO TO 80
  IF (NPEROD .EQ. 0) GO TO 75
  JP1 = JM1
  JP2 = JM2
  GO TO 85
75  JP1 = K
  JP2 = JP1+K
  GO TO 85
80  JP1 = J+K
  JP2 = JP1+K
85  DO 90 I=1,M
      B(I) = 2.*Q(I,J)+Q(I,JM2)+Q(I,JP2)
90  CONTINUE
  CALL TRI (K1,K2,1,M,MM1,BA,BB,BC,B,TUOCOS,D,W)
  DO 95 I=1,M
      T(I) = B(I)+Q(I,JM1)+Q(I,JP1)
      B(I) = T(I)
95  CONTINUE
  CALL TRI (K1,K2,1,M,MM1,BA,BB,BC,B,TUOCOS,D,W)
  DO 100 I=1,M
      Q(I,J) = Q(I,J)+B(I)
      B(I) = T(I)
100 CONTINUE
  CALL TRI (K3,K4,1,M,MM1,BA,BB,BC,B,TUOCOS,D,W)
  DO 105 I=1,M
      Q(I,J) = Q(I,J)+3.*B(I)
105 CONTINUE
110 CONTINUE
115 CONTINUE
120 L = IEX5
  IF (NPEROD .EQ. 1) L = L-1
  IF (L .LE. 0) GO TO 190
  K2 = (N2PW+N2P3P)/2-1
  DO 125 KOUNT=1,L
      K = NP
      NP = 5*NP
      K1 = K2+1
      K2 = K2+K
      K3 = K2+1
      K4 = K2+NP
      JSTART = NP
      IF (NPEROD .EQ. 3) JSTART = 1
      DO 130 J=JSTART,N,NP
          IF (J .NE. 1) GO TO 125
          JM1 = J+K
          JM2 = JM1+K
          JM3 = JM2+K
          JM4 = JM3+K
          GO TO 135

```

```

125      JM1 = J-K
        JM2 = JM1-K
        JM3 = JM2-K
        JM4 = JM3-K
        IF (J .NE. N) GO TO 135
        IF (NPEROD .EQ. 0) GO TO 130
        JP1 = JM1
        JP2 = JM2
        JP3 = JM3
        JP4 = JM4
        GO TO 140
138      JP1 = K
        JP2 = JP1+K
        JP3 = JP2+K
        JP4 = JP3+K
        GO TO 140
135      JP1 = J+K
        JP2 = JP1+K
        JP3 = JP2+K
        JP4 = JP3+K
140      DO 145 I=1,M
        B(I) = 6.*Q(I,J)+4.*(Q(I,JM2)+Q(I,JP2))+Q(I,JM4)+Q(I,JP4)
145      CONTINUE
        CALL TRI (K1,K2,1,M,MM1,BA,BB,BC,B,TWOCOS,D,W)
        DO 150 I=1,M
        B(I) = B(I)+3.*(Q(I,JM1)+Q(I,JP1))+Q(I,JM3)+Q(I,JP3)
150      CONTINUE
        CALL TRI (K1,K2,1,M,MM1,BA,BB,BC,B,TWOCOS,D,W)
        DO 155 I=1,M
        T(I) = B(I)
        B(I) = 2.*Q(I,J)+Q(I,JM2)+Q(I,JP2)+B(I)
155      CONTINUE
        CALL TRI (K1,K2,1,M,MM1,BA,BB,BC,B,TWOCOS,D,W)
        DO 160 I=1,M
        B(I) = B(I)+Q(I,JM1)+Q(I,JP1)
160      CONTINUE
        CALL TRI (K1,K2,1,M,MM1,BA,BB,BC,B,TWOCOS,D,W)
        DO 165 I=1,M
        TEMP = B(I)+Q(I,J)
        B(I) = 4.*Q(I,J)+3.*(Q(I,JM2)+Q(I,JP2))+Q(I,JM4)+
1      Q(I,JP4)-T(I)
        Q(I,J) = TEMP
165      CONTINUE
        CALL TRI (K1,K2,1,M,MM1,BA,BB,BC,B,TWOCOS,D,W)
        DO 170 I=1,M
        B(I) = B(I)+2.*(Q(I,JM1)+Q(I,JP1))+Q(I,JM3)+Q(I,JP3)
170      CONTINUE
        CALL TRI (K3,K4,1,M,MM1,BA,BB,BC,B,TWOCOS,D,W)
        DO 175 I=1,M
        Q(I,J) = Q(I,J)+5.*B(I)
175      CONTINUE
180      CONTINUE
185      CONTINUE
190      IF (NPEROD.EQ.1 .OR. NPEROD.EQ.2) GO TO 235
        IF (NPEROD .EQ. 0) GO TO 205
        DO 195 I=1,M
        B(I) = 2.*Q(I,1)
195      CONTINUE
        CALL TRI (K3,K4,1,M,MM1,BA,BB,BC,B,TWOCOS,D,W)
        DO 200 I=1,M

```

```

      Q(I,N) = Q(I,N)+B(I)
      B(I) = 4.*Q(I,N)
200 CONTINUE
      CALL TRI (K4+1,K4+2*NP-1,2,M,MM1,BA,BB,BC,B,TUOCOS,D,W)
      GO TO 215
205 DO 210 I=1,M
      B(I) = 2.*Q(I,N)
210 CONTINUE
      CALL TRI (K4+1,K4+NP-1,2,M,MM1,BA,BB,BC,B,TUOCOS,D,W)
215 DO 220 I=1,M
      Q(I,N) = Q(I,N)+B(I)
220 CONTINUE
      IF (NPEROD .EQ. 0) GO TO 235
      DO 225 I=1,M
      B(I) = 2.*Q(I,N)
225 CONTINUE
      CALL TRI (K3,K4,1,M,MM1,BA,C3,BC,B,TUOCOS,D,W)
      DO 230 I=1,M
      Q(I,1) = Q(I,1)+B(I)
230 CONTINUE
235 CONTINUE
      IF (IEXS .EQ. 0) GO TO 355
      NP = N2P3P5
      KB = KS5
      IF (NPEROD .EQ. 1) K3 = K3+NP/5
      DO 350 KOUNT=1,IEXS
      K = NP
      NP = NP/5
      K4 = K3-1
      K3 = K3-NP
      K1 = KB+1
      K2 = KB+2*NP
      K5 = K2+1
      K6 = K2+4*NP
      K7 = K6+1
      K8 = K6+2*NP
      JSTART = NP
      IF (NPEROD .EQ. 3) JSTART = 1+NP
      DO 345 J=JSTART,N,K
      JH1 = J-NP
      JP1 = J+NP
      JP2 = JP1+NP
      JP3 = JP2+NP
      JP4 = JP3+NP
      IF (JH1 .NE. 0) GO TO 255
      IF (NPEROD .EQ. 0) GO TO 245
      DO 240 I=1,M
      B(I) = Q(I,JP1)
240 CONTINUE
      GO TO 265
245 DO 250 I=1,M
      B(I) = Q(I,JP1)+Q(I,N)
250 CONTINUE
      GO TO 265
255 DO 260 I=1,M
      B(I) = Q(I,JP1)+Q(I,JH1)
260 CONTINUE
265 CALL TRI (K3,K4,1,M,MM1,BA,BB,BC,B,TUOCOS,D,W)
      DO 270 I=1,M
      Q(I,J) = Q(I,J)+B(I)

```

```

      B(I) = Q(I,JP1)+Q(I,JP3)
270  CONTINUE
      CALL TRI (K3,K4,1,M,MM1,BA,BB,BC,B,TUOCOS,D,W)
      DO 275 I=1,M
        Q(I,JP2) = Q(I,JP2)+B(I)
275  CONTINUE
      IF (JP4 .GT. N) GO TO 285
      DO 280 I=1,M
        B(I) = 2.*Q(I,JP2)+Q(I,J)+Q(I,JP4)
280  CONTINUE
      GO TO 295
285  DO 290 I=1,M
        B(I) = 2.*Q(I,JP2)+Q(I,J)
290  CONTINUE
295  CALL TRI (K1,K2,1,M,MM1,BA,BB,BC,B,TUOCOS,D,W)
      DO 300 I=1,M
        Q(I,JP2) = Q(I,JP2)+B(I)
        B(I) = Q(I,JP2)+Q(I,J)
300  CONTINUE
      CALL TRI (K5,K6,1,M,MM1,BA,BB,BC,B,TUOCOS,D,W)
      DO 305 I=1,M
        Q(I,JP2) = Q(I,JP2)+B(I)
        B(I) = Q(I,J)+Q(I,JP2)
305  CONTINUE
      CALL TRI (K7,K8,1,M,MM1,BA,BB,BC,B,TUOCOS,D,W)
      DO 310 I=1,M
        Q(I,J) = Q(I,J)+B(I)
        B(I) = Q(I,J)+Q(I,JP2)
310  CONTINUE
      CALL TRI (K3,K4,1,M,MM1,BA,BB,BC,B,TUOCOS,D,W)
      DO 315 I=1,M
        Q(I,JP1) = Q(I,JP1)+B(I)
315  CONTINUE
      IF (JP4 .GT. N) GO TO 325
      DO 320 I=1,M
        B(I) = Q(I,JP2)+Q(I,JP4)
320  CONTINUE
      GO TO 335
325  DO 330 I=1,M
        B(I) = Q(I,JP2)
330  CONTINUE
335  CALL TRI (K3,K4,1,M,MM1,BA,BB,BC,B,TUOCOS,D,W)
      DO 340 I=1,M
        Q(I,JP3) = Q(I,JP3)+B(I)
340  CONTINUE
345  CONTINUE
350  CONTINUE
355  IF (IEX3 .EQ. 0) GO TO 455
      NP = N2P3P
      K2 = K53
      IF (NPEROD.EQ.1 .AND. IEX5.EQ.0) K3 = K3+NP/3
      DO 450 KOUNT=1,IEX3
        K = NP
        NP = NP/3
        K4 = K3-1
        K3 = K3-NP
        K1 = K2+1
        K2 = K2+2*NP
        JSTART = NP
        IF (NPEROD .EQ. 3) JSTART = NP+1

```

```

DO 445 J=JSTART,N,K
  JM1 = J-1
  JP1 = J+1
  JP2 = JP1+1
  IF (JM1 .EQ. 0) GO TO 365
  DO 360 I=1,M
    B(I) = Q(I,JP1)+Q(I,JM1)
360  CONTINUE
  GO TO 385
365  IF (HPR0D .EQ. 0) GO TO 375
  DO 370 I=1,M
    B(I) = Q(I,JP1)
370  CONTINUE
  GO TO 385
375  DO 380 I=1,M
    B(I) = Q(I,JP1)+Q(I,N)
380  CONTINUE
385  CALL TRI (K3,K4,1,M,MM1,BA,BB,BC,B,TWOCOS,D,W)
  DO 390 I=1,M
    Q(I,J) = Q(I,J)+B(I)
390  CONTINUE
  IF (JP2 .GT. N) GO TO 400
  DO 395 I=1,M
    Q(I) = Q(I,J)+Q(I,JP2)
395  CONTINUE
  GO TO 410
400  DO 405 I=1,M
    B(I) = Q(I,J)
405  CONTINUE
410  CALL TRI (K1,K2,1,M,MM1,BA,BB,BC,B,TWOCOS,D,W)
  DO 415 I=1,M
    Q(I,J) = Q(I,J)+B(I)
415  CONTINUE
  IF (JP2 .GT. N) GO TO 425
  DO 420 I=1,M
    B(I) = Q(I,J)+Q(I,JP2)
420  CONTINUE
  GO TO 435
425  DO 430 I=1,M
    B(I) = Q(I,J)
430  CONTINUE
435  CALL TRI (K3,K4,1,M,MM1,BA,BB,BC,B,TWOCOS,D,W)
  DO 440 I=1,M
    Q(I,JP1) = Q(I,JP1)+B(I)
440  CONTINUE
445  CONTINUE
450  CONTINUE
455  IF (IEX2 .EQ. 0) GO TO 510
  NP = N2PW
  DO 505 KOUNT=1,IEX2
    K = NP
    IF = I/2
    CG = C-1
    JSTART = NP
    IF (HPR0D .EQ. 3) JSTART = 1+NP
  DO 500 J=JSTART,N,K
    JM1 = J-1
    JP1 = J+1
    IF (JM1 .NE. 0) GO TO 470
    IF (JP1 .GT. N) GO TO 500

```

```

      IF (NPEROD.EQ.1 .OR. NPEROD.EQ.2) GO TO 460
      JM1 = N
      GO TO 480
460    DO 465 I=1,M
          B(I) = 0(I,JP1)
465    CONTINUE
      GO TO 490
470    IF (JP1 .LE. N) GO TO 480
      DO 475 I=1,M
          B(I) = 0(I,JM1)
475    CONTINUE
      GO TO 490
480    DO 485 I=1,M
          B(I) = 0(I,JM1)+0(I,JP1)
485    CONTINUE
490    CALL TRI (NP,K3,1,M,M1,BA,BB,BC,B,TWOCOS,D,W)
      DO 495 I=1,M
          Q(I,J) = 0(I,J)+B(I)
495    CONTINUE
500    CONTINUE
505    CONTINUE
510    RETURN
      END
      SUBROUTINE POINTIT (NPEROD,N, IEX2, IEX3, IEX5, N2P4, N2P3P, N2P3P5, KS3,
1      KSS, NPSTOP, TWOCOS)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION TWOCOS(1)
      PI = 3.14159265358979323800
      NP = N
      IF (NPEROD .EQ. 1) NP = NP+1
      IF (NPEROD .EQ. 3) NP = NP-1
      IEX2 = 0
5    K = NP/2
      IF (2*K .NE. NP) GO TO 10
      IEX2 = IEX2+1
      NP = K
      GO TO 5
10    IEX3 = 0
15    K = NP/3
      IF (3*K .NE. NP) GO TO 20
      IEX3 = IEX3+1
      NP = K
      GO TO 15
20    IEX5 = 0
25    K = NP/5
      IF (5*K .NE. NP) GO TO 30
      IEX5 = IEX5+1
      NP = K
      GO TO 25
30    CONTINUE
      N2P4 = 2**IEX2
      N2P3P = N2P4*(3**IEX3)
      N2P3P5 = N2P3P*(5**IEX5)
      NP = 1
      TWOCOS(1) = 0.
      K = 1
      IF (IEX2 .EQ. 0) GO TO 50
      L = IEX2
      IF (IEX3+IEX5 .NE. 0) GO TO 35
      IF (NPEROD .EQ. 1) L = L-1

```



```

IF (L .EQ. 0) GO TO 145
35 DO 45 KOUNT=1,L
    NP = 2*NP
    DO 40 I=1,NP
        J = K+I
        TWDCOS(J) = 2.*DCOS((FLOAT(I)-.500)*PI/FLOAT(NP))
40 CONTINUE
    K = K+NP
45 CONTINUE
50 IF (IEX3 .EQ. 0) GO TO 70
    L = IEX3
    IF (IEX5 .NE. 0) GO TO 55
    IF (NPEROD .EQ. 1) L = L-1
    IF (L .EQ. 0) GO TO 85
55 DO 65 KOUNT=1,L
    NP = 3*NP
    DO 60 I=1,NP
        J = K+I
        TWDCOS(J) = 2.*DCOS((FLOAT(I)-.500)*PI/FLOAT(NP))
60 CONTINUE
    K = K+NP
65 CONTINUE
70 L = IEX5
    IF (NPEROD .EQ. 1) L = L-1
    IF (L .LE. 0) GO TO 85
    DO 80 KOUNT=1,L
    NP = 5*NP
    DO 75 I=1,NP
        J = K+I
        TWDCOS(J) = 2.*DCOS((FLOAT(I)-.500)*PI/FLOAT(NP))
75 CONTINUE
    K = K+NP
80 CONTINUE
85 IF (NPEROD.EQ.1 .OR. NPEROD.EQ.2) GO TO 105
    IF (NPEROD .EQ. 0) GO TO 95
    NPT2 = 2*NP
    DO 90 I=1,NPT2
        J = K+I
        TWDCOS(J) = 2.*DCOS(FLOAT(I)*PI/FLOAT(NP))
90 CONTINUE
    K = K+NPT2
    GO TO 105
95 DO 100 I=1,NP
    J = K+I
    TWDCOS(J) = 2.*DCOS(2.*FLOAT(I)*PI/FLOAT(NP))
100 CONTINUE
    K = K+NP
105 NP = N2P3PS
    IF (IEX5 .EQ. 0) GO TO 130
    K55 = K
    DO 125 KOUNT=1,IEX5
    NP = NP/5
    NPT2 = 2*NP
    DO 110 I=1,NPT2
        J = K+I
        TWDCOS(J) = 2.*DCOS((FLOAT(I)-.500)*PI/FLOAT(NPT2))
110 CONTINUE
    K = K+NPT2
    DO 115 I=1,NP
        J = K+4*I

```

```

      TWOCOS(J-3) = 2.*DCOS((FLOAT(I)-.8DB)*PI/FLOAT(NP))
      TWOCOS(J-2) = 2.*DCOS((FLOAT(I)-.6DB)*PI/FLOAT(NP))
      TWOCOS(J-1) = 2.*DCOS((FLOAT(I)-.4DB)*PI/FLOAT(NP))
      TWOCOS(J) = 2.*DCOS((FLOAT(I)-.2DB)*PI/FLOAT(NP))
115  CONTINUE
      K = K+4*NP
      DO 120 I=1, NP
        J = K+2*I
        TWOCOS(J-1) = 2.*DCOS(FLOAT(3*I-2)*PI/FLOAT(3*NP))
        TWOCOS(J) = 2.*DCOS(FLOAT(3*I-1)*PI/FLOAT(3*NP))
120  CONTINUE
      K = K+2*NP
125  CONTINUE
130  IF (IEX3 .EQ. 0) GO TO 145
      KS3 = K
      DO 140 KOUNT=1, IEX3
        NP = NP/3
        DO 135 I=1, NP
          J = K+2*I
          TWOCOS(J-1) = 2.*DCOS(FLOAT(3*I-2)*PI/FLOAT(3*NP))
          TWOCOS(J) = 2.*DCOS(FLOAT(3*I-1)*PI/FLOAT(3*NP))
135  CONTINUE
        K = K+2*NP
140  CONTINUE
145  RETURN
      END
      SUBROUTINE TRID (KSTART, KSTOP, ISING, M, MM1, A, B, C, Y, TWOCOS, D, W)
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION A(1), B(1), C(1), Y(1),
1      TWOCOS(1), D(1), W(1)
      DO 15 K=KSTART, KSTOP
        X = -TWOCOS(K)
        D(1) = C(1)/(B(1)+X)
        Y(1) = Y(1)/(B(1)+X)
        DO 5 I=2, M
          Z = B(I)+X-A(I)*D(I-1)
          D(I) = C(I)/Z
          Y(I) = (Y(I)-A(I)*Y(I-1))/Z
5      CONTINUE
        DO 10 IP=1, MM1
          I = M-IP
          Y(I) = Y(I)-D(I)*Y(I+1)
10     CONTINUE
15     CONTINUE
      IF (ISING .EQ. 1) RETURN
      D(1) = C(1)/(B(1)-2.)
      Y(1) = Y(1)/(B(1)-2.)
      DO 20 I=2, MM1
        Z = B(I)-2.-A(I)*D(I-1)
        D(I) = C(I)/Z
        Y(I) = (Y(I)-A(I)*Y(I-1))/Z
20     CONTINUE
      Z = B(M)-2.-A(M)*D(M-1)
      X = Y(M)-A(M)*Y(M-1)
      IF (Z .NE. 0.) GO TO 25
      Y(M) = 0.
      GO TO 30
25  Y(M) = X/Z
30  DO 35 IP=1, MM1
      I = M-IP

```

```

      Y(I) = Y(I)-D(I)*Y(I+1)
35 CONTINUE
      RETURN
      END
      SUBROUTINE TRIDP (KSTART,KSTOP,ISING,M,MM1,A,B,C,Y,TWOCOS,D,W)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION      A(1)      ,B(1)      ,C(1)      ,Y(1)      ,
1      TWOCOS(1) ,D(1)      ,W(1)
      DO 15 K=KSTART,KSTOP
      X = -TWOCOS(K)
      D(1) = C(1)/(B(1)+X)
      W(1) = A(1)/(B(1)+X)
      Y(1) = Y(1)/(B(1)+X)
      BM = B(M)
      Z = C(M)
      DO 5 I=2,MM1
      DEN = B(I)+X-A(I)*D(I-1)
      D(I) = C(I)/DEN
      W(I) = -A(I)*W(I-1)/DEN
      Y(I) = (Y(I)-A(I)*Y(I-1))/DEN
      Y(M) = Y(M)-Z*Y(I-1)
      BM = BM-Z*W(I-1)
      Z = -Z*D(I-1)
5      CONTINUE
      D(MM1) = D(MM1)+W(MM1)
      Z = A(M)+Z
      DEN = BM+X-Z*D(MM1)
      Y(M) = Y(M)-Z*Y(M-1)
      Y(M) = Y(M)/DEN
      Y(MM1) = Y(MM1)-D(MM1)*Y(M)
      DO 10 IP=2,MM1
      I = M-IP
      Y(I) = Y(I)-D(I)*Y(I+1)-W(I)*Y(M)
10      CONTINUE
15      CONTINUE
      IF (ISING.EQ. 1) RETURN
      D(1) = C(1)/(B(1)-2.)
      W(1) = A(1)/(B(1)-2.)
      Y(1) = Y(1)/(B(1)-2.)
      BM = B(M)
      Z = C(M)
      DO 20 I=2,MM1
      DEN = B(I)-2.-A(I)*D(I-1)
      D(I) = C(I)/DEN
      W(I) = -A(I)*W(I-1)/DEN
      Y(I) = (Y(I)-A(I)*Y(I-1))/DEN
      Y(M) = Y(M)-Z*Y(I-1)
      BM = BM-Z*W(I-1)
      Z = -Z*D(I-1)
20      CONTINUE
      D(MM1) = D(MM1)+W(MM1)
      Z = A(M)+Z
      DEN = BM-2.-Z*D(MM1)
      Y(M) = Y(M)-Z*Y(M-1)
      IF (DEN.NE. 0.) GO TO 25
      Y(M) = 0.
      GO TO 30
25      Y(M) = Y(M)/DEN
30      Y(MM1) = Y(MM1)-D(MM1)*Y(M)
      DO 35 IP=2,MM1

```

```

      I = M-IP
      Y(I) = Y(I)-D(I)*Y(I+1)-U(I)*Y(M)
35  CONTINUE
      RETURN
      END
      FUNCTION NCHECK (N)
      NP = N
      5  K = NP/2
      IF (2*K .NE. NP) GO TO 10
      NP = K
      GO TO 5
      10 IF (NP .NE. 1) GO TO 15
      NCHECK = 0
      RETURN
      15 K = NP/3
      IF (3*K .NE. NP) GO TO 20
      NP = K
      GO TO 15
      20 K = NP/5
      IF (5*K .NE. NP) GO TO 25
      NP = K
      GO TO 20
      25 IF (NP .NE. 1) GO TO 30
      NCHECK = 1
      RETURN
      30 NCHECK = 2
      RETURN
      END

```

```

ITACO.EDY      99
C      THIS PROGRAM TESTS SUBROUTINES AINTG AND COSIM.
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ACSPN/CM(31),SM(31),AOUTH(31,31),AOUR(31,31)
      1  ,AOUZ(31,31),AINTH(31,31),AINZ(31,31),AL,AR,PI,M,N
      2  ,I01,I02,I03,M1,N1
      I01 = 6
      I02 = 5
      WRITE(I01,10)
      10  FORMAT(' BEGIN TEST OF AINTG & COSIM')
      READ(I02,20) N,M,AR,AL
      WRITE(I01,20) N,M,AR,AL
      20  FORMAT(2I8,2F8.1)
      CALL COSIM
      N1 = N+1
      READ(I02,30)((AINTH(J,K),AINZ(J,K),J=1,N1),K=1,M)
      30  FORMAT(10F8.1)
      CALL AINTG
      END

```

```

ITGRADZ.EDY    131
C      This program tests subroutine GRADZ.
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ACSPN/CM(30),SM(30),AOUTH(31,30),AOUR(31,30)
      1  ,AOUZ(31,30),AINTH(31,30),AINZ(31,30),AL,AR,PI,M,N,I01
      READ(5,10) N,M,AR,AL
      10  FORMAT(2I8,2F8.1)
      WRITE(6,10) N,M,AR,AL
      N1 = N+1
      M1 = M+1
      PI = 4.*DATAN(1.D0)
      DO 30 I = 1,N1

```

```

DO 20 J = 1,M
AOUTH(1,J) = 0.
AOUZ(1,J) = 0.
AOUR(1,J) = 1 + 2*J
20 CONTINUE
30 CONTINUE
DO 40 I = 1,N1
AOUR(1,M1) = AOUR(1,1)
CALL GRADZ
WRITE(6,50)
50 FORMAT(4X,'I',3X,'J',4X,'AOUR',8X,'AINZ',8X,'AINTH')
WRITE(6,60)((I,J,AOUR(1,J),AINZ(1,J),
1 AINTH(1,J), I=1,N1), J=1,M1)
60 FORMAT(1X,2I4,1P3D12.4)
END
!!

```

REFERENCES

1. Kenneth H. Carpenter, *Solution of the Diffusion Equation by a Perturbation Approach with Polynomial Approximations to the Driving Term*, to be published.
2. H. T. Yeh and K. H. Carpenter, "A Perturbation Expansion with Separated Time Dependence for Eddy Current Calculations," to appear in *Proceedings of Conference on the Computation of Magnetic Fields*, Oxford, March 31 - April 2, 1976.
3. Kenneth H. Carpenter and H. T. Yeh, "Eddy Current Calculations for Thin Cylinders of Finite Length with Driving Fields of Ramp Time Dependence," *IEEE Transactions on Magnetics*, to be published.
4. P. Swarztrauber and R. Sweet, *Efficient FORTRAN Subprograms for the Solution of Elliptic Partial Differential Equations*, NCAR-TN/1A-109, National Center for Atmospheric Research, Boulder, CO (July 1965).