BUREAU OF MINERAL RESOURCES, GEOLOGY AND GEOPHYSICS

RECORD

1981/21

BMR Microform MF163

A COLLECTION OF FORTRAN PROGRAMS FOR THE ANALYSIS OF REGIONAL MAGNETIC AND GRAVITY DATA

by

P.R. GIDLEY

****

****

A COLLECTION OF FORTRAN PROGRAMS FOR THE ANALYSIS OF REGIONAL MAGNETIC AND GRAVITY DATA.

by Peter R Gidley

23/3/1981

Record 1981/21

BMR Microform MF163

****

****

CONTENTS:

1. INTRODUCTION.

The programs listed in this Record have all been written in FORTRAN IV and have been designed to run on the Bureau of Mineral Resources HP 2000 series computer facility. The programs have in most instances been adapted from already existing programs on other machines and where this is the case, recognition has been given.

The programs relate to the processing of regional magnetic and gravity data. If a specific application for the programs is desired but not available in their present state, there is sufficient documentation included to permit modifications to be made. The programs are highly interactive and converse with the user for information on running procedure and input parameters. Some programs may take a considerable amount of time to run, in which case all conversation to the computer is made quickly and then processing begins, leaving the user free to continue other work.

Some routines permit the saving of the program's output so that further processing may be subsequently applied. For instance if a raw magnetc data file were accessed, the regional trend could be removed by program POLYD, output stored, and then a depth to basement program could be run on the saved file.

2. NOTES ON PROGRAMS

These notes outline specific points which may prove helpful for users intending to modify the programs for use on other machines:

1. Some programs use the HP 2000 facility called EMA (Expanded Memory Accessing). This has been done to permit the required source code and reasonably sized arrays to be available within the confined memory size of the HP computer. Declaration of an EMA program is by the second line of source code where $EMA(------) is declared and a COMMON/BLOK/ is used to indicate to the machine the arrays intended for storage and operation in EMA.

Although greatly expanding the capability of the computer, the EMA facility has two main disadvantages:

(a) Programs using EMA are generally slow to run and they dominate the computer while running - to the disdain of other users!

(b) Certain system software cannot be applied to EMA arrays. A number of system plot routines cannot be used in conjunction with EMA arrays, such as automatic scaling. This problem is easily overcome by placing the EMA array into a "dummy", normally stored array, and then operated on conventionally. This situation only applies to a small number of system software functions. The main mathematical and trignometric routines operate as normal.

For users intending to modify programs for other machines with larger core space the EMA declarations can be abolished and the arrays placed in normally dimensioned storage.

2. The HP 2000 series uses an 8-bit word size and so, is not a highly precise device. Consequently, where a program has been adapted from, for example a CDC 7600, double precision has been used. The use of double or even triple precision has been only applied where it has been found that significant differences between the original machine and the HP 2000 output have existed. This situation is especially noticeable where matrix calculations and solving of matrices, result in solution errors. Such precision should not be necessary on a 16- or 32-bit word machine.

3. The HP 2000 series facility was established as an interactive system. Consequently some software functions have been incorporated which permit user interaction between the computer and the user's terminal. The main software routine used is RMPAR which permits computer-terminal conversational interfacing. A sophisticated FILE MANAGER system is also available which permits on-line accessing, storing and processing of disc files while a program is operating. This facility is extensively used in the majority of the described programs.

4. Most interactive questions require a response of "YES", "NO", or a number as otherwise specified.

3. FILE STRUCTURE:

A general purpose but simple file structure was employed which provides the programs with input data. A number of simple questions are asked by the computer to access and read the required file. A single file may contain a number, or series of blocks of data, each easily identifiable and accessed. Each block refers to a geophysical profile of data sequentially recorded, which is preceded immediately by a header record. The header is a single line of information containing a project number (always 18) , date or flight number, line or segment number and the number of values contained in the file of data corresponding to that line or segment.

An example of a header and data record are given below:

col 1-10	col 11-20	col 21-30	col 31-40
PROJECT	DATE	LINE	No. of VALUES

e.g.

18	140279	15	645

57048.00 57049.01 57052.05 57054.22 57056.41

The sequence of questions asked in a program to access the required file is.

WHAT IS THE FILE NAME? - Input as required.

WHAT IS THE SECURITY CODE AND CARTRIDGE NUMBER? - Input as required. Note the cartridge number is normally 9.

WHAT DATA SEGMENT (LINE No.) IS REQUIRED? - This statment requests the line number be inserted. A six character format has been reserved for this number. Consequently the computer must have six characters entered. If, for instance the line number were 15, then the correct method of entering the line number is:
BLANK BLANK BLANK BLANK 15 - note tha the number is right justified because the computer will oterwise interpret the blanks as zero. This style of input was used so alphanumeric characters can be used for line designation.

WHAT IS FORMAT OF DATA? - The appropriate format (enclosed in brackets) is input which describes the mode under which the data records are written on the accessed file. Consequently, for the example above, the correct format input would be (5(F8.2,2X)). Care must be observed where a record has greater than 72 characters per line. A suitable format may have to be found e.g. (10F8.1) = (8(F6.1,2X),2(F6.1,2X)). Where a file is accessed that is the saved result of a previous program the format is (8F10.2).

HOW MANY VALUES PER DATA RECORD? - Input the number of data values per record or line of the accessed data file e.g. 5 in the example above.

NOTES:.

(a) The format-defining questions make the applications of the programs

exremely versatile since virtually any data set may be used or analysed, even if the records contain unrequired values of data.

(b) Two-dimensional array data can be saved by this file structure. For example, program TLAY (see description) requires a two-dimensional array for input and this is simply undertaken by file construction of ROW1,ROW2,ROW3..... where the number of rows equals the number of values in a column.

****

4. PROGRAM DESCRIPTIONS

TN4,B,L,Q,C

```
**************************************
*                                    *
*                                    *
*                                    *
*          PROGRAM  POLYD            *
*                                    *
*                                    *
*                                    *
**************************************
```

EMA(BLOK1,2)

PROGRAM POLYD(3,200)

****

TITLE : POLYD - POLYNOMIAL CURVE FITTING TO ANY ONE-DIMENSIONAL PROFILED DATA.

AUTHOR : PETER R GIDLEY. Bureau of Mineral Resources, CANBERRA,ACT.

REF. : DAVIS, J.C. -1973- STATISTICS AND DATA ANALYSIS IN GEOLOGY. WILEY INTERNATIONAL, NEW YORK.

PURPOSE : THIS PROGRAM DRAWS DATA FROM A DISC FILE AND THEN OFFERS:

1. PLOT OF ORIGINAL DATA
2. CURVILINEAR POLYNOMIAL REGRESSION FIT TO THE DATA TO THE nTH DEGREE.
3. PLOT OF THE REGRESSION LINE.
4. OUTPUT OF VARIOUS STATISTICAL FITTING PARAMETERS.
5. THE REGRESSED DATA CAN BE SAVED IN AN AUTOMATICALLY HEADED DISC FILE.

NOTES:

(a) THE PROGRAM CAN BE APPLIED TO ANY nTH DEGREE POLYNOMIAL HOWEVER IT WILL BE FOUND TO BE IMPRACTIBLE FOR POLYNOMIAL FITTING ABOVE DEGREE 4 DUE TO MACHINE PRECISION.

(b) AS THIS PROGRAM MAYBE THE FIRST APPLIED TO RAW AEROMAGNETIC DATA, AN OPTION IS INCLUDED TO PERMIT THE REVERSAL OF DATA ORDER SO THAT IT MAY BE DEALT WITH IN A

```
          WEST-EAST ORIENTATION.

          WRITTEN BY PETER R. GIDLEY       3/1/1981

*****************************************************

      COMMON/BLOK1/ A(2500,2),B(125,125),D(2500,4),E(2500)
     1,FILE(2500),A1(125,125),B1(125),C(125)
      DIMENSION IDCB(144),IBUF(40),NAME(3),ISEG(3),ISGNO(3)
      DIMENSION LU(5),IPS(3),IB(2),IFORM(40)

      CALL RMPAR(LU)

      READ IN FILE DATA

   50 NAME(2)=2H
      NAME(3)=2H
      DO 100 I=1,2500
      FILE(I)=0.0
  100 CONTINUE
      WRITE(LU,299)
  299 FORMAT(/,"ENTER FILE NAME")
      READ(LU,300) (NAME(I),I=1,3)
  300 FORMAT(3A2)
      WRITE(LU,400)
  400 FORMAT(/,"ENTER SECURITY CODE & CARTRIDGE NUMBER")
      READ(LU,*)ISC,ICR
      IF(ISC.LT.0) STOP
      IF(ICR.LE.0) ICR=9
      CALL OPEN(IDCB,IERR,NAME,0,ISC,ICR)
      IF(IERR.GE.0) GO TO 600

   AN ERROR MESSAGE WILL BE GENERATED IF DIFFICULTY IS ENCOUNTERED IN FILE
   OPENING OR READING

      WRITE(LU,500) IERR
  500 FORMAT(/,"ERROR ",I3," ON OPEN")
      STOP
  600 WRITE(LU,700)
  700 FORMAT(/,"WHAT DATA SEGMENT IS REQUIRED-RIGHT JUSTIFY")
      READ(LU,300) ISGNO
  800 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR.LT.0) GO TO 1500
      CALL CODE
      READ(IBUF,900) IFLAG,IFLGT,ISEG
  900 FORMAT(2I10,4X,3A2)
      DO 1000 I=1,3
      IPS(I)=ISGNO(I)
      IF(ISGNO(I).NE.ISEG(I)) GO TO 800
 1000 CONTINUE
      WRITE(LU,1100)
 1100 FORMAT(/,"WHAT IS FORMAT (F-TYPE) OF DATA--(E.G.(1X,2F6.0))?")
      READ(LU,301) IFORM
  301 FORMAT(40A2)
      WRITE(LU,1155)
 1155 FORMAT(/,"HOW MANY DATA VALUES PER RECORD?")
      READ(LU,*) INO
      DO 1200 I=1,4000,INO
      CALL READF(IDCB,IERR,IBUF,40,LEN)
```

```
      IF(IERR.LT.0) GO TO 1500
      IF(LEN.LT.0) GO TO 1600
      CALL CODE
      READ(IBUF,900) IFLAG
      IF(IFLAG.EQ.18) GO TO 1600
      NXN=I+INO-1
      CALL CODE
      READ(IBUF,IFORM)(FILE(J),J=I,NXN)
 1200 CONTINUE
      WRITE(LU,1300) ISEG
 1300 FORMAT(/,"*** ERROR ERROR ****, MORE THAN 4000 PTS IN ",3A2)
 1600 CALL CLOSE(IDCB,IERR)
      WRITE(LU,1400)
 1400 FORMAT(/,"DOES DATA NEED TO BE DIVIDED BY 1000-eg AIRBORNE FILE?")
      READ(LU,300) ITHOU
      IF(ITHOU.EQ.2HNO) GO TO 1450
      DO 1475 I=1,NXN
      FILE(I)=FILE(I)/1000.0
 1475 CONTINUE

   IF A FLIGHT HAS BEEN FLOWN IN THE WRONG DIRECTION FOR
   PROCESSING THEN WE MUST REVERSE THE ORDER.

      WRITE(LU,601)
  601 FORMAT(/,"DOES DATA REQUIRE REVERSING DUE TO FLIGHT DIRECTION?")
      READ(LU,300) IREV
      IF(IREV.EQ.2HNO) GO TO 1450
      DO 602 I=1,NXN
      E(NXN-I+1)=FILE(I)
  602 CONTINUE
      DO 603 I=1,NXN
      FILE(I)=E(I)
      E(I)=0.0
  603 CONTINUE

       FILE DATA NOW IN,DO YOU WANT A PLOT OR WRITE IT?"

 1450 WRITE(6,1151) NXN
      WRITE(LU,1151) NXN
      WRITE(LU,51)
   51 FORMAT(/,"DO YOU WANT TO PRINT DATA?",
     2/,"----------IF YES , TYPE IN NUMBER OF VALUES REQUIRED",
     3/,"----------IF NO TYPE IN 'NO'.")
      READ(LU,820) IB
      IF(IB.EQ.2HNO) GO TO 833
  820 FORMAT(2A2)
      CALL CODE
      READ(IB,*) NUT
      WRITE(LU,831) (FILE(I),I=1,NUT)
  831 FORMAT(8(F8.2,2X))
 1151 FORMAT(/,"THERE WERE A TOTAL OF ",I5," DATA POINTS READ.")
  833 WRITE(6,115) ISGNO
  115 FORMAT(/,"**********INPUT DATA FROM LINE ",3A2,"**********")
      IF(IPRIN.EQ.2HYE)  WRITE(6,112) (FILE(I),I=1,NXN)
  111 FORMAT(8(F8.2,2X))
  112 FORMAT(13(F8.2,2X))
      WRITE(LU,1150)
 1150 FORMAT(/,"DO YOU WANT TO PLOT ORIGINAL DATA?")
      READ(LU,300) IPLT
      WRITE(LU,255)
```

```
 255 FORMAT(/,"WHAT IS SAMPLE INTERVAL OF DATA?")
     READ(LU,*) DX
     WRITE(LU,40)
  40 FORMAT(/,"WHAT IS REFERENCE LEVEL - EG TOTAL FIELD?")
     READ(LU,*) TOT

     WRITE(LU,110)
 110 FORMAT(/,"WHAT ARE START & END POINTS ALONG PROFILE?")
     READ(LU,*) IST,IEND
     IF(IEND.GT.NXN) IEND=NXN
     IF(IST.LT.0) IST=1
     DO 120 I=IST,IEND
     DIST=(IST+I-2)*DX
     A(I,1)=DIST
     A(I,2)=FILE(I)
 120 CONTINUE
     NXN=1+IEND-IST
     IF(IPLT.EQ.2HNO) GO TO 923
     CALL UPLOT(FILE,NXN,DX,IST,IEND,TOT,LU,IPS)

.... CALL THE POLYNOMIAL FITTING SUBROUTINE

 923 CALL POLY(A,DX,NXN,LU,B,D,E,FILE,C,NUT)

     WRITE(LU,924)
 924 FORMAT(/,"DO YO WANT TO PLOT POLYNOMIAL OF FIT TO DATA?")
     READ(LU,300) IPT
     IF(IPT.EQ.2HNO) GO TO 925
     DO 926 I=1,NXN
     FILE(I)=D(I,3)
  926 CONTINUE
     CALL UPLOT(FILE,NXN,DX,IST,IEND,TOT,LU,IPS)
  925 DO 860 I=1,NXN
     E(I)=D(I,4)+TOT
  860 CONTINUE
     WRITE(LU,879)
  879 FORMAT./,"DO YOU WANT PLOT OF REGRESSION MINUS ORIGINAL?")
     READ(LU,300) IFT
     IF(IFT.EQ.2HNO) GO TO 830
     CALL UPLOT(E,NXN,DX,IST,IEND,TOT,LU,IPS)

   INSERT FILING OPTION

  830 WRITE(LU,450)
  450 FORMAT(/,"DO YOU WISH TO SAVE THE REGRESSION DATA?")
     READ(LU,300) NFIL
     IF(NFIL.EQ.2HYE) CALL UPFIL(E,NXN,ISC,ICR,LU,IFLGT,ISEG)
     WRITE(LU,950)
  950 FORMAT(/,"ANY MORE DATA FILES TO BE POLYNOMIAL FITTED?")
     READ(LU,300) MORE
     IF(MORE.EQ.2HYE) GO TO 50
     STOP
 1500 WRITE(LU,1510) IERR
 1510 FORMAT(/,"THERE HAS BEEN A READ-FROM-FILE ERROR-IERR=",I4)
     CALL CLOSE(IDCB,IERR)
     STOP
     END
     SUBROUTINE UPLOT(FILE,NXN,DX,IST,IEND,TOT,LU,IPS)

**********************************************************************
```

```
    THIS SUBROUTINE PLOTS THE ORIGINAL DATA VERSUS DISTANCE.

**************************************************************************

      EMA FILE(2500)
      DIMENSION LU(5),IPS(3)
    5 WRITE(LU,10)
   10 FORMAT(/,"VERT SCALE IN UNITS/IN,HORIZ SCALE DIST/IN")
      READ(LU,*)SCM,SCD
      SV1=TOT-SCM
      XT=DX*FLOAT(IEND-IST)/SCD
      IF(XT.GT.16.5) 101,103
  101 WRITE(LU 102)
  102 FORMAT(/,"PLOT TOO LONGGGGGGG")
      GO TO 5
  103 CALL PLOTS(XT+10.,12.,0)
      CALL FACTR(1.0)
      CALL PLOT(1.,0.5,-3)
      CALL AXIS(0.,4.,9HAMPLITUDE,9,5.,90.,SV1,SCM)
      CALL AXIS(0.,3.,8HDISTANCE,8,13.,0.,0.,SCD)
      CALL PLOT(0.,5.,-3)
      XP=0.
      DO 600 I=IST,IEND
      YP=(FILE(I)-TOT)/SCM
      CALL PLOT(XP,YP,2)
      XP=XP+DX/SCD
  600  ONTINUE
        S=IPS
       '.L SYMB(6.,3.5,0.1,12HLINE NUMBER=,0.,12)
      CALL SYMB(7.5,3.5,0.1,IPS,0.,6)
      RETURN
      END
      SUBROUTINE POLY (A,DX,N,LU,B,D,E,FILE,C,NUT)
      EMA A(2500,2),B(125,125),D(2500,4),E(2500),FILE(2500)
     1,C(125)
**************************************************************************

    SUBROUTINE POLY FITS A CURVILINEAR POLYNOMIAL
    REGRESSION OF ORDER 'IORD' TO A SET OF DATA.

      ARRAY A    CONTAINS THE VARIABLE DATA 'Y' AND
                 SPATIAL DATA 'X'.
      ARRAY B    CONTAINS TERMS OF COEFFICIENTS OF
                 THE MATRIX DEFINED BY:

                  Xn        b         Y
                   _       _ _       _ _
                 !Xo!      ! !       ! !
                 !X1!      ! !       ! !
                 !X2!      ! !       ! !
                 !  !  .   ! !   =   ! !
                 !. !      ! !       ! !
                 !. !      ! !       ! !
                 !Xn!      ! !       ! !
                   _       _ _       _ _

      ARRAY C    CONTAINS ORIGINALLY THE VECTOR OF THE
                 RHS OF THE NORMAL EQUATIONS ABOVE. AFTER
                 SOLVING THE SET OF NORMAL EQUATIONS,
```

```
                ARRAY C CONTAINS THE COEFFICIENTS OF THE
                REGRESSION EQUATION.
     ARRAY D    CONTAINS X,Y, Y-CALCULATED, AND DEVIATION
                FOR ALL POINTS.

*****************************************************************

      DIMENSION XP(125),LU(5),IK(2)
      WRITE(LU,10)
   10 FORMAT(/"WHAT DEGREE (ORDER) OF POLYNOMIAL FIT IS REQUIRED?")
      READ(LU,*) IORD
      IORD1 = IORD+1

    PRINT OUT ORIGINAL X AND Y DATA

      WRITE(LU,1985)
      WRITE(6,1985)
      CALL PRINS(A,NUT,2,2500,2,LU)

   CALCULATE SUMS FOR LEAST SQUARES SOLUTION

      DO 100 I=1,IORD1
      C(I)=0.0
      DO 101 J=1,IORD1
      B(I,J)=0.0
  101 CONTINUE
  100 CONTINUE
      DO 102 I=1,N
      XP(1)=1.0
      DO 103 J=2,IORD1
      XP(J)=XP(J-1)*A(I,1)
  103 CONTINUE
      DO 104 J=1,IORD1
      DO 105 K=1,IORD1
      B(J,K)=B(J,K)+XP(J)*XP(K)
  105 CONTINUE
      C(J)=C(J)+XP(J)*A(I,2)
  104 CONTINUE
  102 CONTINUE

.... SOLVE SIMULTANEOUS LINEAR EQUATIONS....

      WRITE(6,1001)
      WRITE(LU,1001)
      CALL PRINS (B,IORD1,IORD1,125,125,LU)
      WRITE(LU,1002)
      WRITE(6,1002)
      CALL PRINS (C,1,IORD1,1,125,LU)
      CALL SLE(B,C,IORD1,125,1.0E-06)
      WRITE(6,1003)
      WRITE(LU,1003)
      CALL PRINS (C,1,IORD1,1,125,LU)

.... CALCULATE PREDICTED VALUES....

      DO 106 I=1,N
      D(I,1)=A(I,1)
      D(I,2)=A(I,2)
      XXP=1.0
      YYP=0.0
```

```
      DO 107 J=1,IORD1
      YYP=YYP+XXP*C(J)
      XXP=XXP*D(I,1)
  107 CONTINUE
      D(I,3)=YYP
      D(I,4)=D(I,2)-YYP
  106 CONTINUE
      WRITE(6,1004)
      WRITE(LU,1004)
      CALL PRINS (D,NUT,4,2500,4,LU)

.... CALCULATE ERROR MEASURES...

      SY=0.0
      SY2=0.0
      SYC=0.0
      SYC2=0.0
      DO 108 I=1,N
      SY=SY+D(I,2)
      SY2=SY2+D(I,2)*D(I,2)
      SYC=SYC+D(I,3)
      SYC2=SYC2+D(I,3)*D(I,3)
  108 CONTINUE


... STATISTICAL ANALYSIS....

  903 SST=SY2-SY*SY/FLOAT(N)
      SSR=SYC2-SYC*SYC/FLOAT(N)
      SSD=SST-SSR
      R2=SSR/SST
      R=SQRT(R2)
      WRITE(6,1990)
      WRITE(6,1995) IORD
      WRITE(LU,1995) IORD
      WRITE(6,2000) N
      WRITE(LU,2000) N
      WRITE(6,2001) SST
      WRITE(LU,2001) SST
      WRITE(6,2002) SSR
      WRITE(LU,2002) SSR
      WRITE(6,2003) SSD
      WRITE(LU,2003)  SSD
      WRITE(6,2004) R2
      WRITE(LU,2004) R2
      WRITE(6,2005) R
      WRITE(LU,2005) R
 1001 FORMAT(/,"COEFF. MATRIX OF UNKNOWN PARAMETERS IN EQUATIONS")
 1002 FORMAT(/,"VECTOR OF CROSSPRODUCTS OF X AND Y")
 1003 FORMAT(/,"THE PARAMETERS OF THE REGRESSION EQUATION")
 1004 FORMAT(/," COL 1 = X VARIABLE"/" COL 2 = Y VARIABLE",/,
     1" COL 3 = Y VALUE BASED ON REGRESSION EQUATION",/,
     2" COL 4 = COL 2 - COL 3")
 1985 FORMAT(/," ORIGINAL X AND Y DATA")
 1990 FORMAT(1H1)
 1995 FORMAT(21H0ORDER OF EQUATION=   ,I5)
 2000 FORMAT(/,"NUMBER OF SAMPLES =",I5)
 2001 FORMAT(/,"TOTAL SUMS OF SQUARES = ",F15.4)
 2002 FORMAT(/,"SUMS OF SQUARES DUE TO REGRESSION =",F15.4)
 2003 FORMAT(/,"SUMS OF SQUARES DUE TO DEVIATION =",F15.4)
```

```
 2004 FORMAT(/,"GOODNESS OF FIT =",F15.4)
 2005 FORMAT(/,"CORRELATION COEFFICIENT =",F15.6)
      RETURN
      END
      SUBROUTINE SLE(A1,B1,N,N1,ZERO)
      EMA A1(125,125),B1(125)
**********************************************************

   SUBROUTINE FOR THE SOLUTION OF n SIMULTANEOUS
   EQUATIONS. MATRIX A IS n X n AND B IS A COLUMN
   VECTOR OF n ELEMENTS.
     A  IS CONVERTED TO THE IDENTITY MATRIX.
     B  CONTAINS THE SOLUTION.

**********************************************************

      DO 100 I=1,N
      DIV=A1(I,I)
      IF(ABS(DIV)-ZERO) 99,99,1
    1 DO 101 J=1,N
      A1(I,J)=A1(I,J)/DIV
  101 CONTINUE
      B1(I)=B1(I)/DIV
      DO 102 J=1,N
      IF(I-J) 2,102,2
    2 RATIO=A1(J,I)
      DO 103 K=1,N
      A1(J,K)=A1(J,K)-RATIO*A1(I,K)
  103 CONTINUE
      B1(J)=B1(J)-RATIO*B1(I)
  102 CONTINUE
  100 CONTINUE
   99 RETURN
      END
      SUBROUTINE PRINS(A,N,M,N1,M1,LU)
      EMA A(N1,M1)

**********************************************************

      SUBROUTINE TO PRINT A MATRIX HAVING N ROWS
      AND M COLUMNS.

**********************************************************

      DIMENSION LU(5)
      DO 100 IB=1,M,8
      IE=IB+7
      IF(IE-M) 2,2,1
    1 IE=M
    2 WRITE(6,2000) (I,I=IB,IE)
      WRITE(LU,2000) (I,I=IB,IE)
      DO 101 J=1,N
      WRITE(6,2001) J,(A(J,K),K=IB,IE)
      WRITE(LU,2001) J,(A(J,K),K=IB,IE)
  101 CONTINUE
  100 CONTINUE
 2000 FORMAT(1X,8I15)
 2001 FORMAT(I5,8F15.4)
      RETURN
      END
```

```
      SUBROUTINE UPFIL(E,NXN,ISC,ICR,LU,IFLGT,ISEG)

*****************************************************

    THIS SUBROUTINE ALLOWS THE DATA OF ARRAY E TO BE
    BE PLACED IN A DISC FILE ON CARTRIDGE NUMBER ICR.
    THE DATA FILE HAS A HEADER RECORD OF THE FORMAT:
       COL 10    COL 20   COL 30   COL 40
         18       IFLGT     ISEG     NXN

    THE FOLLOWING DATA IS THEN STORED IN FORMAT (8F10.2)

*****************************************************

      EMA E(2500)
      DIMENSION LU(5),NAME(3),IDCB(144),IBUF(40),ISEG(3)
      WRITE(LU,10)
   10 FORMAT(/,"WHAT DO YOU WANT TO CALL THE CREATED FILE?")
      READ(LU,20) (NAME(I),I=1,3)
   20 FORMAT(3A2)
      CALL CREAT(IDCB,IERR,NAME,32,3,ISC,ICR)
      IF(IERR.LT.0) GO TO 200
      CALL CODE
      WRITE(IBUF,1000) IFLGT,ISEG,NXN
 1000 FORMAT(8X,2H18,I10,4X,3A2,I10,4X)
      CALL WRITF(IDCB,IERR,IBUF,40)
      IF(IERR.LT.0) GO TO 200

    HEADER IS NOW ON FILE

      DO 100 I=1,NXN,8
      DO 90 J=1,40
      IBUF(J)=2H
   90 CONTINUE
      I7=I+7
      IF(I7.GT.NXN) I7=NXN
      CALL CODE
      WRITE(IBUF,2000) (E(K),K=I,I7)
 2000 FORMAT(8F10.2)
      CALL WRITF(IDCB,IERR,IBUF,40)
      IF(IERR.LT.0) GO TO 200

   DATA IS NOW ON FILE

  100 CONTINUE
      CALL CLOSE(IDCB,IERR)
      RETURN
  200 WRITE(LU,300) IERR
  300 FORMAT(/,"THERE WAS AN IERR = ",I4)
      RETURN
      END
      END$
```

```
TN4,B,L,Q,C
```

```
      ************************************
      *                                  *
      *                                  *
      *                                  *
      *         PROGRAM  FILTR           *
      *                                  *
      *                                  *
      *                                  *
      ************************************
```

```
EMA(BLOK1,2)
      PROGRAM FILTR(3,100)
C*************************************************************************
C
C     TITLE     : FILTR  -  CONVOLVES A ONE-DIMENSIONAL RECTANGULAR
C                 FILTER OPERATOR WITH PROFILE DATA. THE TRANSFER FUNCTION
C                 OF THE FILTER OPERATOR CAN BE ANALYSED AND MODIFIED
C                 IF REQUIRED.
C
C     AUTHOR    : PETER R GIDLEY , Bureau of Mineral Resources,CANBERRA, ACT
C
C     REF.      : FRASER, D.C., FULLER, B.D., WARD, S.H. -1966- SOME NUMERICAL
C                 TECHNIQUES FOR APPLICATION IN MINING EXPLORATION.
C                 GEOPHYSICS V.31, No.6, pp.1066-1077.
C
C     PURPOSE   : THIS PROGRAM CONSTRUCTS A 1-DIMENSIONAL
C                 RECTANGULAR FILTER WHICH IS CONVOLVED WITH
C                 DATA WHICH MAY BE READ IN FROM A DISC FILE.
C                 THE ORIGINAL DATA AND BOTH CONVOLVED AND FILTER
C                 COEFFICIENTS MAY BE DISPLAYED.
C
C*************************************************************************
      COMMON/BLOK1/ A(2502),B(2502),C(1001),D(1001)
      DIMENSION IDCB(144),IBUF(40),NAME(3),ISEG(3),ISGNO(3)
      DIMENSION LU(5),IPS(3),IB(2),FILE(2501),IFORM(40)
      DIMENSION F(101),SUM(501),RES0(301),FREQ(301)

      CALL RMPAR(LU)

      READ IN FILE DATA

   50 NAME(2)=2H
      NAME(3)=2H
      YUK=0.0
```

```
      DO 100 I=1,1000
      FILE(I)=0.0
  100 CONTINUE
      WRITE(LU,299)
  299 FORMAT(/,"ENTER FILE NAME")
      READ(LU,300) (NAME(I),I=1,3)
  300 FORMAT(3A2)
      WRITE(LU,400)
  400 FORMAT(/,"ENTER SECURITY CODE & CARTRIDGE NUMBER")
      READ(LU,*)ISC,ICR
      IF(ISC.LT.0) STOP
      IF(ICR.LE.0) ICR=9
      CALL OPEN(IDCB,IERR,NAME,0,ISC,ICR)
      IF(IERR.GE.0) GO TO 600
      WRITE(LU,500) IERR
  500 FORMAT(/,"ERROR ",I3," ON OPEN")
      STOP
  600 WRITE(LU,700)
  700 FORMAT(/,"WHAT DATA SEGMENT IS REQUIRED-RIGHT JUSTIFY?")
      READ(LU,300) ISGNO
  800 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR.LT.0) GO TO 1500
      CALL CODE
      READ(IBUF,900) IFLAG,FLGT,ISEG
  900 FORMAT(2I10,4X,3A2)
      DO 1000 I=1,3
      IPS(I)=ISGNO(I)
      IF(ISGNO(I).NE.ISEG(I)) GO TO 800
 1000 CONTINUE
      WRITE(LU,1100)
 1100 FORMAT(/,"WHAT IS FORMAT (F-TYPE) OF DATA--(E.G.(12F6.0))?")
      READ(LU,301) IFORM
  301 FORMAT(40A2)
      WRITE(LU,1155)
 1155 FORMAT(/,"HOW MANY DATA VALUES PER RECORD?")
      READ(LU,*) INO
      DO 1200 I=1,4000,INO
      CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR.LT.0) GO TO 1500
      IF(LEN.LT.0) GO TO 1600
      CALL CODE
      READ(IBUF,900) IFLAG
      IF(IFLAG.EQ.18) GO TO 1600
      NXN=I+INO-1
      CALL CODE
      READ(IBUF,IFORM)(FILE(J),J=I,NXN)
 1200 CONTINUE
      WRITE(LU,1300) ISEG
 1300 FORMAT(/," **** ERROR ERROR **** MORE THAN 4000 PTS IN ",3A2)
 1600 CALL CLOSE(IDCB,IERR)
      WRITE(LU,1400)
 1400 FORMAT(/,"DOES DATA NEED TO BE DIVIDED BY 1000-eg AIRBORNE FILE?")
      READ(LU,300) ITHOU
      IF(ITHOU.EQ.2HNO) GO TO 1450
      DO 1475 I=1,NXN
      FILE(I)=FILE(I)/1000.0
 1475 CONTINUE

       FILE DATA NOW IN,DO YOU WANT A PLOT OR WRITE IT?"
```

```
1450 WRITE(6,1151) NXN
     WRITE(LU,1151) NXN
     WRITE(LU,51)
  51 FORMAT(/,"DO YOU WANT TO PRINT DATA?",/,
    *"------IF YES TYPE IN NUMBER OF VALUES WANTED.")
     READ(LU,302)IB
 302 FORMAT(2A2)
     IF(IB.EQ.2HNO) GO TO 1149
     CALL CODE
     READ(IB,*)IPRIN
     WRITE(LU,111)(FILE(I),I=1,IPRIN)
     WRITE(6,112) (FILE(I),I=1,IPRIN)
 111 FORMAT(8(F8.2,2X))
 112 FORMAT(13(F8.2,2X))
1151 FORMAT(/,"THERE WERE A TOTAL OF ",I5," DATA POINTS READ")
1149 WRITE(LU,1150)
1150 FORMAT(/,"DO YOU WANT TO PLOT DATA?")
     READ(LU,300) IPLT
     WRITE(LU,255)
 255 FORMAT(/,"WHAT IS SAMPLE INTERVAL OF DATA?")
     READ(LU,*) DX
     IF(IPLT.EQ.2HYE) CALL UPLOT(FILE,NXN,DX,IST,IEND,TOT,LU
    *,YUK,T,F0,DF,IPS)

      CALL UP FILTER COEFFICIENTS.

     WRITE(LU,922)
 922 FORMAT(/,"DO YOU WISH TO FILTER DATA?")
     READ(LU,300) IFIL
     IF(IFIL.EQ.2HNO) GO TO 923
 201 WRITE(LU,350)
 350 FORMAT(/,"INPUT THE FOLLOWING FILTER PARAMETERS:")
      WRITE(LU,360)
 360 FORMAT(/,"  T=HALF-WIDTH OF NOMINATED FILTER LENGTH (SAME UNIT AS",
    *" DX")
     WRITE(LU,370)
 370 FORMAT(/,"  Fo=CENTRE FREQUENCY OF IDEAL BOX-CAR FILTER",/,
    *"                          (CYCLES/UNIT DIST)")
     WRITE(LU,380)
 380 FORMAT(/,"                Df=HALF-WIDTH OF IDEAL BOX-CAR FILTER",/,
    *"                          (CYCLES/UNIT DIST)")
     READ(LU,*) T,F0,DF
     IF(T.LT.0) STOP

     CALCULATE THE FILTER COEFFICIENTS.

     CALL COEFF(T,DX,F0,DF,C,NC,D)
     WRITE(LU,385)T,F0,DF,NC
     WRITE(6,385)T,F0,DF,NC
 385 FORMAT(/,"T=",F8.4,3X,"F0=",F6.4,3X,"DF=",F6.4,3X,"NC=",I4)

     CONVOLVE FILTER COEFFICIENTS WITH DATA

     CALL CONVO(FILE,B,C,NXN,NC,TOT)
     WRITE(LU,386)
 386 FORMAT(/,"DO YOU WANT TO PRINT FILTERED DATA?",/,
    *,"--------IF YES TYPE IN NUMBER OF VALUES REQUIRED.")
     READ(LU,302) IB
     CALL CODE
     READ(IB,*) IFPR
```

```
  SAVE FILTERED DATA IF REQUIRED FOR FURTHER PROCESSING

      IF(IFPR.EQ.2HNO) GO TO 387
      WRITE(LU,121)
      WRITE(6,121)
      WRITE(LU,111) (B(I),I=1,IFPR)
      WRITE(6,112) (B(I),I=1,IFPR)
  121 FORMAT(/,"***FILTERED DATA VALUES FOLLOW******")


      DO 551 I=1,NXN
      FILE(I)=B(I)
  551 CONTINUE
  387 WRITE(LU,405)
  405 FORMAT("DO YOU WANT TO SAVE THE FILTERED DATA?")
      READ(LU,300) ISVE
      IF(ISVE.EQ.2HYE) CALL UPFIL(B,NXN,ISC,ICR,LU,IFLGT,ISEG)
      WRITE(LU,550)
  550 FORMAT(/,"DO YOU WANT A PLOT OF FILTERED DATA?")
      READ(LU,300) IPL
      IF(IPL.EQ.2HYE) YUK=1.0
      IF(IPL.EQ.2HYE) CALL UPLOT(FILE,NXN,DX,IST,IEND,TOT,LU
     *,YUK,T,F0,DF,IPS)
      YUK=0.0

    OUTPUT OF PLOT AND VALUES OF FILTER TRANSFER FUNCTION

      WRITE(LU,650)
  650 FORMAT(/,"DO YOU WANT TO PRINT FILTER COEFFICIENTS?")
      READ(LU,300) IPRN
      IF(IPRN.EQ.2HYE)  WRITE(LU,113) (C(I),I=1,NC)
  113 FORMAT(8(F8.4,2X))
      IF(IPRN.EQ.2HYE) WRITE(6,117)
  117 FORMAT(/,"********* FILTER COEFFICIENTS FOLLOW *********")
      IF(IPRN.EQ.2HYE) WRITE(6,114) (C(I),I=1,NC)
  114 FORMAT(13(F8.5,2X))
      WRITE(LU,775)
  775 FORMAT(/,"DO YOU WANT TO PRINT FILTER FREQUENCY RESPONSE?")
      READ(LU,300) IRES
      IF(IRES.EQ.2HNO) GO TO 849
      WRITE(LU,841)
  841 FORMAT(/,"INPUT START & END FREQUENCIES OVER WHICH TO CALCULATE",
     *" RESPONSE (CYCLES/DX)")
      READ(LU,*)FRST,FREND
      WRITE(LU,842)
  842 FORMAT(/,"WHAT FREQUENCY INTERVAL IS REQUIRED TO CALCULATE THE "
     *"FILTER RESPONSE")
      READ(LU,*) FINT
      FFF=FREND-FRST
      IF(FFF.EQ.0.0) FFF=1.0
      NFREQ=ABS(FFF/FINT)
      DO 843 I=1,NFREQ
      FREQ(I)=FRST+FINT*(I-1)
      FF=FREQ(I)
      CALL FLRES(C,FF,DX,NC,RE)
      RES0(I)=RE
  843 CONTINUE
      WRITE(LU,844)
      WRITE(6,844)
```

```
  844 FORMAT(/,"*FREQUENCY RESPONSE OF FILTER OPERATOR**",/)
      WRITE(LU,846)
      WRITE(6,846)
  846 FORMAT(/,3("FREQUENCY     RESPONSE     "))
      WRITE(LU,848) (FREQ(J),RES0(J),J=1,NFREQ)
      WRITE(6,848) (FREQ(J),RES0(J),J=1,NFREQ)
  848 FORMAT(/,3(F9.4,4X,F8.4,4X))

      WRITE(LU,910)
  910 FORMAT(/,"DO YOU WANT TO PLOT FILTER RESPONSE?")
      READ(LU,300) IREF
      YUKI=0.0
      IF(IREF.EQ.2HYE) CALL REPLT(FREQ,RES0,NFREQ,T,F0,DF,YUKI)

      DO 851 I=1,NFREQ
      FREQ(I)=FREQ(I)/DF
  851 CONTINUE
      YUKI=1.0
      CALL REPLT(FREQ,RES0,NFREQ,T,F0,DF,YUKI)
  849 WRITE(LU,850)
  850 FORMAT(/,"DO YOU WISH TO ALTER FILTER DESIGN?")
      READ(LU,300) INEW
      IF(INEW.EQ.2HYE) GO TO 201
  923 WRITE(LU,950)
  950 FORMAT(/,"ANY MORE DATA FILES TO BE FILTERED?")
      READ(LU,300) MORE
      IF(MORE.EQ.2HYE) GO TO 50
      STOP
 1500 WRITE(LU,1510) IERR
 1510 FORMAT(/,"THERE HAS BEEN A READ-FROM-FILE ERROR-IERR=",I4)
      CALL CLOSE(IDCB,IERR)
      STOP
      END
      SUBROUTINE UPLOT(FILE,NXN,DX,IST,IEND,TOT,LU,YUK,T,F0,DF,IPS)

**********************************************************************

      NORMAL PLOTTING ROUTINE OF AMPLITUDE VERSUS
      DISTANCE. ANNOTATION OF FILTER PARAMETERS IS PROVIDED.

**********************************************************************

      DIMENSION FILE(1),LU(5),IPS(3)
      IF(YUK.EQ.1.0)GO TO 41
    5 WRITE(LU,10)
   10 FORMAT(/,"VERT SCALE IN UNITS/IN,HORIZ SCALE DIST/IN")
      READ(LU,*)SCM,SCD
      WRITE(LU,39)
   39 FORMAT(/,"ENTER START AND END POINTS")
      READ(LU,*) IST,IEND
      IF(IST.EQ.0) IST=1
      IF(IEND.GT.NXN) IEND=NXN
      WRITE(LU,40)
   40 FORMAT(/,"WHAT IS REFERENCE LEVEL-eg TOTAL FIELD")
      READ(LU,*) TOT
   41 SV1=TOT-SCM
      XT=DX*FLOAT(IEND-IST)/SCD
      IF(XT.LT.18.5) GO TO 103
      WRITE(LU,102)
  102 FORMAT(/,"PLOT TOO LONGGGGGGG")
```

```
      GO TO 5
  103 CALL PLOTS(XT+16.,12.,0)
      CALL FACTR(1.0)
      CALL PLOT(1.,0.5,-3)
      CALL AXIS(0.,4.,9HAMPLITUDE,9,5.,90.,SV1,SCM)
      CALL AXIS(0.,3.,8HDISTANCE,8,13.,0.,0.,SCD)
      CALL PLOT(0.,5.,-3)
      XP=0.
      DO 600 I=IST,IEND
      YP=(FILE(I)-TOT)/SCM
      CALL PLOT(XP,YP,2)
      XP=XP+DX/SCD
  600 CONTINUE
      IF(YUK.EQ.0.0) GO TO 700

    ANNOTATE PLOT

      CALL SYMB(8.,3.,0.1,22HHALF-WIDTH (DX UNITS)=,0.,22)
      CALL SYMB(8.,3.5,0.1,18HCENTRE FREQUENCY= ,0.,18)
      CALL SYMB(8.,4.0,0.1,22HHALF-WIDTH(CYC/DIST)= ,0.,22)
      CALL SYMB(8.,4.5,0.1,12HLINE NUMBER=,0.,12)
      CALL SYMB(11.,4.5,0.1,IPS,0.,6)
      CALL NUMB(11.,3.,0.1,T,0.,1)
      CALL NUMB(11.,3.5,0.1,F0,0.,4)
      CALL NUMB(11.,4.0,0.1,DF,0.,4)
  700 RETURN
      END
      SUBROUTINE COEFF(T,DX,F0,DF,C,NC,D)

**********************************************************************

    SUBROUTINE CALCULATES THE FILTER COEFFICIENTS FOR THE
    FILTER AS DEFINED BY Fo, Df, AND T.
       C  = OUTPUT ARRAY OF FILTER COEFFICIENTS.
       D  = WORKSPACE ARRAY.
       NL = NUMBER OF FILTER COEFFICIENTS CALCULATED.

**********************************************************************

      EMA C(1001),D(1001)
      PI=3.1415926536
      D(1)=4.0*DF*DX
      SUMD=D(1)
      ARG1=2*PI*DF*DX
      ARG2=2*PI*F0*DX
      ARG3=PI*DX/T
      MC=T/DX
      NC=MC+1
      DO 10 J=1,MC
      D(J+1)=SIN(J*ARG1)*COS(J*ARG2)*(1+COS(J*ARG3))/(J*PI)
      SUMD=SUMD+2.0*D(J+1)
   10 CONTINUE
      FACTOR=1/SUMD
      DO 20 J=1,NC
      C(J)=D(J)*FACTOR
   20 CONTINUE
      RETURN
      END
      SUBROUTINE CONVO(FILE,B,C,NXN,NC,ZERO)
      EMA B(2502),C(1001)
```

```
      DIMENSION FILE(2501)

**********************************************************************

      ARRAY FILE CONTAINS THE ORIGINAL DATA
      ARRAY B CONTAINS THE DATA PRODUCED BY A CONVOLUTION
         OF FILE BY C.
      ARRAY C  CONTAINS THE FILTER COEFFICIENTS (HALF OF
         A SYMMETRIC OPERATOR)
      NC IS THE NUMBER OF COEFFICIENTS IN ARRAY C.

      ZERO IS A VARIABLE TO WHICH THE NC-1 POINTS AT EITHER
         END OF THE PROFILE WHICH CANNOT BE CONVOLUTED ,ARE
         EQUATED.

**********************************************************************
      ITOP=NXN-NC+1
      DO 10 J=1,NXN
      B(J)=ZERO
   10 CONTINUE
      DO 20 L=NC,ITOP
      B(L)=FILE(L)*C(1)
      DO 30 J=2,NC
      B(L)=B(L)+(FILE(L-J+1)+FILE(L+J-1))*C(J)
   30 CONTINUE
   20 CONTINUE
      RETURN
      END
      SUBROUTINE FLRES(C,F,DX,NC,RESP)
      EMA C(1001)

**********************************************************************

        THIS SUBROUTINE CALCULATES THE FILTER RESPONSE
        AT FREQUENCY 'F'. IF A NUMBER OF DISCRETE
        FILTER FREQUENCY RESPONSES ARE REQUIRED THEN A
        SEPERATE CALL TO THIS SUBROUTINE EACH TIME IS
        MADE. THE RANGE AND INCREMENT OF THE RESPONSES
        TO BE MADE ARE REQUSTED IN THE MAIN PROGRAM.
     **ARRAY C CONTAINS THE FILTER COEFFICIENTS.
     **F IS THE FREQUENCY AT WHICH RESPONSE IS REQURED
     **DX IS THE DISTANCE BETWEEN DATA POINTS.
     **NC IS THE NUMBER OF FILTER COEFFICIENTS IN ARRAY C.
     **RESP IS THE CALCULATED RESPONSE.

**********************************************************************

      PI=3.14159265359
      ARG=2.0*PI*F*DX
      RESP=0.0
      DO 10 J=1,NC
      RESP=RESP+C(J)*COS(ARG*(J-1))
   10 CONTINUE
      RESP=2.0*RESP
      RETURN
      END
      SUBROUTINE REPLT(FREQ,RES0,NFREQ,T,F0,DF,YUKI)

**********************************************************************
```

```
      THIS SUBROUTINE ALLOWS THE CHARACTERISTICS OF FILTER TO HAVE IT'S
      PLOTTED. THE PLOT IS OF FREQUENCY VERSUS
      NORMALISED AMPLITUDE.

**********************************************************
      DIMENSION FREQ(301),RES0(301)
      REAL MAX
      CALL PLOTS(10.,10.,0)
      CALL FACTR(1.0)
      CALL PLOT(1.,1.,-3)
      IF(YUKI.EQ.1.0) GO TO 2
      DO 1 I=1,NFREQ
      FREQ(I)=FREQ(I)*1000.0
    1 CONTINUE
      CALL SCALP(FREQ,8.,NFREQ,1)
      CALL AXIS(0.,0.,16HFREQUENCY * 1000,-16,8.,0.,
     *FREQ(NFREQ+1),FREQ(NFREQ+2))
      DO 3 I=1,NFREQ
      FREQ(I)=FREQ(I)/1000.0
    3 CONTINUE
      GO TO 6
    2 CALL SCALP(FREQ,8.,NFREQ,1)
      CALL AXIS(0.,0.,26HFREQUENCY/CENTRE FREQUENCY,-26,8.,
     *0.,FREQ(NFREQ+1),FREQ(NFREQ+2))
    6 MAX=RES0(1)
      DO 10 I=1,NFREQ
      IF(MAX.LT.RES0(I)) MAX=RES0(I)
   10 CONTINUE
      DO 20 I=1,NFREQ
      RES0(I)=RES0(I)/MAX
   20 CONTINUE
      CALL SCALP(RES0,8.,NFREQ,1)
      CALL AXIS(0.,0.,9HAMPLITUDE,9,8.,90.,RES0(NFREQ+1),RES0(NFREQ+2))
      CALL LINE(FREQ,RES0,NFREQ)
      CALL SYMB(5.,6.,0.2,15HFILTER RESPONSE,0.,15)
      CALL SYMB(5.,3.,0.1,22HHALF-WIDTH (DX UNITS)=,0.,22)
      CALL SYMB(5.,3.5,0.1,18HCENTRE FREQUENCY= ,0.,18)
      CALL SYMB(5.,4.,0.1,21HHALF-WIDTH(CYC/DIST)=,0.,21)
      CALL NUMB(8.,3.,0.1,T,0.,1)
      CALL NUMB(8.,3.5,0.1,F0,0.,4)
      CALL NUMB(8.,4.0,0.1,DF,0.,4)
      RETURN
      END
      SUBROUTINE SCALP(RF,X,N,J)
      DIMENSION Y(301),RF(301)
      DO 10 I=1,N
      Y(I)=RF(I)*10000
   10 CONTINUE
      CALL SCALE(Y,X,N,J)
      RF(N+1)=Y(N+1)/10000
      RF(N+2)=Y(N+2)/10000
      RETURN
      END
      SUBROUTINE LINE(X,Y,N)
      DIMENSION X(301),Y(301)
      IP=3
      DO 10 I=1,N
      XL=(X(I)-X(N+1))/X(N+2)
      YL=(Y(I)-Y(N+1))/Y(N+2)
      CALL PLOT(XL,YL,IP)
```

```
      IP=2
   10 CONTINUE
      RETURN
      END
      SUBROUTINE UPFIL(B,NXN,ISC,ICR,LU,IFLGT,ISEG)

*************************************************************

    THIS SUBROUTINE ALLOWS THE DATA OF ARRAY B TO BE
    BE PLACED IN A DISC FILE ON CARTRIDGE NUMBER ICR.
    THE DATA FILE HAS A HEADER RECORD OF THE FORMAT:
     COL 1-10 COL 11-20 COL 21-30 COL 31-40
         10        IFLGT      ISEG     NXN

    THE FOLLOWING DATA IS THEN STORED IN FORMAT (8F10.2)

*************************************************************

      EMA B(2502)
      DIMENSION LU(5),NAME(3),IDCB(144),IBUF(40),ISEG(3)
      WRITE(LU,10)
   10 FORMAT(/,"WHAT DO YOU WANT TO CALL THE CREATED FILE?")
      READ(LU,20) (NAME(I),I=1,3)
   20 FORMAT(3A2)
      CALL CREAT(IDCB,IERR,NAME,32,3,ISC,ICR)
      IF(IERR.LT.0) GO TO 200
      CALL CODE
      WRITE(IBUF,1000) IFLGT,ISEG,NXN
 1000 FORMAT(8X,2H10,I10,4X,3A2,I10,4X)
      CALL WRITF(IDCB,IERR,IBUF,40)
      IF(IERR.LT.0) GO TO 200

    HEADER IS NOW ON FILE

      DO 100 I=1,NXN,8
      DO 90 J=1,40
      IBUF(J)=2H
   90 CONTINUE
      I7=I+7
      IF(I7.GT.NXN) I7=NXN
      CALL CODE
      WRITE(IBUF,2000) (B(K),K=I,I7)
 2000 FORMAT(8F10.2)
      CALL WRITF(IDCB,IERR,IBUF,40)
      IF(IERR.LT.0) GO TO 200

   DATA IS NOW ON FILE

  100 CONTINUE
      CALL CLOSE(IDCB,IERR)
      RETURN
  200 WRITE(LU,300) IERR
  300 FORMAT(/,"THERE WAS AN IERR = ",I4)
      RETURN
      END
      END$
```

TN4,B,L

```
**********************************
*                                *
*                                *
*                                *
*         PROGRAM  MULTI         *
*                                *
*                                *
*                                *
**********************************
```

PROGRAM MULTI

****

TITLE : MULTI - THE PROGRAM CALCULATES THE MAGNETIC ANOMALY
OVER A SERIES OF DIPPING PRISMS AND PLOTS THE RESULT.
THE RESULTING PROFILE MAY BE SAVED FOR LATER PROCESSING.

AUTHOR : PETER R. GIDLEY Bureau f Mineral Resources, CANBERRA, ACT.

REF. : (a) FROM FORMULAE DERIVED BY HJELT (1972)
HJELT, S.E. 1972 - MAGNETIC ANOMALIES DUE TO PRISM-SHAPED
BODIES WITH ARBITRARY POLARISATION. GEOPHYSICS, v.29,pp 517-531.
(b) R.D, OGILVY Bureau of Mineral Resources, CANBERRA, ACT.
in - AN INTERACTIVE COMPUTER PROGRAM TO CALCULATE THE MAGNETIC
ANOMALY OF A FINITE DIPPING PRISM. B.M.R. Record 1979/28.

PURPOSE : CALCULATES THE MAGNETIC ANOMALY OVER A SERIES OF UP TO 15
DIPPING PRISMS OR TABULAR BODIES. THE PROFILE OF CALCULATED
DATA CAN BE ORIENTED IN ANY DIRECTION RELATIVE TO MAGNETIC
NORTH OR THE BODIES. NATURAL REMANENCE IS ALSO TAKEN INTO
CONSIDERATION IF REQUIRED. SUBROUTINE 'UPFIL' ALLOWS THE
CALCULATED DATA TO BE STORED AS A HEADED DATA FILE.

NOTES : ALL DISTANCES ARE IN METRES
ALL ANGLES ARE IN DEGREES

USE : INPUT PARAMETERS

MN=MODEL NO.
X1=COOD OF FIRST TOP EDGE ALONG X AXIS
X2=COOD OF SECOND TOP EDGE ALONG X AXIS
Y1=COOD OF FIRST EDGE ON Y AXIS
Y2=COOD OF SECOND EDGE ON Y AXIS
Z1=DEPTH TO TOP OF PRISM
Z2=DEPTH TO BASE OF PRISM
XN=X COOD OF FIRST FIELD POINT
YN=Y COOD OF FIRST FIELD POINT

```
ZN=Z COOD OF FIRST FIELD POINT
DX=STATION INCREMENT ALONG PROFILE
PHI=DIP OF PRISM (MEASURED CLOCKWISE FROM +VE X AXIS)
DIP=INCLINATION OF EARTHS FIELD (DEGREES),  -VE IN
SOUTHERN HEMISHERE.
TOTAL=EARTHS TOTAL FIELD(NT)
SUSC=SUSCEPTIBILITY(C.G.S)
ALPHA=ANGLE BETWEEN PRISM STRIKE AND MAGNETIC NORTH
   *MEASURED CLOCKWISE FROM MN TO +VE Y AXIS.
BETA=ANGLE BETWEEN X AXIS AND PROFILE,MEASURED CLOCKWISE FROM
     +VE X AXIS.
NXN=NUMBER OF STATIONS
AT=ARRAY FOR TOTAL MAGNETIC FIELD VALUES
STAT=ARRAY FOR OBSERVATION POINTS
IREM=NO (REMANENCE ABSENT)
IREM=YES(REMANENCE PRESENT)
RI=INCLINATION OF REMANENT VECTOR WITH HORIZONTAL
RA=ANGLE BETWEEN MAGNETIC NORTH AND REMANENT VECTOR
 QKR=KOENIGSBERGER RATIO

*************************************************************************

      DIMENSION BOD1(30),BOD2(30),BOD3(30),BOD4(30),BOD5(30)
      DIMENSION F(8),U(2),V(2),W(2),G(8),B(8),S(8)
      DIMENSION T(8),A(8),AT(500),STAT(500),LU(5),SUM(15,500),SUMT(500)
      INTEGER STR1(8),STR2(7),STR3(9)
      INTEGER STR4(8),STR5(8),STR6(8),STR7(8),STR8(8),STR9(8)
      INTEGER STR10(8),STR11(7),STR12(6),STR13(7),STR14(6)
      DATA STR1/2HTO,2HTA,2HL ,2HFI,2HEL,2HD(,2HNT,2H)./
      DATA STR2/2HDE,2HPT,2HH(,2HME,2HTR,2HES,2H)./
      DATA STR3/2HDI,2HST,2HAN,2HCE,2H (,2HME,2HTR,2HES,2H)./
      DATA STR4/2HMO,2HDE,2HL ,2HNO,2H. ,2H  ,2H  ,2H  /
      DATA STR5/2HIN,2HDU,2HCI,2HNG,2H F,2HIE,2HLD,2H  /
      DATA STR6/2HIN,2HCL,2HIN,2HAT,2HIO,2HN ,2H  ,2H  /
      DATA STR11/2HCR,2HOS,2HS-,2HSE,2HCT,2HIO,2HN /
      DATA STR12/2HPR,2HOF,2HIL,2HE ,2H(Y,2HN)/
      DATA STR13/2HPR,2HOF,2HIL,2HE ,2HAN,2HGL,2HE /

 READ INPUT DATA

      CALL RMPAR(LU)
      MN=1
    1 DO 91 L=1,100
      SUMT(L)=0
      AT(L)=0
      DO 91 K=1,40
      SUM(K,L)=0
   91 CONTINUE
      WRITE(LU,300)
  300 FORMAT(/2X,"INPUT GRID PARAMETERS-XN,YN,ZN,DX,NXN,BETA")
      READ(LU,*)XN,YN,ZN,DX,NXN,BETA
      WRITE(LU,1000)
 1000 FORMAT(/2X,"INPUT MAGNETIC FIELD PARAMETERS -TOTAL,DIP")
      READ(LU,*)TOTAL,DIP
      DIPP=DIP
      BET=BETA
      WRITE(LU,1001)
 1001 FORMAT(/2X,"HOW MANY BODIES-MAXIMUM OF FIFTEEN?")
      READ(LU,*)NBOD
      DO 99 NQ=1,NBOD
```

```
1004 DIP=DIPP
     BETA=BET
     WRITE(LU,1002)NQ
1002 FORMAT(/2X,"INPUT",I2,"TH BODY DATA")
     WRITE(LU,1003)
1003 FORMAT(/2X,"                    X1,X2,Y1,Y2,Z1,Z2,SUSC,PHI,ALPHA")
     READ(LU,*)X1,X2,Y1,Y2,Z1,Z2,SUSC,PHI,ALPHA
     BOD1(NQ)=X1
     BOD2(NQ)=X2
     BOD3(NQ)=Z1
     BOD4(NQ)=Z2
     WRITE(LU,305)
 305 FORMAT(/2X,"REMANENT MAGNETIZATION?")
     READ(LU,17)IREM
  17 FORMAT(A2)
     IF(IREM.NE.2HYE)GO TO 14
  13 WRITE(LU,302)
 302 FORMAT(/2X,"QKR,RI,RA=")
     READ(LU,*)QKR,RI,RA
  14 WRITE(LU,303)
 303 FORMAT( '2X,"MISTAKE IN INPUT DATA?")
     READ(LU,5)NGOOD
   5 FORMAT(A2)
     IF(NGOOD.EQ.2HYE)GO TO 1004

    ALL DATA IS NOW IN - BEGIN PROCESSING
    CONVERT ALL ANGLES TO RADIANS

  29 PI=3.14159
     PHI=PHI*PI/180.
     DIP=DIP*PI/180.
     ALPHA=ALPHA*PI/180.
     RI=RI*PI/180.
     RA=RA*PI/180.
     BETA=BETA*PI/180.
     BOD5(NQ)=PHI
     XM=XN

   LOOP 99 IS THE MAIN PROCESSING LOOP . IT CALCULATES THE MAGNETIC
   VALUE AT EACH SUCCESSIVE POINT ALONG T.' PROFILE.

     DO 99 L=1,NXN
     X=(L-1)*DX*COS(BETA)+XN
     Y=(L-1)*DX*SIN(BETA)+YN
     STAT(L)=XN+(L-1)*DX
     H=Z2-Z1
     U(1)=X-X2
     U(2)=X-X1
     V(1)=Y-Y2
     V(2)=Y-Y1
     W(1)=ZN-Z2
     W(2)=ZN-Z1

CALCULATE F1U ,F1V,F1W,F2U,F2V,F2W

     M=1
     F1U=0.
     F1V=0.
     F1W=0.
     F2U=0.
```

```
      F2V=0.
      F2W=0.
      DO 10 I=1,2
      DO 10 J=1,2
      DO 10 K=1,2
      IF(K.EQ.1)U(I)=U(I)-H/TAN(PHI)
      IF((ABS(U(I))-0.001).LT.0)U(I)=0.001
      IF((ABS(V(J))-0.001).LT.0)V(J)=0.001
      IF((ABS(W(K))-0.001).LT.0)W(K)=0.001
      R=SQRT(U(I)*U(I)+V(J)*V(J)+W(K)*W(K))
      P=U(I)*COS(PHI)+W(K)*SIN(PHI)
      Q=U(I)*SIN(PHI)-W(K)*COS(PHI)
      IF((ABS(Q)-0.001).LT.0)Q=0.001
      F(M)=ATAN((V(J)*P)/(Q*R))
      G(M)=ATAN(((V(J)*V(J)*COS(PHI)-W(K))*Q)/(V(J)*SIN(PHI)*R))
      B(M)=ATAN((V(J)*U(I))/(W(K)*R))
      S(M)=ALOG(U(I)+R)
      T(M)=ALOG(V(J)+R)
      A(M)=ALOG(P+R)
      IJK=I+J+K
      IF(IJK.EQ.4.OR.IJK.EQ.6)GO TO 7
    6 F(M)=-F(M)
      G(M)=-G(M)
      B(M)=-B(M)
      S(M)=-S(M)
      T(M)=-T(M)
      A(M)=-A(M)
    7 IF(K.EQ.1)U(I)=U(I)+H*COS(PHI)/SIN(PHI)
      M=M+1
   10 CONTINUE
      DO 11 N =1,8
      F1U=F1U+F(N)
      F1V=F1V+G(N)
      F1W=F1W+B(N)
      F2U=F2U+S(N)
      F2V=F2V+T(N)
      F2W=F2W+A(N)
   11 CONTINUE

CALCULATE T FACTORS

      T11=COS(PHI)*F2V-SIN(PHI)*F1U
      T12=SIN(PHI)*F2W
      T13=F2V
      T21=F2W
      T22=F1V
      T23=F2U
      T31=COS(PHI)*F1U+SIN(PHI)*F2V
      T32=F2U-COS(PHI)*F2W
      T33=-F1W

CALCULATION OF H FACTORS

      TOX=TOTAL*COS(DIP)*SIN(ALPHA)
      TOY=TOTAL*COS(DIP)*COS(ALPHA)
      TOZ=TOTAL*SIN(DIP)
      IF(IREM.EQ.2HYE)GO TO 613
      AJX=SUSC*TOX
      AJY=SUSC*TOY
      AJZ=SUSC*TOZ
```

```
614 AJ1=AJX*SIN(PHI)-AJZ*COS(PHI)
    AJ2=AJY
    AJ3=AJZ
    HX=AJ1*T11+AJ2*T12+AJ3*T13
    HY=AJ1*T21+AJ2*T22+AJ3*T23
    HZ=AJ1*T31+AJ2*T32+AJ3*T33

CALCULATION OF TOTAL FIELD

    AT(L)=HX*COS(DIP)*SIN(ALPHA)+HY*COS(DIP)*COS(ALPHA)+HZ*SIN(DIP)
    SUM(NQ,L)=AT(L)
 99 CONTINUE
    XN=XM
    GO TO 550

CALCULATION OF REMANENT MAGNETIZATION

613 TR=QKR*TOTAL
    TRX=TR*COS(RI)*SIN(ALPHA-RA)
    TRY=TR*COS(RI)*COS(ALPHA-RA)
    TRZ=TR*SIN(RI)
    AJX=SUSC*(TOX+TRX)
    AJY=SUSC*(TOY+TRY)
    AJZ=SUSC*(TOZ+TRZ)
    GO TO 614

PRINT RESULTS

550 WRITE(LU,3)MN
  3 FORMAT(1H1,7X,9HMODEL NO.,I10)
    WRITE(LU,202)
202 FORMAT(1H0,7X,7HSTATION,10X,11HTOTAL FIELD)
    DO 250 L=1,NXN
    DO 250 NB=1,NBOD
    SUMT(L)=SUM(NB,L)+SUMT(L)
250 CONTINUE
    DO 56 J=1,NXN
    WRITE(LU,60)STAT(J),SUMT(J)
 60 FORMAT(1H0,2X,F10.2,8X,F10.2)
 56 CONTINUE

 PLOT RESULTS IF REQUIRED

    WRITE(LU,65)
 65 FORMAT(/2X,"PLOT RESULTS?")
    READ(LU,5)NGOOD
    IF(NGOOD.EQ.2HNO)GO TO 404
100 WRITE(LU,70)
 70 FORMAT(/2X,"VERT SCALE NT/IN, HORIZ SCALE METRES/IN")
    READ(LU,*)SCM,SCD

 FIND THE DEEPEST BODY

    DO 95 K=1,NBOD
    IF(BOD4(K).GT.BIGZ) BIGZ=BOD4(K)
 95 CONTINUE
    SV1=TOTAL-SCM
    SI1=SCM
    SV2=BIGZ
    SI2=-BIGZ/3.
```

```
      DIST=SCD
      XT=DX*FLOAT(NXN)/SCD
      IF(XT.GT.9.5)101,103
  101 WRITE(LU,102)
  102 FORMAT(/2X,"PLOT TOO LONG CHANGE DISTANCE SCALE")
      GO TO 100

    PLOT AXES

  103 CALL PLOTS(XT+8.,12.,0)
      CALL FACTR(1.0)
      IF(MN.EQ.1)CALL PLOT(1.,0.5,-3)
      CALL AXIS(0.,4.,STR1,16,5.,90.,SV1,SI1)
      CALL AXIS(0.,0.,STR2,14,3.,90.,SV2,SI2)
      CALL AXIS(0.,3.,STR3,18,9.,0.,XM,DIST)

       PLOT  POINTS

      CALL PLOT(0.,5.,-3)
      XP=0.
      DO 600 I=1,NXN
      YP=SUMT(I)/SCM
      CALL SYMB(XP,YP,0.1,7,0.,-1,0.1)
      XP=XP+DX/DIST
  600 CONTINUE

   CONVERT RADIANS BACK TO DEGREES
   PLOT THE VARIOUS BODIES BELOW AXES.

      BETA=BETA*180./PI
      IF(IFIX(BETA).GT.0)GO TO 900
      CALL PLOT(0.,-2.,-3)
      DO 96 N=1,NBOD
      KIK=BOD4(N)-BOD3(N)
      XP1=BOD1(N)/DIST-XM/DIST
      YP1=BOD3(N)/SI2
      YP2=BOD4(N)/SI2
      XP2=BOD2(N)/DIST-XM/DIST
      XP3=(BOD1(N)+KIK/TAN(BOD5(N)))/DIST-XM/DIST
      XP4=(BOD2(N)+KIK/TAN(BOD5(N)))/DIST-XM/DIST
      CALL PLOT(XP1,YP1,3)
      CALL PLOT(XP2,YP1,2)
      CALL PLOT(XP3,YP2,2)
      CALL PLOT(XP4,YP2,2)
      CALL PLOT(XP1,YP1,2)
   96 CONTINUE

    ANNOTATE PLOT

      CALL SYMB(7.,-1.5,0.1,STR11,0.,13)
  900 DIP=DIP*180./PI
      CALL SYMB(7.,5.5,0.1,STR4,0.,16)
      CALL SYMB(7.,5.,0.1,STR5,0.,16)
      CALL SYMB(7.,4.7,0.1,STR6,0.,16)
      CALL SYMB(7.,4.4,0.1,STR12,0.,12)
      CALL SYMB(7.,4.1,0.1,STR13,0.,14)
      CALL NUMB(8.8,5.5,0.1,FLOAT(MN),0.,0)
      CALL NUMB(8.8,5.0,0.1,TOTAL,0.,1).
      CALL NUMB(8.8,4.7,0.1,DIP,0.,2)
      CALL NUMB(8.8,4.4,0.1,YN,0.,2)
```

```
      CALL NUMB(8.8,4.1,0.1,BETA,0.,2)

  OPTION CALL TO SAVE PROFILE FOR LATER PROCESSING.

  404 WRITE(LU,405)
  405 FORMAT(" DO YOU WANT TO SAVE THE CALCULATED PROFILE?")
      READ(LU,81) ISVE
      IF(ISVE.EQ.2HYE) CALL UPFIL(SUMT,NXN,LU)
   TERMINATE PLOT
      CALL PLOT(0.,0.,999)
      CALL GOPLT
      WRITE(LU,80)
   80 FORMAT(/2X,"ANY MORE CASES?")
      READ(LU,81)MORE
   81 FORMAT(A2)
      IF(MORE.EQ.2HYE)GO TO 1
   27 STOP
      END
     SUBROUTINE UPFIL(B,NXN,LU)

**********************************************************

    THIS SUBROUTINE ALLOWS THE DATA OF ARRAY E TO BE
    BE PLACED IN A DISC FILE ON CARTRIDGE NUMBER ICR.
    THE DATA FILE HAS A HEADER RECORD OF THE FORMAT:
       COL 10    COL 20   COL 30   COL 40
         18        IFLGT     ISEG     NXN

     THE FOLLOWING DATA IS THEN STORED IN FORMAT (8F10.2)

**********************************************************

      DIMENSION LU(5),NAME(3),IDCB(144),IBUF(40),ISEG(3),B(200)
      WRITE(LU,10)
   10 FORMAT(/,"WHAT DO YOU WANT TO CALL THE CREATED FILE?")
      READ(LU,20) (NAME(I),I=1,3)
      WRITE(LU,30)
   30 FORMAT(/,"INPUT SECURITY CODE AND CARTRIDGE NUMBER")
      READ(LU,*) ISC,ICR
      WRITE(LU,40)
   40 FORMAT(/,"WHAT IS THE LINE NUMBER -RIGHT JUSTIFY -eg ----15")
      READ(LU,20) ISEG
   20 FORMAT(3A2)
      IFLGT=270
      CALL CREAT(IDCB,IERR,NAME,32,3,ISC,ICR)
      IF(IERR.LT.0) GO TO 200
      CALL CODE
      WRITE(IBUF,1000) IFLGT,ISEG,NXN
 1000 FORMAT(8X,2H18,I10,4X,3A2,I10,4X)
      CALL WRITF(IDCB,IERR,IBUF,40)
      IF(IERR.LT.0) GO TO 200

    HEADER IS NOW ON FILE

      DO 100 I=1,NXN,8
      DO 90 J=1,40
      IBUF(J)=2H
   90 CONTINUE
      I7=I+7
      IF(I7.GT.NXN) I7=NXN
```

```
      CALL CODE
      WRITE(IBUF,2000) (B(K),K=I,I7)
 2000 FORMAT(8F10.2)
      CALL WRITF(IDCB,IERR,IBUF,40)
      IF(IERR.LT.0) GO TO 200

  DATA IS NOW ON FILE

  100 CONTINUE
      CALL CLOSE(IDCB,IERR)
      RETURN

    OUTPUT ERROR MESSAGE IF ONE EXISTED ON FILE CREATION

  200 WRITE(LU,300) IERR
  300 FORMAT(/,"THERE WAS AN IERR = ",I4)
      RETURN
      END
      END$
```

TN4,B,L,Q,C

```
**********************************************
*                                            *
*                                            *
*            PROGRAM  SPECT                  *
*                                            *
*                                            *
**********************************************
```

```
EMA(BLOK1,2)
      PROGRAM SPECT(3,200)
```

****

TITLE : SPECT - CALCULATES THE POWER SPECTRUM OF A ONE-DIMENSIONAL DATA PROFILE. THE PROGRAM MAY ALSO BE USED AUTOMATICALLY BY CALCULATING THE POWER SPECTRA OF A MOVING WINDOW OR SEGMENT OF THE DATA PROFILE.

REF. : THE THEORY AND PROCEDURE INVOLVED IN THE PROCESSING IS DESCPIBED BY :
SPECTOR A. -1968- SPECTRAL ANALYSIS OF AEROMAGNETIC DATA.
Unpublished Ph.D Thesis , University of Toronto, CANADA.

AUTHOR : PROGRAM IS AN HP 2000 SERIES ADAPTATION OF A SPECTRAL ANALYSIS PROGRAM. THE ORIGINAL PROGRAM WAS OBTAINED FROM THE ARGUS LIBRARY AND WAS INITIALLY ESTABLISHED FROM AN ALGORITHM OBTAINED FROM UNI. OF TASMANIA (ALGOL) BY JOHN REES. ----------10/7/1980----------
THE PROGRAM AS PRESENTED HERE WAS WRITTEN BY P. R. GIDLEY.

PURPOSE : THE PROGRAM CONVERTS AND PLOTS A ONE-DIMENSIONAL SPATIAL DOMAIN PROFILE TO ITS FREQUENCY DOMAIN EQUIVALENT. THE POWER SPECTRA DERIVED ARE CALCULATED IN SUBROUTINES 'SPECO', 'FFT1', AND 'FFT2' AND FOLLOW THE PRESENTATION FORMAT AS DESCRIBED BY SPECTOR (1968). THE FAST FOURIER TRANSFORM ALGORITHMS ARE DERIVED FROM COOLEY & TUKEY (SEE SUBROUINE FFT1) . THE PROGRAM FIRST SELECTS THE DATA POINTS (OF A NUMBER DEFINED BY THE ADOPTED WINDOW LENGTH). IT THEN MOVES ONTO THE NEXT WINDOW, CALCULATES, MOVES, AND SO ON, UNTIL ALL THE DATA IS USED. WINDOW OVERLAP IS POSSIBLE AS AN OPTION. IF ONLY ONE SPECTRUM IS REQUIRED FOR THE WHOLE PROFILE, MAKE THE WINDOW LENGTH EQUAL TO THE PROFILE LENGTH.

USE : THE PROGRAM IS DIVIDED INTO A MAIN CALLING PROGRAM AND INTO A NUMBER OF PROCESSING SUBROUTINES. IN THE MAIN PROGRAM, INSTRUCTIONS ARE PASSED TO THE USER (THROUGH

```
                THE TERMINAL HE IS ON) TO NAME THE DATA FILE FOR WHICH
                A NUMBER OF PROCESSING OPTIONS ARE AVAILABLE.OPTIONS
                INCLUDE:
    (1)   TO PLOT THE ORIGINAL PROFILE DATA USING INPUT
          OF SCALES, SAMPLE INTERVAL,START AND END POINTS
          AND THE TOTAL MAGNETIC FIELD.
    (2)   TO CALCULATE AND THEN PLOT THE SPECTRAL (ONE-
          DIMENSIONAL) ANALYSIS OF THE PROFILE DATA  CONTAINED IN EACH WINDOW.
    (3)   THE CALCULATED SPECTRAL PLOT CAN BE EXPANDED
          TO OBTAIN A DETAILED LOOK AT THE SPECTRUM AT
          LOW FREQUENCIES (LONG WAVELENGTHS).
    (4)   A SMOOTHED SPECTRUM CAN BE OBTAINED.
          THIS OPTION PASSES A HANNING WINDOW FILTER
          THROUGH THE ORIGINAL DATA.
    (5)   RETURN TO THE START IS POSSIBLE SHOULD
          ANY MORE FILES REQUIRE PROCESSING.

**************************************************************************

      COMMON/BLOK1/SUMP(4000),FRE(4005),WSP(8000),
     2AMP(4000),ENDEN(4000),DIMDAT(8000)
      DIMENSION IDCB(144),IBUF(40),NAME(3),ISEG(3),ISGNO(3),X(12)
      DIMENSION LU(5),IPS(3),CAT(4002),IB(2),IFORM(40),IK(2)
      INTEGER STR14(6),STR15(2),FLGT,STR1(12),STR2(6),AUTO
      REAL LTH
      DATA STR14/2HCY,2HCL,2HES/
      DATA STR1/2HTO,2HTA,2HL ,2HFI,2HEL,2HD /
      DATA STR2/2HME,2HTR,2HES/
      DATA STR15/2HDB/
      CALL RMPAR(LU)

    INPUT FILE OF DATA AND THE REQUIRED PARAMETERS

  500 NAME(2)=2H
      NAME(3)=2H
      NYE=0
      DO 505 I=1,4000
      SUMP(I)=0.0
  505 CONTINUE
      WRITE(LU,510)
  510 FORMAT(/,"ENTER FILE NAME?")
      READ(LU,520) (NAME(I),I=1,3)
  520 FORMAT(3A2)
      WRITE(LU,530)
  530 FORMAT(/,"ENTER SECURITY CODE,AND CARTRIDGE")
      READ(LU,*) ISC,ICR
      IF(ISC.LT.0) STOP
      IF(ICR.LE.0) ICR=9

      CALL OPEN(IDCB,IERR,NAME,0,ISC,ICR)
      IF(IERR.GE.0) GO TO 540

      WRITE(LU,550) IERR
  550 FORMAT(/,"**** ERROR ",I3," ON OPEN")
      STOP
  540 WRITE(LU,555)
  555 FORMAT(/,"WHAT FLIGHT SEGMENT DO YOU WANT?-RIGHT JUSTIFY")
      READ(LU,557) ISGNO
  557 FORMAT(3A2)
  556 CALL READF(IDCB,IERR,IBUF,40,LEN)
```

```
      IF(IERR.LT.0)  GO TO 1500
      CALL CODE
      READ(IBUF,560) IFLAG,FLGT,ISEG,NPTS
  560 FORMAT(2I10,4X,3A2,I10)
      DO 565 I=1,3
      IPS(I)=ISGNO(I)
      IF(ISGNO(I).NE.ISEG(I)) GO TO 556
  565 CONTINUE
      ICURT=1
  573 WRITE(LU,572)
  572 FORMAT(/,"WHAT IS FORMAT OF INPUT FILE eg (10X,5F12.1)?")
      READ(LU,301)IFORM
  301 FORMAT(40A2)
      WRITE(LU,1155)
 1155 FORMAT(/,"HOW MANY DATA VALUES PER RECORD ON THE FILE?")
      READ(LU,*) INO
      WRITE(LU,600)
  600 FORMAT(/,"DOES DATA NEED TO BE DIVIDED BY 1000-eg AIRBORNE?")
      READ(LU,81) ITHOU
      WRITE(LU,750)
  750 FORMAT(/,"HOW MANY DATA POINTS IN THE WINDOW?")
      READ(LU,*)NP
      WRITE(LU,751)
  751 FORMAT(/,"WHAT IS THE LOCATION OF THE STARTING POINT OF WINDOW?")
      READ(LU,*) ISW
      WRITE(LU,761) NPTS
  761 FORMAT(/,"THERE ARE ",I5," VALUES ON THIS FILE. WHAT IS THE",/,
     *" UPPER LIMIT OF THE NUMBER OF VALUES TO BE PROCESSED. TYPE",/,
     *" IN THE TOTAL NUMBER IF ALL THE LINE IS TO BE PROCESSED")
      READ(LU,*) NPTS
      WRITE(LU,763) ISW,NPTS
  763 FORMAT(/,"THE LINE SHALL BE PROCESSED FROM ",I4," TO ",I4)
      WRITE(LU,70)
   70 FORMAT(/,"FOR PLOTTING-VERT SCALE nT/IN,HORIZ METRES/IN")
      READ(LU,*) SCM,SCD
      WRITE(LU,754)
  754 FORMAT(/,"WHAT PERCENTAGE OVERLAP OF WINDOW IS REQUIRED?")
      READ(LU,*) PERCY
      WRITE(LU,400)
  400 FORMAT(/,"ENTER SPECTRUM'S HORIZ AXIS LENGTH IN INCHES")
      READ(LU,*) LTH
      WRITE(LU,82)
   82 FORMAT(/,"WHAT IS SAMPLE INTERVAL OF DATA?")
      READ(LU,*) DX
      WRITE(LU,83)
   83 FORMAT(/,"WHAT IS TOTAL MAGNETIC FIELD?")
      READ(LU,*) TOT
      WRITE(LU,3075)
 3075 FORMAT(/,"WHAT IS THE D.C. LEVEL OF PROFILE?")
      READ(LU,*) DCLVL
      WRITE(LU,84)
   84 FORMAT(/,"ANY MISTAKES ON INPUTING PARAMETERS?")
      READ(LU,81) MIS
      IF(MIS.EQ.2HYE) GO TO 573
      ME=NP
      DO 757 IKI=1,NPTS
      IF(ME.GE.NPTS) GO TO 755
      ME=ME+(1.0-PERCY/100.0)*NP+0.5
  757 CONTINUE
      WRITE(LU,756)
```

```
756 FORMAT(/,"DISASTER - THERE ARE MORE WINDOWS THAN POINTS****")
755 INUMW=IKI
    IF(ME.GT.NPTS) INUMW=INUMW-1

    WRITE(LU,752)INUMW
752 FORMAT(/,"THE NUMBER OF WINDOWS ON THE LINE IS = ",I3)
    DO 753 IKI=1,INUMW
    IEW=ISW+NP-1

 ISW IS IN RECORD IS, IEW IS IN IE

    IS=(ISW-1)/INO+1

 ICURT=IS-ICURT AND ICURT IS CURRENT RECORD WHICH MUST
 BE INCREMENTED WITH EACH READF.

    IDIFF=IS-ICURT
    CALL POSNT(IDCB,IERR,IDIFF)
    IF(IERR.LT.0) GO TO 1500
    ICURT=IS

  IEW IS THE END OF THE WINDOW
  ISW IS THE START OF THE WINDOW
  NP  IS THE NUMBER OF THE POINT IN THE WINDOW.

    IX=MOD(ISW,INO)
    N1=IX
    NXN=0
    JX=MOD(IEW,INO)
    N2=INO
    DO 570 I=1,4000,INO
    CALL READF(IDCB,IERR,IBUF,40,LEN)
    IF(IERR.LT.0) GO TO 1500
    IF(LEN.LT.0) GO TO 1600
    ICURT=ICURT+1
    CALL CODE
    READ(IBUF,560) IFLAG
    IF(IFLAG.EQ.18) GO TO 1600
    CALL CODE
    READ(IBUF,IFORM) (X(II),II=1,INO)
    IF(NXN+INO+ISW-1.GT.IEW) N2=JX
    DO 571 J=N1,N2
    NXN=NXN+1
    SUMP(NXN)=X(J)
 571 CONTINUE
    IF(NXN+ISW-1.EQ.IEW) GO TO 1600
    N1=1
 570 CONTINUE
    WRITE(LU,575) ISEG
 575 FORMAT(X,"  MORE THAN 4000 PTS IN  ",3A2)
    CALL CLOSE(IDCB,IERR)
1600 CONTINUE
    IF(ITHOU.EQ.2HNO) GO TO 1450

   DIVDE BY 1000 IF USING AIRBORNE FILE DATA.

    DO 1475 I=1,NXN
    SUMP(I)=SUMP(I)/1000.0
1475 CONTINUE
1450 WRITE(6,605) ISGNO
```

```
 605 FORMAT(/,"THE FOLLOWING DATA RELATES TO LINE ",3A2)
     WRITE(LU,1152) IKI,ISW,IEW
     WRITE(6,1152) IKI,ISW,IEW
1152 FORMAT(/,"NOW PROCESSING WINDOW",I3," FROM POINT ",I4," TO ",I4)
     WRITE(LU,1151) NXN
     WRITE(6,1151) NXN
1151 FORMAT(/,"THERE WERE A TOTAL OF ",I6,"  DATA POINTS READ.")
     IF(AUTO.EQ.2HYE) GO TO 950
     WRITE(LU,51)
  51 FORMAT(/,"DO YOU WANT TO PRINT DATA?",
    2/,"---------IF YES TYPE IN THE NUMBER OF VALUES REQUIRED",
    3/,"---------IF NO TYPE IN 'NO'.")
     READ(LU,820) IK
 950 IF(IK.EQ.2HNO) GO TO 833
     CALL CODE
     READ(IK,*) NUT
     WRITE(LU,831) (SUMP(I),I=1,NUT)
 831 FORMAT(8(F8.2,2X))

  PLOT DATA EACH TIME THROUGH A NEW WINDOW IF AUTO IS SET

 833 IF(AUTO.EQ.2HYE)GO TO 951
     WRITE(LU,65)
  65 FORMAT(/2X,"PLOT ARRAY OF INPUT DATA?")
     READ(LU,81) NGOO
 951 IF(NGOO.EQ.2HNO) GO TO 671
 105 SV1=TOT-SCM
     XT=DX*(NXN-1)/SCD
     IF(XT.GT.9.5) 101,103
 101 WRITE(LU,102)
 102 FORMAT(/2X,"PLOT TOO LONGGGGGGGGG")
     WRITE(LU,104)
 104 FORMAT(/,"INPUT NEW PLOT SCALES - VERT nT/IN HORIZ m/IN")
     READ(LU,*) SCM,SCD
     GO TO 105
 103 CALL PLOTS(XT+8.,12.,0)
     CALL FACTR(1.0)
     CALL PLOT(1.,0.5,-3)
     CALL AXIS(0.,4.,STR1,12,5.,90.,SV1,SCM)
     CALL AXIS(0.,3.,STR2,-6,8.,0.,0.,SCD)
     CALL PLOT(0.,5.,-3)
     XP=0.
     DO 601 I=1,NXN
     YP=(SUMP(I)-TOT)/SCM
     CALL PLOT(XP,YP,2)
     XP=XP+DX/SCD
 601 CONTINUE

   NORMALISE DATA TO ZERO IF WANTED

     XP=XP+DX/SCD
 671 DO 672 I=1,NXN
     SUMP(I)=SUMP(I)-DCLVL
 672 CONTINUE
     WRITE(LU,831) (SUMP(J),J=1,NUT)

   CALL SPECTRAL ANALYSIS SUBROUTINES

     CALL SPECO(NXN)
```

```
  PLOT SPECTRAL RESULTS?

      IF(AUTO.EQ.2HYE) GO TO 952
      WRITE(LU,87)
   87 FORMAT(/,"PLOT SPECTRAL RESULTS?")
      READ(LU,81) NYES
  952 IF(NYES.EQ.2HNO) GO TO 75
      NDAT=NXN/2
      ND=NDAT+2
  610 CALL PLOTS(LTH,12.,0)
      CALL FACTR(0.5)
      CALL PLOT(1.,6.,-3)

  ARRAY 'CAT' HAS BEEN USED FOR DATA SO AUTOMATIC SCALING CAN BE
  MADE. THIS IS BECAUSE ARRAY 'SUMP' IS IN EMA.

      DO 620 K=1,NDAT
      CAT(K)=SUMP(K)
  620 CONTINUE
      CALL SCALE(CAT(2),10.,NDAT-1,1)
      SUMP(NDAT+1)=CAT(NDAT+1)
      SUMP(NDAT+2)=CAT(NDAT+2)
      CALL AXIS(0.,0.,STR15,2,10.,90.,CAT(NDAT+1),CAT(NDAT+2))
      DO 7777 I=1,ND
      CAT(I)=FRE(I)*100.
 7777 CONTINUE
      CALL SCALE(CAT(2),LTH,NDAT-1,1)
      DO 779 I=1,NDAT+2
      CAT(I)=CAT(I)/100.0
  779 CONTINUE
      CALL AXIS(0.,0.,STR14,-6,LTH,0.,CAT(NDAT+1),
     3CAT(NDAT+2))
      CALL LINE(CAT(2),SUMP(2),NDAT-1)
      CALL SYMB(8.0,8.0,0.5,IPS,0.,6)
      EW=FLOAT(IEW)
      EST=FLOAT(ISW)
      CALL SYMB(8.0,7.0,0.3,14HWINDOW START =,0.,14)
      CALL SYMB(8.0,6.5,0.3,14HWINDOW ENDS  =,0.,14)
      CALL NUMB(12.5,7.0,0.3,EST,0.,0)
      CALL NUMB(12.5,6.5,0.3,EW,0.0,0)
      IF(NYE.EQ.2HYE) GO TO 670
      IF(ZOT.EQ.1.) GO TO 675
      IF(AUTO.EQ.2HYE) GO TO 953

   EXPAND LONG WAVELENGTH DISPLAY IF REQUESTED

      WRITE(LU,800)
  800 FORMAT(/2X,"DO YOU WANT AN EXPANDED POWER SPECTRUM?",
     9/,"------IF SO TYPE IN CUT OFF FREQUENCY (0-0.5)")
      READ(LU,820) IB
  820 FORMAT(2A2)
  953 IF(IB.EQ.2HNO) GO TO 675
      CALL CODE
      READ(IB,830) FREC
  830 FORMAT(F4.0)
      NDAT=NDAT*(FREC/0.5)
      ZOT=1.0
      GO TO 610
  675 NDAT=NXN/2
      ND=NDAT+2
```

```
      IF(AUTO.EQ.2HYE)GO TO 954
      NYE=0
      ZOT=0.0

   PLOT COMPLETE SMOOTHED SPECTRUM IF REQUESTED

      WRITE(LU,660)
  660 FORMAT(2X,"DO YOU WANT A SMOOTHED SPECTRUM?")
      READ(LU,81)NYE
  954 IF(NYE.EQ.2HNO) GO TO 670
      DO 680 I=2,NDAT-1
      SUMP(I)=SUMP(I)/4.0+SUMP(I+1)/2.0+SUMP(I+2)/4.0
  680 CONTINUE
      WRITE(6,690)
  690 FORMAT(2/,3X,"FREQUENCY(CY/DU)",10X,"SMOOTHED ENERGY DENSITY
     1 SPECTRUM")
      DO 710 J=1,ND-1,10
      JJ=J+9
      IF(JJ.GT.ND-1) JJ=ND-1
      WRITE(6,720) FRE(J),(SUMP( ),I=J,JJ)
  720 FORMAT(5X,E15.6,10E10.3)
  710 CONTINUE
      GO TO 610

  TERMINATE PLOT

  670 CALL PLOT(0.,0.,999)
      CALL GOPLT
   75 IF(LEN.LT.0) GO TO 1520
      IF(IEW.GT.NPTS) GO TO 76
      NEG=(PERCY/100.*FLOAT(NP)-0.5)
      IF(NEG.LE.0) NEG=-1
      ISW=IEW-NEG
      IF(IKI.NE.1) GO TO 848
      WRITE(LU,849)
  849 FORMAT(/,"YOU WANT PROCESSING TO BE AUTOMATIC?")
      READ(LU,81) AUTO
  848 CONTINUE
  753 CONTINUE
      WRITE(LU,1530)
 1530 FORMAT("        ")
   76 WRITE(LU,80)
   80 FORMAT(/2X,"ANY MORE CASES?")
      READ(LU,81)MORE
   81 FORMAT(A2)
      IF(MORE.EQ.2HYE)GO TO 500
      STOP
 1500 WRITE(LU,1510) IERR
 1510 FORMAT(/,"THERE HAS BEEN A READ-FROM-FILE ERROR-IERR=",I4)
 1520 CALL CLOSE(IDCB,IERR)
      STOP
      END
**********************************************************************
EMA(BLOK1,2)
      SUBROUTINE SPECO(NDATA)
======================================================================

   THIS SUBROUTINE IS DESIGNED TO ARRANGE THE NECESSARY
   DATA INTO THE CORRECT FORMAT  FOR PROCESSING BY THE
   FAST FOURIER ROUTINES FFT1 AND FFT2.
```

```
   BEFORE PROCESSING THE DATA PRINTOUT OF ALL THE INPUT
   INFORMATION CAN BE OBTAINED AT A LATER TIME. THE
   ROUTINE ALSO PROVIDES A PRINTOUT FILE OF THE FOLLOWING
      1.  FREQUENCY VERSUS AMPLITUDE.
      2.  FREQUENCY VERSUS ENERGY DENSITY.
      3.  FREQUENCY VERSUS SMOOTHED ENERGY DENSITY.

=================================================================
       COMMON/BLOK1/DATA(4000),FRE(4005),WSP(8000),
      3AMP(4000),ENDEN(4000),DIMDAT(8000)

       DIMENSION IF(30)


       NN=-NDATA
       CALL SETRA(DIMDAT,NDATA,0.)
       WRITE(6,241)
  241  FORMAT(/44X,"---------- INPUT DATA MATRIX ----------",2/)
       WRITE(6,242) (DATA(J),J=1,NDATA)
  242  FORMAT(/2X,10(4X,F6.2))
  CALL SPECTRAL ROUTINE
       CALL FFT2(NN,-1,0,0,IF)
       NF=NDATA/2+1
       DO 2603 J=1,NF
       X=DATA(J)*DATA(J)+DIMDAT(J)*DIMDAT(J)
       WSP(J)=2.0*NF*SQRT(X)
 2603  CONTINUE
       WSP(1)=WSP(1)/2.0

  START OUTPUT TABLE

       NF=NDATA/2+1
       DO 3567 J=1,NF
       WSP(NF+J)=FLOAT((J-1))/FLOAT(NDATA)
 3567  CONTINUE
       WRITE(6,243)
  243  FORMAT(3/,3X,"FREQUENCY(CY/DU)",24X,"AMPLITUDE",2/)
       DO 2478 J=1,NF,10
       JJ=J+9
       IF(JJ.GT.NF) JJ=NF
       WRITE(6,3456) WSP(NF+J),(WSP(I),I=J,JJ)
 3456  FORMAT(5X,E15.6,10E10.3)
 2478  CONTINUE

   THE AMPLITUDE CALCULATION FOLLOWS - THIS IS NOT PLOTTED

       DO 234 I=1,NF
       AMP(I)=20.0*ALOGT(WSP(I))
  234  CONTINUE

    NOW CALCULATE NORMALISED ENERGY DENSITY SPECTRUM
    NORMALISING WITH RESPECT TO VALUE AT ZERO FREQUENCY.

       ENDEN(1)=DATA(1)*DATA(1)+DIMDAT(1)*DIMDAT(1)
       WRITE(6,3457) ENDEN(1)
 3457  FORMAT(" THE NORMALISING ENERGY VALUE = ",E10.3)
       DO 2604 J=2,NF
       Y=(DATA(J)*DATA(J)+DIMDAT(J)*DIMDAT(J))/EN^EN(1)
       ENDEN(J)=10.0*ALOGT(Y)
 2604  CONTINUE
```

```
      ENDEN(1)=0.0
      WRITE(6,244)
  244 FORMAT(3/,3X,"FREQUENCY(CY/DU)",25X,"ENERGY DENSITY",2/)
      DO 290 J=1,NF,10
      JK=J+9
      IF(JK.GT.NF) JK=NF
      WRITE(6,3456) WSP(NF+J),(ENDEN(I),I=J,JK)
  290 CONTINUE
      DO 250 I=1,NF
      FRE(I)=WSP(NF+I)
  250 CONTINUE
      DO 260 J=1,NF
      DATA(J)=ENDEN(J)
  260 CONTINUE

      RETURN
      END
*******************************************************************************
EMA(BLOK1,2)
      SUBROUTINE FFT2(N1,N2,IORI1,IORI2,IB)

===============================================================================

  THIS SUBROUTINE PERFORMS A OND-DIMENSIONAL COMPLEX FFT.
  REA AND IMA ARE ARRAYS CONTAINING RESPECTIVELY THE REAL AND
  IMAGINARY PARTS OF THE INPUT FUNCTION. BOTH HAVE ABS(N1)
  ROWS AND ABS(N2) COLUMNS.
  IORI1,IORI2 IS THE ORIGIN FOR THE TRANSFORMATION.
  WSP IS THE WORKSPACE REQUIRED.
  THE LENGTH OF WSP IS EQUAL TO MAX(4*P,N1*N2), WHERE P IS
  THE LARGEST INTEGER IN THE SET CONTAINING THE PRIME FACTORS
  OF ABS(N1) AND ABS(N2).
  IB IS AN INTEGER ARRAY THAT IS TO CONTAIN THE PRIME FACTORS
  OF ABS(N1) AND ABS(N2). IB IS OF LENGTH NF WHERE
  NF=MAX(NB1,NB2) AND WHERE NB1, NB2 ARE EQUAL TO THE
  NUMBER OF PRIME FACTORS FOR ABS(N1) AND ABS(N2) RESPECTIVELY.
  NB1 AND NB2 ARE BOTH LE.ALOG2(ABS(N1))AND ALOG2(N2)) RESP-
  ECTIVELY. FOR N1 AND N2.LT.0, THE SUBROUTINE EVALUATES
  THE FOURIER COEFFICIENTS OF THE FOURIER ANALYSIS.
  FOR N1 AND N2.GT.0, THE FUNCTION IS SYNTHESISED FROM
  THE FOURIER COEFFICICIENTS.

===============================================================================
      COMMON/BLOK1/REA(4000),FRE(4005),WSP(8000),
     1AMP(4000),ENDEN(4000),IMA(8000)
      DIMENSION IB(1)
      REAL IMA
      NN1=IABS(N1)
      NN2=IABS(N2)

  OBTAIN PRIME FACTORS

      CALL PRFAC(NN1,IB,NB)
      MMAX=IB(NB)
      IF(MMAX.LT.IB(1))MMAX=4
      MAX1=MMAX+1
      MAX2=MAX1+MMAX
      MAX3=MAX2+MMAX
 CALL MAIN SPECTRAL ROUTINE
      DO 1 J2=1,NN2
```

```
      LL=1+(J2-1)*NN1
      CALL FFT1(N1,REA(LL),IMA(LL),IORI1,WSP(1),WSP(MAX1),
     1WSP(MAX2),WSP(MAX3),IB,NB)
    1 CONTINUE
  RETURN IF A ONE DIMENSIONAL ARRAY
      IF(NN2.EQ.1) RETURN
      CALL TRANS(NN1,NN2,REA,WSP)
      CALL TRANS(NN1,NN2,IMA,WSP)
 OBTAIN PRIME FACTORS OF ABS(N2)
      CALL PRFAC(NN2,IB,NB)
      MMAX=IB(NB)
      IF(MMAX.LT.IB(1)) MMAX=4
      MAX1=MMAX+1
      MAX2=MAX1+MMAX
      MAX3=MAX2+MMAX

  CALL SPECTRAL ROUTINE AND SUM WRT 2ND VARIABLE
      DO 2 J1=1,NN1
      LL=1+(J1-1)*NN2
    2 CALL FFT1(N2,REA(LL),IMA(LL),IORI2,WSP(1),WSP(MAX1),
     1WSP(MAX2),WSP(MAX3),IB,NB)
      CALL TRANS(NN2,NN1,REA,WSP)
      CALL TRANS(NN2,NN1,IMA,WSP)
      RETURN
      END
**********************************************************************
      SUBROUTINE FFT1(N,REA,IMA,ORIGIN,RW,RX,IW,IX,B,NB)
======================================================================

  THIS SUBROUTINE IMPLEMENTS THE COOLEY & TUKEY ALGORITHN FOR]
  COMPUTING THE COMPLEX FOURIER SERIES FOR N.LT.0 SUCH THAT     ]
  (NN=ABS(N)), THE SUBROUTINE EVALUATES THE COEFFICIENTS A(K) ]
  OF THE FOURIER ANALYSIS:

          A(K)=1/NN SUM(X(J)*W**(-J*K)) J=0,NN-1

                      FOR K=0,1,2,.............NN-1
  X(J) ARE THE VALUES OF THE SAMPLED FUNCTION.
  FOR N.GT.0 THE FUNCTION X(J), J=0,1,.......NN-1, MAY BE
  SYNTHESISED FROM THE COEFFICIENTS BY:

             X(J)=SUM(A(K)*W**(J*K))  FOR K=0,NN-1

  WHERE IN EACH CASE W=EXP(2*PI*I/NN) AND I=SQRT(-1)

  IN CASE N.LT.0, THEN AT INPUT THE ARRAYS REA AND IMA
  CONTAIN RESPECTIVELY THE REAL AND IMAGINARY PARTS OF THE
  FUNCTION X(J) AT THE SAMPLE POINTS J=0,N-1. AT OUTPUT THE
  SAME ARRAYS CONTAIN THE REAL AND IMAGINARY PARTS OF A(K) FOR
  K=0,N-1, THE FOURIER COEFFICIENTS.
  FOR N.GT.0 THE REVERSE SITUATION APPLIES. THAT IS A(K) IS
  PROVIDED, AND, FOR THE OUTPUT THE SOBROUTINE YIELDS X(J).

  COOLEY & TUKEY HAVE SHOWN THAT IF N=A**P*B**Q*C**R... WHERE
  A,B,C,..... ARE THE PRIME FACTORS OF N ,THEN THE NUMBER OF
  COMPLEX OPERATIONS NEEDED IS N*(P*A+Q*B+C*R...).
  CAPITALISING ON THE BINARY NATURE OF COMPUTERS, A HIGHLY
  EFFICIENT IMPLEMENTATION OF THIS ALGORITHM IS POSSIBLE, IN
  MACHINE CODE, FOR N2**M. THIS SUBROUTINE PLACES NO RESTRCTION
```

ON N BUT IS MOST EFFICIENT FOR THOSE CASES IN WHICH N IS
LARGE COMPARED WITH (P*A+Q*B+C*R.....).
N IS FIRST DECOMPOSED INTO ITS PRIME FACTORS. THE COOLEY &
TUKEY ALGORITHM IS EXECUTED TO OBTAIN THE A(K) OR X(J) BUT
NOT IN YHE CORRECT ORDER. A FINAL ORDERING OF THE ELEMENTS
OF REA AND IMA IS PERFORMED TO ACHIEVE THE CORRECT RESULT.
NOTE THE DESCRIPTION GIVEN ABOVE IS FOR IS FOR ORIGIN=0.

FOR ORIGIN=S THE EXPANSIONS ARE:

A(K)=1/NN SUM(X(J)*W**(-I*K))

FOR J=-S,-S+1,.....N-S-1
AND K=-S,-S+1,.....N-S-1

SIMILAR FORMULA FOLLOW FOR X(J)., WITH THE A(K) AND X(J)
ARRAYS CONTAINED IN REA AND IMA AS APPROPRIATE.
ARRAY B CONTAINS THE PRIME FACTORS OF ABS(N) IN ASCENDING ORDER
NB= NUMBER OF PRIME FACTORS.
DIMENSION OF B IS LE.ALOG2(ABS(N)).
RW,RX,IW,IX, ARE WORKSPACES EACH OF LENGTH P, WHERE P IS THE
LARGEST PRIME FACTOR OF ABS(N).

REFERENCE:
J.W. COOLEY AND J.W. TUKEY,1965. "AN ALGORITHM FOR THE MACHINE
CALCULATION OF COMPLEX FOURIER SERIES" in math.of computing.
vol.10, No 90, pp 297-301.

```
      EMA REA(1),IMA(1),RW(1),RX(1),IW(1),IX(1)


      INTEGER ORIGIN,R,GPSTEP,WG,G,RLESS1,WTK,B(1)
      REAL REA,IMA,IWRZ,IWJ,IWI,ISUM,IW,IX,IM

      NN=IABS(N)
      WTK=NN
      NN=WTK
      NLESS1=NN-1
      TWOPI=6.28318530780
      Z=TWOPI/N
      RLESS1=0
      IF(N.LT.0) TWOPI=-TWOPI
      DO 1 R=1,NB
      GPSTEP=WTK
      KBASE=B(R)
      WTK=WTK/KBASE
      WRZ=TWOPI/KBASE
      RWRZ=COS(WRZ)
      IWRZ=SIN(WRZ)
      DO 2 LL=1,NN,GPSTEP
      G=LL-1
      IF(G.GT.0)GO TO 10
      RWJ=1.0
      IWJ=0.0
      GO TO 11
   10 WG=IREV(G,B,NB,RLESS1)*WTK
      WGZ=WG*Z
      RW(1)=COS(WGZ)
```

```
      RWJ=RW(1)
      IW(1)=SIN(WGZ)
      IWJ=IW(1)
   11 DO 3 J=2,KBASE
      RE=RWRZ*RWJ-IWRZ*IWJ
      IM=RWRZ*IWJ+IWRZ*RWJ
      RW(J)=RE
      RWJ=RW(J)
      IW(J)=IM
      IWJ=IW(J)
    3 CONTINUE
      KEND=G+WTK-1
      IG=G+1
      IKEND=KEND+1
      DO 4 IK=IG,IKEND
      K=IK-1
      JJ=K+ORIGIN
      DO 5 J=1,KBASE
      IF(JJ.GT.NLESS1) JJ=JJ-NN
      RX(J)=REA(JJ+1)
      IX(J)=IMA(JJ+1)
    5 JJ=JJ+WTK
      JJ=K+ORIGIN
      DO 6 I=1,KBASE
      IF(JJ.GT.NLESS1) JJ=JJ-NN
      RWI=RW(I)
      IWI=IW(I)
      RSUM=RX(KBASE)
      ISUM=IX(KBASE)
      KKBASE=KBASE-1
      J=KKBASE
   12 IF(I.EQ.1.AND.G.EQ.0) GO TO 13
      RE=RSUM*RWI-ISUM*IWI
      IM=RSUM*IWI+ISUM*RWI
      RSUM=RE+RX(J)
      ISUM=IM+IX(J)
      GO TO 14
   13 RSUM=RSUM+RX(J)
      ISUM=ISUM+IX(J)
   14 CONTINUE
      J=J-1
      IF(J.GE.1) GO TO 12
      REA(JJ+1)=RSUM
      IMA(JJ+1)=ISUM
      JJ=JJ+WTK
    6 CONTINUE
    4 CONTINUE
    2 CONTINUE
      RLESS1=R
    1 CONTINUE
      IF(N.LT.0) NLESS1=-NLESS1
      CALL JKPER(REA,IMA,NLESS1,ORIGIN,B,NB)
      RETURN
      END
**********************************************************************
      SUBROUTINE SETRA(RARA,NW,CON)

======================================================================

  THIS SUBROUTINE SETS THE REAL ARRAY VALUES. TO DO THIS
```

```
  IT FILLS THE FIRST NW WORDS OF REAL ARRAY RARA WITH
  THE VALUES IN CON.
=======================================================================

      EMA RARA(NW)
      IF(NW.LE.0) GO TO 20
      DO 10 N=1,NW
      RARA(N)=CON
   10 CONTINUE
   20 CONTINUE
      RETURN
      END
***********************************************************************
      SUBROUTINE JKPER(REA,IMA,NA,ORIGIN,B,NB)
=======================================================================
  REA,IMA ARE THE REAL AND IMAGINARY PARTS OF A COMPLEX
  ARRAY A. FOR NA.GT.0 THE SUBROUTINE REARRANGES THE
  ELEMENTS SO THAT A(J)=A(K) WHERE K IS THE J TO K
  TRANSFORMATION DEFINED BY FUNCTION JTOK. THE RADIX
  BASE VECTOR FOR THAT PROCEURE IS ARRAY B(NB).
  ELEMENTS A(K) ARE EXHANGED WITH A(J) IN THE ORDER
  J=0,1,2,3,.....NA PROVIDED K.GT.J.
  IF K.LT.J ELEMENT A(K) IS NOW ELSEWHERE FOLLOWING
  PREVIOUS EXCHANGES, IN THIS CASE REPEATED TRANSFORM-
  ATIONS OF K ARE MADE UNTIL K.GT.J.

  FOR NA.LT.0. THE REARRANGEMENT IS A(J)=A(K)/(ABS(NA)+1).

  THE ABOVE DESCRIPTION APPLIES FOR ORIGIN = 0.
  FOR ORIGIN .NE. 0 THE REARRANGEMENT IS A(J)=A(K), WHERE
  K=JTOK(J-ORIGIN)+ORIGIN, THESE SUMS AND DIFFERENCES
  BEING MODULO(ABS(NA)+1).

=======================================================================
      EMA REA(1),IMA(1)
      INTEGER B(1),ORIGIN
      REAL REA,IMA,NN,ITEMP
      IF(NA.GT.0) GO TO 1
      N=1-NA
      NJ=-NA
      GO TO 2
    1 N=1+NA
      NJ=NA
    2 NN=FLOAT(N)
      NJ1=NJ+1
      DO 3 IJ=1,NJ1
      J=IJ-1
      K=J
   10 K=K-ORIGIN
      IF(K.LT.0) K=K+N
      LL=JTOK(K,B,NB)+ORIGIN
      K=LL
      IF(K.GT.NJ) K=K-N
      IF(K.LT.J) GO TO 10
      IF(J.EQ.K) GO TO 5
      RTEMP=REA(K+1)
      ITEMP=IMA(K+1)
      IF(NA.GE.0)GO TO 6
      RTEMP=RTEMP/NN
```

```
      ITEMP=ITEMP/NN
    6 REA(K+1)=REA(J+1)
      REA(J+1)=RTEMP
      IMA(K+1)=IMA(J+1)
      IMA(J+1)=ITEMP
      GO TO 3
    5 IF(NA.GE.0) GO TO 3
      REA(J+1)=REA(J+1)/NN
      IMA(J+1)=IMA(J+1)/NN
    3 CONTINUE
      RETURN

  REARRANGE REA,IMA TO CORRECT ORDER BY CALLING THE
  SUBROUTINE JKPER.

      END
*********************************************************************
      SUBROUTINE PRFAC(N,F,NF)
=====================================================================

  THIS SUBROUTINE DECOMPOSES N INTO FACTORS OF FOUR (AND
  A POSSIBLE SINGLE FACTOR OF TWO) AND ITS ODD PRIME
  FACTORS. THE FACTORS OCCUPY F(1) TO F(NF) OF ARRAY F
  WHERE NF WILL BE SUFFICIENT IF NF.GE.ALOG2(N).

=====================================================================

      INTEGER F(1),Q,P,D
      I=0
      NN=N
      P=4
      D=-2
      Q=N
   10 IF(Q.LT.P) GO TO 1
      Q=N/P
      IF(N.EQ.Q*P) GO TO 2
      P=P+D
      J=D
      D=2
      IF(J.LT.1)D=1
      GO TO 3
    2 I=I+1
      F(I)=P
      N=Q
    3 GO TO 10
    1 IF(N.EQ.1) GO TO 4
      I=I+1
      F(I)=N
    4 NF=I
      N=NN
      RETURN
      END
*********************************************************************

      SUBROUTINE TRANS(N1,N2,A,WSP)
=====================================================================

  THIS SUBROUTINE TRANSPOSES AN N1*N2 MATRIX CALLED "A"
  INTO AN N2*N1 MATRIX. THE TRANSPOSED MATRIX OCCUPIES
```

THE SAME SPACE AS THE ORIGINAL MATRIX EXCEPT THAT
THE ROWS AND COLUMNS HAVE BEEN TRANSPOSED. THE MATRIX
"A" , TO BE TRANSPOSED HAS TO BE PLACED INTO THE
WORKSPACE "WSP". WSP AND MATRIX A HAVE LENGTH N1*N2.

```
      EMA A(5000),WSP(10000)
      L=N1*N2
      DO 1 J=1,L
      WSP(J)=A(J)
    1 CONTINUE
      DO 2 J1=1,N1
      L=(J1-1)*N2
      DO 2 J2=1,N2
      K=J2+L
      J=J1+(J2-1)*N1
      A(K)=WSP(J)
    2 CONTINUE
      RETURN
      END
      FUNCTION IREV(M,B,N,R)
```

THIS FUNCTION REVERSES THE ORDER OF THE PRIMES.

```
      INTEGER B(1),R,SUM,Q,D
      MM=M
      SUM=0
      I=N
      IF(I.LT.2) GO TO 4
    3 D=B(I)
      Q=M/D
      J=M-Q*D
      IF(I.LE.R) SUM=SUM*D+J
      M=Q
      I=I-1
      IF(I.GE.2) GO TO 3
    4 IREV=SUM*B(1)+M
      M=MM
      RETURN
      END
*************************************************
      FUNCTION JTOK(M,B,N)
```

THIS FUNCTION PERFORMS AN INTEGER TRANSFORMATION
ACCORDING TO THE FOLLOWING RULES:

CONSIDER M EXPRESSED AS (J1,J2,.....Jn) WHERE
J1 IS THE LEAST SIGNIFICANT DIGIT AND Jn IS THE
MOST SIGNIFICANT DIGIT OF THE NDIGIT MIXED
RADIX REPRESENTATION FOR J1 ASSOCIATED WITH
BASE(I).

THE TRANSFORMATION IS APPLIED TO M TO REVERSE

```
      THE ORDER OF SIGNIFICANCE OF THE DIGITS, THAT IS
      Jn IS NOW REGARDED AS THE LEAST SIGNIFICANT
      DIGIT , RETURNING THE RESULT INTO ARRAY JTOK. M
      IS UNCHANGED.

C=====================================================================
      INTEGER B(1),SUM,D,Q
      SUM=0
      NLESS1=N-1
      MM=M
      IF(NLESS1.LE.0) GO TO 2
      DO 1 I=1,NLESS1
      D=B(I)
      Q=M/D
      J=M-Q*D
      SUM=SUM*D+J
    1 M=Q
    2 JTOK=SUM*B(N)+M
      M=MM
      RETURN
      END
      SUBROUTINE LINE(X,Y,N)
      EMA Y(4000)
      DIMENSION X(4005)
      IP=3
      DO 10 I=1,N
      XL=(X(I)-X(N+1))/X(N+2)
      YL=(Y(I)-Y(N+1))/Y(N+2)
      CALL PLOT(XL,YL,IP)
      IP=2
   10 CONTINUE
      RETURN
      END
      END$
```

```
TN4,B,L,Q,C
```

```
**********************************
*                                *
*                                *
*                                *
*          PROGRAM   WERN        *
*                                *
*                                *
*                                *
**********************************
```

```
EMA(BLOK1,2)
      PROGRAM WERN(3,200)

*******AUTOM*****************************************************

     TITLE      : WERN - PERFORMS A CALCULATION TO PRODUCE ESTIMATES
                  OF DEPTH TO MAGNETIC BASEMENT.

     REF.       : WERNER, S. -1953- INTERPRETATION OF MAGNETIC ANOMALIES
                AT SHEET-LIKE BODIES. SVERIGES GEOLOGISKA UNDERSOK.
                SER. C. C.  ARSBOK 43, n06.

     AUTHOR     : (a) HSU & TILBURY - SEE BELOW
                : (b) PETER R GIDLEY, Bureau of Mineral Resources, CANBERRA,ACT.

     PURPOSE    : THIS PROGRAM CAN BRING IN PRESCRIBED SECTIONS OF EQUI-SPACED
                  MAGNETIC DATA AND PERFORMS QUANTITATIVE ANALYSIS ON THE DATA.
                  THE ANALYSIS IS BASED ON A TECHNIQUE CALLED WERNER DECONVOLUTION.
                  THE MODEL ADOPTED IN THIS PROGRAM IS
                  DOUBLE SOURCE/ QUADRATIC INTERFERENCE.

     USE        : THE PROGRAM WAS INITIALLY ADOPTED FROM A BMR RECORD (1977/50)
                  BY HSU & TILBURY :
       A magnetic interpretation program based on werner deconvolution.
     WERNER DECONVOLUTION IS A METHOD OF PRODUCING DIRECT INTERPRETATION
     AND ACTUALLY DETERMINES DEPTHS ,SUSCEPTIBILITIES AND DIPS OF THE
     MAGNETISATION VECTOR.
     TWO MODELS ARE AVAILABLE:
        1. THIN SHEET MODEL
        2. INTERFACE MODEL
     DETAILS AND A MORE COMPLETE DESCRIPTION OF THE THEORY OF THE
     PROGRAM ARE AVAILABLE IN THE ABOVE REFERENCE OR IN:
       Understanding and use of werner deconvolution in aeromagnetic
       interpretation. 1975. J.L.FRIEDBERG, Aero Service Production.

     THE PROGRAM AS ESTABLISHED HERE USES DOUBLE PRECISION THROUGHOUT
     THIS WAS DONE TO ENSURE THE MATRIX INVERSIONS AND SOLUTION
     CALCULATIONS DO NOT COMPOUND THEIR ERRORS IN THE 8-BIT WORD
```

CAPABILITY OF THE H-P COMPUTER. SUCH PRECISION IS NOT REQUIRED ON A LARGER MACHINE. DATA FILES ARE READ IN FROM SUBROUTINE INPUD AND THEN OPERATED ON IN PROGRAM WERNER. A NUMBER OF INPUT CONTROLS ARE REQUIRED FOR OPERATION;

1. NOMINATE THE MODEL TYPE REQUIRED (SEE ABOVE)
2. START AND STOP LEVELS OF INTERPRETATION (SEE HSU & TILBURY)
3. SCANNING STEP .
4. A PLOT OPTION AS DESCRIBED IN THE CALL

NOTE THE ORIGINAL REQUEST (FROM INPUD) OF THE DATA SPACING IS TO BE INPUT IN METRES, HOWEVER THE FINAL PLOT AND PRINTOUT OF DEPTH ESTIMATES IS IN KILOMETRES.

THE PROGRAM WAS ADAPTED FOR THE H-P BY PETER R. GIDLEY 18/2/81.

```
**********************************************************
      COMMON/BLOK1/DXM(2500),DM(1000),TM(2500)
      DOUBLE PRECISION TM,DM,DXM,TS,X,DT,T,C,Y,X,W,V,U,P,BKG,BG1,BG2,A
      DOUBLE PRECISION DET,BB,S,SQ,STT,SHS,STS,STH,DTS,EPC,DPT,TH,XYS
      DOUBLE PRECISION AR,BK,TB,XT,YT,WX,WY,CS,RS,YS,XS,WXS,WYS,FT
      DOUBLE PRECISION ZS,ZX,ZL,ZY,TF,TAA,SMALL,PA,DLC
      DIMENSION POS(502),DEP(502),SS(502),AS(502),DATA(1002)
      DIMENSION A(11,11),C(11,1),T(11)
      DIMENSION P(400),S(10)
      DIMENSION TS(15),B(11,1)
      DIMENSION F(11),FT(10)
      DIMENSION IDCB(144),IBUF(40),NAME(3),ISEG(3),ISGNO(3)
      DIMENSION LU(5),IPS(3),IB(2),IFORM(40)

      CALL RMPAR(LU)

  OBTAIN DATA TO BE PROCESSED FROM SUBROUTINE 'INPUD'.

      CALL INPUD(TM,IDCB,IBUF,NAME,ISEG,ISGNO,LU,IPS,IB,IFORM,NPT
     *,IST,IEND,TOT,DX)

   IT IS NECESSARY TO ENSURE THAT THE NUMBER OF DATA POINTS DOES
   NOT EXCEED THE DIMENSIONED ARRAYS.

      NPT=IEND-IST+1
      IF(NPT.LT.1000) GO TO 61
  116 WRITE(LU,111) NPT
  111 FORMAT(/,"THE NUMBER OF DATA POINTS READ WAS ",I5,/,
     1"SO IT WILL BE NECESSARY TO REDUCE THE NUMBER OF",/,
     2"POINTS TO BE PROCESSED SINCE THE PROGRAM CAN ONLY",/,
     3"OPERATE ON A MAXIMUM OF 1000 POINTS. THEREFORE",/,
     4"*****  INPUT THE NUMBER OF POINTS TO BE MISSED")
      READ(LU,*) NMIS
      K=1
      DO 113 I=1,NPT,NMIS
      TM(K)=TM(I)
      K=K+1
  113 CONTINUE
      NPT=K
      DX=DX*NMIS
      DX1=DX/1000.0
      WRITE(LU,114) NPT,DX,DX1
  114 FORMAT(/,"THE NUMBER OF DATA POINTS IS NOW =","I5,/,
     1"THE SPACING INTERVAL IS =",F6.2," METRES (",F6.2," KILOMETRES)")
      IF(NPT.GT.1000) GO TO 116
```

```
 61 LOS=0
    NBK=0
    PA=57.29578D0
    SMALL=0.000001D0
    NT=11
    NTH=6
    NOD=5*(NT-1)+1
    NF=1
    NL=NT
    NN=NT
    READ IN VARIOUS PARAMETERS
    WRITE(LU,1)
  1 FORMAT(/,"TWO MODELS ARE AVAILABLE FOR PROCESSING. TYPE IN-",/,
   *"       1    THIN SHEETS - ORIGINAL DATA ONLY USED IN ANALYSIS.",/,
   *"       2    INTERFACES - HORIZONTAL DERIVATIVES USED.")
    READ(LU,*) NTYPE
    WRITE(LU,2)
  2 FORMAT(/,"INPUT THE START & STOP LEVELS OF INTERPRETATION")
    READ(LU,*) LVF,LVS
    WRITE(LU,3)
  3 FORMAT(/,"WHAT SCANNING STEP IS REQUIRED?")
    READ(LU,*) ISTEP
    WRITE(LU,900)
900 FORMAT(/," DO YOU WANT PLOTS OF DEPTH ESTIMATES?",/,
   1"      TYPE IN  0   IF ONLY WANT A SINGLE PLOT OF ALL RESULTS,",/,
   2"      TYPE IN  1   IF WANT PLOTS FOR EACH LEVEL'S RESULTS,",/,
   3"      TYPE IN  2   IF NO PLOTS ARE REQUIRED.")
    READ(LU,*) MOKE
  4 FORMAT(///,"           ********************************** ",/,
   1"                  I AM PROCESSING YOUR REQUEST",/,"
   2 PLEASE WAIT",/,"           **********************************")
    WRITE(LU,901)
901 FORMAT("DO YOU WANT A PRINT LISTING OF RESULTS AUTOMATICALLY?")
    READ(LU,301) IPRNT

    DT=1.
    NSL=1
    NSU=NPT
    DO 121 II=1,NPT
    DATA(II)=TM(II)
121 CONTINUE
    IF(ISTEP.LE.0) ISTEP=2
    IF(NTYPE.EQ.1) GO TO 100

    CALL HTDER TO COMPUTE HORIZONTAL DERIVATIVES.

    CALL HTDER(TM,DXM,DT,NSL,NPT)

    CHECK THE SIZE OF SCANNING STEP.


    PUT HORIZONTAL DERIVATIVES IN ARRAY DM.

    DO 90 LM=1,NPT
    DM(LM)=DXM(LM)
 90 CONTINUE
    GO TO 120

    PUT ORIGINAL DATA IN ARRAY DM.
```

```
100 DO 110 LM=1,NPT
    DM(LM)=TM(LM)-TOT
110 CONTINUE
120 CONTINUE
    IF(IPRNT.EQ.2HYE) WRITE(16,128) ISEG
128 FORMAT(" THE DATA AND RESULTS FOLLOWING ARE FROM LINE  ",3A2,/,
   1" ***   ***   ***   ***   ***   ***   ***   *** ")
    IF(IPRNT.EQ.2HYE.AND.ITYPE.EQ.1) WRITE(16,126)
    IF(IPRNT.EQ.2HYE.AND.ITYPE.EQ.2) WRITE(16,127)
126 FORMAT(" *** MODEL ASSUMED IS A THIN SHEET *** ",/,
   *" *** -ORIGINAL DATA ONLY USED IN ANALYSIS *** ")
127 FORMAT(" *** MODEL ASSUMED WERE INTERFACES *** ",/,
   *" *** -HORIZONTAL DERIVATIVES USED *** ")
    WRITE(LU,115)
115 FORMAT(/,"DO YOU WANT TO PRINT DATA?",
   2/,"----IF YES ,TYPE IN NUMBER OF VALUES REQUIRED,",
   3/,"---- IF NO  ,TYPE IN NO.")
    READ(LU,125) IB
125 FORMAT(2A2)
    IF(IB.EQ.2HNO) GO TO 118
    CALL CODE
    READ(IB,*) NUT
    WRITE(LU,119) (DM(I),I=1,NUT)
    WRITE(6,119) (DM(I),I=1,NUT)
119 FORMAT(11F10.4)
118 LVL=LVF-1

    SET LEVEL OF ANALYSIS,LVL.

130 LVL=LVL+1

    IF NO MORE LEVEL, GO TO PRINT OUT AND PLOT RESULTS, AND THEN


    GO TO NEXT LINE

    IF(LVL.GT.LVS) GO TO 460
    WRITE(LU,4)
    LV=LVL-1

    SET SAMPLE INTERVAL.

    INV=2**LV
    DLC=LV*INV/LVL
    NOC=(INV-1)*6+1
    NC=(INV-1)*3+1
    NCS=NC
    NLT=NPT-NC+1
    ZL=INV
    IF(LVL.EQ.1) ZL=0.D0

  CALL UPWARD CONTINUATION

    CALL UPCON(NOC,DM,TM,DT,ZL,NC,NLT)
140 NS=0
    JN=-ISTEP

    SHIFT SAMPLE ARRAY ACROSS PROFILE.
```

```
150 JN=JN+ISTEP
    NX=NCS+JN
    NL=NX+INV*10
    IF(NL.GT.NLT) GO TO 410

    SAMPLE PROFILE.

    DO 160 J=1,11
    JJ=NX+(J-1)*INV
    T(J)=TM(JJ)
160 CONTINUE

    GENERATE ELEMENTS IN LINEAR EQUATION MATRIX.

    NN=NT
    NF=1
    NL=NT
    DO 180 N=1,11
    X=N-NTH
    TS(4)=T(N)
    TS(3)=T(N)*X
    TS(2)=TS(3)*X
    TS(1)=TS(2)*X
    TS(5)=X*X
    TS(6)=X
    TS(7)=1.
    TS(8)=X**3
    TS(9)=X**4
    TS(10)=TS(9)*X
    TS(11)=TS(10)*X
    C(N,1)=TS(1)*X
    DO 170 J=1,11
    A(N,J)=TS(J)
170 CONTINUE
180 CONTINUE
    N=11
    L=1

    CALL MATIN--MATRIX INVERSION ROUTINE

    CALL MATIN(A,N,C,L,DET,IRR)
190 FORMAT(8F12.4)
    IF(IRR.EQ.1) GO TO 150

    RETURN WITH SOLUTION IN ARRAY A(11,11)


    COMPUTE THE COEFFICENTS OF QUADRATIC INTERFERENCE.

    BG2=-C(11)
    BG1=-C(10)-C(11)*C(1)
    BKG=-C(9)+BG1*C(1)-C(11)*C(2)

    SOLVE UNKNOWNS FOR ONE OR TWO SOURCES--
    POSITION,
    DEPTH,

220 C(1)=C(1)*0.5D0
    C(2)=-C(2)
    C(3)= C(3)*0.5D0
```

```
      C(4)=-C(4)
      DO 230 I=21,81
      X=0.1*(I-NOD)
      Y=C(1)-X
      W=C(2)*Y-4.D0*X*Y*Y-C(3)
      V=C(2)*X-4.D0*X*X*Y-C(3)
      U=C(4)*(X-Y)*(X-Y)
      P(I)=U+V*W
  230 CONTINUE
      MR=0
      DO 250 I=21,81
      IF(P(I).EQ.0.0) GO TO 240
      FR=P(I+1)*P(I)
      IF(FR.GE.0.0) GO TO 250
  240 MR=MR+1
      S(MR)=0.1*(I-NOD)-0.1D0*P(I)/(P(I+1)-P(I))
  250 CONTINUE
      IF(MR.EQ.0) GO TO 150
      DO 400 M=1,MR
      X=S(M)
      Y=C(1)-X
      ZX=999999.D0
      WXS=999999.D0
      ZY=999999.D0
      WYS=999999.D0
      XYS=(X-Y)*(X-Y)
      IF(XYS.LT.SMALL) GO TO 260
      WX=(C(2)*X-4.D0*X*X*Y-C(3))/(X-Y)
      WY=(C(2)*Y-4.D0*X*Y*Y-C(3))/(Y-X)
      GO TO 270
  260 CS=C(2)-4.D0*X*Y
      RS=CS*CS-4.D0*C(4)
      IF(RS.LT.SMALL) GO TO 290
      WX=(CS+DSQRT(RS))/2.D0
      WY=(CS-DSQRT(RS))/2.D0
  270 YS=Y*Y
      XS=X*X
      IF(WX.LT.XS) GO TO 280
      WXS=DSQRT(WX)
      ZS=WX-XS
      ZX=DSQRT(ZS)
  280 IF(WY.LT.YS) GO TO 290
      WYS=DSQRT(WY)
      ZS=WY-YS
      ZY=DSQRT(ZS)
  290 CONTINUE

      INTENSITY,
      ANGLE OF MAGNETIZATION.

      DO 310 I=1,11
      C(I,1)=0.0
      DO 315 J=1,11
      A(I,J) =0.
  315 CONTINUE
  310 CONTINUE
      MF=NF
      ML=NL
      LF=1
      LL=4
```

```
      IF(ZY.EQ.999999.D0) LF=1
      IF(ZY.EQ.999999.D0) LL=2
      IF(ZX.EQ.999999.D0) LF=3
      IF(ZX.EQ.999999.D0) LL=4
      DO 330 I=1,11
      XN=I-NTH
      XA=XN-X
      XB=XN-Y
      DXO=XA*XA+ZX*ZX
      DY=XB*XB+ZY*ZY
      TS(1)=ZX/DXO
      TS(2)=XA/LXO
      TS(3)=ZY/DY
      TS(4)=XB/DY
      TF=T(I)+BKG+BG1*XN+BG2*XN*XN
      DO 335 J=LF,LL
      C(J,1)=C(J,1)+TF*TS(J)
      DO  320 K=LF,LL
      A(K,J)=A(K,J)+TS(J)*TS(K)
  320 CONTINUE
  335 CONTINUE
  330 CONTINUE
      NN=LL-LF+1
      DO 340 I=1,NN
      DO 345 J=1,NN
      A(J,I)=A(J+LF-1,I+LF-1)
  345 CONTINUE
  340 CONTINUE
      CALL MATIN(A,NN,BB,0,DET,IRR)
      IF(IRR.EQ.1)GO TO 150
      DO 350 L=LF,LL
      DO 355 K=LF,LL
      A(K,L)=A(K-LF+1,L-LF+1)
  355 CONTINUE
  350 CONTINUE
      DO 360 L=1,4
      FT(L)=0.
  360 CONTINUE
      DO 370 L=LF,LL
      DO 375 K=LF,LL
      FT(L)=FT(L)+A(K,L)*C(K,1)
  375 CONTINUE
  370 CONTINUE

      CHECK SOLUTIONS. IF UNFIT, REJECT THEM.

      STS=0.D0
      STH=0.D0
      DO 380 N=1,11
      XN=N-NTH
      XA=XN-X
      XB=XN-Y
      TAA=(FT(4)*XB+FT(3)*ZY)/(XB*XB+ZY*ZY)
      TH=(FT(2)*XA+FT(1)*ZX)/(XA*XA+ZX*ZX)+TAA
      BK=BKG+BG1*XN+BG2*XN*XN
      TDT=T(N)-TH+BK
      TB=TDT+TH
      STH=STH+TB*TB
      DTS=TDT*TDT
      STS=STS+DTS
```

```
380 CONTINUE
    STT=DSQRT(STS)
    SHS=DSQRT(STH)
    EPC=STT/SHS*100.D0
    IF(EPC.GT.10.D0) GO TO 150

    CHECK RANGE OF DEPTH AND POSITION, IF OUTSIDE,REJECT IT.

    XT=(3.D0-X)*(X+3.D0)
    YT=(3.D0-Y)*(Y+3.D0)
    IF(XT.LT.0.D0.OR.ZX.GT.4.5D0) GO TO 390

    CONVERT POSITION, DEPTH, INTENSITY AND ANGLE INTO TRUE VALUES
    FOR SOURCE 1.

    DPT=ZX*INV-ZL
    IF(DPT.LT.DLC) GO TO 390
    NS=NS+1
    DEP(NS)=DPT*DX/1000.0
    POS(NS)=(NX+(X+5.0)*INV)*DX/1000.0
    SQ=FT(1)*FT(1)+FT(2)*FT(2)
    SS(NS)=DSQRT(SQ)*INV
    IF(FT(1).EQ.0.D0) FT(1)=0.000001D0
    AR=FT(2)/FT(1)
    AS(NS)=DATAN(AR)*PA
390 IF(YT.LT.0.D0.OR.ZY.GT.4.5D0) GO TO 400

    FOR SOURCE 2.

    DPT=ZY*INV-ZL
    IF(DPT.LT.DLC) GO TO 400
    NS=NS+1
    DEP(NS)=DPT*DX/1000.0
    POS(NS)=(NX+(Y+5.0)*INV)*DX/1000.0
    SQ=FT(3)*FT(3)+FT(4)*FT(4)
    SS(NS)=DSQRT(SQ)*INV
    IF(FT(3).EQ.0.0D0) FT(3)=0.000001D0
    AR=FT(4)/FT(3)
    AS(NS)=DATAN(AR)*PA
400 CONTINUE

    SAVE ESTIMATES IN ARRAYS POS,DEP,AIS,SIS.

    GO TO 150
410 WRITE(LU,420) LVL,MC,JN
420 FORMAT(9X," LEVEL",I5,10X," OFFSET",I5,10X,"NO OF SECTIONS",I5//)
430 FORMAT(10X,4F15.4)
    WRITE(LU,431) NS
    IF(IPRNT.EQ.2HYE) WRITE(16,431) NS
431 FORMAT(10X," NS=",I5)
    DUC=4.5*INV
    WRITE(LU,432) LVL,INV,DLC,DUC
    IF(IPRNT.EQ.2HYE) WRITE(16,432) LVL,INV,DLC,DUC
432 FORMAT("INTERP LEVEL=",I3," SAMPLE INT=",I3," LOW RANGE=",I5,
   *" UPPER RANGE=",I5)
    IF(NS.LE.0) GO TO 450

   SORT THE DATA ARRAYS ACCORDING TO DISTANCE.

    CALL SORT(POS,DEP,AS,SS,NS)
```

```
      WRITE(LU,440)
      IF(IPRNT.EQ.2HYE) WRITE(16,440)
  440 FORMAT(16X,"POSITION",10X,"DEPTH",10X,"ANGLE",10X,"INTENSITY"/)

      PRINT ESTIMATED SOURCES

      WRITE(LU,430)(POS(L),DEP(L),AS(L),SS(L),L=1,NS)
      IF(IPRNT.EQ.2HYE) WRITE(16,430) (POS(L),DEP(L),AS(L),SS(L),L=1,NS)
  450 CONTINUE

   CALL PLOTTING SUBROUTINE

      IF(MOKE.EQ.2) GO TO 925
      CALL WPLOT(POS,DEP,AS,SS,LU,NS,DATA,NPT,DX,MOKE,SP1,SP2,
     *ISEG,NTYPE)
  925 GO TO 130
  460 CONTINUE
  470 FORMAT(12X,I5,2F15.2,10X,I5,2F15.2)
  480 CONTINUE
  301 FORMAT(3A2)
      STOP
      END
      SUBROUTINE HTDER(TM,DXM,DT,NSL,NSU)

**********************************************************
    THIS SUBROUTINE COMPUTES HORIZONTAL DERIVATIVES
    FOR EQUI - SPACED DATA.
        TM - ARRAY ON WHICH HORIZONTAL DERIVATIVES ARE CALCULATED
        DXM- HORIZONTAL DERIVATIVE ARRAY
        DT - DATA UNIT (ALWAYS EQUAL TO 1.0)
        NSL- NUMBER OF STARTING DATA POINT
        NSU- NUMBER OF END DATA POINT
**********************************************************

      EMA TM(2500),DXM(2500)
      DOUBLE PRECISION TM,DXM,DT
      NSN=NSL+2
      NSS=NSU-2
      COMPUTE HORIZONTAL DERIVATIVES
      FOR FIRST POINT.
      DXM(NSL)=(TM(NSL+1)-TM(NSL))/DT
      FOR SECOND POINT.
      DXM(NSL+1) =(-TM(NSL)/3.-TM(NSL+1)/2.+TM(NSL+2)-TM(NSL+3)/6.)/DT
      FOR SECOND LAST POINT.
      DXM(NSU-1)=(-TM(NSU-3)/3.-TM(NSU-2)/2.+TM(NSU-1)-TM(NSU)/6.)/DT
      FOR LAST POINT.
      DXM(NSU)=(TM(NSU)-TM(NSU-1))/DT
      DO 10  I=NSN,NSS
      FOR ALL INTERNAL POINTS.
      DX=TM(I-2)/6.-4.*TM(I-1)/3.-TM(I+2)/6.+4.*TM(I+1)/3.
      DXM(I)=DX/(2.*DT)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE MATIN (A,N,B,L,D,IRROR)
      DOUBLE PRECISION A,B,AMAX,D
      DIMENSION A(11,11),B(11,1),IPIV(11),IND(11,2)

**********************************************************
```

```
      A IS AN NXN MATRIX TO BE INVERTED,OR CONTAINING EQUATION COEFFS
      B IS AN NXM RHS MATRIX FOR EQUATIONS
      IF L=0,INVERSE ONLY GIVEN.L POSITIVE,SOLUTIONS ONLY.L NEGATIVE
       BOTH.    M=ABS(L).
      D CONTAINS THE DETERMINANT OF THE A MATRIX ON EXIT
      A IS REPLACED BY THE INVERSE ,B BY THE SOLUTIONS.
      METHOD OF GAUSS-JORDON PIVOTAL ELIMINATION

**********************************************************

      M=IABS(L)
      D=1.0
      DO 10 I=1,N
      IPIV(I)=0
   10 CONTINUE
      DO 220 I=1,N
      AMAX=0.0

      SEARCH SUB-MATRIX FOR LARGEST ELEMENT AS PIVOT

      DO  60 J=1,N
      IF(IPIV(J)) 70,20,60
   20 DO  50 K=1,N
      IF(IPIV(K)-1) 30,50,70

      THIS ROW COLUMN HAS NOT BEEN A PIVOT

   30 IF(DABS (A(J,K))-AMAX) 50,50,40
   40 IROW=J
      ICOL=K
      AMAX=DABS (A(J,K))
   50 CONTINUE
   60 CONTINUE

      PIVOT FOUND
      PIVOT FOUND

      IPIV(ICOL)=IPIV(ICOL)+1
      IF(AMAX-1.0E-37) 70,70,80

      MATRIX SINGULAR,ERROR RETURN

   70 IRROR=1
      RETURN
   80 IF(IROW-ICOL) 90,130,90

      MAKE PIVOT A DIAGONAL ELEMENT BY ROW INTERCHANGE.

   90 D=-D
      DO 100 K=1,N
      AMAX=A(IROW,K)
      A(IROW,K)=A(ICOL,K)
      A(ICOL,K)=AMAX
  100 CONTINUE
      IF(M) 130,130,110
  110 DO 120 K=1,M
      AMAX=B(IROW,K)
      B(IROW,K)=B(ICOL,K)
      B(ICOL,K)=AMAX
  120 CONTINUE
```

```
130  IND(I,1)=IROW
     IND(I,2)=ICOL
     AMAX=A(ICOL,ICOL)
     D=D*AMAX
     A(ICOL,ICOL)=1.0

     DIVIDE PIVOT ROW BY PIVOT

     DO 140 K=1,N
     A(ICOL,K)=A(ICOL,K)/AMAX
140  CONTINUE
     IF(M) 170,170,150
150  DO 160 K=1,M
     B(ICOL,K)=B(ICOL,K)/AMAX
160  CONTINUE

     REDUCE NON-PIVOT ROWS

170  DO 220 J=1,N
     IF(J-ICOL) 180,220,180
180  AMAX=A(J,ICOL)
     A(J,ICOL)=0.0
     DO 190  K=1,N
     A(J,K)=A(J,K)-A(ICOL,K)*AMAX
190  CONTINUE
     IF(M)  220,220,200
200  DO  210 K=1,M
     B(J,K)=B(J,K)-B(ICOL,K)*AMAX
210  CONTINUE
220  CONTINUE

     AFTER N PIVOTAL CONDENSATIONS,SOLUTIONS LIE IN B MATRIX

     IF(L) 230,230,270

     FOR INVERSE OF A, INTERCHANGE COLUMNS

230  DO 260 I=1,N
     J=N+1-I
     IF(IND(J,1)-IND(J,2)) 240,260,240
240  IROW=IND(J,1)
     ICOL=IND(J,2)
     DO 250  K=1,N
     AMAX=A(K,IROW)
     A(K,IROW)=A(K,ICOL)
     A(K,ICOL)=AMAX
250  CONTINUE
260  CONTINUE
270  IRROR=0
     RETURN
     END
     SUBROUTINE UPCON(NOC,DM,TM,DT,ZL,NSL,NSU)

**********************************************************

     THIS SUBROUTINE UPWARD CONTINUES THE MAGNETIC DATA.

     NOC - LENGTH OF UPWARD CONTINUATION FILTER
     TM  - UPWARD CONTINUATION ARRAY
     DM  - INPUT DATA ARRAY
```

```
C     DT   - DATA UNIT (SET =1.0)
C     ZL   - HEIGHT OF UPWARD CONTINUATION
C     NSL  - NUMBER OF STARTING DATA POINT
C     NSU  - NUMBER OF END DATA POINT
C
C*************************************************************
      EMA DM(1000),TM(2500)
      DOUBLE PRECISION DM,TM,DT,DL,CA,SUM,XNORM,SMALL,PI,ZL
      DIMENSION CA(300)
      DATA SMALL/1.0D-8/
      DATA PI/3.1415927D0/
      NC=(NOC-1)/2
      XNORM=0.0D0
C
C    IF NO UPWARD CONTINUATION, SET COEFFS TO 1.0
C
      IF(ZL.GT.SMALL.AND.NOC.GT.1) GO TO 10
      CA(1)=1.D0
      XNORM=1.D0
      GO TO 30
C
C     COMPUTE COEFFS OF UPWARD CONTINUATION
C
   10 DO 20 II=1,NOC
      I=II-NC
      DL=DT*(I-1)
      CA(II)=ZL/(PI*(ZL*ZL+DL*DL))
C     NORMALISATION CONSTANT
      XNORM=XNORM+CA(II)
   20 CONTINUE
C      CONVOLUTION WITH DATA OF UPWARD CONTINUATION ARRAY
   30 DO 50 I=NSL,NSU
      SUM=0.D0
      DO 40 J=1,NOC
      NM=I+J-1-NC
      SUM=SUM+DM(NM)*CA(J)/XNORM
   40 CONTINUE
      TM(I)=SUM
   50 CONTINUE
      RETURN
      END
C
      SUBROUTINE INPUD(TM,IDCB,IBUF,NAME,ISEG,ISGNO,LU,IPS,IB,IFORM,
     *NXN,IST,IEND,TOT,DX)
C*************************************************************
C
C      THIS PROGRAM DRAWS DATA FROM A DISC FILE
C      AND THEN OFFERS:
C       1.   PLOT OF ORIGINAL DATA
C       2.   OUTPUT OF DATA VALUES TO CHECK SEGMENT etc.
C
C*************************************************************
C
      EMA TM(2500)
      DOUBLE PRECISION TM
      DIMENSION IDCB(144),IBUF(40),NAME(3),ISEG(3),ISGNO(3)
      DIMENSION LU(5),IPS(3),IB(2),IFORM(40)
C
C     READ IN FILE DATA
```

```
      NAME(2)=2H
      NAME(3)=2H
      DO 100 I=1,1000
      TM(I)=0.0
  100 CONTINUE
      WRITE(LU,299)
  299 FORMAT(/,"ENTER FILE NAME")
      READ(LU,300) (NAME(I),I=1,3)
  300 FORMAT(3A2)
      WRITE(LU,400)
  400 FORMAT(/,"ENTER SECURITY CODE & CARTRIDGE NUMBER")
      READ(LU,*)ISC,ICR
      IF(ISC.LT.0) STOP
      IF(ICR.LE.0) ICR=9
      CALL OPEN(IDCB,IERR,NAME,0,ISC,ICR)
      IF(IERR.GE.0) GO TO 600
      WRITE(LU,500) IERR
  500 FORMAT(/,"ERROR ",I3," ON OPEN")
      STOP
  600 WRITE(LU,700)
  700 FORMAT(/,"WHAT DATA SEGMENT IS REQUIRED-RIGHT JUSTIFY")
      READ(LU,300) ISGNO
  800 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR.LT.0) GO TO 1500
      CALL CODE
      READ(IBUF,900) IFLAG,FLGT,ISEG
  900 FORMAT(2I10,4X,3A2)
      DO 1000 I=1,3
      IPS(I)=ISGNO(I)
      IF(ISGNO(I).NE.ISEG(I)) GO TO 800
 1000 CONTINUE
      WRITE(LU,1100)
 1100 FORMAT(/,"WHAT IS FORMAT (F-TYPE) OF DATA--(E.G.(12F6.0))?")
      READ(LU,301) IFORM
  301 FORMAT(40A2)
      WRITE(LU,1155)
 1155 FORMAT(/,"HOW MANY DATA VALUES PER RECORD?")
      READ(LU,*) INO
      DO 1200 I=1,4000,INO
      CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR.LT.0) GO TO 1500
      IF(LEN.LT.0) GO TO 1600
      CALL CODE
      READ(IBUF,900) IFLAG
      IF(IFLAG.EQ.18) GO TO 1600
      NXN=I+INO-1
      CALL CODE
      READ(IBUF,IFORM)(TM(J),J=I,NXN)
 1200 CONTINUE
      WRITE(LU,1300) ISEG
 1300 FORMAT(/,"MORE THAN 1000 PTS IN ",3A2)
 1600 CALL CLOSE(IDCB,IERR)
      WRITE(LU,1400)
 1400 FORMAT(/,"DOES DATA NEED TO BE DIVIDED BY 1000-eg AIRBORNE FILE?")
      READ(LU,300) ITHOU
      IF(ITHOU.EQ.2HNO) GO TO 1450
      DO 1475 I=1,NXN
      TM(I)=TM(I)/1000.0
 1475 CONTINUE
```

```
      FILE DATA NOW IN,DO YOU WANT A PLOT OR WRITE IT?"

 1450 WRITE(6,1151) NXN
      WRITE(LU,1151) NXN
      WRITE(LU,51)
   51 FORMAT(/,"DO YOU WANT TO PRINT DATA?",
     2/,"----------IF YES , TYPE IN NUMBER OF VALUES REQUIRED",
     3/,"----------IF NO TYPE IN 'NO'.")
      READ(LU,820) IB
      IF(IB.EQ.2HNO) GO TO 833
  820 FORMAT(2A2)
      CALL CODE
      READ(IB,*) NUT
      WRITE(LU,831) (TM(I),I=1,NUT)
  831 FORMAT(8(F8.2,2X))
 1151 FORMAT(/,"THERE WERE A TOTAL OF ",I5," DATA POINTS READ.")
  833 WRITE(6,115) ISGNO
  115 FORMAT(/,"**********INPUT DATA FROM LINE ",3A2,"*********")
      IF(IB.EQ.2HNO) GO TO 113
      WRITE(6,112) (TM(I),I=1,NUT)
  112 FORMAT(13(F8.2,2X))
  113 WRITE(LU,255)
  255 FORMAT(/,"WHAT IS SAMPLE INTERVAL OF DATA?")
      READ(LU,*) DX
      WRITE(LU,256)
  256 FORMAT(/,"WHAT IS THE REFERENCE LEVEL-eg TOTAL FIELD?")
      READ(LU,*) TOT

      WRITE(LU,110)
  110 FORMAT(/,"WHAT ARE START & END POINTS ALONG PROFILE?")
      READ(LU,*) IST,IEND
      IF(IEND.GT.NXN) IEND=NXN
      IF(IST.LT.0) IST=1
      RETURN
 1500 WRITE(LU,1510) IERR
 1510 FORMAT(/,"THERE HAS BEEN A READ-FROM-FILE ERROR-IERR="I4)
      CALL CLOSE(IDCB,IERR)
      RETURN
      END
      SUBROUTINE WPLOT(POS,DEP,AS,SS,LU,NN,DATA,NPT,DX,MOKE,SP1,SP2
     *,ISEG,NTYPE)

**************************************************************

      THIS SUBROUTINE PLOTS THE OUTPUT OF THE PROGRAM WERN.
      THE ARRAYS ARE AUTOMATICALLY SCALED AND THEN PLOTTED
      IN TWO SEPERATE OPERATIONS.
        1. THE AXES AND PROFILE ARE PLOTTED FOR THE ORIGINAL
           TOTAL FIELD DATA.
        2. THE DEPTH INTERPRETATIONS ARE PLOTTED BELOW THE
           PROFILE IN A KILOMETRES/IN DEPTH SCALE (Q)
      THE DEPTH ESTIMATES ARE PLOTTED AS A SYMBOL WHICH INDICATES
    THE SIZE OF MAGNETISATION (PLOTTED BY SIZE
    OF FROM 0.0 TO 0.4 INCHES). THE DIRECTION OF MAGNETISATION
    IS ALSO PLOTTED AS A TICK IN THE DIRECTION OF THE VECTOR.

    A PLOT OPTION IS AVAILABLE AND INDICATED BY THE INTEGER MOKE.
    ACCORDING TO MOKE=0  A SINGLE PLOT OF ALL LEVELS IS PRODUCED
                         ON TWO SHEETS OF PLOT PAPER.
                 MOKE=1  A PLOT IS PRODUCED FOR EACH LEVEL OF
```

```
                    INTERPRETATION. ONLY SINGLE SHEETS.
*****************************************************************************
      DIMENSION POS(502),DEP(502),AS(502),SS(502),LU(5),ISEG(3)
      DIMENSION DATA(1002),SPACE(1002)

  SCALE THE DISTANCE ARRAY AND SELECT OPTION

      DO 10 I=1,NPT
      SPACE(I)=(I-1)*DX/1000.
 10 CONTINUE

   DECIDE ON PLOT OPTION

      IF(MOKE.EQ.0) GO TO 50
      IF(MOKE.EQ.3) GO TO 25

  INITIALISE PLOT PAGE AND PLOT BORDER

      CALL PLOTS(9.5,12.0,0)
      CALL FACTR(1.0)
      CALL PLOT(1.,6.,-3)
      CALL PLOT(0.,6.,3)
      CALL PLOT(8.5,6.,2)
      CALL PLOT(8.5,-6.,2)
      CALL PLOT(0.,-6.,2)
      CALL PLOT(0.,0.,3)

  SCALE AND PLOT DISTANCE SCALE

      CALL SCALE(SPACE,8.5,NPT,1)
      SP1=SPACE(NPT+1)
      SP2=SPACE(NPT+2)
      CALL AXIS(0.,0.,12HDISTANCE(KM),12,8.5,0.,SP1,SP2)

  SCALE AND PLOT PROFILE

 15 CALL SCALE(DATA,6.,NPT,1)
      CALL AXIS(0.,0.,22HMAGNETIC INTENSITY(NT),22,6.,90.,
     *DATA(NPT+1),DATA(NPT+2))
      CALL LINE(SPACE(1),DATA(1),NPT,1,0)

   SCALE AND PLOT DEPTH AXES

      CALL PLOT(0.,-6.,-3)
      CALL AXIS(0.,0.,9HDEPTH(KM),9,6.,90.,3.0,-0.50)
      CALL PLOT(0.,6.,-3)

 INSERT ANNOTATION ONTO PLOT

      CALL CODE
      READ(ISEC,55) FIPS
 55 FORMAT(F6.0)
      CALL SYMB(0.5,5.0,0.1,12HLINE NUMBER=,0.,12)
      CALL NUMB(2.0,5.0,0.1,FIPS,0.,0)
      IF(NTYPE.EQ.1)CALL SYMB(0.5,4.5,0.1,23HMODEL TYPE = THIN SHEET,
     *0.,23)
      IF(NTYPE.EQ.2)CALL SYMB(0.5,4.5,0.1,23HMODEL TYPE = INTERFACES,
     *0.,23)
```

```
PLOT POINTS AND DEPTH WHERE ESTIMATED

25 POS(NN+1)=SP1
   POS(NN+2)=SP2

 CALCULATE MAXIMUM AND MINIMUM MAGNETISATIONS & ARRANGE ANGLES FOR PLOT

   SINS=SS(1)
   SAXS=SS(1)
   DO 20 I=1,NN
   IF(SS(I).GT.SAXS) SAXS=SS(I)
   IF(SS(I).GT.SINS) SINS=SS(I)
   AS(I)=AS(I)+45.0
20 CONTINUE

SELECT PLOT RANGE FOR MAGNETISATION

   SINK=SAXS-SINS
   DO 30 I=1,NN
   SS(I)=(SS(I)/SAXS)*0.4
30 CONTINUE

SCALE DEPTH VALUES FOR PLOTTING ON 6 INCH VERTICAL SCALE

   DO 35 I=1,NN
   DEP(I)=-1.0*DEP(I)*2.0
35 CONTINUE

PLOT INDIVIDUAL DEPTH ESTIMATES

   DO 40 I=1,NN
   CALL SYMB((POS(NN+1)+POS(I)/POS(NN+2)),DEP(I),SS(I),81,AS(I),-1)
40 CONTINUE
   GO TO 45

THIS SECTION IS JUMPED TO IF A SINGLE PLOT ONLY IS REQUIRED

50 CALL PLOTS(20.5,12.0,0)
   CALL FACTR(1.0)
   CALL PLCT(1.,6.,-3)
   CALL PLOT(0.,6.,3)
   CALL PLOT(19.5,6.,2)
   CALL PLOT(19.5,-6.,2)
   CALL PLOT(0.,-6.,2)
   CALL PLOT(0.,0.,3)

SCALE THE DISTANCE AXIS FOR INCREASED PLOT LENGTH

   CALL SCALE(SPACE,19.5,NPT,1)
   CALL AXIS(0.,0.,12HDISTANCE(KM),12,19.5,0.,SPACE(NPT+1),
  *SPACE(NPT+2))
   SP1=SPACE(NPT+1)
   SP2=SPACE(NPT+2)
   MOKE=3
   GO TO 15

 END PLOT ROUTINE AND RETURN TO PROGRAM

45 RETURN
```

```
      END
      SUBROUTINE SORT(POS,DEP,AS,SS,NS)
*************************************************************

    THIS SUBROUTINE SORTS ARRAYS DEP,POS,AS,SS INTO
   ASCENDING ORDER ACCORDING TO THE DISTANCE ALONG
   POS (OR POSITION)

   ISWAP IS USED TO COUNT THE NUMBER OF TIMES A SWAP IS
   MADE IN EACH PASS THROUGH THE ARRAY.
    IF NO SWAPS HAVE OCCURRED (ISWAP=0) THEN ALL ELEMANTS
    OF THE ARRAY ARE IN CORRECT ORDER.

*************************************************************

      DIMENSION POS(502),DEP(502),AS(502),SS(502)
   50 ISWAP=0
      DO 100 I=1,NS
      IF(POS(I).GE.POS(I-1)) GO TO 100

   NEED TO SWAP

      A=POS(I)
      POS(I)=POS(I-1)
      POS(I-1)=A

      A=DEP(I)
      DEP(I)=DEP(I-1)
      DEP(I-1)=A

      A=AS(I)
      AS(I)=AS(I-1)
      AS(I-1)=A

      A=SSI)
      SS(I)=SS(I-1)
      SS(I-1)=A

      ISWAP=ISWAP+1
  100 CONTINUE

  THAT PASS COMPLETED , WERE ANY SWAPS MADE THAT TIME?

      IF(ISWAP.GT.0) GO TO 50
      RETURN
      END
      END$
```

```
TN4,B,L,Q,C,Y
```

```
                    ************************************
                    *                                  *
                    *                                  *
                    *                                  *
                    *          PROGRAM  ADEPT          *
                    *                                  *
                    *                                  *
                    *                                  *
                    ************************************
```

```
EMA(BLOK1,2)
      PROGRAM ADEPT(3,200)

************ ***************************************************************

     TITLE     : ADEPT - THIS PROGRAM ESTIMATES THE DEPTHS TO MAGNETIC
                 BASEMENT BY USE OF THE HILBERT TRANSFORM.

     AUTHOR    : J. PHILLIPS, USGS, DENVER, COLORADO, U.S.A.

     MODS      : PETER R GIDLEY , Bureau of Mineral Resources , CANBERRA, ACT.

     PURPOSE   : TO ESTIMATE THE DEPTHS TO MAGNETIC BASEMENT FROM SAMPLED
                 DISCRETE DATA.

     USE       :
    RESULTS MUST BE INTERPRETED USING THE CONVERGENCE OF THE DEPTH ESTIMATES.
    SUBROUTINES REQUIRED:  DEPTH, FORK, DPLOT, INPUD, UPLOT, ALPLT.
    THE PROGRAM INITIALLY READS IN DATA FROM A DISC FILE
    ACCORDING TO A USER DEFINED FORMAT (WHICH IS THE SAME FORMAT
    AS THE DI3C FILE)
    THE PROGFAM THEN CALCULATES THE HILBERT TRANSFORM AND A
    BURG AUTOCORRELATION  WHICH ESTIMATES DEPTHS TO SOURCES
    AS THE PROGRAM SCANS ACRSSS THE DATA PROFILE. THESE
    DEPTH ESTIMATES AND DIFFERENCES ARE LISTED, PLOTTED AND
    LINE PRINTER PLOTTED IF A QUICK PLOT IS DESIRED.
    WRITTEN BY J.PHILLIPS.

    MODIFIED BY PETER R. GIDLEY       19/1/1981.

   THE ORIGINAL PROGRAM AND ASSOCIATED ALGORITHMS WERE
   TAKEN FROM THE PAPER BY R. H. GODSON:
     "Depth to magnetic basement system" USGS REPORT, OCTOBER, 1978
  A COPY OF THE PROGRAM AND ITS INTERPRETATIONAL PROCEDURE IS
  LOCATED ON OPEN FILE USGS MICROFISCHE REORT 79/367.
  A TWO DIMENSIONAL BASEMENT SURFACE CAN BE CONSTRUCTED BY
  LAMINATING TOGETHER A LARGE NUMBER OF VERY THIN VERTICAL
  OR NEAR VERTICAL DYKES. THE DYKES ARE OF INFINITE Y-DIRECTION
```

```
 AND Z-DIRECTION EXTENT AND TERMINATE AT A DEPTH Z1(X) BELOW
 THE SURFACE Z=0. EACH DYKE HAS A MAGNETISATION INTENSITY "M"
 WHICH MAY DIFFER FROM THOSE OF ADJOINING DYKES. A CROSS SECTION
 IN THE X-DIRECTION REVEALS THE TOPOGRAPHY OF THE BASEMENT Z1(X).
 THE MAGNETISATION OF THE BASEMENT IS EXPRESSED AS M(X).
 THE MAGNETIC ANOMALY PROFILE H(X) OBSERVED AT THE SURFACE Z
 =0 IS THE SUPERPOSITION OF THE ANOMALIES PRODUCED BY EACH THIN
 DYKE. A MOVING WINDOW IS PASSED ALONG THE DATA PROFILE AND DEPTHS
 CALCULATED. WITHIN THIS WINDOW THE MAGNETIC ANOMALY CAN BE EXPRESSED
 AS THE CONVOLUTION OF THE MAGNETISATION M(X) AND AN IMPULSE RESPONSE
 G(X) WHICH IS EQUIVALENT TO THE ANOMALY OF A SINGLE DYKE AT DEPTH Z1.
 CONSEQUENTLY THE AUTOCORRELATION 0h(X) OF THE MAGNETIC PROFILE IS
 GIVEN BY THE CONVOLUTION OF THE AUTOCORRELATION 0m(X) OFTHE
 MAGNETISATION AND THE AUTOCORRELATION 0g(X) OF THE IMPULSE RESPONSE.
 A PROGRESSIVE LAG OF THE AUTOCORRALATIONS IS CREATED BY THE MOVEMENT
 OF THE WINDOW AND ITS LENGTH ACROSS THE PROFILE AND CONSEQUENTLY
 A PRIMARY AND SECONDARY ALGORITHM OF DEPTHS IS APPLIED TO CALCULATE
 DEPTHS WITHIN THE WINDOW. IF THE DEPTH ESTIMATES REMAIN NEARLY THE
 SAME FOR ALL LAGS, THEN THE CALCULATED AUTOCORRELATION FUNCTION
 HAS THE FORM OF THE THEORETICAL AUTOCORRELATION FUNCTION, AND WE CAN
 ASSUME THAT BOTH THE MODEL AND THE DEPTH ESTIMATES ARE VALID.

PROGRAM EXECUTION:
    THE PROGRAM DRAWS ORIGINAL DATA FROM A DISC FILE SO THE
    FOLLOWING INPUT STEPS ARE REQUIRED:
       1.  INPUT THE FILE NAME
       2.  INPUT THE SECURITY CODE AND CARTRIDGE NUMBER CONTAINING FILE
       3.  INPUT THE LINE OR SEGMENT NUMBER REQUIRED.N.B. THIS
           REQUIRED TO BE RIGHT JUSTIFIED WITH EG 4 BLANKS TO THE
           LEFT OF A TWO DIGIT LINE NUMBER EG ^^^^50.
       4.  INPUT IN BRACKETS, THE FORMAT THAT THE MAGNETIC VALUES
           ARE RECORDED ON THE FILE
       5.  HOW MANY VALUES ARE THERE ON THE FILE LINE
       6.  DOES THE DATA REQUIRE DIVIDING BY 1000?
       7.  TO CHECK THE INPUT VALUES  AN OPTION IS AVAILABLE FOR PRINTING
       8.  DO YOU WANT TO PLOT THE DATA?
       9.  DO YOU WANT TO PLOT?
      10.  INPUT SAMPLE INTERVAL
      11.  INPUT REFERENCE LEVEL OF TOTAL FIELD.
      12.  INPUT START AND STOP POINTS OF PROFILE
      13.  DO YOU WANT TO REMOVE A LEAST SQUARES FIT.
      14.  DO YOU WANT TO PROCESS EVERY POINT- A RESTRICTION IS
           IMPOSED BY THE DIMENSIONS OF THE ARRAYS. THESE ARE LIMITED
           TO 500 POINTS.
      15. AN OPTION OF PLOTTING NEW DATA SPACING
      16.  INPUT NUMBER OF POINTS IN THE WINDOW EG 15.
      17.  DO YOU WANT TO USE NOISY DATA OPTION . SEE PAPER.

**************************************************

      COMMON/BLOK1/ X(2500),A(5,500),R(5,500),V(500)
     2,D(4,500),ND(500),AA(500),IDAT(2500),XC(500),F(500),B(500)
     3,FB(500),CX(500)
      COMMON C
      COMPLEX A,R,C,XC,F,B,FB,FTEM,CONJG,CM,CX,CEXP
      COMPLEX XP
      DIMENSION LU(5),IPS(3),IB(2),ISEC(3),NAME(3),WE(50),WO(50)
      DIMENSION PLT(502),DIF(502),DEPP(502),DEPS(502),DIFS(502)
      DOUBLE PRECISION AP,XINC,DET,RC,WE,TEMP
```

```
      REAL IDAT

READ INPUT DATA AND PREPARE IT FOR PROCESSING

      CALL RMPAR(LU)
      CALL INPUD(IDAT,IPS,ISEG,NAME,LU,NXN,IST,IEND,TOT,DX,LUN)
      DELX=DX/1000.
      NC=4
      NXN=IEND-IST+1
      NCOEF=16
      ILINE=1
      IZAP=1
      LX=NXN
      WRITE(LU,53)
53    FORMAT(/,"DO YOU WANT A LEAST SQUARES FIT REMOVED FROM DATA?")
      READ(LU,300) ISQ
      IF(ISQ.EQ.2HNO) ILINE=0
      DO 57 I=1,NXN
      X(I)=IDAT(I)
57    CONTINUE
      WRITE(LU,65)
65    FORMAT(/,"DO YOU WANT EVERY DATA POINT PROCESSED?",/,
     2"------IF NO , INPUT THE NUMBER OF POINTS TO BE MISSED.",/,
     3"------A MAXIMUM OF 500 POINTS ONLY IS ALLOWED, DUE TO ",/,
     4"------MEMORY SIZE RESTRICTIONS. DELETE A NUMBER OF VALUES ",/,
     5"------TO ALLOW FOR THIS BY TAKING EVERY nTH VALUE OF THE ",/,
     6"------DATA. BEFORE THIS IS DONE THE DATA IS AUTOMAT.CALLY",/,
     7"------FILTERED BY A HANNING FILTER.")
      READ(LU,300) IB
300   FORMAT(2A2)
      IF(IB.EQ.2HYE) GO TO 76
      CALL CODE
      READ(IB,*) NOT
      DO 54 I=1,NXN
      IDAT(I)=IDAT(I)/4.0+IDAT(I+1)/2.0+IDAT(I+2)/4.0
54    CONTINUE
      MN=0
      DO 80 I=1,NXN-2,NOT
      MN=MN+1
      X(MN)=IDAT(I)
80    CONTINUE
      LX=MN
      DELX=DELX*NOT
      DELM=DELX*1000.
      WRITE(LU,86)
86    FORMAT(/,"DO YOU WANT A PLOT OF PROFILE WITH POINTS MISSED?")
      READ(LU,300) IPO
      IF(IPO.EQ.2HNO) GO TO 76
      CALL UPLOT(X,LX,DELM,1,LX,TOT,LU,IPS)
76    WRITE(LUN,102) DELX,LX,ILINE
      WRITE(LU,102) DELX,LX,ILINE
      WRITE(LUN,112)
      WRITE(LUN,106) (X(I),I=1,MN)
      WRITE(LUN,122)
122   FORMAT("  PREDICTION FILTER COEFFICIENTS")
102   FORMAT(2X,"INPUT PARAMETERS-",/,"   DELX= ",F10.7,"   LX= ",I5
     1,"   ILINE= ",I5)
112   FORMAT("   INPUT DATA ARRAY  ")
      WRITE(LU,55)
55    FORMAT(/,"HOW MANY POINTS IN THE WINDOW?")
```

```
      READ(LU,*) IW
      WRITE(LU,200)
      READ(LU,300) INOIS
      WRITE(LU,113)
113   FORMAT(/////,"           ********************************** ",/,
     1"                I AM PROCESSING YOUR REQUEST",/,"
     2PLEASE  WAIT",/,/,"           **********************************")
      LXM1=LX-1
      SLOPE=0.
      DB=0.
      DO 5 I=1,LX
      DB=DB+X(I)
    5 CONTINUE
      DB=DB/LX
      IF(ILINE.NE.1) GO TO 11
      SLOPE=(X(LX)-X(1))/LXM1
      DB=X(1)
      DO 6 I=1,LX
      X(I)=X(I)-(I-1)*SLOPE-DB
    6 CONTINUE

 REMOVE LEAST SQUARES LINE

      ISUM=LXM1*LX/2
      SQIS=FLOAT(LXM1)*FLOAT(LX)*FLOAT(LX+LX-1)/6.
      SQIS=AINT(SQIS)
      XIND=0.
      XSUM=X(LX)
      DO 10 I=1,LXM1
      XIND=XIND+X(I+1)*I
      XSUM=XSUM+X(I)
   10 CONTINUE
      DET=LX*SQIS-FLOAT(ISUM)*FLOAT(ISUM)
      SLOPE=(LX*XIND-FLOAT(ISUM)*XSUM)/DET
      DB=(SQIS*XSUM-ISUM*XIND)/DET
   11 CONTINUE
      DO 12 I=1,LX
      X(I)=X(I)-(I-1)*SLOPE-DB
      XC(I)=X(I)
   12 CONTINUE

 DETERMINE PREDICTION FILTER

      DO 20 I=1,LX
      AA(I)=X(I)
      V(I)=X(I)
  20  CONTINUE
      WE(1)=1.
      DO 23 J=2,NCOEF
      AP=0.
      XIND=0.
      DO 21 I=J,LX
      AP=AP+AA(I)*AA(I)+V(I-J+1)*V(I-J+1)
      XIND=XIND+AA(I)*V(I-J+1)
  21  CONTINUE
      RC=-2.*XIND/AP
      DO 22 I=J,LX
      TEMP=AA(I)
      AA(I)=AA(I)+RC*V(I-J+1)
      V(I-J+1)=V(I-J+1)+RC*TEMP
```

```
 22 CONTINUE
    WE(J)=0.
    JH=(J+1)/2
    DO 23 I=1,JH
    K=J-I+1
    TEMP=WE(K)+RC*WE(I)
    WE(I)=WE(I)+RC*WE(K)
    WE(K)=TEMP
 23 CONTINUE

 EXTEND PROFILE

    K=LX+100
    DO 24 I=1,11
    LC=2**I
    IF(LC.GT.K) GO TO 25
 24 CONTINUE
 25 LP=LX+1
    JH=(LP+LC)/2
    DO 26 I=LP,JH
    XC(I)=0.
    XC(LC-I+LP)=0.
    DO 26 J=2,NCOEF
    XC(I)=XC(I)-WE(J)*XC(I-J+1)
    XC(LC-I+LP)=XC(LC-I+LP)-WE(J)*XC(MOD(LC-I+LP+J-2,LC)+1)
 26 CONTINUE
200 FORMAT(/,"IS DATA NOISY- see Program Explanation")
    IF(INOIS.EQ.2HYE) GO TO 201

 BEGIN HILBERT TRANSFORM

    CALL FORK(LC,XC,-1.)
    K=LC/2
    DO 30 J=2,K
    XC(J)=XC(J)+XC(J)
    XC(LC+2-J)=0.
 30 CONTINUE
    CALL FORK(LC,XC,+1.)
 END HILBERT TRANSFORM

    GO TO 121

201 IZAP=0

 BEGIN HILBERT TRANSFORM FOR NOISY DATA

    CALL FORK(LC,XC,-1.)
    K=LC/4
    DO 130 J=2,K
    XC(J)=XC(J)+XC(J)
130 XC(LC+2-J)=0.
    K=K+1
    L=LC/2+1
    DO 131 J=K,L
    ARG=3.1415926S*FLOAT(J-K)/(L-K)
    XC(J)=XC(J)*(1.0+COS(ARG))
131 XC(LC+2-J)=0.0
    CALL FORK(LC,XC,+1.0)

 END HILBERT TRANSFORM FOR NOISY DATA
```

```
121 WRITE(LUN,104)(WE(J),J=1,NCOEF)
    WRITE(LUN,105)
    WRITE(LUN,106)(XC(I),I=LP,LC)
104 FORMAT(12F10.7)
105 FORMAT(1H0,'PREDICTED EXTENSION',/)
106 FORMAT(12F10.2)
    DO 40 I=1,LX
    V(I)=1.
    A(1,I)=1.
    R(1,I)=1.
    DO 40 J=1,NC
    D(J,I)=0.
 40 CONTINUE
    IS=IW/2
    IW=IS+IS+1
    IWM1=IW-1
    IWM2=IW-2
    IWM3=IW-3
    IWP1=IW+1
    ISP1=IS+1
    LXMS=LX-IS
    LMSP=LXMS+1

INITIALIZE WEIGHTS

    ZE=4.
    DO 41 J=1,IS
    RO=J-IS
    RE=RO-0.5
    WO(J)=RO*RO+ZE
    WO(IWM1-J)=WO(J)
    WE(J)=RE*RE+ZE
    WE(IW-J)=WE(J)
 41 CONTINUE
    DXS=ZE

BEGIN ADAPTIVE PROCESSING

    DO 42 J=1,LX
    F(J)=XC(J)
    B(J)=XC(J)
42 CONTINUE

ENTER MAIN LOOP

    DO 52 I=1,NC,2
    IP=I+1
    IB=(I-1)*IS+2
    IE=LX-IB+I+1
    DO 43 J=IB,IE
    FB(J)=F(J)*CONJG(B(J-I))
    AA(J)=F(J)*CONJG(F(J))+B(J-I)*CONJG(B(J-I))
43 CONTINUE
    IE=IE-IWM2
    DO 48 K=IB,IE
    XP=0.
    AP=0.
    DO 44 J=1,IWM2
    XP=XP+FB(K+J-1)/WO(J)
```

```
      AP=AP+AA(K+J-1)/WO(J)
   44 CONTINUE
      CM=(XP+XP)/AP
      XP=0.
      AP=0.
      DO 45 J=1,IWM1
      XP=XP+FB(K+J-1)/WE(J)
      AP=AP+AA(K+J-1)/WE(J)
   45 CONTINUE
      C=(XP+XP)/AP
      IK=K+IS-IP/2
      CALL DEPTH(I,IK,IZAP)

   SET WEIGHTS

      ZP=D(I,IK)*D(I,IK)
      IF(ZP.LT.DXS) ZP=DXS
      DO 46 J=1,IWM1
      WE(J)=WE(J)-ZE+ZP
   46 CONTINUE
      DO 47 J=1,IWM2
      WO(J)=WO(J)-ZE+ZP
   47 CONTINUE
      ZE=ZP
      IK=IK+I/2
      FTEM=F(IK)
      F(IK)=F(IK)-CM*B(IK-I)
      B(IK-I)=B(IK-I)-CONJG(CM)*FTEM
   48 CONTINUE
      IK=IK+IP/2
      F(IK)=F(IK)-C*B(IK-I)
      IB=IB+IS
      IE=IE+IS
      DO 49 J=IB,IE
      FB(J)=F(J)*CONJG(B(J-IP))
      AA(J)=F(J)*CONJG(F(J))+B(J-IP)*CONJG(B(J-IP))
   49 CONTINUE
      IE=IE-IWM3
      DO 52 K=IB,IE
      XP=0.
      AP=0.
      DO 50 J=1,IWM2
      XP=XP+FB(K+J-1)/WO(J)
      AP=AP+AA(K+J-1)/WO(J)
   50 CONTINUE
      C=(XP+XP)/AP
      IK=K+IS-IP/2-1
      CALL DEPTH(IP,IK,IZAP)

   SET WEIGHTS

      ZP=D(IP,IK)*D(IP,IK)
      IF(ZP.LT.DXS) ZP=DXS
      DO 51 J=1,IWM2
      WO(J)=WO(J)-ZE+ZP
   51 CONTINUE
      ZE=ZP
      IK=IK+IP/2
      FTEM=F(IK)
      F(IK)=F(IK)-C*B(IK-IP)
```

```
      B(IK-IP)=B(IK-IP)-CONJG(C)*FTEM
   52 CONTINUE

   PROCESSING COMPLETED

   BEGIN OUTPUT
      IF(INOIS.NE.2HYE) GO TO 205

   BEGIN OUTPUT FOR NOISY DATA IF REQUESTED

      WRITE(LUN,167)
  167 FORMAT(/,"THE FOLLOWING RESULTS ARE CALCULATED USING THE",/,
     *"          NOISY DATA OPTION")
      WRITE(LUN,107)
      DO 160 I=1,IS
      X(I)=0.
      AA(I)=(I-1)*DELX
  160 WRITE(LUN,108) AA(I),XC(I)
      DO 165 I=ISP1,LXMS
      AA(I)=(I-1)*DELX
      J=ND(I)
      DO 161 K=1,J
  161 D(K,I)=D(K,I)*DELX
      IFLAG=1
      IF(J.GT.2) GO TO 162
      V(I)=D(1,I)-D(J,I)
      GO TO 164
  162 V(I)=D(1,I)-D(2,I)
      JM1=J-1
      DO 163 K=1,JM1
      KP1=K+1
      DO 163 L=KP1,J
      CS=D(K,I)-D(L,I)
      IF(ABS(CS).GT.ABS(V(K))) GO TO 163
      V(I)=CS
      IFLAG=K
  163 CONTINUE
  164 WRITE(LUN,108) AA(I),XC(I),V(I),(D(K,I),K=1,J)
  165 X(I)=D(IFLAG,I)
      IDEP=ISP1
      KJ=1
      DO 168 I=ISP1,LX
      DEPP(KJ)=D(1,I)
      DIF(KJ)=V(I)
      KJ=KJ+1
  168 CONTINUE
      DO 166 I=LMSP,LX
      X(I)=0.0
      AA(I)=(I-1)*DELX
  166 WRITE(LUN,108) AA(I),XC(I)
      GO TO 169

   END OUTPUT ROUTINE FOR NOISY DATA

   NOTE:  THERE IS NO SECONDARY RESULT CALCULATED

   205 WRITE(LUN,107)
      DO 60 I=1,IS
      AA(I)=(I-1)*DELX
      WRITE(LUN,108) AA(I),XC(I)
```

```
 60 CONTINUE
    IDEP=ISP1
    DO 62 I=ISP1,LXMS
    AA(I)=(I-1)*DELX
    J=ND(I)
    DO 61 K=1,J
    D(K,I)=D(K,I)*DELX
 61 CONTINUE
    V(I)=3.*(D(1,I)-D(J,I))/(FLOAT(J-1)+.1E-05)
    WRITE(LUN,108) AA(I),XC(I),V(I),(D(K,I),K=1,J)
 62 CONTINUE
    INPT=LXMS-ISP1
    KJ=1
    DO 64 J=ISP1,LX
    DEPP(KJ)=D(1,J)
    DIF(KJ)=V(J)
    KJ=KJ+1
 64 CONTINUE
    DO 63 I=LMSP,LX
    AA(I)=(I-1)*DELX
    WRITE(LUN,108) AA(I),XC(I)
 63 CONTINUE
169 DO 66 I=1,LX
    PLT(I)=AA(I)
 66 CONTINUE
    WRITE(LU,123)
123 FORMAT(/,"DO YOU WANT A LINE PRINTER PLOT OF OUTPUT?")
    READ(LU,300) NOE
    IF(NOE.EQ.2HNO) GO TO 114
    WRITE(LU,116)
116 FORMAT(/,"INPUT ORDINATE SCALE FACTOR FOR PROFILE & ORDINATE FOR",
   */,"PLOTTING THE DEPTH ESTIMATES----ASCAL,DSCAL.")
    READ(LU,*) ASCAL,DSCAL
    CALL DPLOT(X,1,1,LX,1,2,DELX,ASCAL,0.)
    CALL DPLOT(D,4,NC,LX,-1,1,DELX,DSCAL,0.)
114 CONTINUE
107 FORMAT(1H0,'PRIMARY RESULT',//,'  LOCATION',5X,'ANOMALY',3X,
   1       'HILBERT',4X,'DIF',6X,'DEPTH1',3X,'DEPTH2...'/)
108 FORMAT(2H   ,F7.2,2X,2F10.2,3X,F7.3,6(3X,F6.3))
    IF(INOIS.EQ.2HYE) GO TO 170

 OBTAIN SECONDARY RESULT

    WRITE(LUN,109)
    DXS=DELX*DELX
    ISP1=IS+IS
    LXMS=LX-ISP1+1
    KK=1
    DO 75 K=ISP1,LXMS
    N=ND(K)
    DO 70 I=1,N
    WE(I)=D(I,K)*D(I,K)
 70 CONTINUE
    D(N,K)=0.
    N=N-1
    DO 71 I=1,N
    IP=I+1
    IP2=IP*IP
    I2=I*I
    ARG=(IP2*WE(I)-I2*WE(IP))/(IP2-I2+4.*(WE(IP)-WE(I))/DXS)
```

```
      IF(ARG.LT.0.) ARG=0.
      D(I,K)=SQRT(ARG)
   71 CONTINUE
      AA(K)=(K-1)*DELX
      IFLAG=1
      IF(N.GT.2) GO TO 72
      V(K)=D(1,K)-D(N,K)
      GO TO 74
   72 V(K)=D(1,K)-D(2,K)
      NM1=N-1
      DO 73 I=1,NM1
      IP1=I+1
      DO 73 J=IP1,N
      CS=D(I,K)-D(J,K)
      IF(ABS(CS).GT.ABS(V(K))) GO TO 73
      V(K)=CS
      IFLAG=I
   73 CONTINUE
   74 ND(K)=IFLAG
      WRITE(LUN,110) AA(K),IFLAG,V(K),(D(I,K),I=1,N)
      DEPS(KK)=D(1,K)
      DIFS(KK)=V(K)
      KK=KK+1
   75 CONTINUE
  170 NC=NC-1
      IF(NOE.EQ.2HNO) GO TO 115
      CALL DPLOT(D,4,NC,LX,-1,1,DELX,DSCAL,0.)
  115 CONTINUE
  109 FORMAT(1H0,'SECONDARY RESULT',//,'  LOCATION',1X,
     1        'FLAG',5X,'DIF',5X,'DEPTH1',3X,'DEPTH2...'/)
  110 FORMAT(2H   ,F7.2,I5,3X,F7.3,4(3X,F6.3))
      WRITE(LU,117)
  117 FORMAT(/,"DO YOU WANT A PLOT OF OUTPUT?")
      READ(LU,300) IYPE
      IF(IYPE.EQ.2HNO) GO TO 118
      IHT=1
      CALL ALPLT(DEPP,KJ,IDEP,DEPS,KK,ISP1,DIF,PLT,LX,LU,IHT,IZAP)
      IF(INOIS.EQ.2HYE) GO TO 118
      IHT=0
      CALL ALPLT(DEPP,KJ,IDEP,DEPS,KK,ISP1,DIFS,PLT,LX,LU,IHT,IZAP)

  118 STOP
      END
EMA(BLOK1,2)
      SUBROUTINE DEPTH(I,IK,IZAP)
      COMMON/BLOK1/ X(2500),A(5,500),R(5,500),V(500),
     2D(4,500),ND(500),AA(500),IDAT(2500),XC(500),F(500),B(500),
     3FB(500)
      COMMON C
      COMPLEX A,R,C,BOT,CONJG

  DETERMINE NEW AUTOCORRELATION COEFFICIENT

      IP=I+1
      A(IP,IK)=0.
      R(IP,IK)=C*V(IK)
      V(IK)=V(IK)*(1.-C*CONJG(C))
      DO 20 J=2,IP
      R(IP,IK)=R(IP,IK)-A(J,IK)*R(IP-J+1,IK)
   20 CONTINUE
```

```
      IG=(IP+1)/2
      DO 21 J=1,IG
      BOT=A(IP-J+1,IK)-C*CONJG(A(J,IK))
      A(J,IK)=A(J,IK)-C*CONJG(A(IP-J+1,IK))
      A(IP-J+1,IK)=BOT
   21 CONTINUE

   DETERMINE DEPTH

      IF(IZAP.NE.0) GO TO 22
      ARG=(2*I-1)/(REAL(R(I,IK))/REAL(R(IP,IK))-1.)-(I-1)*(I-1)
      GO TO 23
   22 ARG=I*I/(1./REAL(R(IP,IK))-1.)
   23 IF(ARG.LT.0.) ARG=0.
      D(I,IK)=0.5*SQRT(ARG)
      ND(IK)=I
      RETURN
      END
      SUBROUTINE FORK(LX,CX,SIGNI)

*****************************************************************
                          LX
   CX(K) = SQRT(1/LX) SUM (CX(J)*EXP(2*PI*SIGNI*I*(J-1)*(K-1)/LX))
                          J=1

                                        FOR K=1,2,...,(LX=2**INTEGER)
   WRITTEN BY J.F.CLAERBOUT.
*****************************************************************

      EMA CX
      COMPLEX CX(LX),CARG,CW,CTEMP
      COMPLEX CEXP
      J=1
      SC=SQRT(1./LX)
      DO 5 I=1,LX
      IF(I.GT.J)GO TO 2
      CTEMP=CX(J)*SC
      CX(J)=CX(I)*SC
      CX(I)=CTEMP
    2 M=LX/2
    3 IF(J.LE.M)GO TO 5
      J=J-M
      M=M/2
      IF(M.GE.1)GO TO 3
    5 J=J+M
      L=1
    6 ISTEP=2*L
      DO 8 M=1,L
      CARG=(0.,1.)*(3.14159265*SIGNI*(M-1))/L
      CW=CEXP(CARG)
      DO 8 I=M,LX,ISTEP
      CTEMP=CW*CX(I+L)
      CX(I+L)=CX(I)-CTEMP
      CX(I)=CX(I)+CTEMP
    8 CONTINUE
      L=ISTEP
      IF(L.LT.LX)GO TO 6
      RETURN
      END
      SUBROUTINE INPUD(IDAT,IPS,ISEG,NAME,LU,NXN,IST,IEND,TOT,DX,LUN)
```

```
**********************************************************************

      THIS PROGRAM DRAWS DATA FROM A DISC FILE
      AND THEN OFFERS:
       1.   PLOT OF ORIGINAL DATA
       2.   OUTPUT OF DATA VALUES TO CHECK SEGMENT etc.

**********************************************************************

      EMA IDAT(2500)
      DIMENSION IDCB(144),IBUF(40),NAME(3),ISEG(3),ISGNO(3)
      DIMENSION LU(5),IPS(3),IB(2),IFORM(40)

      REAL IDAT
      READ IN FILE DATA

      LUN=6
      NAME(2)=2H
      NAME(3)=2H
      DO 100 I=1,500
      IDAT(I)=0.0
  100 CONTINUE
      WRITE(LU,299)
  299 FORMAT(/,"ENTER FILE NAME")
      READ(LU,300) (NAME(I),I=1,3)
  300 FORMAT(3A2)
      WRITE(LU,400)
  400 FORMAT(/,"ENTER SECURITY CODE & CARTRIDGE NUMBER")
      READ(LU,*)ISC,ICR
      IF(ISC.LT.0) STOP
      IF(ICR.LE.0) ICR=9
      CALL OPEN(IDCB,IERR,NAME,0,ISC,ICR)
      IF(IERR.GE.0) GO TO 600
      WRITE(LU,500) IERR
  500 FORMAT(/,"ERROR ",I3," ON OPEN")
      STOP
  600 WRITE(LU,700)
  700 FORMAT(/,"WHAT DATA SEGMENT IS REQUIRED- eg _____15 ie SIX SPACES")
      READ(LU,300) ISGNO
  800 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR.LT.0) GO TO 1500
      CALL CODE
      READ(IBUF,900) IFLAG,FLGT,ISEG
  900 FORMAT(2I10,4X,3A2)
      DO 1000 I=1,3
      IPS(I)=ISGNO(I)
      IF(ISGNO(I).NE.ISEG(I)) GO TO 800
 1000 CONTINUE
      WRITE(LU,1100)
 1100 FORMAT(/,"WHAT IS FORMAT (F-TYPE) OF DATA--(E.G.(12F6.0))?")
      READ(LU,301) IFORM
  301 FORMAT(40A2)
      WRITE(LU,1155)
 1155 FORMAT(/,"HOW MANY DATA VALUES PER RECORD?")
      READ(LU,*) INO
      DO 1200 I=1,4000,INO
      DO 1250 J=1,40
      IBUF(J)=0
 1250 CONTINUE
```

```
      CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR.LT.0) GO TO 1500
      IF(LEN.LT.0) GO TO 1600
      CALL CODE
      READ(IBUF,900) IFLAG
      IF(IFLAG.EQ.18) GO TO 1600
      NXN=I+INO-1
      CALL CODE
      READ(IBUF,IFORM)(IDAT(J),J=I,NXN)
 1200 CONTINUE
      WRITE(LU,1300) ISEG
 1300 FORMAT(/,"MORE THAN 4000 PTS IN ",3A2)
 1600 CALL CLOSE(IDCB,IERR)
      WRITE(LU,1400)
 1400 FORMAT(/,"DOES DATA NEED TO BE DIVIDED BY 1000-eg AIRBORNE FILE?")
      READ(LU,300) ITHOU
      IF(ITHOU.EQ.2HNO) GO TO 1450
      DO 1475 I=1,NXN
      IDAT(I)=IDAT(I)/1000.0
 1475 CONTINUE
 1450 NUM=NXN
      DO 1480 I=1,NUM
      IF(IDAT(I).EQ.0.0) NXN=NXN-1
 1480 CONTINUE

       FILE DATA NOW IN,DO YOU WANT A PLOT OR WRITE IT?"

      WRITE(LU,123)
  123 FORMAT(/,"DO YOU WANT RESULTS AUTOMATICALLY PRINTED?")
      READ(LU,300) IOUT
      IF(IOUT.EQ.2HYE) I[illegible]=16
      WRITE(LUN,1151) NXN
      WRITE(LU,1151) NXN
      WRITE(LU,51)
   51 FORMAT(/,"DO YOU WANT TO PRINT DATA?",
     2/,"----------IF YES , TYPE IN NUMBER OF VALUES REQUIRED",
     3/,"----------IF NO TYPE IN 'NO'.")
      READ(LU,820) IB
      IF(IB.EQ.2HNO) GO TO 833
  820 FORMAT(2A2)
      CALL CODE
      READ(IB,*) NUT
      WRITE(LU,831) (IDAT(I),I=1,NUT)
  831 FORMAT(8(F8.2,2X))
 1151 FORMAT(/,"THERE WERE A TOTAL OF ",I5," DATA POINTS READ.")
  833 WRITE(LUN,115) ISGNO
  115 FORMAT(/,"**************INPUT DATA FROM LINE ",3A2,"**********")
      IF(IB.EQ.2HNO) GO TO 113
      WRITE(LUN,112) (IDAT(I),I=1,NUT)
  112 FORMAT(13(F8.2,2X))
  113 WRITE(LU,1150)
 1150 FORMAT(/,"DO YOU WANT TO PLOT ORIGINAL DATA?")
      READ(LU,300) IPLT
      WRITE(LU,255)
  255 FORMAT(/,"WHAT IS SAMPLE INTERVAL OF DATA?")
      READ(LU,*) DX
      WRITE(LU,256)
  256 FORMAT(/,"WHAT IS THE REFERENCE LEVEL-eg TOTAL FIELD?")
      READ(LU,*) TOT
```

```
      WRITE(LU,110)
  110 FORMAT(/,"WHAT ARE START & END POINTS ALONG PROFILE?")
      READ(LU,*) IST,IEND
      IF(IEND.GT.NXN) IEND=NXN
      IF(IST.LT.0) IST=1
      IF(IPLT.EQ.2HNO) GO TO 923
      CALL UPLOT(IDAT,NXN,DX,IST,IEND,TOT,LU,IPS)
  923 RETURN
 1500 WRITE(LU,1510) IERR
 1510 FORMAT(/,"THERE HAS BEEN A READ-FROM-FILE ERROR-IERR=",I4)
      CALL CLOSE(IDCB,IERR)
      RETURN
      END
      SUBROUTINE UPLOT(IDAT,NXN,DX,IST,IEND,TOT,LU,IPS)

**********************************************************************

   PLOTS INPUT DATA PROFILE OF AMPLITUDE VERSUS DISTANCE.

**********************************************************************

      EMA IDAT(2500)
      DIMENSION LU(5),IPS(3)
      REAL IDAT
    5 WRITE(LU,10)
   10 FORMAT(/,"VERT SCALE IN UNITS/IN,HORIZ SCALE DIST/IN")
      READ(LU,*)SCM,SCD
      SV1=TOT-SCM
      XT=DX*FLOAT(IEND-IST)/SCD
      IF(XT.GT.16.5) 101,103
  101 WRITE(LU,102)
  102 FORMAT(/,"PLOT TOO LONGGGGGGG")
      GO TO 5
  103 CALL PLOTS(XT+10.,12.,0)
      CALL FACTR(1.0)
      CALL PLOT(1.,0.5,-3)
      CALL AXIS(0.,4.,9HAMPLITUDE,9,5.,90.,SV1,SCM)
      CALL AXIS(0.,3.,8HDISTANCE,8,13.,0.,0.,SCD)
      CALL PLOT(0.,5.,-3)
      XP=0.
      DO 600 I=IST,IEND
      YP=(IDAT(I)-TOT)/SCM
      CALL PLOT(XP,YP,2)
      XP=XP+DX/SCD
  600 CONTINUE
      YPS=IPS
      CALL SYMB(6.,3.5,0.1,12HLINE NUMBER=,0.,12)
      CALL SYMB(7.5,3.5,0.1,IPS,0.,6)
      RETURN
      END
      SUBROUTINE DPLOT(F,NDIM,N,M,JSIGN,NSIDE,DELX,SCALE,FILL,LUN)

**********************************************************************

   A HIGH DENSITY PRINTER PLOT ROUTINE,.
   SUCCESIVE VALUES ARE PLOTTED AS APOSTRAPHIES AND FULL STOPS
   UNLESS THY OCCUPY THE SAME SPACE IN WHICH CASE A VERTICAL
   SLASH IS USED.

       JSIGN = +1 FOR POSITIVE UP
```

```
                -1 FOR NEGATIVE DOWN
      NSIDE = 1 FOR ONE SIDED INPUT
                2 FOR TWO SIDED INPUT
       FILL = 1. FOR NO OVERLAP
                0. FOR TOTAL OVERLAP
      SCALE = UNITS/INCH


*************************************************************************

      EMA F(NDIM,M)
      DIMENSION IL(2),IR(2),IM(2),IB(122),LU(5)
      INTEGER LINE(121),LINE1(121),ITEML(9),ITEMR(9)
      DATA IL/2H'',2H''/
      DATA IR/2H..,2H../
      DATA IM/2H||,2H||/
      DATA IB/122*2H  /
      DATA LINE/121*2H  /
      DATA LINE1/121*2H--/
      WRITE(LUN,904) SCALE
      WRITE(LUN,906) LINE1
      SC=10.*JSIGN/SCALE
      ISET=61-JSIGN*60
      IF(NSIDE.EQ.2) ISET=61-FILL*JSIGN*(60-120/(2*N))
      IBIAS=120*FILL*JSIGN*60
      DO 16 K=2,M,2
      DO 11 J=1,N
      IMAGL=ISET+(J-1)*IBIAS
      LINE(IMAGL)=IM
      IMAGL=MIN0(MAX0(1,IFIX(F(J,K-1)*SC)+ISET+(J-1)*IBIAS),121)
      ITEML(J)=IMAGL
   11 LINE(IMAGL)=IL
      DO 13 J=1,N
      IMAGR=MIN0(MAX0(1,IFIX(F(J,K)*SC)+ISET+(J-1)*IBIAS),121)
      ITEMR(J)=IMAGR
      IF(LINE(IMAGR).EQ.IR) GO TO 13
      IF(LINE(IMAGR).NE.IB) GO TO 12
      LINE(IMAGR)=IR
      GO TO 13
   12 LINE(IMAGR)=IM
   13 CONTINUE
      IF(MOD(K-2,10).EQ.0) GO TO 14
      WRITE(LUN,907) LINE
      GO TO 15
   14 XL=(K-2)*DELX
      WRITE(LUN,908)XL,LINE
   15 CONTINUE
      DO 16 J=1,N
      LINE(ITEML(J))=IB
   16 LINE(ITEMR(J))=IB
      WRITE(LUN,906) LINE1
      DO 17 J=1,N
      IMAGL=ISET+(J-1)*IBIAS
   17 LINE(IMAGL)=IB
      RETURN
  904 FORMAT(1H1,50X,F8.3," UNITS/INCH",/)
  906 FORMAT(10X,121A1)
  907 FORMAT(9X,"|",121A1,"|")
  908 FORMAT(1H ,F8.3,"+",121A1,"+")
      END
```

```
      SUBROUTINE ALPLT(DEPP,KJ,IDEP,DEPS,KK,ISP1,DIF,PLT,LX,LU,IHT,IZAP)

C*********************************************************************

C     THIS SUBROUTINE PRODUCES A PLOT OF THE PRIMARY AND
C     SECONDARY DEPTH ESTIMATES AND THEIR RESPECTIVE
C     DIFFERENCES AGAINST DISTANCE ALONG THE TRAVERSE.

C*********************************************************************

      DIMENSION LU(5),DIF(502),PLT(502),DEPP(502),DEPS(502)

C     INITIALISE PLOT FOR DIFFERENCES AND PLOT AXES.

      CALL PLOTS(9.5,12.,0)
      CALL FACTR(1.0)
      CALL PLOT(1.,6.,-3)
      CALL PLOT(0.0,6.0,3)
      CALL PLOT(8.5,6.0,2)
      CALL PLOT(8.5,-6.0,2)
      CALL PLOT(0.0,-6.0,2)
      CALL PLOT(0.0,3.0,3)
      CALL PLOT(8.5,3.0,2)
      CALL PLOT(0.0,0.0,3)
      CALL SCALE(PLT,10.,LX,1)
      IF(IHT.EQ.1) INOK=KJ-IDEP
      IF(IHT.EQ.0) INOK=KK-1
      DO 4 I=1,INOK
      IF(DIF(I).LT.-1.5) DIF(I)=-1.5
      IF(DIF(I).GT.1.5) DIF(I)=1.5
    4 CONTINUE
      DIF(INOK+1)=-1.5
      DIF(INOK+2)=0.5
      CALL AXIS(0.,0.,14HDIFFERENCE(KM),14,6.,90.,DIF(INOK+1),DIF(INOK+2
     *))
      CALL AXIS(0.,0.,12HDISTANCE(KM),12,10.,0.,PLT(LX+1),PLT(LX+2))

C     PLOT DIFFERENCE DATA

      IF(IHT.EQ.0) GO TO 5
      PLT(IDEP+INOK)=PLT(LX+1)
      PLT(IDEP+INOK+1)=PLT(LX+2)
      CALL LINE(PLT(IDEP),DIF(1),INOK,1,0)
      GO TO 7
    5 PLT(ISP1+INOK)=PLT(LX+1)
      PLT(ISP1+INOK+1)=PLT(LX+2)
      CALL LINE(PLT(ISP1),DIF(1),INOK,1,0)
    7 CONTINUE
      CALL SYMB(5.,5.,0.2,15HDIFFERENCE PLOT,0.,15)

C     RESET ORIGIN FOR DEPTH PLOT AND PLOT AXES.

      CALL PLOT(0.,-6.,-3)
      IF(IZAP.EQ.0) GO TO 25
      CALL SCALE(DEPS(1),6.,KK-1,-1)
      DEPP(KJ-IDEP+1)=DEPS(KK)
      DEPP(KJ-IDEP+2)=DEPS(KK+1)
   15 CALL AXIS(0.,0.,9HDEPTH(KM),9,6.,90.,DEPS(KK),DEPS(KK+1))

C     PLOT DEPTH DATA
```

```
      IF(IHT.EQ.0) GO TO 10
      CALL LINE(PLT(IDEP),DEPP(1),INOK,1,10,3,.1)
 10   PLT(ISP1+KK-1)=PLT(LX+1)
      PLT(ISP1+KK)=PLT(LX+2)
      CALL LINE(PLT(ISP1),DEPS(1),KK-1,1,0)
      CALL SYMB(5.,5.,0.2,11HDEPTH PLOTS,0.,11)
      IF(IHT.EQ.0) GO TO 20
      CALL SYMB(5.0,4.5,0.1,14H---+---+=PRIMARY,0.,14)
 20   CALL SYMB(5.0,4.0,0.1,16H-------=SECONDARY,0.,16)
      CALL PLOT(0.,0.,999)
      RETURN

SCALE DEPP IF NOISY DATA OPTION IS CALLED

 25   CALL SCALE(DEPP(1),6.,INOK,-1)
      DEPS(KK)=DEPP(INOK+1)
      DEPS(KK+1)=DEPP(INOK+2)
      GO TO 15
      END
      END$
```

TN4,B,L,Q,C,Y

```
**********************************
*                                *
*                                *
*                                *
*          PROGRAM  GRVGD        *
*                                *
*                                *
*                                *
*                                *
**********************************
```

PROGRAM GRVGD

****

TITLE : GRVGD - THIS PROGRAM CALCULATES THE GRAVITY ANOMALY OF A SERIES OF THREE-DIMENSIONAL DIPPING PRISMD OR BLOCKS.

AUTHOR : PETER R. GIDLEY, Bureau of Mineral Resources, CANBERRA, ACT.

REF. : (a) HJELT, S.E. -1974- THE GRAVITY ANOMALY OF A DIPPING PRISM. GEOEXPLORATION v.12(1), pp.29-39.

(b) R.D. OGILVY - AN INTERACTIVE COMPUTER PROGRAM TO CALCULATE THE GRAVITY ANOMALY OF A FINITE HORIZONTAL PRISM. B.M.R. Record 1979/29.

PURPOSE : THIS PROGRAM CALCULATES THE GRAVITY ANOMALY OF A SERIES OF 3-DIMENSIONAL DIPPING PRISMS, OR TRAPEZOIDAL STRUCTURES.

USE : THE PROGRAM USES THE FORMULAE DESCRIBED BY THE PAPER BY HJELT (1974 - "THE GRAVITY ANOMALY OF A DIPPING PRISM. - GEOEXPLORATION V.12, pp29-39)
AS THE PROGRAM IS PRESENTLY STRUCTURED AND IIMENSIONED IT IS CAPABLE OF CALCULATING THE GRAVITY ANOMALY OF UP TO FIVE INDIVIDUAL 3-DIMENSIONAL BODIES. THE CALCULATIONS ARE CARRIED OUT WITH RESPECT TO EACH BODY, SUMMED, AND THEN OUTPUT AS EITHER LINE PRINTER FILE, PLOT OR AS A DISPLAYED FILE ON THE TERMINAL OF THE USER. THE OUTPUT GRAVITY EFFECT APPEARS IN ARRAY "SUMG" WHICH IS THE SUM TOTAL OF THE GRAVITY EFFECTS OF ALL THE BODIES - B(T,L) WHERE L IS THE BODY NUMBER AND T JS THE GRAVITY EFFECT.
A CONVENTIONAL RIGHT HANDED COORDINATE SYSTEM IS USED FOR THE PLACEMENT OF THE AXES AND BODIES.

ALL DIMENSIONS ARE IN METRES.
NOTE: THIS EVEN APPLIES TO A REGIONAL SITUATION e.g.10km=10000m.

MN=MODEL NUMBER

```
                X1=CO-OD OF FIRST TOP EDGE ON X AXIS
                X2=CO-OD OF SECOND TOP EDGE ON X AXIS
                Y1=CO-OD OF FIRST EDGE ON Y AXIS
                Y2=CO-OD OF SECOND EDGE ON Y AXIS
                Z1=DEPTH TO TOP OF PRISM
                Z2=DEPTH TO BASE OF PRISM
                XN=X CO-OD OF FIRST FIELD POINT
                YN=Y CO-OD OF FIRST FIELD POINT
                ZN=Z CO-OD OF FIRST FIELD POINT
                NBOD= NUMBER OF BODIES

                ALL ANGLES MEASURED IN DEGREES.
                DX=STATION INCREMENT ALONG X AXIS
                THETA =DIP OF PRISM (OR TRAPEZOID ), EDGE NEAREST TO ORIGIN
                ETA=DIP OF PRISM EDGE FURTHEST FROM ORIGIN
                BETA=ANGLE BETWEEN X AXIS AND PROFILE, MEASURED CLOCKWISE
                FROM +VE X AXIS.
                DEN=DENSITY CONSTRAST( C.G.S)
                GC=GRAVITATIONAL CONSTANT (C.G.S)
                NXN=NUMBER OF STATIONS
                AG=ARRAY FOR GRAVITY VALUES
                STAT=ARRAY FOR OBSERVATION POINTS

**********************************************************************

      DIMENSION X(6,10),STAT(400),F1(8),F2(8),F3(8),F4(8),LU(5)
      DIMENSION F5(8),U(2),V(2),W(2),SUMG(400),B(6,400)
      DOUBLE PRECISION GC,PI,X,H,U,V,W,PHI,R,B,XA,YA,YN,ZN
      DOUBLE PRECISION F1,F2,F3,F4,F5,G,SUMG,DX,XN,Q,P
      INTEGER MN,NXN,NBOD,T,Z
      INTEGER STR1(10),STR2(7),STR3(8),STR4(8),STR14(7)
      DATA STR1/2HGR,2HAV,2HIT,2HY  ,2HVA,2HLU,2HE(,2HMG,2HAL,2HS)/
      DATA STR2/2HDE,2HPT,2HH(,2HME,2HTR,2HES,2H)./
      DATA STR3/2HDI,2HST,2HAN,2HCE,2H(M,2HET,2HRE,2HS)/
      DATA STR4/2HMO,2HDE,2HL ,2HNO,2H. ,2H  ,2H  ,2H  /

       READ INPUT DATA

      CALL RMPAR(LU)
1004  WRITE(LU,1005)
1005  FORMAT(/2X,"INPUT GRID PARAMETERS-MN,XN,YN,ZN,DX,NXN")
      READ(LU,*)MN,XN,YN,ZN,DX,NXN
      WRITE(LU,1001)
1001  FORMAT(/2X,"HOW MANY BODIES - MAX NUMBER IS 5?")
      READ(LU,*)NBOD
      DO 99 T=1,NBOD
      IF(T.EQ.1)GO TO 1006
      IF(T.EQ.2)GO TO 1008
      IF(T.EQ.3)GO TO 1010
      IF(T.EQ.4)GO TO 1012
      IF(T.EQ.5)GO TO 1014
1006  WRITE(LU,1007)
1007  FORMAT(/2X,"FIRST BODY DATA-X1,X2,Y1,Y2,Z1,Z2,DEN,ETA,THETA,BETA")
1019  READ(LU,*)X(T,1),X(T,2),X(T,3),X(T,4),X(T,5),X(T,6),X(T,7),
     *X(T,8),X(T,9),X(T,10)
      GO TO 14
1008  WRITE(LU,1009)
1009  FORMAT("  SECOND BODY DATA-X1,X2,Y1,Y2,Z1,Z2,DEN,ETA,THETA,BETA")
      GO TO 1019
1010  WRITE(LU,1011)
```

```
1011 FORMAT(/2X,"THIRD BODY DATA-X1,X2,Y1,Y2,Z1,Z2,DEN,ETA,THETA,BETA")
     GO TO 1019
1012 WRITE(LU,1013)
1013 FORMAT("  FOURTH BODY DATA-X1,X2,Y1,Y2,Z1,Z2,DEN,ETA,THETA,BETA")
     GO TO 1019
1014 WRITE(LU,1015)
1015 FORMAT(/2X,"FIFTH BODY DATA-X1,X2,Y1,Y2,Z1,Z2,DEN,ETA,THETA,BETA")
     GO TO 1019
  14 WRITE(LU,303)
 303 FORMAT(/2X,"MISTAKE IN INPUT DATA?")
     READ(LU,6)NGOOD
   6 FORMAT(A2)
     IF(NGOOD.EQ.2HYE)GO TO 1006

      CONVERT INPUT DATA TO RADIANS, CORRECT DEPTH DATA AND
      SPACE STATIONS CORRECTLY.

     GC=6.67D0/10000 0000 D0
     PI=3.14159D0
     X(T,8)=X(T,8)*PI/180.D0
     X(T,9)=X(T,9)*PI/180.D0
     X(T,10)=X(T,10)*PI/180.D0
     DO 99 L=1,NXN
     XA=XN+(L-1)*DX*DCOS(X(T,10))
     YA=YN+(L-1)*DX*DSIN(X(T,10))
     STAT(L)=XN+(L-1)*DX
     H=X(T,6)-X(T,5)
     U(1)=XA-X(T,2)
     U(2)=XA-X(T,1)
     V(1)=YA-X(T,4)
     V(2)=YA-X(T,3)
     W(1)=ZN-X(T,6)
     W(2)=ZN-X(T,5)

     CALCULATE GRAVITY FUNCTION

     M=1
     G=0.
     DO 10 I=1,2
     DO 10 J=1,2
     DO 10 K=1,2
     IF((DABS(W(K))-0.001D0).LT.0)H(K)=0.001D0
     IF(I.EQ.1)PHI=X(T,8)
     IF(I.EQ.2)PHI=X(T,9)
     IF(K.EQ.1)U(I)=U(I)-H/DTAN(PHI)
     R=DSQRT(U(I)*U(I)+V(J)*V(J)+W(K)*W(K))
     P=U(I)-W(K)/DTAN(PHI)
     IF((DABS(P)-0.001D0).LT.0)P=0.001D0
     Q=W(K)+U(I)/DTAN(PHI)
     F1(M)=W(K)*DATAN((V(J)*U(I))/(W(K)*R))
     F2(M)=-P*DSIN(PHI)*DCOS(PHI)*DATAN((V(J)*Q)/(P*R))
     F3(M)=-P*DSIN(PHI)*DSIN(PHI)*DLOG(V(J)+R)
     F4(M)=-V(J)*DLOG(U(I)+R)
     F5(M)=V(J)*DCOS(PHI)*DLOG(Q*DSIN(PHI)+R)
     IJK=I+J+K
     IF(IJK.EQ.4.OR.IJK.EQ.6)GO TO 19
     F1(M)=-F1(M)
     F2(M)=-F2(M)
     F3(M)=-F3(M)
     F4(M)=-F4(M)
```

```
      FS(M)=-FS(M)
   19 IF(K.EQ.1)U(I)=U(I)+H/DTAN(PHI)
      M=M+1
   10 CONTINUE
      DO 11 N=1,8
      G=G+F1(N)+F2(N)+F3(N)+F4(N)+FS(N)
   11 CONTINUE

      CALCULATION OF GRAVITY VALUE - SUM INDIVIDUAL BODY CONTRIBUTIONS.

      B(T,L)=-100000.D0*GC*X(T,7)*G
      SUMG(L)=SUMG(L)+B(T,L)
   99 CONTINUE

       PRINT RESULTS
      SELECT OUTPUT FORMAT ACCORDING TO NUMBER OF BODIES.

      WRITE(LU,3)MN
    3 FORMAT(1H1,7X,9HMODEL NO.,I10)
      IF(NBOD.EQ.1)GO TO 71
      IF(NBOD.EQ.2)GO TO 72
      IF(NBOD.EQ.3)GO TO 73
      IF(NBOD.EQ.4)GO TO 74
      WRITE(LU,202)
      WRITE(6,202)
  202 FORMAT(1H0,2X,7HSTATION,5X,6HBODY 1,5X,6HBODY 2,5X,6HBODY 3,5X,
     *6HBODY 4,5X,6HBODY 5,5X,5HTOTAL)
      GO TO 77
   71 WRITE(LU,81)
      WRITE(6,81)
   81 FORMAT(1H0,2X,7HSTATION,5X,6HBODY 1,5X,5HTOTAL)
      GO TO 77
   72 WRITE(LU,82)
      WRITE(6,82)
   82 FORMAT(1H0,2X,7HSTATION,5X,6HBODY 1,5X,6HBODY 2,5X,5HTOTAL)
      GO TO 77
   73 WRITE(LU,83)
      WRITE(6,83)
   83 FORMAT(1H0,2X,7HSTATION,5X,6HBODY 1,5X,6HBODY 2,5X,6HBODY 3,
     *5X,5HTOTAL)
      GO TO 77
   74 WRITE(LU,84)
      WRITE(6,84)
   84 FORMAT(1H0,2X,7HSTATION,5X,6HBODY 1,5X,6HBODY 2,5X,6HBODY 3,
     *5X,6HBODY 4,5X,5HTOTAL)
   77 DO 56 L=1,NXN
      WRITE(LU,204)STAT(L),(B(J,L),J=1,NBOD),SUMG(L)
      WRITE(6,204)STAT(L),(B(J,L),J=1,NBOD),SUMG(L)
  204 FORMAT(1H0,2X,F7.2,5X,F6.2,5X,F6.2,5X,F6.2,5X,F6.2,5X,F6.2,
     *5X,F6.2)
   56 CONTINUE
      WRITE(LU,65)
   65 FORMAT(/2X,"PLOT RESULTS?")
      READ(LU,66)NGOOD
   66 FORMAT(A2)
      IF(NGOOD.EQ.2HNO)GO TO 75
  100 WRITE(LU,76)
   76 FORMAT(/2X,"VERT SCALE MGAL/IN, HORIZ SCALE METRES/IN")
      READ(LU,*)SCM,SCD
      SI1=SCM
```

```
 FIND THE DEEPEST BODY

      ZMIN=X(1,5)
      ZMAX=X(1,6)
      DO 750 J=1,NBOD
      IF(X(J,6).GE.ZMAX) ZMAX=X(J,6)
      IF(X(J,5).LE.ZMIN) ZMIN=X(J,5)
  750 CONTINUE
      SV2=ZMAX
      SI2=-ZMAX/3.0
      DIST=SCD
      XT=DX*FLOAT(NXN)/SCD
      IF(XT.GT.19.0)101,103
  101 WRITE(LU,102)
  102 FORMAT(/2X,"PLOT TOO LONG.....CHANGE DISTANCE SCALE")
      GO TO 100
  103 XYZ=XT+18
      CALL PLOTS(XYZ,12.,0)
      CALL FACTR(1.0)

       PLOT AXIS

      CALL PLOT(1.,0.5,-3)
      CALL AXIS(0.,4.,STR1,20,5.,90.,0.,SI1)
      CALL AXIS(0.,0.,STR2,14,3.,90.,SV2,SI2)
      CALL AXIS(0.,3.,STR3,16,10.,0.,XN,DIST)

       PLOT POINTS

      CALL PLOT(0.,4.,-3)
      DO 600 J=1,NBOD
      XP=0.
      YP=0.
      DO 600 I=1,NXN
      YP=B(J,I)/SCM
      CALL SYMB(XP,YP,0.1,0,0.,-1,0.1)
      XP=XP+DX/DIST
  600 CONTINUE
      CALL PLOT(0.,0.,3)
      XP=0.
      DO 488 I=1,NXN
      YP=SUMG(I)/SCM
      CALL SYMB(XP,YP,0.1,7,0.,-2,0.1)
      XP=XP+DX/DIST
  488 CONTINUE

      PLOT THE VARIOUS BODIES.

      BETA=BETA*180./PI
      IF(IFIX(BETA).GT.0.001)GO TO 800
      CALL PLOT(0.,-1.,-3)
      DO 800 J=1,NBOD
      XP1=X(J,1)/DIST-XN/DIST
      YP1=X(J,5)/SI2
      XP2=X(J,2)/DIST-XN/DIST
      XP3=(X(J,2)+(X(J,6)-X(J,5))/DTAN(X(J,8)))/DIST-XN/DIST
      XP4=(X(J,1)+(X(J,6)-X(J,5))/DTAN(X(J,9)))/DIST-XN/DIST
      YP2=(X(J,6)-X(J,5))/SI2
```

```
      CALL PLOT(XP1,YP1,3)
      CALL PLOT(XP2,YP1,2)
      CALL PLOT(XP3,YP2,2)
      CALL PLOT(XP4,YP2,2)
      CALL PLOT(XP1,YP1,2)
  800 CONTINUE
  900 THETA=THETA*180./PI
      ETA=ETA*180./PI
      CALL SYMB(11.,8.0,0.1,STR4,0.,16)
      CALL NUMB(12.8,8.0,0.1,FLOAT(MN),0.,0)

       TERMINATE PLOT AND ASK IF ANY MORE CASES ARE REQUIRED.

      CALL PLOT(0.,0.,999)
      CALL GOPLT
   75 WRITE(LU,20)
   20 FORMAT(/2X,"ANY MORE CASES?")
      READ(LU,30)MORE
   30 FORMAT(A2)
      IF(MORE.EQ.2HYE)GO TO 12
   12 DO 13 L=1,NXN
      SUMG(L)=0
   13 CONTINUE
      IF(MORE.EQ.2HYE)GO TO 1004
   27 STOP
      END
      END$
```

```
TN4,B,L,Q,C
```

```
**********************************
*                                *
*                                *
*                                *
*          PROGRAM   TLAY        *
*                                *
*                                *
*                                *
*                                *
**********************************
```

```
EMA(BLOK1,2)
     PROGRAM TLAY(3,200)

*****************************************************************************

      TITLE     : TLAY - CALCULATION OF THE INTERFACE TOPOGRAPHY IN
                  SEDIMENTARY BASINS.

      AUTHOR    : DONALD WHITEHILL, CITY SERVICE, TULSA, OKLAHOMA. USA.

      MODS      : (a) MAY 1977 - BJORN PAULSSON, ENG. GEOSCIENCE, UNI. of
                  CALIFORNIA, BERKELEY, USA.
                  (b) MARCH 1981 - PETER R GIDLEY, BUREAU of MINERAL RESOURCES
                  CANBERRA,ACT. MODIFIED TO RUN ON HP 2000 SERIES COMPUTER.

      PURPOSE   : THIS PROGRAM MAKES A TWO LAYER, THREE-DIMENSIONAL, MODEL
                  INTERPRETATION. IT CALCULATES THE TOPOGRAPHY OF A BURIED
                  INTERFACE BETWEEN THE TWO CONSTANT DENSITY LAYERS. INPUT
                  IS A TWO DIMENSIONAL ARRAY 'GOBS' OF RESIDUAL GRAVITY
                  DATA. THIS ARRAY IS OBTAINED FROM A DISC FILE.

      USE       : PROGRAM INPUT REQUIRED.
           GOBS(I,J)=GRIDDED GRAVITY VALUES (IN MILLIGALS).
           DX=GRID SIZE IN N-S DIRECTION (IN FEET).
           DY=GRID SIZE IN E-W DIRECTION (IN FEET).
           MAX=MAXIMUM VALUE OF I.
           NAY=MAXIMUM VALUE OF J.
           I=N-S GRID NUMBER WITH I=1 THE SOUTHERN EDGE.
           J=E-W GRID NUMBER WITH J=1 THE WESTERN EDGE.
           ZB=MEAN DEPTH TO INTERFACE BETWEEN LAYERS.
           RHO=DENSITY CONTRAST. BOTTOM LAYER DENSITY-TOP
               LAYER DENSITY.(IN GRAMS/CUBIC CENT.)
           ITMAX=MAXIMUM NUMBER OF ITERATIONS.
           ERMS=RMS DEVIATION TERMINATION CRITERION.
           EMEAN=MEAN ABSOLUTE ERROR TERMINATION CRITERION.
           EMAX=MAXIMUM GRID ERROR TERMINATION CRITERIUN.
           IRROR= 0 - TERMINATES AFTER ITMAX ITERATIONS.
                = 1 - ERMS CONTROLLED TERMINATION.
```

```
          = 2 - EMEAN CONTROLLED TERMINATION.
          = 3 - EMAX CONTROLLED TERMINATION.
      NO VALUES OF GOBS(I,J)=0.0 EXACTLY SHOULD BE INPUT
      FOR AN (I,J) WHICH IS A DATA POINT.
      IF GOBS(I,J) =0.0 THE PROGRAM
      ADJUSTS THE VALUE TO GOBS(I,J) = 0.01
      PROGRAM KEYS ON ZERO FOR Z(I,J) WHEN THERE
      ARE GRID CORNERS FOR WHICH NO DATA POINT
      EXISTS. THE PROGRAM ADJUST THE DEPTH TO
      INTERFACE BELOW EACH (i,J) DATA POINT TO
      OBTAIN A GOOD FIT TO THE DATA WITH A
      MODEL WHICH HAS MEAN DEPTH OF ZB AND
      DENSITY CONTRAST RHO. THE MODEL EXTENDS
      ALL BOUNDARY PRISMS TO INFINITY IN
      HORZONTAL DIRECTION ALONG ROW
      OR COLUMN AWAY FROM DATA. THERE
      SHOULD BE NO ZERO VALUES OF GOBS(I,J)
      EMBEDDED IN DATA SUCH THAT A PATH ALONG
      A ROW OR COLUMN CROSSES DATA
      POINTS AND ESCAPES THE GRID BOUNDARIES.
      FOUR CORNER PRISMS
      WHICH HAVE TWO EDGES REMOVED TO INFINITE
      DISTANCE MUST BE LOCATED. THESE PRISMS
      SHOULD HAVE ONLY THE ONE NON-ZERO
      VALUE FOR GOBS(I,J), GIVEN AT THE
      SPECIFIED LOCATIONS, WITHIN THEIR
      BOUNDARIES.
      (ILB,JLB)-LOCATION OF LEFT-BOTTOM PRISM
           WHICH HAS SOUTHERN AND WESTERN EDGES AT INFINITY.
      (ILT,JLT)-LOCATION OF LEFT-TOP PRISM WHICH HAS
           NORTHERN AND WESTERN EDGES AT INFINITY.
      (IRB,JRB)-LOCATION OF RIGHT-BOTTOM PRISM WHICH HAS
           SOUTHERN AND EASTERN EDGES AT INFINITY.
      (IRT,JRT)-LOCATION OF RIGHT-TOP PRISM WHICH HAS
           NORTHERN AND EASTERN EDGES AT INFINITY.
           DESIGNATION OF NORTH AND EAST DIRECTIONS IS ONLY
           FOR CONVENIENCE AND DOES NOT HAVE TO BE TRUE DIRECTIONS.
      KEY(I,J)-ARRAY SET UP FROM GOBS(I,J) TO
           ESTABLISH BOUNDARY PRISM LOCATIONS.
      ITNUM-START ITERATION NUMBER.
           .EQ.0 - NORMAL USE STARTING WITH GOBS(I,J)
           .LT.0 - CALCULATES GMOD(I,J) FOR SPECIFIC
                Z(I,J) IS INPUT. GOBS(I,J)
                INPUT WITH APPROPRIATE ZEROS TO
                ESTABLISH KEY(I,J).
           .GT.0 - CONTINUES WITH Z(I,J)*S FROM
                PREVIOUS RUN. REQUIRES Z(I,J)
               AND GMOD(I,J) FROM PREVIOUS RUN.
   INPUT ARRAY IS STRUCTURED AS BELOW:

     X11 X12 X13 ..................X1d
     X21 X22     ..................X2c
     :                             :
     :                             :
     :                             :
     Xr1 Xr2     ..................Xrc

   THE PROGRAM READS THE DISC FILE AS:
  col10     col20     col30
col 1-10  col 11-20  col 21-30
```

```
        18        279         15
  X11 X12 X13 X14 ..............Xrc.....
  ..................Xrc
         18

     PROGRAM OUTPUT AFTER EACH ITERATION.
          ITNUM=ITERATION NUMBER.
          DGM=MEAN ABSOLUTE DEVIATION.
          RMSD=RMS DEVIATION
          DGMAX=MAXIMUM ABSOLUTE DEVIATION.
          IMAX,JMAX=GRID LOCATION OF DGMAX.
          Z(I,J)=GRIDDED DEPTHS TO INTERFACE
          ZKM(I,J)=GRIDDED DEPTHS TO INTERFACE IN KILOMETRES
         PRINTED IN A FORMAT WHICH CAN EASILY IS HAND CONTOURED.


          GMOD(I,J)=GRIDDED TWO-LAYER MODEL GRAVITY VALUES.
          GMOD(I,J)-GOBS(I,J) IS PRINTED AFTER LAST ITERATION
     OTHER VARIABLES.
          DZDG=DEPTH CORRECTION FACTOR
          DZBO=BOUGUER SLAB DEPTH CORRECTION FACTOR
              =(2*PI*RHO*G(GRAVITATIONAL CONSTANT)**-1
          DZDG=RBC*DZBO

          AVOID BIGGER GRID THAN 26*26 DUE TO COST, THIS DEMANDS
          200 CP SEC ON A CDC 7600 - ABOUT 1-2 HOURS ON THE HP 2000.
          THE PROGRAM TAKES A LONG TIME TO RUN - DEPENDING ON SIZE
          OF DATA ARRAY, PRIORITY USED, AND NUMBER OF OTHER JOBS
          RUNNING ON THE SYSTEM. AN OPTION IS INCORPORATED INTO
          THE PROGRAM WHICH PERMITS THE USER TO OBTAIN THE RESULTS
          ON THE PRINTER AUTOMATICALLY - THIS PERMITS THE USER
          TO LEAVE THE TERMINAL ONCE HE HAS INPUT THE REQUIRED DATA.

**************************************************************************


     COMMON/BLOK1/GOBS(51,51),GMOD(51,51),ZKM(51,51),KEY(51,51)
     COMMON DX,DY,RHO,ZB
     DIMENSION LU(5),IPS(3),IB(2),ISEG(3),NAME(5),IDAT(2601),Z(51,51)
     REAL IDAT
  10 FORMAT(4F10.3,5A4,I5)
  20 FORMAT(2I5)
  30 FORMAT(F10.2)
  40 FORMAT (I5,2F9.3,F9.2,28X,5A4)
  45 FORMAT(10F8.3)
  50 FORMAT(3I5,5X,3F10.3)
  60 FORMAT(//1X,"DX=",F10.2,"  FEET. N-S GRID WIDTH"/1X,"DY=",F10.2,"
    1 FEET. E-W GRID WIDTH"/1X,"ZB=",F10.2,"  FEET. MEAN DEPTH TO INTER
    2FACE",/1X,"RHO=",F9.2,"  GM/CC. DENSITY CONTRAST"/1X,"MAX =",I5,"
    3NUMBER OF N-S GRID LINES"/1X,"NAY= ",I5," NUMBER OF E-W GRID LINES
    4"/1X,"NUMB=",I5," NUMBER OF GRID GRAVITY VALUES",//)
  61 FORMAT(1X,"ILB=",I4,5X,"JLB=",I4)
  62 FORMAT(1X,"ILT=",I4,5X,"JLT=",I4)
  63 FORMAT(1X,"IRB=",I4,5X,"JRB=",I4)
  64 FORMAT(1X,"IRT=",I4,5X,"JRT=",I4,//,1X,"KEY(I,J) ARRAY")
  70 FORMAT(1X,100I1)
  80 FORMAT(1HO,"MEAN VALUE OF OBSERVED GRAVITY=",E10.3,1X,"MILLIGALS
    1")
  90 FORMAT(1X"ITERATION NUMBER=",I5,/,1X,"MEAN ABS DEV=",E10.3," MIL"
    1"LIGALS",/1X,"RMS DEV=",E10.3," MILLIGALS",/1X,"MAX DEV=",E10.3,
```

```
     2" MILLIGALS",5X,"IMAX=",I5,5X,"JMAX=",I5)
  100 FORMAT(//1X,"OBSERVED GRAVITY VALUES ADJUSTED TO ZERO MEAN")
  110 FORMAT(/15(1X,F7.1))
  115 FORMAT (10(1X,F7.1))
  120 FORMAT(1HO,"BOUGUER CORRECTION=",E10.3," FEET/MILLIGAL"//1X,"MU"
     1,"LTPLY FACTOR=",E10.3,//1X,"DZDG=",E10.3," FEET/MILLIGAL",//)
  130 FORMAT(/1X,"GRIDDED DEPTHS TO INTERFACE")
  140 FORMAT(//1X,"MODEL GRIDDED GRAVITY VALUES")
  150 FORMAT(//1X,"CONVERGENCE CRITERION,R.M.S. DEV. .LE.",E10.3,"MI
     ILLIGALS HAS BEEN SATISFIED")
  160 FORMAT(//1X,"CONVERGENCE CRITERION, MEAN ABS DEV .LE.",E10.3,"
     1MILLIGALS HAS BEEN SATISFIED")
  170 FORMAT(//1X,"CONVERGENCE CRITERION, MAX ERROR .LE.",E10.3," MIL"
     1,"LIGALLS HAS BEEN SATISFIED")
  180 FORMAT(//1X,"THE MAXIMUM NUMBER OF ITERATIONS,",I5,",HAS BEEN REAC
     1HED")
  181 FORMAT(1X,"THE MEAN ABS DEV INCREASED.")
  182 FORMAT(//1X,"THE RMS DEV INCREASED AND DZDG WAS HALVED"/1X,"NEW VA
     1LUE FOR DZDG=",E10.3,//)
  190 FORMAT(//1X,"FINAL GRAVITY ERROR ARRAY, GMOD(I,J: GOBS(I,J)")
  219 FORMAT(1X,"THERE ARE ",I5,"EMBEDDED ZEROS FOR GOBS(I,J)")
  600 FORMAT (1H0,4F10.2,5X,5A4)
  605 FORMAT (1H0,3I5,5X,3F10.2)
  610 FORMAT (1H0,15F8.2)
  620 FORMAT (1H0,45X,"INPUT ARRAY",//)
  701 FORMAT(1X,"CALCUATED VALUE OF Z(",I3,",",I3,") WAS .LE.0.0",/1X,
     1"THE VALUE WAS CHANGED TO Z(I,J) = 1.0"/1X,"THIS CHANGES THE MEAN
     2DEPTH TO INTERFACE",//)
  702 FORMAT(/,1X,".......IN KILOMETRES.......")
  703 FORMAT(//,25(1X,F4.1))

    INPUT ALL THE REQUIRED INFORMATION BEFORE RUNNING JOB

      CALL RMPAR(LU)
  990 CALL INPUD(IDAT,IPS,ISEG,NAME,LU,NXN,LUN)
      WRITE(LU,1000)
 1000 FORMAT(/,"TYPE IN THE FOLLOWING:",/,
     1"    (a)  DISTANCE (IN FEET) BETWEEN STATIONS IN X DIRECTION",/,
     2"    (b)  DISTANCE (IN FEET) BETWEEN STATIONS IN Y DIRECTION",/,
     3"    (c)  ZB  - MEAN DEPTH TO INTERFACE (IN FEET)",/,
     4"    (d)  RHO - DENSITY CONTRAST BETWEEN LAYERS (gm/cc3)")
      READ(LU,*) DX,DY,ZB,RHO
      WRITE(LU,1010)
 1010 FORMAT(/,"TYPE IN THE FOLLOWING:",/,
     1"    (a)   NUMBER OF VALUES ALONG X-AXIS",/,
     2"    (b)   NUMBER OF VALUES ALONG Y-AXIS")
      READ(LU,*) MAX,NAY
      WRITE(LU,1020)
 1020 FORMAT(/,"TYPE IN THE FOLLOWING:.....(see notes)",/,
     1"(ILB,JLB) - LOCATION OF LEFT BOTTOM PRISM:_")
      READ(LU,*) ILB,JLB
      WRITE(LU,1030)
 1030 FORMAT(/,"(ILT,JLT) - LOCATION OF LEFT TOP PRISM:_")
      READ(LU,*) ILT,JLT
      WRITE(LU,1040)
 1040 FORMAT(/,"(IRB,JRB) - LOCATION OF RIGHT BOTTOM PRISM:_")
      READ(LU,*) IRB,JRB
      WRITE(LU,1050)
 1050 FORMAT(/,"(IRT,JRT) - LOCATION OF RIGHT TOP PRISM:_")
      READ(LU,*) IRT,JRT
```

```
      WRITE(LU,1060)
1060  FORMAT(/,"INPUT MAXIMUM NUMBER OF ITERATIONS:_")
      READ(LU,*) ITMAX
      ITNUM = 0
      WRITE(LU,1070)
1070  FORMAT(/,"THE FOLLOWING JOB TERMINATION OPTIONS ARE AVAILABLE:",/,
     1"    (a)  TYPE 0 TO TERMINATE AFTER MAXIMUM ITERATIONS",/,
     2"    (b)  TYPE 1 TO TERMINATE BY RMS DEVIATION",/,
     3"    (c)  TYPE 2 TO TERMINATE BY MEAN ABSOLUTE ERROR",/,
     4"    (d)  TYPE 3 TO TERMINATE BY MAX GRID ERROR - (see notes)")
      READ(LU,*) IRROR
      IF(IRROR.EQ.0) GO TO 1110
      IF(IRROR.EQ.1) GO TO 1080
      IF(IRROR.EQ.2) GO TO 1090
      IF(IRROR.EQ.3) GO TO 1100
1080  WRITE(LU,1120)
1120  FORMAT(/,"INPUT RMS DEVIATION VALUE:_")
      READ(LU,*) ERMS
      GO TO 1110
1090  WRITE(LU,1130)
1130  FORMAT(/,"INPUT MEAN ABSOLUTE ERROR TERMINATION VALUE:_")
      READ(LU,*) EMEAN
      GO TO 1110
1100  WRITE(LU,1140)
1140  FORMAT(/,"INPUT MAXIMUM GRID ERROR TERMINATION VALUE:_")
      READ(LU,*) EMAX
1110  CONTINUE
      WRITE(LU,1150)
1150  FORMAT(/,"ANY MISTAKES ON INPUT OF DATA?")
      READ(LU,1160) IMIS
1160  FORMAT(2A2)
      IF(IMIS.EQ.2HYE) GO TO 990
      WRITE(LU,194)
 194  FORMAT(////,"           ********************************** ",/,
     1"                I AM PROCESSING YOUR REQUEST",/,"
     2PLEASE WAIT",/,"           **********************************")
      IT=1
      IR=0
 192  DO 191 J=1,MAX
      IR=IR+1
      GOBS(IT,J)=IDAT(IR)
 191  CONTINUE
      IT=IT+1
      IF(IT.LE.NAY) GO TO 192

   CONTROL DATA HAS NOW BEEN INPUT - RUN PROGRAM

      INUM = 0
      ZT = 0.0
      WRITE (6,600) DX,DY,ZB,RHO
      WRITE (6,605) MAX,NAY

   RBO IS A MULTIPLIKATION FAKTOR

      RBO=1.0
      WRITE (6,600) RBO
      WRITE (6,620)
      DO 660 I=1,MAX
      WRITE (6,610) (GOBS(I,J),J=1,NAY)
660   CONTINUE
```

```
      DO 650 I=1,MAX
      DO 640 J=1,NAY
      GOBS(I,J)=GOBS(I,J)/10.0
      IF(GOBS(I,J).NE.0.00) GO TO 640
      GOBS(I,J) =GOBS(I,J) +0.01
  640 CONTINUE
  650 CONTINUE

    DEFINE THE BOUNDARIES OF THE ARRAY 'KEY'.

      DO 201 I=1,MAX
      DO 201 J=1,NAY
      KEY(I,J)=1
  201 CONTINUE
      DO 202 I=1,ILB
      DO 202 J=1,JLB
  202 KEY(I,J)=0
      KEY(ILB,JLB)=2
      NZERO=ILB*JLB-1
      DO 203 I=ILT,MAX
      DO 203 J=1,JLT
  203 KEY(I,J)=0
      KEY(ILT,JLT)=4
      NZERO=NZERO+(MAX-ILT+1)*JLT-1
      DO 204 I=IRT,MAX
      DO 204 J=JRT,NAY
  204 KEY(I,J)=0
      KEY(IRT,JRT)=6
      NZERO=NZERO+(MAX-IRT+1)*(NAY-JRT+1)-1
      DO 205 I=1,IRB
      DO 205 J=JRB,NAY
  205 KEY(I,J)=0
      KEY(IRB,JRB)=8
      NZERO=NZERO+IRB*(NAY-JRB+1)-1
      IO=ILB+1
      IF=ILT-1
      IF(IF.LT.IO)GO TO 2006
      DO 206 I=IO,IF
      DO 207 J=1,NAY
      IF(GOBS(I,J).EQ.0.0)GO TO 207
      KEY(I,J)=3
      NZERO=NZERO+J-1
      GO TO 206
  207 KEY(I,J)=0
  206 CONTINUE
 2006 CONTINUE
      IO=IRB+1
      IF=IRT-1
      IF(IF.LT.IO)GO TO 2008
      DO 208 I=IO,IF
      DO 209 JJ=1,NAY
      J=NAY+1-JJ
      IF(GOBS(I,J).EQ.0.0)GO TO 209
      KEY(I,J)=7
      NZERO=NZERO+JJ-1
      GO TO 208
  209 KEY(I,J)=0
  208 CONTINUE
 2008 CONTINUE
      JO=JLB+1
```

```
      JF=JRB-1
      IF(JF.LT.JO)GO TO 2010
      DO 210 J=JO,JF
      DO 211 I=1,MAX
      IF(GOBS(I,J).EQ.0.0)GO TO 211
      KEY(I,J)=9
      NZERO=NZERO+I-1
      GO TO 210
  211 KEY(I,J)=0
  210 CONTINUE
 2010 CONTINUE
      JO=JLT+1
      JF=JRT-1
      IF(JF.LT.JO)GO TO 2012
      DO 212 J=JO,JF
      DO 213 II=1,MAX
      I=MAX+1-II
      IF(GOBS(I,J).EQ.0.0)GO TO 213
      KEY(I,J)=5
      NZERO=NZERO+II-1
      GO TO 212
  213 KEY(I,J)=0
  212 CONTINUE
 2012 CONTINUE
      NUMB=MAX*NAY-NZERO
      WRITE(LUN,60)DX,DY,ZB,RHO,MAX,NAY,NUMB
      WRITE(LUN,61)ILB,JLB
      WRITE(LUN,62)ILT,JLT
      WRITE(LUN,63)IRB,JRB
      WRITE(LUN,64)IRT,JRT
      DO 214 II=1,MAX
      I=MAX+1-II
  214 WRITE(LUN,70) (KEY(I,J),J=1,NAY)
      SOBS=0.0
      DO 215 I=1,MAX
      DO 215 J=1,NAY
      SOBS=SOBS+GOBS(I,J)
      Z(I,J)=ZB
      GMOD(I,J)=0.0
      IF(KEY(I,J).EQ.0)Z(I,J)=0.0
  215 CONTINUE
      SOBS=SOBS/NUMB
      WRITE(LUN,80)SOBS
      DGMS=0.0
      RMSDS=0.0
      DGMAX=0.0
      INBED=0

    ****  THIS OPTION IS NOT AT PRSENT AVAILABLE. IT
    ****  COULD EASILY BE INCLUDED BY THE INCLUSION OF
    ****  A FILE SAVE AND READ ROUTINE.
           HERE YOU READ IN AN ARRAY OF Z(I,J)
           IF YOU HAVE READ IN THE Z(I,J) ARRAY YOU ALSO HERE
           READ IN CORPESPONDING GMOD(I,J) ARRAY
           IF ITNUM IS BEGGER THAN 0.
           IF ITNUM IS LESS THAN 0 . YOU CALCULATE GMOD(I,J)
           FOR SPECIFIC Z(I,J). Z(I,J) IS INPUT. GOBS(I,J)
           IS INPUT WITH APPROPRIATE ZEROS TO
           ESTABLISH KEY(I,J)
```

```
      IF(ITNUM.EQ.0)GO TO 250
      DO 251 I=1,MAX
      READ(5,45)(Z(I,J),J=1,NAY)
  251 CONTINUE
      IF(ITNUM.LT.0)GO TO 350
      DO 252 I=1,MAX
      READ(5,45)(GMOD(I,J),J=1,NAY)
  252 CONTINUE
  250 CONTINUE
      DO 216 I=1,MAX
      DO 216 J=1,NAY
      IF(KEY(I,J).'Q.0)GO TO 216
      IF(GOBS(I,J).EQ.0.0)INBED=INBED+1
      GOBS(I,J)=GOBS(I,J)-SOBS
      DG=ABS(GOBS(I,J))
      DGMS=DGMS+DG
      RMSDS=DGMS+DG**2
      IF(DG.LT.DGMAX)GO TO 216
      DGMAX=DG
      IMAX=I
      JMAX=J
  216 CONTINUE
      IF (INBED.EQ.0)GO TO 218
      WRITE(LUN,219)INBED
      GO TO 553
  218 CONTINUE
      DGMS=DGMS/NUMB
      RMSDS=SQRT(RMSDS/NUMB)
      WRITE(LUN,90)ITNUM,DGMS,RMSDS,DGMAX,IMAX,JMAX
      WRITE(LU,90)ITNUM,DGMS,RMSDS,DGMAX,IMAX,JMAX
      WRITE(LUN,100)
      DO 217 II=1,MAX
      I=MAX+1-II
      WRITE(LUN,110)(GOBS(I,J),J=1,NAY)
  217 CONTINUE
      DZBO=1.0E+4/(6.283185*6.67*3.048*RHO)
      DZDG=RBO*DZBO
      WRITE(LUN,120)DZBO,RBO,DZDG
  900 SMOD = 0.0
      ITNUM=ITNUM+1
      LOOP 300 MAKES ADJUSTMENT TO DEPTH TO INTERFACE.
      DO 300 I=1,MAX
      DO  300 J=1,NAY
      IF(KEY(I,J).EQ.0)GO TO 300
      Z(I,J)=Z(I,J)+DZDG*(GMOD(I,J)-GOBS(I,J))
      GMOD(I,J)=0.0
      IF(Z(I,J).LE.0.0)GO TO 700
      GO TO 300
  700 Z(I,J)=1.0
      WRITE(LUN,701)I,J
  300 CONTINUE
  350 CONTINUE
      LOOP 301 CALCULATES GMOD(I,J) FOR NEW Z(I,J) VALUES.
        THIS LOOP IS THE MAIN CALCULATING OF THE DEPTH ESTIMATES.
        ACCORDING TO ARRAY 'KEY' SUCCESIVE CALLS ARE MADE TO THE
        GRAVITY RESPONSE ROUTINES (GPRIJ....) FOR EACH DATA
        POINT - 1 DATA POINT -> 1 CALL.
      DO 301 K=1,MAX
      DO 301 L=1,NAY
      IF(KEY(K,L).EQ.0)GO TO 301
```

```
      X=(K-1)*DX
      Y=(L-1)*DY
      GSUM=0.0
      DO 302 I=1,MAX
      DO 302 J=1,NAY
      IF(KEY(I,J).EQ.0)GO TO 302
      NTYP=KEY(I,J)
      GO TO (1,2,3,4,5,6,7,8,9),NTYP
    1 CALL GPRIJ(I,J,Z(I,J),X,Y,G)
      GO TO 303
    2 CALL GPR1'(I,J,Z(I,J),X,Y,G)
      GO TO 303
    3 CALL GPRI1(I,J,Z(I,J),X,Y,G)
      GO TO 303
    4 CALL GPRM1(I,J,Z(I,J),X,Y,G)
      GO TO 303
    5 CALL GPRMJ(I,J,Z(I,J),X,Y,G)
      GO TO 303
    6 CALL GPRMN(I,J,Z(I,J),X,Y,G)
      GO TO 303
    7 CALL GPRIN(I,J,Z(I,J),X,Y,G)
      GO TO 303
    8 CALL GPRIN(I,J,Z(I,J),X,Y,G)
      GO TO 303
    9 CALL GPRIJ(I,J,Z(I,J),X,Y,G)
  303 GSUM=GSUM+G
  302 CONTINUE
      GMOD(K,L)=GSUM
      SMOD=SMOD+GSUM
  301 CONTINUE
      IF(ITNUM.LT.0)GO TO 351
      SMOD=SMOD/NUMB
      DGM=0.0
      RMSD=0.0
      DGMAX=0.0
      DO 304 I=1,MAX
      DO 304 J=1,NAY
      IF(KEY(I,J).EQ.0)GO TO 304
      GMOD(I,J)=GMOD(I,J)-SMOD
      DG=ABS(GOBS(I,J)-GMOD(I,J))
      DGM=DGM+DG
      RMSD=RMSD+DG**2
      IF(DG.LT.DGMAX)GO TO 304
      DGMAX=DG
      IMAX=I
      JMAX=J
  304 CONTINUE
      DGM=DGM/NUMB
      RMSD=SQRT(RMSD/NUMB)
      WRITE(LUN,90)ITNUM,DGM,RMSD,DGMAX,IMAX,JMAX
      WRITE(LU,90)ITNUM,DGM,RMSD,DGMAX,IMAX,JMAX
  351 CONTINUE
      WRITE(LUN,130)
      WRITE(LU,130)
      DO 305 II=1,MAX
      I=MAX+1-II
      WRITE(LUN,110)(Z(I,J),J=1,NAY)
      WRITE(LU,110)(Z(I,J),J=1,NAY)
  305 CONTINUE
      WRITE(LU,194)
```

```
   OUTPUT DEPTHS IN KILOMETRES

      WRITE(LUN,130)
      WRITE(LUN,702)
      DO 307 I=1,MAX
      DO 307 J=1,NAY
      ZKM(I,J)=Z(I,J)*.0003048
  307 CONTINUE
      DO 308 II=1,MAX
      I=MAX+1-II
      WRITE(LUN,703)(ZKM(I,J),J=1,NAY)
  308 CONTINUE
      WRITE(LUN,140)
      DO 306 II=1,MAX
      I=MAX+1-II
      WRITE(LUN,110)(GMOD(I,J),J=1,NAY)
  306 CONTINUE
      IF(ITNUM.LT.0)GO TO 553
      IF(IRROR.EQ.0) GO TO 440
      GO TO (410,420,430),IRROR
  410 IF(RMSD.GT.ERMS)GO TO 440
      WRITE(LUN,150)ERMS
      GO TO 550
  420 IF(DGM.GT.EMEAN)GO TO 440
      WRITE(LUN,160)EMEAN
      GO TO 550
  430 IF(DGMAX.GT.EMAX)GO TO 440
      WRITE(LUN,170)EMAX
      GO TO 550
  440 IF(ITNUM.LT.ITMAX)GO TO 500
      WRITE(LUN,180)ITMAX
      GO TO 550
  500 CONTINUE
      IF(DGM.GT.DGMS)WRITE(LUN,181)
      DGMS=DGM
      IF(RMSD.GT.RMSDS)GO TO 501
      RMSDS=RMSD
      GO TO 900
  501 DZDG=DZDG/2.
      WRITE(LUN,182)DZDG
      GO TO 900
  550 WRITE(LUN,190)
      DO 551 I=1,MAX
      DO 551 J=1,NAY
  551 GOBS(I,J)=GMOD(I,J)-GOBS(I,J)
      DO 552 II=1,MAX
      I=MAX+1-II
      WRITE(LUN,110)(GOBS(I,J),J=1,NAY)
  552 CONTINUE
      DO 255 I=1,MAX
      DO 254 J=1,NAY
      IF (Z(I,J).EQ.0.0) GO TO 254
      INUM = INUM + 1
      X = (J-1)*2.0
      Y = (I-1)*2.0
  254 CONTINUE
  255 CONTINUE
      ANY DESIRED PLOT OUTPUT SHOULD BE INSERTED HERE.
  553 CONTINUE
```

```
      WRITE(LU,554)
  554 FORMAT(/////,"          ********************************",/,
     1"          THE JOB HAS FINALLY FINISHED     ",/,"          ******",
     2"********************************")
      STOP
      END
      SUBROUTINE GPR11(I,J,ZT,X,Y,G)

**************************************************************************

      CALCULATES G(X,Y) FOR A VERTICAL PRISM AS DESCRIBED.
      X-NORTH. Y-EAST. Z-VERTICAL DOWN.
      NORTHERN EDGE OF PRISM IS AT X=(I-.5)*DX.
      EASTERN EDGE OF PRISM IS AT Y=(J-.5)*DY
      WESTERN AND SOUTHERN EDGES ARE INFINITELY FAR AWAY.
      ZT IS DEPTH TO TOP OF PRISM, AND ZB IS DEPTH TO BOTTOM.
      ALL DISTANCES MUST BE GIVEN IN FEET.
      RHO IS DENSITY IN GRAMS/CUBIC CENTIMETER
      G(X,Y)=G-MILLIGALS.

**************************************************************************


      COMMON DX,DY,RHO,ZB
      U1=(FLOAT(I)-0.5)*DX-X
      V1=(FLOAT(J)-0.5)*DY-Y
      Z1=ZB
      Z0=ZT
      R111=SQRT(U1*U1+V1*V1+Z1*Z1)
      R110=SQRT(U1*U1+V1*V1+Z0*Z0)
      G=-U1*ALOG((-V1+R110)/(-V1+R111))
     1-V1*ALOG((-U1+R110)/(-U1+R111))
     2-Z0*ATAN(U1*V1/(Z0*R110))+Z1*ATAN(U1*V1/(Z1*R111))
     3+1.570796326B*(Z1-Z0)+Z1*ATAN(U1/Z1)
     4+Z1*ATAN(V1/Z1)-Z0*ATAN(U1/Z0)-Z0*ATAN(V1/Z0)
      SCALE=3.048
      G=G*RHO*6.67*SCALE*1.E-04
      GR=G
      RETURN
      END
      SUBROUTINE GPR1N(I,J,ZT,X,Y,G)


**************************************************************************

      CALCULATES G(X,Y) FOR A VERTICAL PRISM AS DESCRIBED.
      X-NORTH. Y-EAST. Z-VERTICAL DOWN.
      NORTHERN EDGE OF PRISM IS AT X=(I-.5)*DX.
      WESTERN EDGE OF PRISM IS AT Y=(J-1.5)*DY.
      SOUTHERN AND EASTERN EDGES ARE INFINITELY FAR AWAY.
      TOP OF PRISM AT ZT, AND BOTTOM AT ZB.
      ALL DISTANCES MUST BE GIVEN IN FEET.
      RHO IS DENSITY IN GRAMS/CUBIC CENTIMETERS.
      G=G(X,Y) IS GIVEN IN MILLIGALS.

**************************************************************************


      COMMON DX,DY,RHO,ZB
      U1=(FLOAT(I)-0.5)*DX-X
```

```
      V0=(FLOAT(J)-1.5)*DY-Y
      Z0=ZT
      Z1=ZB
      R101=SQRT(U1*U1+V0*V0+Z1*Z1)
      R100=SQRT(U1*U1+V0*V0+Z0*Z0)
      G=-U1*ALOG((-V0+R101)/(-V0+R100))
     1-V0*ALOG((-U1+R101)/(-U1+R100))
     2+Z0*ATAN(U1*V0/(Z0*R100))-Z1*ATAN(U1*V0/(Z1*R101))
     3+1.5707963268*(Z1-Z0)+Z1*ATAN(U1/Z1)
     4-Z0*ATAN(U1/Z0)-Z1*ATAN(V0/Z1)+Z0*ATAN(V0/Z0)
     5+U1*ALOG((Z1*Z1+U1*U1)/(Z0*Z0+U1*U1))
      SCALE=3.048
      G=G*RHO*6.67*SCALE*1.E-04
      GR=G
      RETURN
      END
      SUBROUTINE GPRM1(I,J,ZT,X,Y,G)

**********************************************************************

      CALCULATES G(X,Y) FOR THE VERTICAL PRISM DESCRIBED.
      X-NORTH. Y-EAST. Z-VERTICAL DOWN.
      SOUTHERN EDGE OF PRISM IS AT X=(I-1.5)*DX.
      EASTERN EDGE OF PRISM IS AT Y=(J-.5)*DY.
      NORTHERN AND WESTERN EDGES ARE INFINITELY FAR AWAY.
      TOP IS AT ZT. BOTTOM IS AT ZB.
      ALL DISTANCES MUST BE GIVEN IN FEET.
      RHO IS DENSITY IN GRAMS/CC.
      G=G(X,Y) IS GIVEN IM MILLIGALS.

**********************************************************************

      COMMON DX,DY,RHO,ZB
      U0=(FLOAT(I)-1.5)*DX-X
      V1=(FLOAT(J)-.5)*DY-Y
      Z0=ZT
      Z1=ZB
      R011=SQRT(U0*U0+V1*V1+Z1*Z1)
      R010=SQRT(U0*U0+V1*V1+Z0*Z0)
      G=-U0*ALOG((-V1+R011)/(-V1+R010))
     1-V1*ALOG((-U0+R011)/(-U0+R010))
     2+Z0*ATAN(U0*V1/(Z0*R010))-Z1*ATAN(U0*V1/(Z1*R011))
     3+1.5707963268*(Z1-Z0)+Z1*ATAN(V1/Z1)
     4-Z0*ATAN(V1/Z0)-Z1*ATAN(U0/Z1)+Z0*ATAN(U0/Z0)
     5+V1*ALOG((Z1*Z1+V1*V1)/(Z0*Z0+V1*V1))
      SCALE=3.048
      G=G*RHO*6.67*SCALE*1.E-04
      GR=G
      RETURN
      END
      SUBROUTINE GPRMN(I,J,ZT,X,Y,G)

**********************************************************************

      CALCULATES G(X,Y) FOR THE VERTICAL PRISM DESCRIBED.
      SOUTHERN EDGE OF PRISM IS AT X=(I-1.5)*DX
      WESTERN EDGE OF PRISM IS AT Y=(J-1.5)*DY
      NORTHERN AND EASTERN EDGES ARE INFINITELY FAR AWAY.
      TOP IS AT ZT. BOTTOM IS AT ZB.
      ALL DISTANCES MUST BE GIVEN IN FEET.
```

```
      RHO IS DENSITY IN GRAMS/CC.
      G=G(X,Y) IS GIVEN IN MILLIGALS.

**********************************************************************

      COMMON DX,DY,RHO,ZB
      Z0=ZT
      U0=(FLOAT(I)-1.5)*DX-X
      V0=(FLOAT(J)-1.5)*DY-Y
      Z1=ZB
      R001=SQRT(U0*U0+V0*V0+Z1*Z1)
      R000=SQRT(U0*U0+V0*V0+Z0*Z0)
      G=-U0*ALOG((-V0+R000)/(-V0+R001))
     1-V0*ALOG((-U0+R000)/(-U0+R001))
     2-Z0*ATAN(U0*V0/(Z0*R000))+Z1*ATAN(U0*V0/(Z1*R001))
     3+1.5707963268*(Z1-Z0)-Z1*ATAN(U0/Z1)
     4-Z1*ATAN(V0/Z1)+Z0*ATAN(U0/Z0)+Z0*ATAN(V0/Z0)
     5       -U0*ALOG((Z1*Z1+U0*U0)/(Z0*Z0+U0*U0))
     6       -V0*ALOG((Z1*Z1+V0*V0)/(Z0*Z0+V0*V0))
      SCALE=3.048
      G=G*RHO*6.67*SCALE*1.E-04
      GR=G
      RETURN
      END
      SUBROUTINE GPR1J(I,J,ZT,X,Y,G)

**********************************************************************

      CALCULATES G(X,Y) FOR THE VERTICAL PRISM DESCRIBED.
      X-NORTH. Y-EAST. Z-VERTICAL DOWN.
      NORTHERN EDGE OF PRISM AT (I-.5)*DX.
      EASTERN EDGE OF PRISM AT (J-.5)*DY.
      WESTERN EDGE OF PRISM AT (J-1.5)*DY.
      SOUTHERN EDGE OF PRISM INFINITELY FAR AWAY.
      TOP IS AT ZT. BOTTOM AT ZB.
      ALL DISTANCES ARE IN FEET.
      RHO IS DENSITY IN GRAMS/CC.
      G=G(X,Y) IS GIVEN IN MILLIGALS.

**********************************************************************

      COMMON DX,DY,RHO,ZB
      Z0=ZT
      U1=(FLOAT(I)-0.5)*DX-X
      V1=(FLOAT(J)-0.5)*DY-Y
      V0=(FLOAT(J)-1.5)*DY-Y
      Z1=ZB
      R111=SQRT(U1*U1+V1*V1+Z1*Z1)
      R101=SQRT(U1*U1+V0*V0+Z1*Z1)
      R110=SQRT(U1*U1+V1*V1+Z0*Z0)
      R100=SQRT(U1*U1+V0*V0+Z0*Z0)
      G=-U1*ALOG((-V1+R110)/(-V0+R100)*(-V0+R101)/(-V1+R111))
     1-V1*ALOG((-U1+R110)/(-U1+R111))-V0*ALOG((-U1+R101)/(-U1+R100))
     2-Z0*ATAN(U1*V1/(Z0*R110))+Z0*ATAN(U1*V0/(Z0*R100))
     3+Z1*ATAN(U1*V1/(Z1*R111))-Z1*ATAN(U1*V0/(Z1*R101))
     6+Z1*ATAN(V1/Z1)-Z1*ATAN(V0/Z1)
     7-Z0*ATAN(V1/Z0)+Z0*ATAN(V0/Z0)
      SCALE=3.048
      G=G*RHO*6.67*SCALE*1.E-04
      GR=G
```

```
      RETURN
      END
      SUBROUTINE GPRMJ(I,J,ZT,X,Y,G)

**********************************************************************

      CALCULATES G(X,Y) FOR THE VERTICAL PRISM DESCRIBED.
      X-NORTH. Y-SOUTH. Z-DOWNWARD.
      SOUTHERN EDGE OF PRISM IS AT X=(I-1.5)*DX.
      EASTERN EDGE OF PRISM IS AT Y=(J-.5)*DY.
      WESTERN EDGE OF PRISM IS AT Y=(J-1.5)*DY.
      NORTHERN EDGE OF PRISM IS INFINITELY FAR AWAY.
      ALL DISTANCES MUST BE GIVEN IN FEET.
      TOP IS AT ZT. BOTTOM AT ZB.
      RHO IS DENSITY IN GRAMS/CC.
      G=G(X,Y) IS GIVEN IN MILLIGALS.

**********************************************************************

      COMMON DX,DY,RHO,ZB
      Z0=ZT
      U0=(FLOAT(I)-1.5)*DX-X
      V1=(FLOAT(J)-0.5)*DY-Y
      V0=(FLOAT(J)-1.5)*DY-Y
      Z1=ZB
      R011=SQRT(U0*U0+V1*V1+Z1*Z1)
      R001=SQRT(U0*U0+V0*V0+Z1*Z1)
      R010=SQRT(U0*U0+V1*V1+Z0*Z0)
      R000=SQRT(U0*U0+V0*V0+Z0*Z0)
      G=-U0*ALOG((-V0+R000)/(-V1+R010)*(-V1+R011)/(-V0+R001))
     1+V1*ALOG((U0+R011)/(U0+R010))-V0*ALOG((U0+R001)/(U0+R000))
     2+Z0*ATAN(U0*V1/(Z0*R010))-Z0*ATAN(U0*V0/(Z0*R000))
     3-Z1*ATAN(U0*V1/(Z1*R011))+Z1*ATAN(U0*V0/(Z1*R001))
     6+Z1*ATAN(V1/Z1)-Z1*ATAN(V0/Z1)
     7-Z0*ATAN(V1/Z0)+Z0*ATAN(V0/Z0)
      SCALE=3.048
      G=G*RHO*6.67*SCALE*1.E-04
      GR=G
      RETURN
      END
      SUBROUTINE GPRI1(I,J,ZT,X,Y,G)

**********************************************************************

      CALCULATES G(X,Y) FOR THE VERTICAL PRISM DESCRIBED.
      X-NORTH. Y-EAST. Z-DOWNWARD.
      NORTHERN EDGE OF PRISM AT X=(I-.5)*DX.
      SOUTHERN EDGE OF PRISM AT X=(I-1.5)*DX.
      EASTERN EDGE OF PRISM AT Y=NJ-.5)*DY.
      WESTERN EDGE OF PRISM IS INFINITELY FAR AWAY.
      TOP IS AT ZT. BOTTOM AT ZB.
      ALL DISTANCES MUST BE GIVEN IN FEET.
      RHO IS DENSITY IN GRAMS/CC.
      G=G(X,Y) IS GIVEN IN MILLIGALS.

**********************************************************************

      COMMON DX,DY,RHO,ZB
      Z0=ZT
      U1=(FLOAT(I)-0.5)*DX-X
```

```
      U0=(FLOAT(I)-1.5)*DX-X
      V1=(FLOAT(J)-0.5)*DY-Y
      Z1=ZB
      R111=SQRT(U1*U1+V1*V1+Z1*Z1)
      R011=SQRT(U0*U0+V1*V1+Z1*Z1)
      R110=SQRT(U1*U1+V1*V1+Z0*Z0)
      R010=SQRT(U0*U0+V1*V1+Z0*Z0)
      G=-V1*ALOG((-U1+R110)/(-U0+R010)*(-U0+R011)/(-U1+R111))
     1-U1*ALOG((-V1+R110)/(-V1+R111))-U0*ALOG((-V1+R011)/(-V1+R010))
     2-Z0*ATAN(U1*V1/(Z0*R110))+Z0*ATAN(U0*V1/(Z0*R010))
     3+Z1*ATAN(U1*V1/(Z1*R111))-Z1*ATAN(U0*V1/(Z1*R011))
     6+Z1*ATAN(U1/Z1)-Z1*ATAN(U0/Z1)
     7-Z0*ATAN(U1/Z0)+Z0*ATAN(U0/Z0)
      SCALE=3.048
      G=G*RHO*6.67*SCALE*1.E-04
      GR=G
      RETURN
      END
      SUBROUTINE GPRIN(I,J,ZT,X,Y,G)

**********************************************************************

      CALCULATES G(X,Y) FOR THE VERTICAL PRISM DESCRIBED.
      X-NORTH.  Y-EAST.  Z-DOWNWARD.
      NORTHERN EDGE OF PRISM IS AT X=(I-.5)*DX.
      SOUTHERN EDGE OF PRISM IS AT X=(I-1.5)*DX.
      WESTERN EDGE OF PRISM IS AT Y=(J-1.5)*DY.
      EASTERN EDGE OF PRISM IS INFINITELY FAR AWAY.
      ALL DISTANCES MUST BE GIVEN IN FEET.
      RHO IS DENSITY IN GRAMS/CC.
      TOP IS AT ZT.  BOTTOM AT ZB.
      G=G(X,Y) IS GIVEN IN MILLIGALS.

**********************************************************************

      COMMON DX,DY,RHO,ZB
      Z0=ZT
      U1=(FLOAT(I)-0.5)*DX-X
      U0=(FLOAT(I)-1.5)*DX-X
      V0=(FLOAT(J)-1.5)*DY-Y
      Z1=ZB
      R101=SQRT(U1*U1+V0*V0+Z1*Z1)
      R001=SQRT(U0*U0+V0*V0+Z1*Z1)
      R000=SQRT(U0*U0+V0*V0+Z0*Z0)
      R100=SQRT(U1*U1+V0*V0+Z0*Z0)
      G=-V0*ALOG((-U0+R000)/(-U1+R100)*(-U1+R101)/(-U0+R001))
     1+U1*ALOG((R101+V0)/(R100+V0))-U0*ALOG((R001+V0)/(R000+V0))
     2+Z0*ATAN(U1*V0/(Z0*R100))-Z0*ATAN(U0*V0/(Z0*R000))
     3-Z1*ATAN(U1*V0/(Z1*R101))+Z1*ATAN(U0*V0/(Z1*R001))
     6+Z1*ATAN(U1/Z1)-Z1*ATAN(U0/Z1)
     7-Z0*ATAN(U1/Z0)+Z0*ATAN(U0/Z0)
      SCALE=3.048
      G=G*RHO*6.67*SCALE*1.E-04
      GR=G
      RETURN
      END
      SUBROUTINE GPRIJ(I,J,ZT,X,Y,GR)

**********************************************************************
```

```
      CALCULATES G(X,Y) FOR A VERTICAL PRISM CENTERED AT X=NI-1)*DX, Y=(
      J-1)*DY,WITH WIDTHS DX,DY, DEPTH TO TOP Z0, DEPTH TO BOTTOM Z1, DE
      NSITY RHO GM/CC. DISTANCES ARE IN FEET, G(X,Y) IS IN MILLIGALS.
      X-DIRECTION IS NORTH. Y DIRECTION IS EAST. Z-DIRECTION IS DOWNWARD

*************************************************************************

      COMMON DX,DY,RHO,ZB
      U1=(FLOAT(I)-.5)*DX-X
      U0=(FLOAT(I)-1.5)*DX-X
      V1=(FLOAT(J)-.5)*DY-Y
      V0=(FLOAT(J)-1.5)*DY-Y
      Z1=ZB
      Z0=ZT
      R111=SQRT(U1*U1+V1*V1+Z1*Z1)
      R101=SQRT(U1*U1+V0*V0+Z1*Z1)
      R011=SQRT(U0*U0+V1*V1+Z1*Z1)
      R001=SQRT(U0*U0+V0*V0+Z1*Z1)
      R110=SQRT(U1*U1+V1*V1+Z0*Z0)
      R100=SQRT(U1*U1+V0*V0+Z0*Z0)
      R010=SQRT(U0*U0+V1*V1+Z0*Z0)
      R000=SQRT(U0*U0+V0*V0+Z0*Z0)
      G=-U1*ALOG((-V1+R110)/(-V0+R100)*(-V0+R101)/(-V1+R111))
     1-U0*ALOG((-V0+R000)/(-V1+R010)*(-V1+R011)/(-V0+R001))
     2-V1*ALOG((-U1+R110)/(-U0+R010)*(-U0+R011)/(-U1+R111))
     3-V0*ALOG((-U0+R000)/(-U1+R100)*(-U1+R101)/(-U0+R001))
     4-Z0*ATAN(U1*V1/(Z0*R110))
     5+Z0*ATAN(U1*V0/(Z0*R100))
     6+Z0*ATAN(U0*V1/(Z0*R010))
     7-Z0*ATAN(U0*V0/(Z0*R000))
     8+Z1*ATAN(U1*V1/(Z1*R111))
     9-Z1*ATAN(U1*V0/(Z1*R101))
     A-Z1*ATAN(U0*V1/(Z1*R011))
     B+Z1*ATAN(U0*V0/(Z1*R001))
      SCALE=3.048
      G=G*RHO*6.67*SCALE*1.E-04
      GR=G
      RETURN
      END
      SUBROUTINE INPUD(IDAT,IPS,ISEG,NAME,LU,NXN,LUN)

*************************************************************************

       THIS PROGRAM DRAWS DATA FROM A DISC FILE
       AND THEN OFFERS:
        1.  PLOT OF ORIGINAL DATA
        2.  OUTPUT OF DATA VALUES TO CHECK SEGMENT etc.

*************************************************************************

      DIMENSION IDCB(144),IBUF(40),NAME(3),ISEG(3),ISGNO(3),IDAT(2601)
      DIMENSION LU(5),IPS(3),IB(2),IFORM(40)

      REAL IDAT
      READ IN FILE DATA

      NAME(2)=2H
      NAME(3)=2H
      DO 100 I=1,2601
      IDAT(I)=0.0
```

```
  100 CONTINUE
      WRITE(LU,299)
  299 FORMAT(/,"ENTER FILE NAME")
      READ(LU,300) (NAME(I),I=1,3)
  300 FORMAT(3A2)
      WRITE(LU,400)
  400 FORMAT(/,"ENTER SECURITY CODE & CARTRIDGE NUMBER")
      READ(LU,*)ISC,ICR
      IF(ISC.LT.0) STOP
      IF(ICR.LE.0) ICR=9
      CALL OPEN(IDCB,IERR,NAME,0,ISC,ICR)
      IF(IERR.GE.0) GO TO 600
      WRITE(LU,500) IERR
  500 FORMAT(/,"ERROR ",I3," ON OPEN")
      STOP
  600 WRITE(LU,700)
  700 FORMAT(/,"WHAT DATA SEGMENT IS REQUIRED - eg ____15 ie SIX SPACES")
      READ(LU,300) ISGNO
  800 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR.LT.0) GO TO 1500
      CALL CODE
      READ(IBUF,900) IFLAG,FLGT,ISEG
  900 FORMAT(2I10,4X,3A2)
      DO 1000 I=1,3
      IPS(I)=ISGNO(I)
      IF(ISGNO(I).NE.ISEG(I)) GO TO 800
 1000 CONTINUE
      WRITE(LU,1100)
 1100 FORMAT(/,"WHAT IS FORMAT (F-TYPE) OF DATA--(E.G.(12F6.0))?")
      READ(LU,301) IFORM
  301 FORMAT(40A2)
      WRITE(LU,1155)
 1155 FORMAT(/,"HOW MANY DATA VALUES PER RECORD?")
      READ(LU,*) INO
      WRITE(LU,1160)
 1160 FORMAT(/,"DO YOU WANT THE RESULTS PRINTED AUTOMATICALLY?")
      READ(LU,300) IPNT
      LUN=6
      IF(IPNT.EQ.2HYE) LUN=16
      DO 1200 I=1,2600,INO
      DO 1250 J=1,40
      IBUF(J)=0
 1250 CONTINUE
      CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR.LT.0) GO TO 1500
      IF(LEN.LT.0) GO TO 1600
      CALL CODE
      READ(IBUF,900) IFLAG
      IF(IFLAG.EQ.18) GO TO 1600
      NXN=I+INO-1
      CALL CODE
      READ(IBUF,IFORM)(IDAT(J),J=I,NXN)
 1200 CONTINUE
      WRITE(LU,1300) ISEG
 1300 FORMAT(/,"MORE THAN 2600 PTS IN ",3A2)
 1600 CALL CLOSE(IDCB,IERR)
 1450 NUM=NXN
      DO 1480 I=1,NUM
      IF(IDAT(I).EQ.0.0) NXN=NXN-1
 1480 CONTINUE
```

```
      FILE DATA NOW IN,DO YOU WANT A PLOT OR WRITE IT?"

      WRITE(LUN,1151) NXN
      WRITE(LU,1151) NXN
      WRITE(LU,51)
   51 FORMAT(/,"DO YOU WANT TO PRINT DATA?",
     2/,"----------IF YES , TYPE IN NUMBER OF VALUES REQUIRED",
     3/,"----------IF NO TYPE IN 'NO'.")
      READ(LU,820) IB
      IF(IB.EQ.2HNO) GO TO 833
  820 FORMAT(2A2)
      CALL CODE
      READ(IB,*) NUT
      WRITE(LU,831) (IDAT(I),I=1,NUT)
  831 FORMAT(8(F8.2,2X))
 1151 FORMAT(/1X,"THERE WERE A TOTAL OF ",I5," DATA POINTS READ.")
  833 WRITE(LUN,115) ISGNO
  115 FORMAT(/,"**********INPUT DATA FROM LINE ",3A2," *********")
      IF(IB.EQ.2HNO) GO TO 923
      WRITE(LUN,112) (IDAT(I),I=1,NUT)
  112 FORMAT(13(F8.2,2X))

  923 RETURN
 1500 WRITE(LU,1510) IERR
 1510 FORMAT(/,"THERE HAS BEEN A READ-FROM-FILE ERROR-IERR=",I4)
      CALL CLOSE(IDCB,IERR)
      RETURN
      END
      END$
```

END OF ALL PROGRAM DESCRIPTIONS