# MAGDEP - A Computer Program for Finding Depth To Basement From Total Field Marine Magnetic Profiles

By

Alan K. Cooper

USGS Open File 1974
74-1017

# INTRODUCTION

The purpose of the Fortran program MAGDEP is to provide a method for finding the depth to two-dimensional magnetic source bodies when given only the total field magnetic intensity profile. The assumptions made in this depth analysis are that the causitive source body is:

1. A two-dimensional structure
2. Uniformly magnetized
3. A finite or infinite polygon in cross section

Unlike other methods for magnetic depth determinations which assume a given body shape, this technique allows the shape of the source body to emerge during the analysis. The final result is a sequence of spot depths which locate the position and depth to the corners of the magnetic source body. The correct outline of the body is controlled by the shape of the magnetic intensity profile which has been reduced to the pole. This reduced polar profile is also determined by program MAGDEP. Two depth analyses are made by the program; one for thick polygonal bodies and the other for thin (thickness less than depth) or narrow bodies. The determination of the correct magnetic source body shape depends upon a careful interpretation of the results from both depth analyses.

## Theoretical Aspects

The method used by program MAGDEP is based on the theoretical work of Nabighian (1972). The reader interested in the details of the theory should refer to his paper ; this section will only outline the theoretical aspects which have been applied to program MAGDEP. Nabighian (1972) shows that all bodies of polygonal cross section can be obtained by the superposition of a finite number of magnetized steps. He gives the magnetic attraction for such a magnetized step as $\Delta M(x)$ (in eq. 1) and then uses it as the starting point for the analysis. Differentiating the attraction of the magnetized step with respect to x yields the horizontal derivative and this represents the magnetic anomaly due to a thin infinite sheet: $T(x)=\frac{\partial}{\partial x}\Delta M(x)$. The vertical derivative of the magnetized step is obtained by differentiating with respect to y: $T_1(x)=\frac{\partial}{\partial y}\Delta M(x)$. A comparison of the two derivatives shows that $T(x)$ is the negative Hilbert transform of $T_1(x)$. When one forms the complex analytic signal function from the two derivatives: $A(x)=T(x) - i\,T_1(x)$ many interesting properties arise. Those properties which are used by MAGDEP include:

1. The amplitude $|A(x)|$ is a symmetric function with respect to $x=0$.
2. The complex function $A(x)$ has simple poles at each corner of the polygon.
3. The bell-shaped function $a(x) = |A(x)|^2 = T(x)^2 + T_1(x)^2$ has the property that its half-maximum half-width is equal to the depth to the polygon corner.

From this brief introduction to the theory, the procedure for computing the depths to the magnetic source bodies becomes more defined. In the depth analysis for thick source bodies:

1. The total field intensity profile (residual values) is differentiated with respect to x to yield T(x).

2. A fourier transformation, spectrum modification, and inverse fourier transformation is performed to obtain the complex analytic signal function: $A(x) = T(x) - iT_1(x)$

3. The amplitude square of the signal function is computed to give: $a(x) = T(x)^2 + T_1(x)^2$. The deconvolution of the a(x) curve into a finite number of bell-shaped symmetric curves gives the locations and depths to the corners of the magnetic source body.

The depth analysis for thin (thickness less than depth) magnetic source bodies is accomplished by the same procedure with the exception that in step 1, the total field intensity profile is not differentiated.

In the general case, where the relief of the magnetic source body is irregular, each corner of the body will contribute a symmetric bell-shaped component to the $a(x) = T(x)^2 + T_1(x)^2$ curve. To identify each of these symmetric components, it is necessary to deconvolve the a(x) curve using a nonlinear least square optimization routine. The procedure used by MAGDEP was derived by Marquardt (1963) and is similar to the technique described by Johnson (1969). See Appendix I for more complete details. The amplitude square curve can also be written as: $a(x) = \frac{\alpha^2}{h^2+(x-c)^2}$ where $\alpha$ is a proportional to the magnetic susceptibility and the dip of the polygon side, h is the depth to the polygon corner, x is the dependent position variable, and c is the location of the polygon corner along the profile. Each corner of the magnetic source body will contribute a bell-shaped symmetric curve controlled by the three parameters $\alpha$, h, and c. At the start of the least square optimazation process, initial values must be assigned to each of the parameters; the 'goodness of fit' of the final solution depends to a large extent upon the quality of the initial guesses. Program MAGDEP uses the following procedure for locating the symmetric bell-shaped curves and determining the initial values for each of the parameters (no. of parameters = 3 x no. of bell-shaped curves):

1. All local maxima of the a(x) curve are checked to see if:
   a. The a(x) value is larger than an arbitrary cutoff value.
   b. The second derivative of the a(x) curve is large enough. (controlled by maximum expected source depth ZMAXKM)
2. If step 1 is true, the initial parameter values are found according to:
   a. h(depth): find the local half-maximum half-width from the a(x) curve
   b. $\alpha$: use the a(x) value at the peak of the bell-shaped curve $(=\alpha^2/h^2)$
   c. c (position): use the x-position of the peak of the bell-shaped curve

After the initial symmetric curves are located, the least square routine is used to obtain a trial solution. The residual values (a(x) - computed values) are then examined by a similar procedure to determine whether more bell-shaped curves are present. The final solution is the 'best fit' of the a(x) curve by the least squares computed curve (using all bell-shaped curves). Depths and locations of the magnetic source body corners are derived from this final least squares approximation.

## Using the Program

Program MAGDEP has been designed so that it can be used in one or two sections depending upon the desires of the user. The first section computes the a(x) curve and the magnetic profile reduced to the pole; the second section does the least squares analysis of the a(x) curve (see theory section). Experience working with MAGDEP has shown that section 2 generally is from 5 to 15 times more time consuming (and expensive) than section 1 and under some circumstances, where high precision is not required, satisfactory results can be obtained by omitting section 2 and using only a visual inspection of the a(x) curve. A second point is that since computation time in the least squares routine increases rapidly with the addition of each bell-shaped curve, the user should consider examining the a(x) curve for each new profile prior to using the least squares routine. Knowing the approximate number of bell-shaped curves present will not only save computation time but will also reduce the chances of obtaining a poor solution because too few bell-shaped curves were specified by the user.* With these notes of warning, the following procedure is suggested for the new user of MAGDEP:

1. Read the theory section in this program description of MAGDEP to become familiar with the terminology used here
2. Look over the program output given in the example (Appendix II)
3. Use the program thru section 1 to obtain the plot of the a(x) curve
4. Examine the a(x) curve to get a feel for the number and type of bell-shaped curves present. The shapes and sizes of the curves present will give a rough estimate of the depths to the body

*Example of computation time for 100 data points on an IBM 360/70 system:
Section 1 only: 4 sec. execution
Section 2 only: thick body analysis only, 7 bell-shaped curves, 4 LSQR iterations: 20 sec. execution

5. Refer to the do's and don'ts section for help in deciding whether the profile should be broken into smaller segments for analysis.
6. Resubmit those parts of the profile which require precision depth estimates using the least squares routine.

Input constants required: (must be right-justified)

| | | | | |
|---|---|---|---|---|
| Card 1: (required) (20A4) | Cols: | 1-80 | Heading | |
| Card 2: (required) (F10.1,2I10) | Cols: | 1-10 | SPDKT | : ship speed in knots |
| | | 11-20 | ID | : data interpolation flag<br>=1 want data to be interpolated (must use card 4)<br>=2 use input data as is. (data must be evenly spaced) |
| | | 21-30 | ISECT | : Number of program sections to use (see above)<br>=1 compute a(x) curve and profile reduced to the pole (optional)<br>=2 do this and use the least squares routine |
| Card 3: (required) (I10,F10.1,2I10,F10.3) | Cols: | 1-10 | IBOD | : Types of bodies to do analysis for<br>=1 Thick bodies only<br>=2 Thin bodies only<br>=3 Both thick and thin bodies |
| | | 11-20 | ZMAXKM | : Maximum estimated depth to the magnetic source body in kilometers |
| | | 21-30 | IPOLE | : Profile reduced to the pole flag<br>=0 do not want the reduced profile<br>=1 do want the reduced profile (must use Card 5) |

| | | | |
|---|---|---|---|
| | Cols: 31-40 | INPLT | : Request for the initial plot containing the a(x) curve, the initial parameter curve, and the residual total field profile. (Recommended as being useful)<br>=0 do not want the plot<br>=1 do want the plot |
| | 41-50 | CUTAX | : A factor which is to be multiplied by the internal value of the a(x) curve cutoff point. This is used in determining whether local maxima of the a(x) curve are to be considered as bell-shaped curves. (Recommend using CUTAX = 1. on first run) |
| Card 4: (use only if ID = 1)<br>(2F10.2, F10.1, I10) | Cols: 1-10 | XSHR | : Starting point for the data interpolation in hours of elapsed time from the start of the profile (must be ≥ 0.) |
| | 11-20 | XEHR | : Ending point for the data interpolation in hours of elapsed time from the start of the profile (can be larger than the time of the last data point) |
| | 21-30 | DXMIN | : New time increment at which data is to be sampled (in minutes) (recommend DXMIN = 2. or 5. for ships magnetic profiles) |
| | 31-40 | NP | : Maximum number of points desired to be in the data set used for the depth analysis (must be less than or equal to 200 - the absolute maximum number of data points allowed) |

Card 5: (use only if IPOLE = 1)
(2F10.1) Cols: 1-10 FINC : Inclination of the earth's magnetic field where the magnetic profile was measured (in degrees)

Cols: 11-20 ANG : Angle between magnetic north and the ship's heading (in degrees where 180 > ANG > 0)

Card 6: (use only if ISECT = 2)
(4I10,F10.3) Cols: 1-10 NRES : The number of residual checks which are to be made during the execution of section 2. The normal cycle is: 1st sqr.-residual check - 1st sq - residual check - 1st sq - etc., depending upon NRES. (Recommen initial use of NRES=1)

11-20 NPEAKD : Thick body analysis: The number of bell-shaped curves which are to be used by the least square routine during its first series of iterations. If NPEAKD=0, all bell-shaped curves (max. of 10) with peak values greater than(AXCUT * internal cutoff) will be used. NPEAKD must be less than or equal to 15. (Recommend either: 1. use the two step procedure described above to determine NPEAKD or 2. use NPEAKD=0)

21-30 NPEAKT : Thin body analysis: (same description, requirements, and recommendations as for NPEAKD)

31-40 IOUT : Flag to control the extra output for the values for: 1. the least square approximation and 2. the residual values (a(x) - computed)
=0 no output is desired
=1 output is desired
(Recommend initial use of IOUT=1)

| | | | |
|---|---|---|---|
| | Cols: 41-50 | CUTRSD | : A factor which is to be multiplied by the internal value of the residual curve cutoff point. This is an arbitrary criteria used in determining whether a local maxima in the residual curve is large enough to allow another bell-shaped curve to be added. (Recommend using CUTRSD=1. on the first run) |
| Card 7: (use only if ISECT=2) (8I10) | Cols: 1-10 (required) | ITIR(1) | : The number of iterations of the least squares routine which are to be made during the first call of MARQ1. (MARQ1 is the name used for the least squares routine) (recommend using ITIR(I) ≤ 3 on first run) |
| | 11-20 (only if NRES>0) | ITIR(2) | : The number of iterations of the least squares routine which are to be made after the first residual check and during the second call of MARQ1. |
| | (as required up to ITIR(NRES+1)) | | : The number of iterations of the least squares routine which are to be made after the NRES residual check and during the NRES+1 call of MARQ1. |

Card 8 to end of data set: Input of time and total field values
(T60,I2,T6,I3,I2,2F2.0,F5.0)

| | | |
|---|---|---|
| 6-8 | ID | : Julian day (sequential starting from 1 on January 1) |
| 9-10 | IH | : Hour (based on 24 hour day) |
| 11-12 | AM | : Minute |
| 13-14 | AS | : Second |

| | | |
|---|---|---|
| Cols: 15-19 | TOTMAG | : Measured total field intensity value. (The program will subtract a constant field (= average field value) from all measured total field values. If residual field intensity values are used, a minor change will have to be made to the main program (those statements directly following the data interpolation section)) |
| 60-61 | IY | : Year |

<u>Note</u>: Data cards of a different format can readily be used by changing statement 100 in Subroutine INPUT to fit the new data format.

| | CARD 2 | | | CARD 3 | | | | | CARD 4 | | | | 5 | CARD 6 | | | | | CARD 7 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RUN | SPDKT | ID | ISECT | IBOD | ZMAXKM | IPOLE | INPLT | CUTAX | XSHR | XEHR | DXMIN | NP | FINC / ANG | NRES | NPEAKD | NPEAKT | IOUT | CUTRSD | ITIR (1) | ITIR (2) | ITIR (3) etc | COMMENTS |
| 1 | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | | | | | | | | | | |

## Some Do's and Don'ts

At the time of writing of this 'suggestion' section, the program MAGDEP has had limited use; consequently, many of the following suggestions may eventually become obsolete or possibly even misleading. It is hoped that helpful additions and deletions will be made to this description as they are discovered. The current state of the art in computers and algorithms to do non-linear least squares requires a great deal of computing time and memory storage space. It is thus advantageous for the user of MAGDEP and other similar programs, to take a few extra precautions to insure a less expensive and higher quality solution on the first attempt. The two step procedure, as described earlier, is a worthy precaution and consequently a highly recommended DO. Some observations concerning the operation of the least squares routine, which might be useful are:

### Least Squares

1. The better the initial guesses to the parameter values, the faster the routine will reach a good solution. It should be noted that the two techniques described by Nabighian (1972) for determining trial depth values (Eqs. 19 and 21) were attempted and found to give adequate results. Unfortunately, occasional bad depth estimates (probably the result of noise in the spectral transformations) were encountered which adversely affected the least square solution. The current method for finding trial depths, although basically simple, has provided more reliable trial values for the least square solution.

2. The least squares routine converges most rapidly for the amplitude of the symmetric curve, less rapidly towards the width of the curve, and least rapidly for the position of the curve.

3. The rapid amplitude convergence also means that smaller amplitude curves adjacent to or on the flanks of large amplitude curves, are often wiped out during the first series of iterations of the least square routine. These small peaks seem to have a better chance for survival after the large peaks are 'locked in'. The residual analysis has been designed to restore these small peaks and other less obvious ones so that they will be included in the depth analysis.

4. To improve a solution when there is a region of small amplitude symmetric curves near an area of large amplitude peaks, it is often best to break the profile into two parts and analyze them separately.

5. Since the number of multiplications for I iterations of the least square routine is approximately equal to $13 \times I \times K \times N$, where K is the number of parameters (i.e., 3 x number of bell-shaped curves) and N is the number of data points, it is advantageous to either:
   a. Use the largest possible sample interval which does not jepordize the depth resolution or
   b. use a smaller sample interval for only a short time period. Both possibilities will minimize the total number of data points.

A list of some other do's and don'ts includes:

Data Input and Interpolation

| DO | DON'T |
| --- | --- |
| 1. Do modify the subroutine INPUT to accommodate a different data input format if necessary. | 1. Don't input residual magnetic values without first changing three statements in the main program. |
| 2. Do interpolate raw data to the largest sample interval which will not affect the depth resolution. (max resolution is probably about 1/2 the sample interval) | 2. Don't use unevenly sampled data without using the data interpolation option. |
| 3. Do use raw data sampled at the closest interval possible. It is far better to interpolate to a larger sample interval from raw data than to use fewer data cards. | 3. Don't use more than 200 input raw data cards or request more than 200 interpolated data points. |
| 4. Do check the sample listing of the raw data and interpolated data to insure against any input reading errors. | |

## Computing a(x) and initial deconvolution

**DO**

1. Do request initial plot on the first run for any new data set (INPLT=1).

2. Do use a two step procedure and for the first step let the program decide the number of bell-shaped curves to use (let NPEAKT and NPEAKD = 0).

3. Do consider carefully what the maximum source depth might be ± 2 km (i.e., don't guess wildly and set to 100).

4. Do examine the plot of the a(x) curve carefully for all symmetric curves which might be detected visually.

5. Do break up a data profile into smaller segments or change the sample interval if the a(x) curve has groups or areas of high and low amplitude peaks.

**DON'T**

1. Don't request more than 15 bell-shaped curves (i.e., don't let NPEAKD or NPEAKT = 16 or more).

2. Don't request a large number (>7) of bell-shaped curves on a first run, unless the least squares routine will not be used.

3. Don't set the cutoff for the a(x) curve too low (i.e., AXCUT); otherwise noise becomes a problem.

4. Don't be afraid to request a plot of the profile reduced to the pole - its cheap.

5. Don't worry if the initial parameter curve looks like a poor fit to the a(x) curve - least squares does a good job.

## Least squares and residual analysis

**DO**

1. Do remember that the time required for each least squares analysis will be proportional to the number of multiplications performed (about 13 x # iter x # data pts x # parameters).

**DON'T**

1. Don't use a cutoff value for the residuals which is too low (i.e., RSDCUT), it may cause a false symmetric curve to be added which may affect the least squares analysis. The internal cutoff has worked all right so far (RSDCUT=L).

2. Do use at least one residual analysis.

3. Do use enough iterations (min. of 3) on the first least square analysis to allow the residual values to develop adequately.

4. Do expect that the least square solution may 'seek' or oscillate through the correct value from iteration to iteration.

2. Don't be discouraged if a seemingly poor fit is obtained after only a few iterations. The program will inform you if major problems arise and will give words of encouragement and advice.

## Flow Diagrams

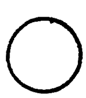
The detailed flow diagrams for the entire program and some of the more important subroutines are included for two reasons:

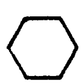
1. To give the interested (or confused) user an opportunity to look into the 'black box' of MAGDEP

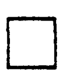
2. To assist in trouble shooting should difficulties arise

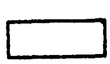
For the purpose of illustration, the entire program has been broken into the two sections which were described previously. The physical layout of the program is such that this separation into two sections is not readily apparent. Those subroutines which do not have a detailed flow diagram included here are indicated by dashed boxes; those with flow diagrams are indicated by double solid boxes. The name of the subroutine lies in one corner of the outer most box.

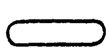
<u>Explanation for flow diagrams</u>:

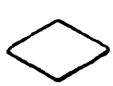
 Start and stop

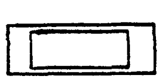
 Return from subroutines

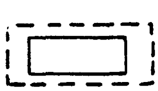
 Go to

 Computational procedure

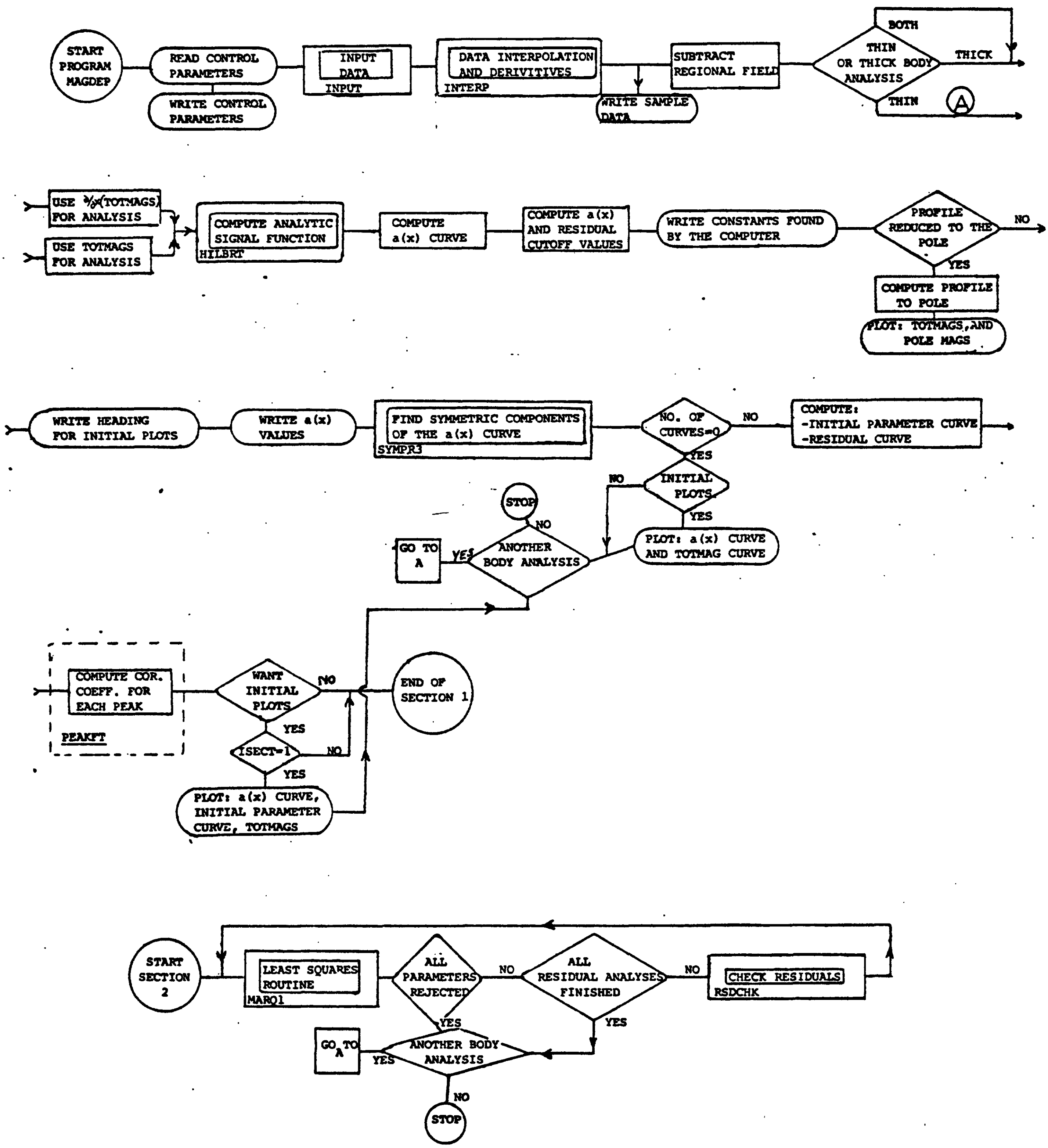
 Input and output

 Decision

 Description of a subroutine which has a detailed flow diagram

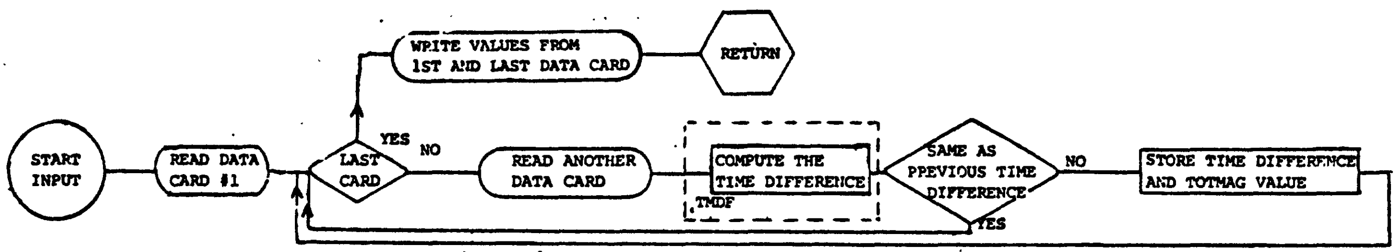
 Description of a subroutine which does not have a detailed flow diagram

Flow diagram of entire program MAGDEP:

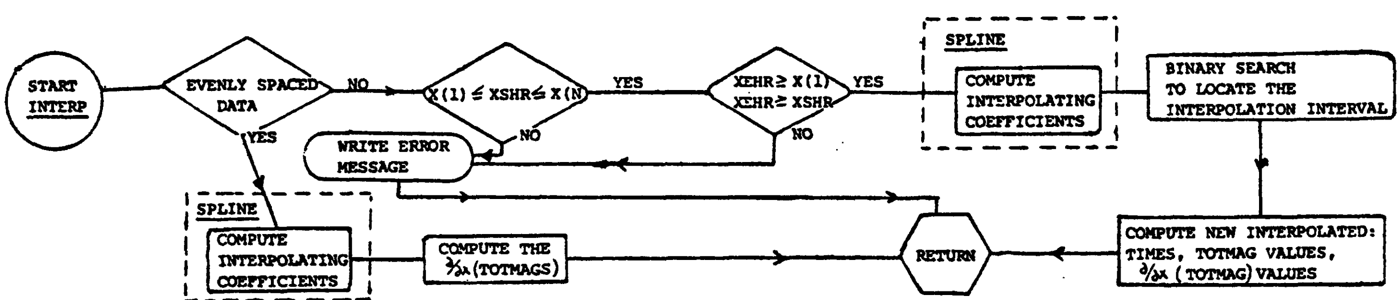


Flow diagrams of important subroutines:

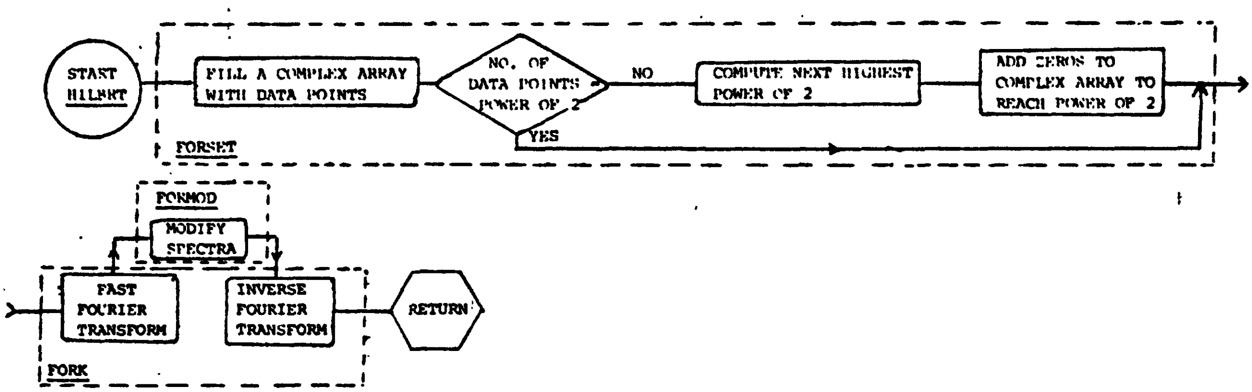
SUBROUTINE INPUT



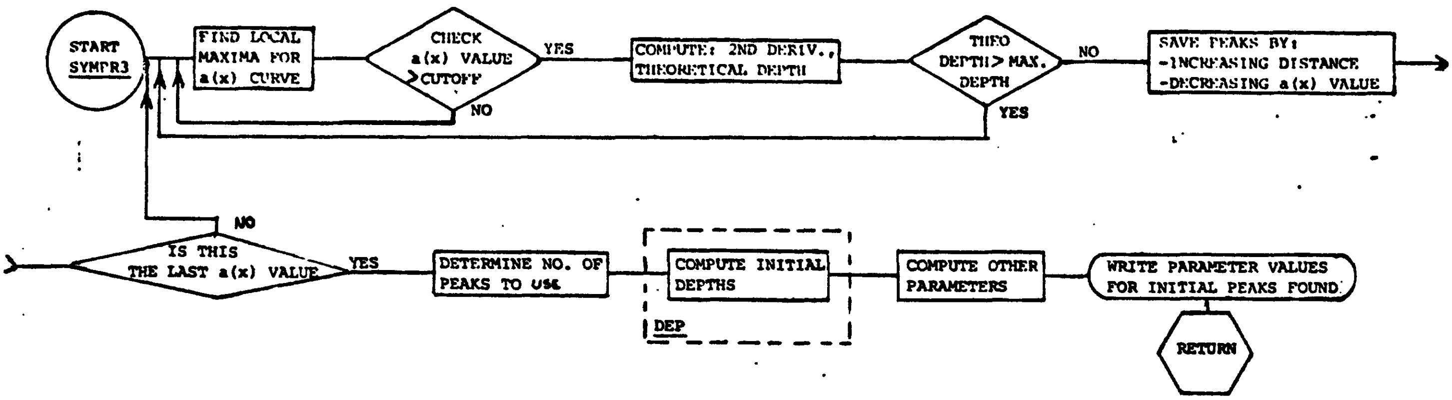

SUBROUTINE INTERP



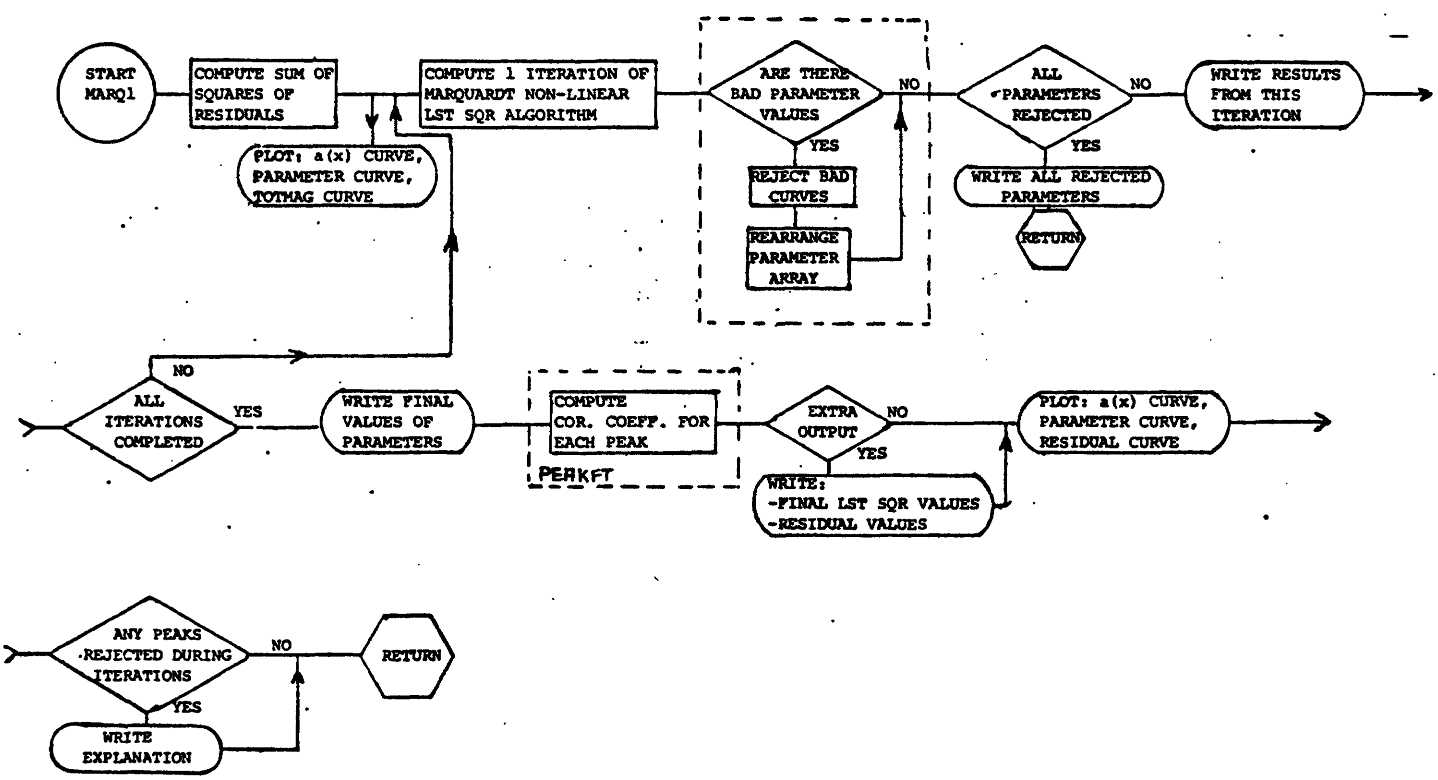

SUBROUTINE HILBRT



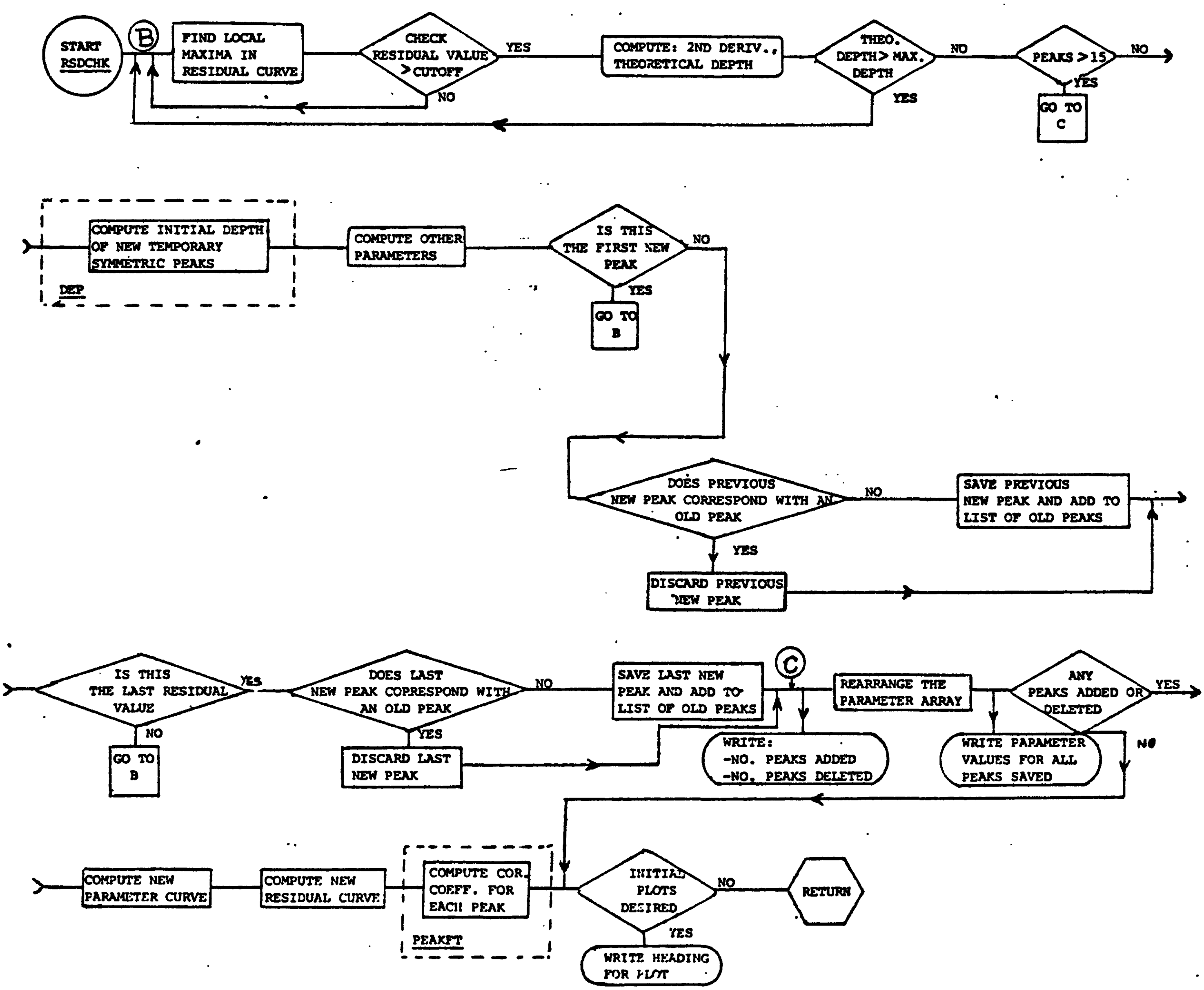
## SUBROUTINE SYMPR3




## SUBROUTINE MARQ1



SUBROUTINE RSDCHK



## Programming Considerations

Listed below are suggestions for making minor changes to program MAGDEP which might be desired at some later time:

A. To change input data format:

1. Modify statement 100 in subroutine INPUT.

B. To use residual magnetic values as input:

1. Modify the statement immediately preceeding statement 120 in the main program. Do not subtract AVG from an input (or interpolated) values.

C. To increase the number of data points which may be used, up to N, modify the following statements: Note: wherever N appears, use the desired integer number for N.

1. Main Program:

```
COMMON TIME (N), B(N), C(N), D(N), E(N), F(N)
COMMON /PRMCRV/XF(N)
DIMENSION AX(N), TMAG(N), DTMAG(N)
COMPLEX CAX (next even power of 2 above N)
```

2. Subroutine INTERP

```
COMMON XAXIS(N), B(N), C(N), D(N), E(N), XY(N)
```

3. Subroutine SPLINE

```
REAL*4 H(N), SPP(N)
```

4. Subroutine HILBRT

```
COMPLEX AXM (next even power of 2 above N)
```

5. Subroutine POLRED

```
COMMON X(N), XX(4N)
```

6. Subroutine SYMPR3

   COMMON X(N), Q(N), T(N), XX(3N)

7. Function DEP

   COMMON X(N), XX(5N)

8. Subroutine PEAKFT

   COMMON X(N), Y(N), C(N), D(N), DIFF(N), XX(N)

   COMMON /PRMCRV/ F(N)

9. Subroutine PLOT

   COMMON X(N), XX(5N)

   DIMENSION U(1), V(1), W(1), Y(N,3), SYM(3)

10. Subroutine PRLPLT

   DIMENSION SYM(NY), ZY(11), GRAPH(101), Y(N,3), X(1)

11. Subroutine MARQ1

   COMMON X(N), TM(N), SS(N), TEM(N), DIFF(N), XX(N)

   COMMON /PRMCRV/ F(N)

   DIMENSION P(N,45), Y(1)

12. Subroutine DERIV

   COMMON X(N), XX(5N)
   DIMENSION P(N,45)

13. Subroutine FUNCT

   COMMON X(N), XX(5N)

14. Subroutine PRMCHK

   COMMON X(N), TT(N), TM(N), XX(3N)

15. Subroutine RSDCHK

   COMMON X(N), Q(N), REJ(N), W(N), R(N), F(N)

   COMMON /PRMCRV/ FF(N)

16. SUBROUTINE INPUT
   IF(N.GT N) Go to 10

D. To increase the number of parameters allowed to KMAX (i.e., 3 times the no. of symmetric curves) change the following: Note: wherever KMAX appears, use the desired integer number for KMAX.

1. Main Program

   COMMON /SYM/ PARM(KMAX), NPAR

2. Subroutine SYMPR3

   COMMON /SYM/ B(KMAX), K

   DIMENSION AX(1), IDIST (KMAX), IAX(KMAX)

   KMAX='KMAX' (directly below statement 104)

3. Subroutine PEAKFT

   COMMON /SYM/ B(KMAX), K

4. Subroutine MARQ1

   COMMON /SYM/ B(KMAX, K)

   DIMENSION P(200,KMAX)

   DIMENSION GC(KMAX), GS(KMAX), GU(KMAX), D(KMAX), DS(KMAX),
   DG(KMAX), B1(KMAX), A(KMAX, KMAX), AS(KMAX, KMAX)

5. Subroutine DERIV

   COMMON /SYM/ B(KMAX), K
   DIMENSION P(200, KMAX)

6. Subroutine PRMCHK

   COMMON /SYM/ B(KMAX), K

7. Subroutine RSDCHK

   COMMON /SYM/ B(KMAX), K

   KMAX='KMAX' (directly after statement 108)

8. SUBROUTINE MARQGS

   DIMENSION A(KMAX, KMHX), B(1),X(1)

E. The following subroutines have been written so that they may be used outside of MAGDEP; the only changes required are to dimension or common statements:

INTERP (SPLINE)

SPLINE

HILBRT (FORK, FORSET, FORMOD)

FORSET

FORK

FORMOD

CORLAT

MINMAX

PLOT (MINMAX, PRLPLT)

PRLPLT

Note: other subroutines which are called by the main subroutine are enclosed in parenthesis.

# APPENDIX II - Program Listing for MAGDEP

NOTE: INPUT DATA FORMAT! NOT BARTLETT

AS STORED at Ames UNDER TKKR15MD 11/28/73

~~MADE From Source Deck 'COPY 1' of MAGDEP in~~ Cooper's Files

LEVEL 20 MAIN DATE = 73332 17/28/02

```
C*****************************************************************************
C       ----------------FORTRAN PROGRAM MAGDEP-----------------------
C                          FORTRAN IV  LEVEL G
```

USGS OPEN FILE 74-1017

```
C*****************************************************************************
C     THE PROGRAM IS DESIGNED TO PROVIDE A METHOD FOR FINDING THE DEPTH TO
C     CORNERS OF A TWO-DIMENSIONAL MAGNETIC SOURCE BODY WHEN GIVEN ONLY THE TOTAL
C     FIELD MAGNETIC INTENSITY PROFILE. ASSUMPTIONS MADE ABOUT CAUSITIVE SOURCE
C     BODY:     1. A TWO-DIMENSIONAL STRUCTURE
C               2. UNIFORMLY MAGNETIZED
C               3. A FINITE OR INFINITE POLYGON IN CROSS SECTION
C     TWO DEPTH ANALYSES ARE MADE BY THE PROGRAM:   1. THICK BODIES  AND  2. THIN
C     BODIES(THICKNESS LESS THAN DEPTH). THE DETERMINATION OF THE CORRECT SOURCE
C     BODY SHAPE DEPENDS UPON A CAREFUL INTERPRETATION OF RESULTS FROM BOTH DEPTH
C     ANALYSES. THE TECHNIQUES USED BY THE PROGRAM ARE BASED ON THE WORK OF:
C       1. THEORY:  NABIGHIAN, MISAC N.,GEOPHYSICS,V.37,PP507-517,1972
C       2. LEAST SQUARES:  MARQUARDT, DON W,JCUR.SOC.APPL.MATH,V11,P431-441,1963
C                    AND   JOHNSON, WILLIAM W,GEOPHYSICS,V34,NO1,PP65-74,1969
C     FOR FURTHER DETAILS SEE DOCUMENTATION AVAILABLE FROM OFF. MARINE GEOLOGY
C
C     USE OF PROGRAM:    THE PROGRAM CAN BE USED IN ONE OR TWO SECTIONS WHICH:
C       SECTION 1:  COMPUTES A(X) CURVE, MAG PROFILE REDUCED TO POLE, INITIAL
C                   TRIAL VALUES FOR PARAMETERS OF THE SYMMETRIC CURVES (1 CURVE
C                   PER BODY CORNER AND 3 PARAMETERS PER CURVE(A,H,C))
C       SECTION 2:  LEAST SQUARE ANALYSIS, RESIDUAL ANALYSIS
C     GENERAL PROCEDURE TO FOLLOW WHEN EXAMINING A NEW DATA PROFILE:
C       1. USE PROGRAM THRU SECT 1 TO SEE PLOT OF A(X) CURVE - USE 5 MIN VALUES
C       2. EXAMINE A(X) CURVE FOR NO. AND TYPES OF SYMMETRIC CURVES. THE HALF-MAX
C          HALF-WIDTH IS DEPTH TO BODY CORNER. CHOOSE GOOD VALUE FOR CUTAX
C       3. BREAK PROFILE INTO SMALLER SEGMENTS: A. TO SEPARATE AREAS OF LARGE AND
C          SMALL PEAKS  B. TO DELETE OR LOOK CLOSER AT 'FLAT' AREAS  C. TO  STAY
C          WITHIN THE MAX. OF POINTS (INPUT OR INTERPOLATION: MAX=200)
C       4. COMPUTE THRU SECTION 2  USING ITIR(1&2)=SMALL NO.   DO ONLY 1 RESIDUAL
C          ANALYSIS.  AFTER RUN, CHECK VALUE OF CUTRSD
C       5. MAKE FINAL RUN BASED UPON RESULTS FROM STEPS 1 THRU 4
C     NOTE: THE EXECUTION TIME FOR THE LEAST SQUARES ROUTINE(SECT 2) IS USUALLY
C           FROM 5 TO 15 TIMES GREATER THAN FOR SECT 1. UNDER SOME CIRCUMSTANCES,
C           WHERE HIGH PRECISION IS NOT REQUIRED, SATISFACTORY RESULTS CAN BE
C           OBTAINED BY OMITTING SECT 2 AND USING ONLY A VISUAL INSPECTION OF
C           THE A(X) CURVE.
C     INPUT CONSTANTS REQUIRED:
C     CARD 1:     HEADING              FORMAT(20A4)          REQUIRED
C     CARD 2:       REQUIRED           FORMAT(F10.1,2I10)
C               SPDKT = SHIP SPEED IN KNOTS
```

```
C              ID = USER DESIRES DATA INTERPOLATION  =1 YES (MUST USE CARD 4)
C                   =2 NO (DATA MUST BE EVENLY SPACED)
C              ISECT = PROGRAM SECTIONS TO USE   =1 THRU SECT 1   =2 THRU SECT 2
C     CARD 3:   REQUIRED          FORMAT(I10,F10.1,2I10,F10.3)
C              IBOD = BODY TYPES TO ANALYSE FOR:  =1 THICK   =2 THIN   =3 BOTH
C              ZMAXKM = MAX. ESTIMATED DEPTH TO SOURCE BODY IN KM
C              IPOLE = WANT PROFILE REDUCED TO POLE:  =0 NO  =1 YES (USE CARD 5)
C              INPLT= WANT INITIAL PLOT (CONTAINS A(X), INITIAL PARAMETER, AND
C                     RESIDUAL TOTAL FIELD PROFILE):   =0 NO   =1 YES
C              CUTAX = FACTOR WHICH WILL BE MULTIPLIED BY THE INTERNAL VALUE FOR
C                      THE A(X) CURVE CUTOFF POINT AND USED TO DETERMINE WHETHER
C                      LOCAL MAXIMA IN A(X) CURVE WILL BE KEPT AS SYMMETRIC CURVE
C                      (ON FIRST RUN USE CUTAX=1.)
C     CARD 4:   USE ONLY IF ID=1     FORMAT(3F10.2,I10)
C              XSHR = START INTERPOLATION--HRS OF ELAPSED TIME FROM START OF LINE
C              XEHR = END   INTERPOLATION--HRS OF ELAPSED TIME FROM START OF LINE
C              DXMIN = TIME INCREMENT AT WHICH DATA IS TO BE INTERPOLATED IN MIN
C              NP = MAX. NO. POINTS WANTED IN DATA SET (MUST BE LESS THAN 201)
C     CARD 5:   USE ONLY IF IPOLE=1            FORMAT(2F10.1)
C              FINC = INCLINATION OF EARTHS MAG FIELD WHERE PROFILE WAS MEASURED
C              ANG = ANGLE BETWEEN MAG NORTH AND SHIP HEADING(0)ANG)180 DEG)
C     CARD 6:   USE ONLY IF ISECT = 2           FORMAT(4I10,F10.3)
C              NRES = NO. RESIDUAL CHECKS TO BE MADE  (REC NRES=1 ON FIRST RUN)
C              NPEAKD = THICK BODY: NO. OF A(X) PEAKS TO BE USED FOR 1ST LST SQRS
C              NPEAKT = THIN  BODY: NO. OF A(X) PEAKS TO BE USED FOR 1ST LST SQRS
C              IOUT = WANT EXTRA OUTPUT (LST SQR VAL & RESID VAL): =0 NO =1 YES
C              CUTRSD = FACTOR USED IN RESIDUAL ANALYSIS (SAME EXPLAN AS CUTAX)
C     CARD 7:   USE ONLY IF ISECT=2              FORMAT(8I10)
C              ITIR(1) = NO. ITERATIONS TO BE MADE DURING 1ST CALL OF LST SQRS
C              ITIR(2) = NO. ITERATIONS OF LST SQRS TO BE MADE AFTER RESIDUAL
C                        NUMBER 1 AND DURING LST SQRS ANALYSIS NUMBER 2
C              ITIR(N) = AS REQUIRED UP TO ITIR(NRES+1) = NO. OF ITERATIONS TO BE
C                        MADE AFTER THE NRES RESIDUAL CHECK AND DURING THE
C                        NRES+1 CALL OF LST SQRS.
C     DATA CARDS:      FORMAT(T60,I2,T6,I3,I2,2F2.0,F5.0)
C     THE REQUIRED INPUT VALUES ON EACH DATA CARD (ACCORDING TO THE FORMAT) ARE:
C     IY = YEAR   ID = JULIAN DAY   IH = HOUR (24 HR DAY)   AM = MINUTE
C     AS = SECOND   TOTMAG = MEASURED TOTAL FIELD INTENSITY VALUE
C     NOTE: A MAX. OF 200 DATA CARDS OR 200 INTERPOLATED VALUES ARE ALLOWED
C           THE PROGRAM SUBTRACTS A CONSTANT REGIONAL FIELD FROM THE DATA
C
C     DEVELOPED BY:  ALAN K COOPER
C                    OFFICE OF MARINE GEOLOGY
```

```
C                    U.S. GEOLOGICAL SURVEY
C                    345 MIDDLEFIELD ROAD
C                    MENLO PARK, CALIFORNIA  94025        DECEMBER 1973
C*******************************************************************************
      COMMON TIME(200),B(200),C(200),D(200),E(200),F(200)
      COMMON /LSTSQ/ XLAM,XU,EPS,OUTPT,SPD,ITER,INPLT
      COMMON /RESID/ ZMAX,ICALL
      COMMON /SYM/ PARM(45),NPAR
      COMMON /PRMCRV/XF(200)
      COMMON /WORK/ AA(100),BB(100)
      COMMON /CUT/ CUTAX,CUTRSD,AMIN,AMAX
      COMMON /LSQERR/ IERR
      DIMENSION AX(200),TMAG(200),DTMAG(200),ITIR(10)
      DIMENSION TITLE(20)
      COMPLEX CAX(256)
      LOGICAL OUTPT
C    -----------FORMAT STATEMENTS----------
  800 FORMAT(F10.1,2I10)
  801 FORMAT(I10,F10.1,2I10,F10.3)
  802 FORMAT(2F10.2,F10.1,I10)
  803 FORMAT(2F10.1)
  804 FORMAT(4I10,F10.3)
  805 FORMAT(8I10)
  806 FORMAT (20A4)
  900 FORMAT(///,' VALUES FOR THE   A(X) = T(X)**2 + T1(X)**2   CURVE:',
     $/,10(/,1P5E20.9))
  901 FORMAT(1H1,'ESTIMATION OF DEPTH TO THICK MAGNETIC SOURCE BODIES -
     $USING THE HORIZONTAL DERIVITIVE OF THE TOTAL FIELD ',/,' MAGNETICS
     $ FOR THE ANALYSIS:')
  902 FORMAT(1H1,'ESTIMATION OF DEPTH TO THIN MAGNETIC SOURCE BODIES - U
     $SING THE TOTAL FIELD MAGNETICS FOR THE ANALYSIS:')
  903 FORMAT(//,1X,'THE MATRIX IN SUBROUTINE MARQ1 IS SINGULAR',/,
     $' CHECK THE INITIAL VALUES OF THE PARAMETERS OR REFER TO THE MARQ1
     $ PROGRAM DESCRIPTION')
  904 FORMAT(1H1,'PLOT OF A(X) CURVE(+), INITIAL PARAMETER CURVE(*), AND
     $ THE RESIDUAL TOTAL FIELD MAGNETIC CURVE(.):')
  905 FORMAT(///,' CORRELATION COEFFICIENT BETWEEN A(X) AND INITIAL PARA
     $METER CURVE=',E17.6)
  906 FORMAT(1H1,'PLOT OF A(X) CURVE(+) AND THE RESIDUAL TOTAL FIELD MAG
     $NETICS CURVE(*):')
  907 FORMAT(//,  ' SAMPLE OF INPUT DATA AND INTERPOLATED DATA:  (FIRST
     $20 DATA POINTS)',//,' NO. OF INPUT DATA POINTS=',I4,15X,
     $'NO. OF INTERPOLATED DATA POINTS=',I4,//,10X,'TIME',7X,'TOTMAGS',
```

```
      $27X,'TIME',7X,'TOTMAGS',/,(2F15.3,15X,2F15.3))
  908 FORMAT(///,' THE DEPTH ANALYSIS FOR THICK BODIES - BASED ON THE HO
     $RIZONTAL DERIVITIVE OF THE TOTAL FIELD MAGNETICS - HAS TERMINATED
     $SUCCESSFULLY')
  909 FORMAT(///,' THE DEPTH ANALYSIS FOR THIN BODIES - BASED ON THE TOT
     $AL FIELD MAGNETICS - HAS TERMINATED SUCCESSFULLY')
  910 FORMAT(/, ' INPUT INFORMATION:',/,' PARAMETERS SUPPLIED BY THE USE
     $R:',//,' SHIP SPEED(KTS):  SPDKT=',F6.1,/,' DATA INTERPOLATION:  I
     $D=',I4,/,' PROGRAM USE:  ISECT=',I4,//,
     $' BODY TYPES:  IBOD=',I4,/,' MAX.BODY DEPTH(KM):  ZMAXKM=',
     $F6.1,/,' PROFILE TO POLE:  IPOLE=',I4,/,' INITIAL PLOTS:  INPLT=',
     $I4,/,' A(X) CUTOFF:  CUTAX=',F7.3)
  911 FORMAT(/,' START INTERPOLATION(HRS):  XSHR=',F7.2,/,
     $' END INTERPOLATION(HRS): XEHR=',F7.2,/,' SAMPLE INCREMENT(MIN):
     $DXMIN=',F6.1,/,' MAX.NO.POINTS:  NP=',I6)
  912 FORMAT(/,' FIELD INCLINATION(DEG):  FINC=',F7.1,/,' MAG HEADING(DE
     $G):  ANG=',F7.1)
  913 FORMAT(/,' NO. RESIDUAL CHECKS:  NRES=',I4,/,' NO. PEAKS-THICK BOD
     $IES:  NPEAKD=',I4,/,' NO. PEAKS-THIN BODIES:  NPEAKT=',I4,/,
     $' EXTRA OUTPUT:  IOUT=',I4,/,' RESIDUAL CUTOFF:  CUTRSD=',F7.3,
     $/,' ITERATIONS:  ITIR(NRES+1)=',(10I6))
  914 FORMAT(1H1,' PLOT OF THE INPUT TOTAL FIELD MAGNETICS(+) AND THE TO
     $TAL FIELD MAGNETICS REDUCED TO THE POLE(*):  (CONSTANT REGIONAL FI
     $ELD REMOVED)')
  915 FORMAT(1H1,(20A4))
  916 FORMAT(////,' CONSTANTS DETERMINED BY THE COMPUTER:',//,' SHIP SPE
     $ED(KM/HR)=',F6.1,/,' SAMPLE INCREMENT(MIN)=',F6.1,/,17X,
     $'(HRS)=',F9.4,//,' CONSTANT REGIONAL FIELD',/,' REMOVED FROM INPUT
     $DATA=',F8.0,/,' MAX. BODY DEPTH(HRS)=',F8.4,//,' A(X) CUTOFF VALUE
     $=',E15.7,/,' RESIDUAL CUTOFF VALUE=',E15.7)
  917 FORMAT(//,' DATA AFTER INTERPOLATION FROM',F6.2,' TO',F6.2,' HRS:'
     $)
C     ----------READ IN PARAMETERS----------
      READ(5,806)(TITLE(I),I=1,20)
      WRITE(6,915)(TITLE(I),I=1,20)
      READ(5,800) SPDKT,ID,ISECT
      READ(5,801) IBOD,ZMAXKM,IPOLE,INPLT,CUTAX
      WRITE(6,910) SPDKT,ID,ISECT,IBOD,ZMAXKM,IPOLE,INPLT,CUTAX
      IF(ID.EQ.2) GO TO 10
      READ(5,802) XSHR,XEHR,DXMIN,NP
      WRITE(6,911) XSHR,XEHR,DXMIN,NP
      DX=DXMIN/60.
      XS=XSHR
```

```
      XE=XEHR
   10 IF(IPOLE.EQ.0) GO TO 20
      READ(5,803) FINC,ANG
      WRITE(6,912) FINC,ANG
   20 IF(ISECT.EQ.1) GO TO 90
      READ(5,804) NRES,NPEAKD,NPEAKT,IOUT,CUTRSD
      OUTPT=.TRUE.
      IF(IOUT.EQ.0) OUTPT=.FALSE.
      KT=NRES+1
      READ(5,805) (ITIR(I),I=1,KT)
      WRITE(6,913) NRES,NPEAKD,NPEAKT,IOUT,CUTRSD,(ITIR(I),I=1,KT)
   90 CONTINUE
C     -----------INPUT CONSTANTS-----------
      IF(ISECT.EQ.1) NPEAKT=0
      IF(ISECT.EQ.1) NPEAKD=0
      IF(ISECT.EQ.1) CUTRSD=0.
      SPD=1.852*SPDKT
      ZMAX=ZMAXKM/SPD
      IERR=0
      IPM=0
      EPS=0.15
      XU=10.
      XLAM=.01
C     ---------INPUT DATA----------
      CALL INPUT(N,TIME,TMAG)
C     ----------INTERPOLATE DATA-----------
      CALL INTERP(N,TIME,TMAG,XS,XE,DX,NI,E,F,DTMAG,ID,NP)
INSERT->  IF(ID.EQ.2) GO TO 70
      WRITE(6,917) XSHR,XEHR
      CALL PRINT(E(1),E(NI))
INSERT->  70 CONTINUE
      WRITE(6,907) N,NI,(TIME(I),TMAG(I),E(I),F(I),I=1,20)
      N=NI
      CALL MINMAX(N,F,TMIN,TMAX)
      AVG=(TMAX+TMIN)/2.
      DO 120 I=1,N
      TIME(I)=E(I)
      TMAG(I)=F(I)-AVG
  120 CONTINUE
      LE=1
      IF(IBOD.EQ.3) LE=2
      DO 1001 JK=1,LE
      ICALL=1
      ITK=1
      KBOD=IBOD+JK
```

```
C     ----------COMPUTE A(X) CURVE---------
      GO TO (200,200,201,200,201), KBOD
  200 CALL HILBRT(N,NC,DTMAG,CAX)
      GO TO 210
  201 CALL HILBRT(N,NC,TMAG,CAX)
  210 CONTINUE
      DO 211 I=1,N
  211 AX(I)=CABS(CAX(I))**2
C     ----------COMPUTE CUTOFF POINT FOR A(X) AND RESIDUAL CURVES-----------
      IF(JK.EQ.2) GO TO 212
      CALL MINMAX(N,AX,AMIN,AMAX)
      CUTOFF=SQRT(AMAX*AMIN)
      CUTAX=CUTOFF*CUTAX
      CUTRSD=CUTOFF*CUTRSD                          IF(ID.EQ.2) DX=TIME(2)-TIME(1)
C     ----------OUTPUT COMPUTED CONSTANTS----------  IF(ID.EQ.2) DXMIN=60.*DX
insert WRITE(6,916) SPD,DXMIN,DX,AVG,ZMAX,CUTAX,CUTRSD
  212 CONTINUE
C     ----------MAGNETIC PROFILE REDUCED TO THE POLE----------
      IF(IPOLE.EQ.0.OR.IPM.NE.0) GO TO 228
      IF(ANG.EQ.90.) GO TO 221
      CR=.01745329
      BINC=ATAN(TAN(FINC*CR)/ABS(COS(ANG*CR)))/CR
      GO TO 222
  221 BINC=90.
  222 PHI2=180.-2.*BINC
  225 CALL POLRED(N,CAX,PHI2)
      WRITE(6,914)
      CALL PLOT(N,2,TMAG,B,B)
  228 IPM=1
C     ----------FIND SYMMETRIC COMPONENTS OF A(X)-----------
      GO TO (231,231,232,231,232), KBOD
  231 WRITE(6,901)
      WRITE(6,900) (AX(I),I=1,N)
      CALL SYMPR3(N,NPEAKD,AX)
      GO TO 230
  232 WRITE(6,902)
      WRITE(6,900) (AX(I),I=1,N)
      AX(1)=AX(2)
      CALL SYMPR3(N,NPEAKT,AX)
  230 CONTINUE
      IF(NPAR.EQ.0) GO TO 233
      CALL FUNCT(PARM,NPAR,XF,N)
      DO 234 JT=1,N
```

```
  234 E(JT)=AX(JT)-XF(JT)
      CALL PEAKFT(N,AX)
C     ----------PLOT A(X), TOTMAGS, & INITIAL PARAMETER CURVE-----------
  233 IF(INPLT.EQ.0) GO TO 250
      CALL MINMAX(N,AX,AMIN,AMAX)
      CALL SETUP(N,TMAG,B,AMAX,AMIN)
      IF(NPAR.EQ.0) GO TO 235
      WRITE(6,904)
      IF(ISECT.GE.2) GO TO 252
      CALL FUNCT(PARM,NPAR,C,N)
      CALL CORLAT(N,AX,C,VAL)
      WRITE(6,905) VAL
      GO TO 237
  235 WRITE(6,906)
      CALL PLOT(N,2,AX,B,AX)
      GO TO 1000
  237 CALL PLOT(N,3,AX,C,B)
      GO TO 1000
  250 IF(ISECT.LT.2) GO TO 1000
  252 CONTINUE
C     ----------LEAST SQUARES ROUTINE---------
      ITER=ITIR(ITK)
  300 CALL MARQ1(N,AX,$310)
      IF(NPAR.EQ.0) GO TO 1000
      GO TO 320
  310 WRITE(6,903)
      GO TO 1000
  320 IF(NRES-ITK.LT.0) GO TO 1000
      ITK=ITK+1
C     ----------CHECK RESIDUALS----------
      CALL RSDCHK(N,AX)
      GO TO 252
 1000 CONTINUE
      GO TO (400,400,401,400,401),KBOD
  400 WRITE(6,908)
      GO TO 1001
  401 WRITE(6,909)
 1001 CONTINUE
      STOP
      END
```

```
      SUBROUTINE INTERP(N,X,Y,XSTART,XSTOP,XINC,N1,X1,Y1,DY1,ID,NP)
C     THIS SUBROUTINE WILL INTERPOLATE THE VALUES FROM ARRAY Y(N) FROM XSTART
C     TO XSTOP AT INCREMENTS OF XINC.  THREE ARRAYS ARE RETURNED :
C             X1(N1) = REAL ARRAY CONTAINING THE INTERPOLATED X VALUES
C             Y1(N1) = REAL ARRAY CONTAINING THE EVENLY SPACED INTERPOLATED VALUE
C             DY1(N1)= REAL ARRAY CONTAINING THE DERIVITIVES OF Y(N) AT EVENLY
C                      SPACED INCREMENTS XINC(IE AT SAME LOCATIONS AS Y1(N1))
C     THE OTHER VARIABLES INCLUDE:
C             X(N) = REAL ARRAY CONTAINING THE INITIAL VALUES ALONG THE X-AXIS
C             Y(N) = REAL ARRAY CONTAINING THE INITIAL VALUES TO BE INTERPOLATED
C             XSTART = FIRST X VALUE WHERE THE INTERPOLATION IS TO START
C             XSTOP = LAST X VALUE FOR THE INTERPOLATION
C             XINC = INCREMENT OF X AT WHICH EVENLY SPACED DATA WILL BE RETURNED
C             N = NUMBER OF DATA POINTS IN THE INPUT ARRAY Y(N)
C             N1 = NUMBER OF DATA POINTS IN THE OUTPUT ARRAY Y1(N1) AND DY1(N1)
C             NP = MAXIMUM NUMBER OF POINTS ALLOWED IN OUTPUT ARRAYS
C             ID = IDENTIFICATION MARKER TO INSTRUCT THE SUBROUTINE, THE VALUES:
C                  =0  : DO AN INTERPOLATION USING ALL PARAMETERS, THREE ARRAYS
C                        ARE RETURNED, AND SUBROUTINE SPLINE IS NOT CALLED
C                  =1  : SAME AS FOR 0 EXCEPT SUBROUTINE SPLINE IS CALLED
C                  =2  : THE SUBROUTINE WILL ONLY RETURN THE DERIVITIVE OF THE
C                        INPUT CURVE AT EACH OF THE EVENLY SPACED X VALUES
C      THIS IS A GENERAL SUBROUTINE FOR USE OUTSIDE OF THIS PROGRAM
C     MAXIMUM COMMON USAGE:  XAXIS(0),B(N),C(N),D(N),E(N),XY(0)
      COMMON XAXIS(200),B(200),C(200),D(200),E(200),XY(200)
      DIMENSION X(1),Y(1),X1(1),Y1(1),DY1(1)
  100 FORMAT(//1X,'XSTART IS OUT OF THE RANGE OF THE INPUT DATA')
  101 FORMAT(//1X,'XSTOP IS OUT OF THE RANGE OF THE INPUT DATA')
      IEND=0
      IF(ID.EQ.2) GO TO 15
C     ----------CHECK INTERPOLATION RANGE----------
      IF(XSTART.LE.X(N).AND.XSTART.GE.X(1)) GO TO 8
      WRITE(6,100)
      IEND=1
    8 IF(XSTOP.GE.X(1).AND.XSTOP.GE.XSTART) GO TO 9
      WRITE(6,101)
      RETURN
    9 IF(IEND.EQ.1) RETURN
      IF(ID.EQ.0) GO TO 1
C
C     ----------COMPUTE THE INTERPOLATING CONSTANTS----------
      CALL SPLINE(N,X,Y,B,C,D)
C     ----------BINARY SEARCH----------
```

```
      1 XX=XSTART
        I=1
        J=N
      2 IF(J-1.EQ.1) GO TO 6
        K=(I+J)/2
        IF(XX-X(K)) 3,5,4
      3 J=K
        IF(X(K)-XX.GT.X(K)-X(K-1)) GO TO 2
        J=K-1
        GO TO 7
      4 I=K
        IF(XX-X(K).GT.X(K+1)-X(K)) GO TO 2
        J=K
        GO TO 7
      5 J=K
        GO TO 7
      6 J=I
      7 IF(XSTOP.GT.X(N)) XSTOP=X(N)
        M=(XSTOP-XSTART)/XINC+1
        IF(M.GT.NP) M=NP
C     ----------DO INTERPOLATION - COMPUTE DERIVITIVES----------
     11 DO 10 I=1,M
     12 IF(XX.LT.X(J+1)) GO TO 13
        IF(J+1.NE.N) GO TO 14
        N1=I-1
        RETURN
     14 J=J+1
        GO TO 12
     13 IF(XX.GT.XSTOP) RETURN
        N1=I
        DX=XX-X(J)
        Y1(I)=Y(J)+DX*(B(J)+DX*(C(J)+DX*D(J)))
        DY1(I)=B(J)+DX*(2.*C(J)+3.*DX*D(J))
        X1(I)=XX
        XX=XX+XINC
     10 CONTINUE
        RETURN
C
C     ----------ONLY COMPUTE DERIVITIVES----------
     15 CALL SPLINE(N,X,Y,B,C,D)
        KK=N-1
        N1=N
        DO 16 I=1,N
```

```
      X1(I)=X(I)
   16 Y1(I)=Y(I)
      DO 17 I=1,KK
   17 DY1(I)=B(I)
      DY1(N)=B(KK)
   20 RETURN
      END
```

```
      SUBROUTINE SPLINE(N,X,Y,B,C,D)                                      1.
C     THIS SUBROUTINE COMPUTES THE SPLINE COEFFICIENTS NECESSARY TO DO DATA
C     INTERPOLATION.  THE INPUT PARAMETERS ARE:
C         N = NO. OF DATA POINTS
C         X(N) = REAL ARRAY CONTAINING THE ABSCISSA OF THE DATA POINTS
C         Y(N) = REAL ARRAY CONTAINING THE ORIDNATES OF THE DATA POINTS
C         B(N-1), C(N-1), & D(N-1) = REAL ARRAYS CONTAINING THE SPLINE
C                                    COEFFICIENTS
      REAL*4 X(1), Y(1), B(1), C(1), D(1)                                 2.
      REAL*8 DELTA,DELTAI
      REAL*4 H(200),SPP(200)
C                                                                        36.
      NM1 = N-1                                                          37.
      DO 10 I=1,NM1                                                      38.
          H(I) = X(I+1) - X(I)                                           39.
          DELTAI = (Y(I+1) - Y(I))/H(I)                                  40.
          IF (I.NE.1) B(I) = DELTAI - DELTA                              41.
          DELTA = DELTAI                                                 42.
   10 CONTINUE                                                           43.
C     ----------FORWARD ELIMINATION----------
      C(2) = 2.0D0*(H(1) + H(2))                                         45.
      DO 20 I = 3,NM1                                                    46.
          C(I) = 2.0D0*(X(I+1) - X(I-1)) - H(I-1)**2/C(I-1)              47.
          B(I) = B(I) - H(I-1)*B(I-1)/C(I-1)                             48.
   20 CONTINUE                                                           49.
C     ----------BACK SUBSTITUTION----------
      SPP(1) = 0.0D0                                                     51.
      SPP(N) = 0.0D0                                                     52.
      SPP(N-1) = B(N-1)/C(N-1)                                           53.
      NM2 = N-2                                                          54.
      DO 30 I = 2,NM2                                                    55.
          J = N-I                                                        56.
          SPP(J) = (B(J) - H(J)*SPP(J+1))/C(J)                           57.
   30 CONTINUE                                                           58.
C                                                                        59.
      DO 40 I = 1,NM1                                                    60.
          B(I) = (Y(I+1) - Y(I))/H(I) - (SPP(I+1) + 2.0D0*SPP(I))*H(I)   61.
          C(I) = 3.0D0*SPP(I)                                            62.
          D(I) = (SPP(I+1) - SPP(I))/H(I)                                63.
   40 CONTINUE                                                           64.
      RETURN                                                             65.
      END
```

```
      SUBROUTINE FORSET(N,F,NC,CF)
C     THIS SUBROUTINE PREPARES THE DATA SET F(N) FOR ENTRY INTO THE FAST
C     FOURIER TRANSFORM SUBROUTINE FORK.  NC IS THE NUMBER OF POINTS IN THE
C     MODIFIED DATA SET(IE ZEROS ARE ADDED TO THE NEAREST POWER OF 2).  CF(NC)
C     IS A COMPLEX ARRAY WHICH CONTAINS THE DATA READY FOR INPUT INTO FORK
      DIMENSION F(1)
      COMPLEX CF(1)
C     ----------FIND NEXT HIGHEST POWER OF 2----------
      IR=ALOG(1.*N)/.69315
      NC=2**(IR+1)
      NX=N+1
C     ----------FILL THE COMPLEX ARRAY----------
      DO 1 I=1,N
      CF(I)=F(I)
    1 CONTINUE
      IF(N.EQ.NC) RETURN
C     ----------ADD ZEROS----------
      DO 2 I=NX,NC
      CF(I)=0.
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE HILBRT(N,NC,FX,FXM)
C     THIS SUBROUTINE WILL COMPUTE THE COMPLEX ARRAY AX(NC) = T(X) - IT1(X)
C     FROM A REAL ARRAY FX(N).  THE REAL ARRAY IS FOUR. TRANS., MODIFIED, AND
C     INVERSE FOUR. TRANS. SUCH THAT: THE REAL PART OF THE COMPLEX OUTPUT ARRAY
C     EQUALS THE INPUT ARRAY, AND THE OMAGINARY PART OF THE OUTPUT ARRAY IS THE
C     HILBERT TRANSFORM OF THE INPUT ARRAY
C        FX(N) = A 1-DIMENSIONAL REAL ARRAY CONTAINING THE DATA TO BETRANSFORMED
C       FXM(NC) = 1-DIMENSIONAL COMPLEX ARRAY CONTAINING THE TRANSFORM
      DIMENSION FX(1)
      COMPLEX FXM(1),AXM(256)
C     ----------FAST FOURIER TRANSFORM----------
      CALL FORSET(N,FX,NC,AXM)
      CALL FORK(NC,AXM,-1.)
C     ----------MODIFY SPECTRUM----------
      CALL FORMOD(NC,AXM,FXM)
C     ----------INVERSE FAST FOURIER TRANSFORM----------
      CALL FORK(NC,FXM,1.)
      RETURN
      END
```

```
      SUBROUTINE FORK(LX,CX,SIGNI)                                      FORK0010
C     FAST FOURIER TRANSFORM ROUTINE WRITTEN BY JON CLARBOUT, GEOPHYSICS
C     DEPARTMENT, STANFORD UNIVERSITY, 1970.
C                                                                       FORK0020
C                             LX                                        FORK0030
C   CX(K) = SQRT(1/LX) SUM (CX(J)*EXP(2*PI*SIGNI*I*(J-1)*(K-1)/LX))     FORK0040
C                            J=1                                        FORK0050
C                                                                       FORK0060
C                                    FOR K=1,2,...,(LX=2**INTEGER)      FORK0070
      COMPLEX CX(LX),CARG,CEXP,CW,CTEMP                                 FORK0080
      J=1                                                               FORK0090
      SC=SQRT(1./LX)                                                    FORK0100
      DO 5 I=1,LX                                                       FORK0110
      IF(I.GT.J)GO TO 2                                                 FORK0120
      CTEMP=CX(J)*SC                                                    FORK0130
      CX(J)=CX(I)*SC                                                    FORK0140
      CX(I)=CTEMP                                                       FORK0150
    2 M=LX/2                                                            FORK0160
    3 IF(J.LE.M)GO TO 5                                                 FORK0170
      J=J-M                                                             FORK0180
      M=M/2                                                             FORK0190
      IF(M.GE.1)GO TO 3                                                 FORK0200
    5 J=J+M                                                             FORK0210
      L=1                                                               FORK0220
    6 ISTEP=2*L                                                         FORK0230
      DO 8 M=1,L                                                        FORK0240
      CARG=(0.,1.)*(3.14159265*SIGNI*(M-1))/L                           FORK0250
      CW=CEXP(CARG)                                                     FORK0260
      DO 8 I=M,LX,ISTEP                                                 FORK0270
      CTEMP=CW*CX(I+L)                                                  FORK0280
      CX(I+L)=CX(I)-CTEMP                                               FORK0290
    8 CX(I)=CX(I)+CTEMP                                                 FORK0300
      L=ISTEP                                                           FORK0310
      IF(L.LT.LX)GO TO 6                                                FORK0320
    9 RETURN                                                            FORK0330
      END                                                               FORK0340
```

```
      SUBROUTINE POLRED(N,A,PHI2)
C     THIS SUBROUTINE TAKES THE COMPLEX ANALYTIC FUNCTION: A(X)=T(X)-IT1(X)
C     AND USES IT TO OBTAIN A MAGNETIC PROFILE WHICH HAS BEEN REDUCED TO THE
C     POLE. THE PROFILE OUTPUT IS STORED IN ARRAY B
C     MAXIMUM COMMON USAGE:  X(N),B(N),XX(0)
      COMMON X(200),B(200),XX(800)
      COMPLEX A(1)
      DX=X(2)-X(1)
      CON=3.141597/180.
      PHI2=PHI2*CON
      R1=COS(PHI2)
      R2=SIN(PHI2)
      POLMAG=0.
      WRITE (6,100)
  100 FORMAT ('0 INPUT DATA REDUCED TO THE POLE')
      DO 1 I=1,N
      POLMAG=POLMAG+(REAL(A(I))*R1-AIMAG(A(I))*R2)*DX
      B(I)=POLMAG
      WRITE (6,200) I,B(I)
  200 FORMAT (5X,I3,5X,F12.5)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FORMOD(NC,CF,CFM)
C     THIS SUBROUTINE WILL MODIFY THE FOURIER TRANSFORM INTO A HILBERT
C     TRANSFORM BY MAKING THE FOLLOWING MODIFICATIONS:
C                         H(W)=F(W) FOR W=0
C                         H(W)=2F(W) FOR F(W)>0
C                         H(W)=0 FOR F(W))0
C     NC IS THE NUMBER OF POINTS IN CF(NC)
C     THE ORIGINAL ARRAY CF(NC) IS SAVED AND THE MODIFIED TRANSFORM ARRAY CFM(NC)
C     IS RETURNED
      COMPLEX CF(1),CFM(1)
      NB=NC/2+1
      NA=NB+1
      CFM(1)=CF(1)
      DO 1 I=2,NB
      CFM(I)=2.*CF(I)
    1 CONTINUE
      DO 2 I=NA,NC
      CFM(I)=0.
    2 CONTINUE   <- CFM(NB)=0.
INSERT ->
      RETURN
      END
```

```
      SUBROUTINE SYMPR3(N,IPEAK,AX)
C     THIS SUBROUTINE FINDS THE LOCATION OF EACH SYMMETRIC COMPONENT OF THE A(X)
C     CURVE BY THE FOLLOWING CRITERIA:   1. FIND THE LOCAL MAXIMA OF A(X) - TEST
C     ADJACENT POINTS.   2.CHECK THE 2ND DERIVITIVE AT EACH MAXIMA - A THEORET-
C     ICAL DEPTH BASED ON A CONTINUOUS 2ND DERIVITIVE IS COMPUTED AND THIS MUST
C     BE LESS THAN ZMAX.   THE SUBROUTINE COMPUTES THE PARAMETERS FOR A NUMBER OF
C     THESE SYMMETRIC COMPONENTS WHICH IS DETERMINED BY:   1. THE USER SPECIFYING
C     A VALUE FOR IPEAK,  OR   2, SETTING IPEAK=0 WHICH USES ALL COMPONENTS WHICH
C     HAVE AN A(X) VALUE LARGER THAN THE A(X) CUTOFF VALUE.  OTHER PARAMETERS ARE
C           N = NUMBER OF DATA POINTS
C           IPEAK = 0  INDICATES USER WISHES THE COMPUTER TO USE THE LARGEST
C                      A(X) PEAKS IN DESCENDING ORDER. (MAX OF 10 PEAKS)
C                 = ANY NUMBER ) 16: USER WISHES THIS NUMBER OF PEAKS TO BE
C                      USED. (IN DESCENDING ORDER - MAX OF 15 PEAKS)
C           AX(N) = REAL ARRAY CONTAINING THE A(X) VALUES
C     OTHER PARAMETERS USED IN THE SUBROUTINE
C           ZMAX = ESTIMATE OF MAXIMUM DEPTH TO MAGNETIC SOURCE
C           K = NUMBER OF PARAMETERS ( 3 * NUMBER OF SYMMETRIC COMPONENTS)
C           B(?) = REAL ARRAY CONTAING THE PARAMETER VALUES
C           KMAX = MAXIMUM NUMBER OF PARAMETERS ALLOWED BY DIMENSION STATEMENTS
C                  (MUST BE DIVISIBLE BY THREE)
C     A MAXIMUM OF 15 PEAKS (45 PARAMETERS) IS ALLOWED FOR THE LEAST SQUARE
C     ROUTINE(MARQ1)
C     MAXIMUM COMMON USAGE:  X(N),Q(2*KMAX),T(K/3),XX(0),B(KMAX)
      COMMON X(200),Q(200),T(200),XX(600)
      COMMON /RESID/ ZMAX,ICALL
      COMMON /SYM/ B(45),K
      COMMON /CUT/ CUTAX,CUTRSD,AMIN,AMAX
      DIMENSION AX(1),IDIST(50),IAX(50)
C     ----------FORMAT STATEMENTS----------
  100 FORMAT(1H1,'PARAMETERS USED FOR THE DECOMPOSITION OF THE A(X)=',
     $'T(X)**2+T1(X)**2 CURVE INTO ITS SYMMETRIC COMPONENTS;',//,
     $' MAXIMUM VALUE OF A(X) =',1PE14.6,/,' MINIMUM VALUE OF A(X) =',
     $1PE14.6,/,' CUTOFF VALUE FOR A(X) =',1PE14.6,/,' NUMBER OF '
     $,'SYMMETRIC PEAKS WITH A(X) VALUES LARGER THAN THE CUTOFF =',
     $I5,/,' NUMBER OF PEAKS REQUESTED =',I5,/,' TOTAL NUMBER OF ',
     $'PEAKS DETERMINED =',I5,/,' NUMBER OF PEAKS USED FOR LEAST ',
     $'SQUARE FIT =',I5,//)
  101 FORMAT(1H ,'THE LOCATION AND A(X) VALUES FOR ALL PEAKS DETERMINED
     $ARE:',/,' X - POSITION          A(X) VALUE',/,(F9.3,10X,E15.7))
  102 FORMAT(//1H ,'PARAMETERS OF THE SYMMETRIC CURVES IN A(X): (USED AS
     $ THE INITIAL VALUES IN THE LEAST SQUARE FIT PROCESS)',/,10X,
     $'ALPHA',4X,'DEPTH(HOURS)',4X,'X-DIST(HOURS)',7X,'A(X) VALUE')
```

```
  103 FORMAT(1H ,(3F15.3,E19.7))
  104 FORMAT(//,' NO PEAKS WERE FOUND IN THE A(X) CURVE')
  105 FORMAT(//,' ****WARNING****   NO SYMMETRIC PEAKS HAVE A(X) VALUES
     $> CUTOFF   ---   ANALYSIS ALLOWED TO CONTINUE')
      KMAX=45
      L=2
      M=N-1
      LL=1
      KL=0
      DX=X(2)-X(1)
C     ----------FIND LOCAL MAXIMA----------
      DO 1 I=L,M
      IF(AX(I).LE.AX(I+1).OR.AX(I).LE.AX(I-1)) GO TO 1
C     ----------CHECK 2ND DERIVITIVE CRITERIA----------
      DER2=ABS((AX(I+1)-2.*AX(I)+AX(I-1)))
      DEP1=SQRT(2.*AX(I)/DER2)*DX
      IF(DEP1.GT.ZMAX) GO TO 1
      IF(LL.GT.KMAX) GO TO 1
C     ----------STORE PEAK LOCATIONS (INCREASING X-DIST----------
      IF(AX(I).GE.CUTAX) KL=KL+1
      IDIST(LL)=I
      IF(LL.NE.1) GO TO 3
      IAX(1)=I
      GO TO 10
    3 KK=1
C     ----------STORE PEAK LOCATIONS (DECREASING VALUES OF AX(I)----------
    4 IT=IAX(KK)
      IF(AX(I).GT.AX(IT)) GO TO 5
      KK=KK+1
      IF(KK.GE.LL) GO TO 6
      GO TO 4
    6 IAX(KK)=I
      GO TO 10
C     ----------REARRANGE ARRAY IAX----------
    5 J=KK
      JJ=J
      IM=IAX(J)
    7 JJ=JJ+1
      IF(JJ.EQ.LL) GO TO 8
      IL=IAX(JJ)
      IAX(JJ)=IM
      IM=IL
      GO TO 7
```

```
    8 IAX(JJ)=IM
      IAX(J)=I
   10 LL=LL+1
C     ----------DETERMINE NO, OF PEAKS TO UES----------
    1 CONTINUE
      LL=LL-1
      IF(IPEAK.NE.0) GO TO 9
      KUSE=KL
      IF(KL.GT.10) KUSE=10
      GO TO 11
    9 KUSE=IPEAK
       IF(KUSE.GT,LL) KUSE=LL
      IF(IPEAK.GT.KMAX/3) KUSE=KMAX/3
   11 CONTINUE
      WRITE(6,100) AMAX,AMIN,CUTAX,KL,IPEAK,LL,KUSE
      NO=-1
      IF(LL.EQ.0) GO TO 50
C     ----------FIND PARAMETERS (INCREASING X-DIST)----------
      DO 20 J=1,LL
      NO=NO+2
      I=IDIST(J)
      Q(NO)=X(I)
      Q(NO+1)=AX(I)
   20 CONTINUE
      NO=NO+1
      WRITE(6,101) (Q(I),I=1,NO)
      NO=-2
      IF(KUSE.NE.0) GO TO 21
      WRITE(6,105)
      KUSE=1
   21 CONTINUE
C     ----------FIND PARAMETERS (DECREASING AX(I) VALUES----------
      DO 30 J=1,KUSE
      NO=NO+3
      I=IAX(J)
C     ----------COMPUTE INITIAL DEPTHS----------
      TEM=DEP(N,I,AX)
      B(NO)=SQRT(TEM**2*AX(I))
      B(NO+1)=TEM
      B(NO+2)=X(I)
      T(J)=AX(I)
   30 CONTINUE
      WRITE(6,102)
```

```
      L=-2
      DO 40 J=1,KUSE
      L=L+3
      M=L+2
      WRITE(6,103) (B(I),I=L,M),T(J)
   40 CONTINUE
C     ----------FINAL NO. OF PARAMETERS USED----------
      K=KUSE*3
      RETURN
   50 WRITE(6,104)
      K=0
      RETURN
      END
```

```
LEVEL  20                CORLAT              DATE = 73332          17/28/02

      SUBROUTINE CORLAT(N,A,B,COR)
C     THIS SUBROUTINE COMPUTES THE PRODUCT-MOMENT CORRELATION COEFFICIENT BETWEEN
C     TWO DATA ARRAYS A(N) AND B(N).  THE MATHEMATICAL EXPRESSION IS:
C                COR = (SUM A*B)/SQRT((SUM A**2)(SUM B**2))
      DIMENSION A(1),B(1)
      REAL*8 X,Y,Z
      X=0.D0
      Y=0.D0
      Z=0.D0
      DO 1 I=1,N
      X=X+A(I)*B(I)
      Y=Y+A(I)**2
    1 Z=Z+B(I)**2
C     ----------CORRELATION COEFFICIENT----------
      COR=DABS(X/DSQRT(Y*Z))
      RETURN
      END
```

```
      FUNCTION DEP(N,I,AX)
C     THIS FUNCTION ESTIMATES THE INITIAL DEPTHS TO THE MAGNETIC SOURCE BY
C     FINDING THE 1/2 WIDTH OF THE SYMMETRIC PEAKS IN THE AX(N) CURVE
C         N = NO. OF DATA POINTS IN A(N)
C         I = THE SUBSCRIPT OF THE ARRAY A(I) AT WHICH THE INITIAL DEPTH IS
C             TO BE ESTIMATED
C         AX(N)= A REAL ARRAY CONTAINING THE DATA CURVE
C     MAXIMUM COMMON USAGE:  X(N),XX(O)
      COMMON X(200),XX(1000)
      DIMENSION AX(1)
      DX=X(2)-X(1)
      JK=I-1
      IF(N-I.LT.I) JK=N-I
      LO=1
      DO 21 JL=1,JK
      IF(LO.EQ.JK) GO TO 25
      IF(AX(I+1+LO).GT.AX(I+LO).OR.AX(I-1-LO).GT.AX(I-LO)) GO TO 25
      LO=LO+1
   21 CONTINUE
C     ----------COMPUTE 1/2 WIDTH----------
   25 DEP=(DX*FLOAT(LO))/2.
      RETURN
      END
```

```
      SUBROUTINE PEAKFT(N,AX)
C     THIS SUBROUTINE GIVES A QUANTITATIVE ESTIMATE AS TO THE QUALITY OF THE
C     FIT BETWEEN THE A(X) CURVE AND THE COMPUTED CURVE - FOR EACH SYMMETRIC
C     PEAK SEPARATELY.  THE ANALYSIS FOR EACH PEAK IS DONE BETWEEN:
C         PEAK POSITION +/- 3.*COMPUTED DEPTH
C     AT THESE DISTANCES, THE CURVE FOR THE SYMMETRIC PEAK SHOULD(THEORETICALLY)
C     BE AT 10( OF ITS MAXIMUM VALUE.   THE PARAMETERS USED ARE:
C         N=NO. OF DATA POINTS IN THE AX(N) CURVE
C         AX(N)=REAL ARRAY CONTAINING THE A(X) CURVE
C     MAXIMUM COMMON USAGE:  X(N),Y(0),C(0),D(0),E(0),DIFF(N),B(KMAX)
C                            F(N),AA(100),BB(100),XX(0)
      REAL AA,BB
      REAL MARQIP
      COMMON X(200),Y(200),C(200),D(200),DIFF(200),XX(200)
      COMMON /SYM/B(45),K
      COMMON /PRMCRV/F(200)
      COMMON /WORK/ AA(100),BB(100)
      DIMENSION AX(1)
  100 FORMAT(//,' ESTIMATION OF THE "GOODNESS OF FIT" FOR EACH SYMMETRIC
     $ PEAK - BASED UPON THE COMPARISON OF THE A(X) CURVE AND THE',/,
     $' COMPUTED PARAMETER CURVE IN THE VICINITY OF THE SYMMETRIC PEAK',
     $//,' PEAK LOCATION(HRS)   COMPARISON PTS   RANGE(HRS): FROM',
     $'     TO          SUM OF SQUARES OF RESIDUALS   CORRELATION COEFFICIE
     $NT')
  102 FORMAT(6X,F6.3,T27,I3,T51,F5.2,2X,F5.2,T73,E16.7,T103,E16.7)
      WRITE(6,100)
      DX=X(2)-X(1)
      DO 1 I=1,K,3
      TDEP=3.*B(I+1)
      DISP=B(I+2)-X(1)
      BDIST=(DISP-TDEP)/DX
      FDIST=(DISP+TDEP)/DX
      IA=IFIX(FDIST)+2
      IB=IFIX(BDIST)+1
      IF(IA.GT.N) IA=N
      IF(IB.LT.1) IB=1
      NPT=IA-IB+1
      IF(NPT.LT.100) GO TO 4
      NR=(NPT-99)/2
      IA=IA-NR
      IB=IB-NR
      NPT=IA-IB+1
    4 DO 2 J=1,NPT
```

```
      AA(J)=AX(IB+J-1)
    2 BB(J)=F(IB+J-1)
      CALL CORLAT(NPT,AA,BB,COR)
      DO 3 J=1,NPT
    3 AA(J)=DIFF(IB+J-1)
      SQR=MARQIP(AA,AA,NPT)
      WRITE(6,102) B(I+2),NPT,X(IB),X(IA),SQR,COR
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SETUP(N,TMAG,V,YMAX,YMIN)
C     THIS SUBROUTINE SETS UP THE SCALE FACTORS AND MODIFIES TMAG(N) SO THAT THE
C     TOTAL FIELD MAGNETICS CAN BE PLOTTED WITH THE /A(X)/**2 CURVE. THE SCALE
C     IS SET SUCH THAT 25 SMALL DIVISIONS ON THE Y-AXIS =1000 GAMMA (2000 GAMMA
C     IN SPECIAL CASES).    V(N) CONTAINS THE THE MODIFIED TMAG VALUES, YMIN IS
C     THE MINIMUM VALUE OF THE /A(X)/**2 CURVE,  YMAX IS THE MAXIMUM VALUE.
      DIMENSION TMAG(1),V(1)
      CALL MINMAX(N,TMAG,TMIN,TMAX)
      CS=(YMAX-YMIN)/4000.
      CP=(YMAX+YMIN)/2.
      IF((TMAX-TMIN).GT.2000.) GO TO 2
      DO 1 I=1,N
    1 V(I)=((TMAG(I)-TMIN)*CS)+CP
      RETURN
    2 CS=0.5*CS
      DO 3 I=1,N
    3 V(I)=((TMAG(I)-TMIN)*CS)+CP
      RETURN
      END
```

```
      SUBROUTINE MINMAX(N,X,XMIN,XMAX)
C     THIS SUBROUTINE FINDS THE MINIMUM AND MAXIMUM VALUES WITHIN A 1-DIMENSIONAL
C     ARRAY X(N)
      DIMENSION X(1)
      XMIN=X(1)
      XMAX=X(1)
      DO 10 I=2,N
      IF(X(I).GT.XMAX) XMAX=X(I)
      IF(X(I).LT.XMIN) XMIN=X(I)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PLOT(N,NCUR,U,V,W)
C     THIS WILL PLOT UP TO A MAXIMUM OF THREE(3) CURVES: U(N),V(N),W(N) WITH A
C     COMMON X-AXIS GIVEN BY X(N).  THE ARRAY X(N) MUST BE STORED IN COMMON
C     CURVES ARE PLOTTED USING SYMBOLS   +  *  .    RESPECTIVELY
C        NCUR = NO. OF CURVES TO BE PLOTTED
C     THIS SUBROUTINE REQUIRES THE USE OF SUBROUTINE PRLPLT
C     MAXIMUM COMMON USAGE:  X(N),XX(0)
      COMMON X(200),XX(1000)
      DIMENSION U(1),V(1),W(1),Y(200,3),SYM(3)
      DATA SYM(1)/'+'/,SYM(2)/'*'/,SYM(3)/'.'/
C     ----------SETUP PLOT ARRAYS----------
    2 DO 1 I=1,N
      Y(I,1)=U(I)
      Y(I,2)=V(I)
    1 Y(I,3)=W(I)
C     ----------FIND SMALLEST AND LARGEST VALUE----------
   10 CALL MINMAX(N,U,YMIN,YMAX)
      CALL MINMAX(N,V,XYMIN,XYMAX)
      CALL MINMAX(N,W,ZYMIN,ZYMAX)
      CALL MINMAX(N,X,XMIN,XMAX)
      YMIN=AMIN1(YMIN,XYMIN,ZYMIN)
      YMAX=AMAX1(YMAX,XYMAX,ZYMAX)
C     ----------PLOT THE CURVES----------
      CALL PRLPLT(XMAX,XMIN,YMAX,YMIN,NCUR,N,N,SYM,Y,X)
      RETURN
      END
```

```
      SUBROUTINE PRLPLT(XMAX,XMIN,YMAX,YMIN,NY,LINES,LAST,SYM,Y,X)
C    THIS SUBROUTINE IS A PLOT ROUTINE WHICH HAS THE PARAMETERS:
C            NY = NUMBER OF PLOTS TO BE MADE (FOR EACH PLOT A SYMBOL MUST
C                 BE ASSIGNED IN THE CALLING PROGRAM - EX. DATA SYM(1)/'*'/
C            LINES = THE TOTAL NUMBER OF LINES ALONG THE X AXIS TO BE PLOTTED
C            LAST = THE NUMBER OF THE LAST POINT TO BE PLOTTED
C            SYM = AN ARRAY CONTAINING THE SYMBOLS TO BE USED IN PLOTTING
C            Y = A 2-DIMENSIONAL ARRAY CONTAINING THE Y VALUES TO BE PLOTTED
C            X = A 1 CIMENSIONAL ARRAY CONTAINING THE X AXIS VALUES
      DATA BLANK/' '/,CR/'-'/,PL/'+'/                                   PRLPL019
      DIMENSION SYM(NY),ZY(11),GRAPH(101),Y(200,3),X(1)
   15 FORMAT(///,8X,1P11E10.2)
   20 FORMAT(1H ,3X,'X VALUES',2X,20('+....'),'+',5X,'Y VALUES')
   75 FORMAT(1H ,2X,1PE9.2,2X,101A1,2(1PE9.2))
   76 FORMAT(1H ,2X,1PE9.2,2X,101A1,2X,1PE11.4)
   85 FORMAT(1H ,8X,1P11E10.2)                                          PRLPL066
   90 FORMAT(1H0,3X,'YSCALE =',1PE11.4,4X,'XSCALE =',1PE11.4)           PRLPL068
      XSCALE=(XMAX-XMIN)/(LINES-1.)                                     PRLPL021
      YSCALE=(YMAX-YMIN)/100.                                           PRLPL022
      DO 10 K=1,11                                                      PRLPL023
   10 ZY(K)=10.*(K-1)*YSCALE+YMIN                                       PRLPL024
      WRITE(6,15) (ZY(K),K=1,11)                                        PRLPL025
      WRITE(6,20)                                                       PRLPL027
      INDEX=1                                                           PRLPL029
   40 X1=XMIN                                                           PRLPL036
      DO 80 I=1,LINES                                                   PRLPL037
      IF(MOD(I,5).EQ.1) GO TO 45                                        PRLPL038
      GRAPH(1)=CR                                                       PRLPL039
      GRAPH(101)=CR                                                     PRLPL040
      GO TO 50                                                          PRLPL041
   45 GRAPH(1)=PL                                                       PRLPL042
      GRAPH(101)=PL                                                     PRLPL043
   50 DO 55 J=2,100                                                     PRLPL044
   55 GRAPH(J)=BLANK                                                    PRLPL045
   57 DO 60 J=1,NY
      IY=(Y(INDEX,J )-YMIN)/YSCALE+1.5                                  PRLPL052
      GRAPH(IY)=SYM(J)                                                  PRLPL053
   60 CONTINUE                                                          PRLPL054
      INDEX=INDEX+1                                                     PRLPL055
      GO TO 70                                                          PRLPL056
   70 IF(NY.EQ.1) GO TO 71
      WRITE(6,75)X(I),(GRAPH(J),J=1,101),Y(I,1),Y(I,2)
      GO TO 80
```

```
   71 WRITE(6,76)X(I),(GRAPH(J),J=1,101),Y(I,1)
   80 CONTINUE                                                    PRLPL063
      WRITE(6,20)                                                 PRLPL064
      WRITE(6,85) (ZY(K),K=1,11)                                  PRLPL065
      WRITE(6,90) YSCALF,XSCALE                                   PRLPL067
      RETURN                                                      PRLPL069
      END
```

```
      SUBROUTINE MARQ1(N,Y,*)
C                                                                       MARQ1002
C      MODIFICATION OF STANFORD UNIVERSITY
C     LIBRARY PROGRAM NUMBER  C008                                      MARQ1003
C     WILLIAM E RIDDLE (SCC)                                            MARQ1004
C     MARCH 9,1967   REVISED APRIL 4,1968                               MARQ1005
C                                                                       MARQ1006
C    THIS SUBROUTINE DOES A NONLINEAR LEAST SQUARE ANALYSIS BASED ON THE METHOD
C    GIVEN BY MARQUARDT.  THE ROUTINE FINDS THE BEST FIT FOR THE K PARAMETERS
C    (OR K/3 SYMMETRIC CURVES) OF THE AX(N) CURVE.  THE INITIAL VALUES OF THE
C    PARAMETERS (FOUND BY SYMPR3) ARE IMPROVED BY MINIMIZING THE SUM OF THE
C    SQUARES OF THE DIFFERENCE BETWEEN THE AX(N) CURVE AND THE LEAST SQUARE
C    CURVE.   THE EFFECTIVENESS OF THE METHOD IS HIGHLY DEPENDENT UPON THE
C    QUALITY OF THE INITIAL VALUES FOR THE PARAMETERS.
C    FOR EACH ITERATION, THE 1ST PARTIAL DERIVITIVES OF ALL PARAMETERS ARE
C    COMPUTED ONCE
C    FOR EACH FUNCTION EVALUATION, THE PARAMETER VALUES ARE USED TO COMPUTE THE
C    LEAST SQUARE APPROXIMATION CURVE.
C    MAXIMUM COMMON USAGE:  X(N),TM(N),SS(5*KMAX/3),TEM(0),DIFF(N),XX(0)
C                                B(KMAX),F(N)
C                                                                       MARQ1015
C SUBROUTINE PARAMETERS                                                 MARQ1016
      INTEGER N,K,IMAX                                                  MARQ1017
      REAL LAMBDP,NU,EPS,P,B,Y,F                                        MARQ1018
C LOCAL VARIABLES                                                       MARQ1021
      REAL TAU,FACT,SUM,S,NORMD,NORMG,DIFF,G,GS,GU,D,DS,DG,B1,A,AS      MARQ1022
      REAL MARQIP,LAMBDA                                                MARQ1023
      INTEGER C1,C2,I,J,L,C3,I1                                         MARQ1026
      DOUBLE PRECISION T1,T2,TSUM
      COMMON X(200),TM(200),SS(200),TEM(200),DIFF(200),XX(200)
      COMMON /LSTSQ/ LAMBDA,NU,EPS,OUTPT,SPD,IMAX,INPLT
      COMMON /RESID/ ZMAX,ICALL
      COMMON /SYM/ B(45),K
      COMMON /PRMCRV/F(200)
      COMMON /LSQERR/ IERR
      DIMENSION P(200,45),Y(1)
      LOGICAL OUTPT
      DIMENSION G(45),GS(45),GU(45),D(45),DS(45),DG(45),B1(45),
     $A(45,45),AS(45,45)
C SUPBROGRAMS CALLED: MARQGS(SUBROUTINE),MARQIP(FUNCTION)               MARQ1027
C                                                                       MARQ1028
C    ----------FORMAT STATEMENTS----------
   89 FORMAT(1H1, ' MARQUARDT NONLINEAR LEAST SQUARES ESTIMATION OF THE
```

```
      $SYMMETRIC COMPONENTS OF THE A(X) CURVE:',/)
   90 FORMAT(///,' GOODNESS OF FIT PRIOR TO LEAST SQUARE PROCESS:',/,
     $' SUM OF SQUARES OF RESIDUALS=',E16.7,/,
     $' CORRELATION COEFFICIENT=',E18.7,//,' NUMBER OF ITERATIONS REQUES
     $TED=',I5)
  551 FORMAT(//,' THE LEAST SQUARE ROUTINE HAS CONVERGED TO A PROPER SOL
     $UTION DURING ITERATION',I3,/)
  601 FORMAT(/' RESULTS FROM ITERATION=',I4,/,' NO. OF FUNCTION EVALUATI
     $ONS=',I4,/,' SUM OF SQUARES OF RESIDUALS=',  E16.8,/,' CORRELATION
     $ COEFFICIENT-A(X) AND COMPUTED CURVE=',E17.7,/,' NO. OF CURVES REJ
     $ECTED DURING THIS ITERATION:',I4)
  901 FORMAT(//,' FINAL APPROXIMATION TO THE PARAMETERS OF THE SYMMETRIC
     $ CURVES - AFTER',I3,' ITERATIONS.',/,9X,'ALPHA',5X,'DEPTH(HOURS)',
     $4X,'X-DIST(HOURS)', 9X,'DEPTH(KM)',4X,'X-DIST(KM)',/,(3F15.5,5X,
     $2F15.5))
  902 FORMAT(1H1,'THE VALUES FOR THE APPROXIMATION TO THE A(X) CURVE ARE
     $:', 10(/,1P5E20.9))
  903 FORMAT(//,' -------ALL SYMMETRIC CURVES HAVE BEEN REJECTED DURING
     $ITERATION',I3,' OF THE LEAST SQUARE ALGORITHM--------',/,
     $' NO. OF FUNCTION EVALUATIONS=',I4,/,' PARAMETERS REJECTED DURING
     $THIS ITERATION:',/,(3F15.5))
  904 FORMAT(///,' THE RESIDUAL VALUES (A(X) - COMPUTED) ARE:',
     $10(/,1P5E20.9))
  905 FORMAT (1H1,' PLOT OF THE LEAST SQUARE APPROXIMATION TO THE A(X) C
     $URVE:  (AFTER',I3,' CHECKS OF THE RESIDUAL VALUES)',
     $/,' (+) A(X) CURVE   (*) LST SQ CURVE   (.) RESIDUAL CURVE')
  906 FORMAT(' PARAMETERS OF REJECTED CURVES:',/,(3F15.5))
  907 FORMAT(' --------RECHECK THE A(X) CURVE AND THE INITIAL VALUES OF
     $THE PARAMETERS--------')
  908 FORMAT(//,' NOTE:  THIS PLOT DOES NOT INCLUDE THE',I3,' SYMMETRIC
     $CURVES WHICH HAVE BEEN REJECTED DURING THE LEAST SQUARE PROCESS.',
     $/,' TO IMPROVE THE RESULTS REFER TO THE FOLLOWING EXPLANATION:',
     $///,9X,'REJECTION CRITERIA',27X,'POSSIBLE CAUSE',34X,
     $'REMEDY',//,' COMPUTED POSITION OUT OF DATA RANGE:',T45,'SMALL PEA
     $K NEAR OR ON LARGE PEAK OR',T90,'USE MORE PEAKS FOR ANALYSIS OR',
     $/,' COMPUTED DEPTH NEGATIVE:',T45,'MANY LARGE PEAKS CLOSE TOGETHER
     $',T90,'USE ADDITIONAL RESIDUAL ANALYSIS',/,' COMPUTED DEPTH LARGER
     $ THAN ZMAXKM:',T45,'ZMAXKM TOO SMALL OR LARGE',T90,'INCREASE ZMAXK
     $M')
  909 FORMAT(///,' ****ERROR****    THE LEAST SQUARE SOLUTION HAS BLOWN
     $UP. PROGRAM EXECUTION HAS TERMINATED ON AN EXPONENT OVERFLOW',
     $/,' OR UNDERFLOW. THE ERROR HAS BEEN CAUSED BY THE REJECTION OF AN
     $ IMPORTANT SYMMETRIC CURVE, THUS CAUSING THE LEAST SQUARE',/,
```

```
C     ----------FILL ARRAY G(K)----------
      TSUM=0.D0
      DO 121 JJ=1,N
      T1=DIFF(JJ)
      T2=P(JJ,I)
  121 TSUM=TSUM+T1*T2
  120 G(I)=TSUM
C     ----------CONTINUE WITH ITERATIVE PROCEDURE----------
      NORMG = MARQIP(G,G,K)                                    MARQ1046
      DO 150 I=1,K                                             MARQ1047
  150 DG(I) = 1.0 / SQRT(A(I,I))                               MARQ1048
      DO 160 I=1,K                                             MARQ1049
        GS(I) = G(I) * DG(I)                                   MARQ1050
        DO 160 J=1,K                                           MARQ1051
  160   AS(I,J) = A(I,J) * DG(I) * DG(J)                       MARQ1052
      FACT = 1.0                                               MARQ1053
      IF (LAMBDA.GT.(1.0E-12)) FACT = 1.0 / NU                 MARQ1054
C                                                              MARQ1055
C                                                              MARQ1056
  200 DO 220 I=1,K                                             MARQ1057
        DO 210 J=1,K                                           MARQ1058
  210     A(I,J) = AS(I,J)                                     MARQ1059
        A(I,I) = AS(I,I) + LAMBDA * FACT                       MARQ1060
  220   GU(I) = GS(I)                                          MARQ1061
      MM=K
      CALL MARQGS (MM,A,GU,DS,$1010)
      DO 230 I=1,K                                             MARQ1063
  230 D(I) = DS(I) * DG(I)                                     MARQ1064
C                                                              MARQ1065
C                                                              MARQ1066
  300 DO 310 I=1,K                                             MARQ1067
  310 B1(I) = B(I) + D(I)                                      MARQ1068
C     ----------EVALUATE THE FUNCTION----------
      CALL  FUNCT(B1,K,F,N)                                    MARQ1069
      C1 = C1+1                                                MARQ1070
      DO 320 I=1,N                                             MARQ1071
  320 DIFF(I) = Y(I) - F(I)                                    MARQ1072
      S = MARQIP(DIFF,DIFF,N)                                  MARQ1073
C                                                              MARQ1074
C                                                              MARQ1075
C     ----------CHECK SUM OF SQUARES----------
      IF (S-SUM) 400,400,500                                   MARQ1076
C                                                              MARQ1077
```

```
      $' SOLUTION TO BECOME UNSTABLE.  REMEDY: CHANGE THE NECESSARY INPUT
      $ PARAMETERS ()ZMAXKM,SHPSPD> OR )NPEAKT,NPEAKD> OR )CUTRSD>) SO',
      $/,' THAT THE CURVE IS NOT REJECTED.  CHECK AND CHANGE:',/,5X,
      $'1. ZMAXKM,SHPSPD - IF THE SYMMETRIC CURVE WAS REJECTED BECAUSE TH
      $E (COMPUTED DEPTH(HRS) * SHPSPD) EXCEEDED ZMAXKM.',/,5X,
      $'2.  NPEAKT,NPEAKD - IF ERROR OCCURED BEFORE A RESIDUAL ANALYSIS.
      $MORE SYMMETRIC CURVES ARE PROBABLY',/,26X,' NEEDED FOR THE LEAST S
      $QUARE SOLUTION.',/,5X,'3,  CUTRSD - IF ERROR OCCURED AFTER A RESID
      $UAL ANALYSIS. TOO MANY SYMMETRIC CURVES (OR CURVES TOO CLOSE TOGET
      $HER)',/,18X,'HAVE BEEN ADDED. TRY LOWERING THE RESIDUAL PEAK CUTOF
      $F LEVEL BY MAKING CUTRSD SMALLER.',/,' FOR HELP IN FINDING THE PRO
      $BLEM, THE FINAL VALUES OF THE PARAMETERS FOR ALL SYMMETRIC',
      $/,' CURVES, AT THE TIME THE  ERROR OCCURED ARE: (ALPHA,DEPTH(HRS),
      $X-POSITION(HRS))',/,(3F15.5))
       IEX=0
       IP=ICALL-1
       TAU = 1.0E-3                                                     MARQ1030
       C2 = 0                                                           MARQ1031
C     ----------COMPUTE INITIAL PARAMETER CURVE----------
       C1 = 1                                                           MARQ1033
       SUM = MARQIP(DIFF,DIFF,N)                                        MARQ1036
C                                                                       MARQ1037
       IF(INPLT.EQ.0) GO TO 70
       CALL PLOT(N,3,Y,F,TM)
   70  CALL CORLAT(N,Y,F,VAL)
       WRITE(6,90) SUM,VAL,IMAX
       WRITE(6,89)
C                                                                       MARQ1038
C     ----------START ITERATIVE PROCEDURE----------
C     ----------COMPUTE 1ST PARTIAL DERIVITIVES----------
  100  CALL DERIV(N,P)
       NREJ=0
       C2 = C2 + 1                                                      MARQ1040
       C3 = 0                                                           MARQ1041
       DO 120 I=1,K                                                     MARQ1042
         DO 110 J=1,K                                                   MARQ1043
C     ----------FILL ARRAY A(K,K)----------
       TSUM=0.D0
       DO 111 JJ=1,N
       T1=P(JJ,I)
       T2=P(JJ,J)
  111  TSUM=TSUM+T1*T2
  110  A(I,J)=TSUM
```

```
C                                                              MARQ1078
C     ----------EXIT FROM ROUTINE----------
  400 DO 410 I=1,K                                             MARQ1079
  410 B(I) = B1(I)                                             MARQ1080
      LAMBDA = LAMBDA * FACT                                   MARQ1081
      SUM = S                                                  MARQ1082
      GO TO 600                                                MARQ1083
C                                                              MARQ1084
C                                                              MARQ1085
  500 IF (C3.NE.0) GO TO 550                                   MARQ1086
C                                                              MARQ1087
C     ----------CHANGE CONVERGENCE FACTOR----------
      NORMD = MARQIP(D,D,K)                                    MARQ1088
      IF(IERR.EQ.0) GO TO 125
      WRITE(6,909) (B1(I),I=1,K)
      STOP
  125 CONTINUE
      S = NORMD * NORMG                                        MARQ1089
      S = MARQIP(D,G,K) / SQRT(S)                              MARQ1090
      IF (S.GT.(.70107)) GO TO 570                             MARQ1091
C                                                              MARQ1092
C     ----------CONTINUE WITH ROUTINE----------
        FACT = FACT * NU                                       MARQ1093
        GO TO 200                                              MARQ1094
C                                                              MARQ1095
C                                                              MARQ1096
  550 IF (C3.NE.40) GO TO 570                                  MARQ1097
      WRITE(6,551) C2
      GO TO 600
C                                                              MARQ1101
C                                                              MARQ1102
  570 C3 = C3 + 1                                              MARQ1103
      DO 580 I=1,K                                             MARQ1104
  580   D(I) = D(I)/NU                                         MARQ1105
      GO TO 300                                                MARQ1106
C                                                              MARQ1107
C                                                              MARQ1108
C     ----------CHECK PARAMETERS----------
  600 CALL PRMCHK(N,NREJ)
      IREJ=NREJ*3
      IF(NREJ.NE.0) IEX=IEX+NREJ
      IF(K.EQ.0) GO TO 700
      CALL CORLAT(N,Y,F,VAL)
```

```
C     ----------PRINT RESULTS----------
      WRITE(6,601) C2,C1,SUM,VAL,NREJ
      IF(IREJ.NE.0) WRITE(6,906) (SS(I),I=1,IREJ)
  602 IF(C2.EQ.IMAX) GO TO 900
      IF(VAL.LT.0.98) GO TO 100
  900 J=1
      DO 910 I=1,K,3
      DO 911 JJ=1,3
      SS(J)=B(I+JJ-1)
  911 J=J+1
      DO 910 JJ=1,2
      SS(J)=B(I+JJ)*SPD
  910 J=J+1
      KK=5*K/3
      WRITE(6,901) C2,(SS(I),I=1,KK)
      IF(NREJ.EQ.0) GO TO 915
      CALL FUNCT(B,K,F,N)
  915 CALL PEAKFT(N,Y)
      IF(.NOT.OUTPT) GO TO 710
      WRITE(6,902) (F(I),I=1,N)
      WRITE(6,904) (DIFF(I),I=1,N)
C                                                                  MARQ1136
C                                                                  MARQ1137
  710 WRITE(6,905) IP
C     ----------PLOT RESULTS----------
      CALL PLOT(N,3,Y,F,DIFF)
      IF(IEX.EQ.0) GO TO 1000
      WRITE(6,908) IEX
      GO TO 1000
  700 WRITE(6,903) C2,C1, (SS(I),I=1,IREJ)
      WRITE(6,907)
 1000 RETURN                                                       MARQ1138
 1010 RETURN 1                                                     MARQ1139
C     LAST CARD OF MARQ1                                           MARQ1140
      END
```

```
      SUBROUTINE MARQGS(M,A,B,X,*)                                      MARQG000
C     CALLED BY MARQ1 - LIBRARY PROGRAM C 008                           MARQG001
C     SOLVES, BY GAUSSIAN ELIMINATION, THF SET OF SIMUL. EQNS. A*X=B    MARQG002
C                                                                       MARQG003
C PARAMETERS                                                            MARQG004
      INTEGER M                                                         MARQG005
      REAL A,B,X                                                        MARQG006
      DIMENSION A(45,45),B(1),X(1)
C LOCAL VARIABLES                                                       MARQG008
      INTEGER I,J,IMAX,K,L,II                                           MARQG009
      REAL MX,T,QUOT                                                    MARQG010
C                                                                       MARQG011
C                                                                       MARQG012
      DO 500 K= 1,M                                                     MARQG013
      MX = 0.0                                                          MARQG014
      IMAX = K                                                          MARQG015
      DO 100 I = K,M                                                    MARQG016
        IF (MX.GT.ABS(A(I,K))) GO TO 100                                MARQG017
          MX = ABS(A(I,K))                                              MARQG018
          IMAX = I                                                      MARQG019
  100 CONTINUE                                                          MARQG020
      IF (MX.EQ.0.0) GO TO 1010                                         MARQG021
      I = K                                                             MARQG022
      IF (K.EQ.IMAX) GO TO 400                                          MARQG023
C                                                                       MARQG024
C                                                                       MARQG025
      J=IMAX                                                            MARQG026
      T=B(I)                                                            MARQG027
      B(I)=B(J)                                                         MARQG028
      B(J)=T                                                            MARQG029
      DO 200 L=1,M                                                      MARQG030
        T = A(I,L)                                                      MARQG031
        A(I,L) = A(J,L)                                                 MARQG032
  200   A(J,L) = T                                                      MARQG033
C                                                                       MARQG034
C                                                                       MARQG035
  400 L=K+1                                                             MARQG036
      IF(L.GT.M) GO TO 500
      DO 450 J=L,M                                                      MARQG037
        QUOT = A(J,K) / A(I,K)                                          MARQG038
        B(J) = B(J) - B(K)*QUOT                                         MARQG039
        DO 450 II=L,M                                                   MARQG040
  450     A(J,II) = A(J,II) - QUOT*A(K,II)                              MARQG041
```

```
C                                                            MARQG042
C                                                            MARQG043
  500 CONTINUE                                               MARQG044
C                                                            MARQG045
      DO 590 K=1,M                                           MARQG046
      I= M + 1 - K                                           MARQG047
      T = 0                                                  MARQG048
      J = I+1                                                MARQG049
      IF(J.GT.M) GO TO 590
      DO 580 L=J,M                                           MARQG050
  580   T = T + A(I,L) * X(L)                                MARQG051
  590 X(I) = (B(I)-T)/A(I,I)                                 MARQG052
C                                                            MARQG053
      RETURN                                                 MARQG054
 1010 RETURN 1                                               MARQG055
      END
```

```
LEVEL  20                    MARQIP              DATE = 73332         17/28/02

       FUNCTION  MARQIP  (V1,V2,N)                                    MARQP000
C      CALLED BY MARQ1 - LIBRARY PROGRAM NUMBER C 008                 MARQP001
C      CALCULATES DOUBLE PRECISION INNER PRODUCT                      MARQP002
C                                                                     MARQP003
C      PARAMETERS                                                     MARQP004
       COMMON /LSQERR/ IERR
       INTEGER N                                                      MARQP005
       REAL V1, V2, MARQIP                                            MARQP006
       DIMENSION V1(N),V2(N)
C      LOCAL VARIABLES                                                MARQP008
       INTEGER I                                                      MARQP009
       DOUBLE PRECISION T1, T2, SUM                                   MARQP010
C                                                                     MARQP011
       SUM=0.D0
       DO 10 I=1,N                                                    MARQP013
       T1 = V1(I)                                                     MARQP014
       T2 = V2(I)                                                     MARQP015
       IF(T1*T2.LT.1.E60.AND.T1*T2.GT.1.E-60) GO TO 8
       IERR=1
       RETURN
     8 CONTINUE
    10 SUM = SUM + T1 * T2                                            MARQP016
       MARQIP = SUM                                                   MARQP017
       RETURN                                                         MARQP018
       END
```

```
      SUBROUTINE DERIV(N,P)
C     THIS SUBROUTINE IS USED BY SUBROUTINE MARQ1 TO COMPUTE THE DERIVITIVES
C     OF THE ANALYTINC EXPRESSION FOR THE SYMMETRIC BELL SHAPED CURVES WHICH
C     COMPOSE THE AX(X) CURVE.  THIS EXPRESSION IS F(X)=ALPHA**2/H**2+(X-POS)**2
C     N=NO. OF DATA POINTS
C     P(N,K)= ARRAY CONTAINING THE FIRST PARTIAL DERIVITIVES OF THE FUNCTION
C     OTHER PARAMETERS USED BY THE SUBROUTINE ARE:
C     K= NO. OF PARAMETERS TO BE MINIMIZED
C     B(K)= ARRAY CONTAINING THE PARAMETERS
C     F(X) WITH RESPECT TO EACH OF THE PARAMETERS AND AT EACH DATA POINT
C     MAXIMUM COMMON USAGE:  X(N),XX(O),B(KMAX)
      COMMON X(200),XX(1000)
      COMMON /SYM/ B(45),K
      REAL*8 D,C,CON,CON2
      DIMENSION P(200,45)
      L=K/3
      DO 10 I=1,N
      DO 10 J=1,K,3
      DIST=X(I)
      ALPHA=B(J)
      DEPTH=B(J+1)
      PKPOS=B(J+2)
C     ----------COMPUTE 1ST PARTIAL DERIVITIVES----------
      D=2.DO*ALPHA**2
      C=DIST-PKPOS
      CON=DEPTH**2+C**2
      CON2=CON**2
C     ----------ALPHA(A)----------
      P(I,J)=D/(CON*ALPHA)
C     ----------DEPTH(H)----------
      P(I,J+1)=-D*DEPTH/CON2
C     ----------POSITION(C)----------
   10 P(I,J+2)=D*C/CON2
      RETURN
      END
```

```
      SUBROUTINE FUNCT(B,K,F,N)
C     THIS SUBROUTINE IS USED BY SUBROUTINE MARQ1 TO DETERMINE THE VALUES OF THE
C     FUNCTION (WHICH IS BEING USED FOR THE LEAST SQUARE MINIMAZATION) AT EACH
C     OF THE DATA POINTS
C     K=NO. OF PARAMETERS BEING MINIMIZED
C     N= NO. OF DATA POINTS
C     B(K)= ARRAY CONTAINING THE PARAMETER VALUES
C     F(N)= ARRAY CONTAINING THE OUTPUT FUNCTION VALUES
C     MAXIMUM COMMON USAGE:  X(N),XX(0)
      COMMON X(200),XX(1000)
      REAL*8 SUM
      DIMENSION B(1),F(1)
      SUM=0.D0
      DO 3 I=1,N
      DO 2 J=1,K,3
      DIST=X(I)
      ALPHA=B(J)
      DEPTH=B(J+1)
      PKPOS=B(J+2)
C     ----------SUM OF ALL CURVES AT POINT X----------
    2 SUM=SUM+(ALPHA**2)/((DEPTH**2)+(DIST-PKPOS)**2)
      F(I)=SUM
    3 SUM=0.D0
      RETURN
      END
```

```
      SUBROUTINE PRMCHK(N,NREJ)
C     THIS SUBROUTINE CHECKS THE PARAMETERS OF THE SYMMETRIC CURVES TO SEE IF:
C       1. SOURCE DEPTHS ARE NEGATIVE
C       2. SOURCE DEPTHS ARE GREATER THAN ZMAX
C       3. X-POSITIONS LIE OUTSIDE THE RANGE OF THE DATA
C     IF ANY CONDITION IS FOUND, THE ROUTINE THROWS AWAY THAT SYMMETRIC CURVE
C     AND REARRANGES THE ARRAY CONTAINING THE PARAMETERS
C     N=NO. OF DATA POINTS
C     NREJ=NO. OF SYMMETRIC CURVES REJECTED
C     OTHER PARAMETERS USED IN THE SUBROUTINE ARE:
C     K=NO. OF PARAMETERS
C     B(K)=REAL ARRAY CONTAINING THE PARAMETERS
C     MAXIMUM COMMON USAGE:  X(N),TT(0),TM(KMAX),XX(0)
      COMMON X(200),TT(200),TM(200),XX(600)
      COMMON /RESID/ ZMAX,ICALL
      COMMON /SYM/ B(45),K
      NT=1
      L=K
C     ----------CHECK PARAMETERS----------
      DO 11 I=1,K,3
      IF(B(I+2).GT.X(N).OR.B(I+2).LT.X(1)) GO TO 12
      IF(B(I+1).LT.0..OR.B(I+1).GT.ZMAX) GO TO 12
      GO TO 11
   12 L=K-3*NT
      NP=3*NREJ
      NREJ=NREJ+1
      NT=NT+1
C     ----------REJECT BAD CURVES----------
      DO 13 J=1,3
      TM(NP+J)=B(I+J-1)
   13 B(I+J-1)=0.
   11 CONTINUE
      NT=1
C     ----------REARRANGE PARAMETER ARRAY----------
      DO 14 I=1,K
      IF(B(I).EQ.0.) GO TO 14
      B(NT)=B(I)
      NT=NT+1
   14 CONTINUE
      K=L
      RETURN
      END
```

```
      SUBROUTINE RSDCHK(N,AX)
C     THIS SUBROUTINE CHECKS THE RESIDUAL CURVE (IE AX(I)-COMPUTED) FOR
C     ADDITIONAL SYMMETRIC PEAKS WHICH DO NOT OCCUR AS LOCAL MAXIMA ON THE
C     A(X) CURVE.  NEW PEAKS FOUND IN THE RESIDUAL CURVE (BY THE SAME CRITERIA
C     AS THE A(X)) ARE ADDED TO THE LIST OF OLD SYMMETRIC PEAKS IF THEY ARE
C     LOCATED A DISTANCE > 2 * DX AWAY FROM AN OLD PEAK. (STATEMENTS 51 & 52)
C     NEW PEAKS ARE NOT KEPT IF THEY ARE LESS THAN THIS DISTANCE
C         N = NO. OF DATA POINTS IN A(X)
C         AX(N) = REAL ARRAY CONTAINING THE A(X) CURVE
C     OTHER PARAMETERS USED IN THE SUBROUTINE ARE:
C         R(N) = REAL ARRAY CONTAINING THE RESIDUAL VALUES  (A(X) - COMPUTED)
C         K = NO. OF PARAMETERS
C         B(K) = REAL ARRAY CONTAING THE PARAMETER VALUES
C         ZMAX = MAXIMUM ESTIMATED DEPTH TO THE MAGNETIC SOURCE BODY
C         NADD = NO. OF CURVES ADDED FROM RESIDUAL ANALYSIS
C         ND = NO. OF CURVES DELETED BY THE RESIDUAL ANALYSIS
C         KMAX = MAXIMUM NO. OF PARAMETERS ALLOWED
C     MAXIMUM COMMON USAGE:   X(N),Q(O),REJ(KMAX),W(2*KMAX),R(N),F(KMAX)
C                             B(KMAX),FF(N)
      COMMON X(200),Q(200),REJ(200),W(200),R(200),F(200)
      COMMON /LSTSQ/ LAMBDA,NU,EPS,OUTPT,SPD,IMAX,INPLT
      COMMON /RESID/ ZMAX,ICALL
      COMMON /SYM/ B(45),K
      COMMON /PRMCRV/FF(200)
      COMMON /CUT/ CUTAX,CUTRSD,AMIN,AMAX
      DIMENSION AX(1)
C     ----------FORMAT STATEMENTS----------
  100 FORMAT(//,' ----------THE TOTAL NUMBER OF SYMMETRIC PEAKS HAS REAC
     $HED',I3,' - THE MAXIMUM ALLOWABLE----------')
  101 FORMAT(1H1,'RESULTS FROM ANALYSIS NUMBER',I3,' OF THE RESIDUAL CUR
     $VE:  (AFTER THE LEAST SQUARE PROCESS)',//,' THE CUTOFF LEVEL FOR T
     $HE RESIDUAL PEAKS=',1PE14.6)
  102 FORMAT(/,' THE',I3,' NEW SYMMETRIC CURVES ADDED HAVE THE PARAMETER
     $S:  (ONLY RESIDUAL PEAKS WITH VALUES > CUTOFF ARE CONSIDERED)',/,
     $(3F15.5))
  103 FORMAT(/,' THE',I3,' OLD SYMMETRIC CURVES REMOVED HAVE THE PARAMET
     $ERS:',/,(3F15.5))
  104 FORMAT(/,' THE',I3,' NEW PARAMETERS OF THE',I3,' SYMMETRIC CURVES
     $(AFTER RESIDUAL ANALYSIS) ARE:',/,(3F15.5))
  105 FORMAT(/,' NO NEW SYMMETRIC CURVES HAVE BEEN ADDED')
  106 FORMAT(/,' NO OLD SYMMETRIC CURVES HAVE BEEN REMOVED')
  107 FORMAT(/,' THE VALUES OF THE PARAMETERS REMAIN UNCHANGED AFTER THE
     $ RESIDUAL ANALYSIS')
```

```
  108 FORMAT(1H1,'PLOT OF THE A(X) CURVE(+), NEW PARAMETER CURVE(*), AND
     $ TOTMAG CURVE(.):  (AFTER RESIDUAL ANALYSIS',I2,' & BEFORE LST SQR
     $ ANALYSIS',I2,')')
      KMAX=45
      NADD=0
      NN=1
      M=N-1
      KN=K+1
      DX=X(2)-X(1)
      ND=0
      WRITE(6,101) ICALL,CUTRSD
      DO 10 I=1,K
   10 W(I)=B(I)
      DO 1 I=2,M
C     ----------FIND LOCAL MAXIMA---------
      IF(R(I).LT.CUTRSD) GO TO 1
      IF(R(I).LE.R(I+1).OR.R(I).LE.R(I-1)) GO TO 1
C     ----------CHECK 2ND DERIVITIVE CRITERIA----------
      DER2=ABS(R(I+1)-2.*R(I)+R(I-1))
      DEP1=SQRT(2.*R(I)/DER2)*DX
      IF(DEP1.GT.ZMAX) GO TO 1
      IF(KN-3*ND.GT.KMAX) GO TO 7
      NADD=NADD+1
C     ----------FIND INITIAL DEPTH----------
      T=DEP(N,I,R)
C     ----------ADD TEMPORARY NEW PEAKS----------
      F(NN)=SQRT(T**2*AX(I))
      F(NN+1)=T
      F(NN+2)=X(I)
C     ----------CHECK FOR REPETITIVE PEAKS---------
      IF(NN.EQ.1) GO TO 4
C     ----------DELETE NEW PEAK IF WITHIN 2*DX OF OLD PEAK----------
    2 DO 3 JJ=1,K,3
   51 IF(ABS(F(NN-1)-W(JJ+2)).GE.2.*DX) GO TO 3
      NADD=NADD-1
      GO TO 4
    3 CONTINUE
C     ----------ADD NEW RESIDUAL PEAKS----------
      DO 15 J=1,3
   15 W(KN+J-1)=F(NN+J-4)
      KN=KN+3
    4 NN=NN+3
      LL=I
```

```
    1 CONTINUE
      IF(NADD.EQ.0) GO TO 8
C     ----------CHECK LAST TEMPORARY PEAK----------
      DO 16 JJ=1,K,3
   52 IF(ABS(F(NN-1)-W(JJ+2)).GE.2.*DX) GO TO 16
      NADD=NADD-1
      GO TO 8
   16 CONTINUE
      DO 17 J=1,3
   17 W(KN+J-1)=F(NN+J-4)
      KN=KN+3
      GO TO 8
    7 KPMAX=KMAX/3
      WRITE(6,100) KPMAX
    8 NT=1
      KR=ND*3
      KL=K+1
C     ----------OUTPUT RESULTS----------
      IF(NADD.NE.0) GO TO 20
      WRITE(6,105)
      GO TO 21
   20 K=KN-1
      WRITE(6,102) NADD,(W(I),I=KL,K)
   21 IF(ND.NE.0) GO TO 22
      WRITE(6,106)
      IF(NADD+ND.NE.0) GO TO 23
      WRITE(6,107)
      GO TO 30
   22 WRITE(6,103) ND,(REJ(I),I=1,KR)
   23 CONTINUE
C     ----------REARRANGE THE PARAMETER ARRAY----------
      DO 6 I=1,K
      IF(W(I).EQ.0.) GO TO 6
      B(NT)=W(I)
      NT=NT+1
    6 CONTINUE
      K=K-3*ND
      KL=K/3
C     ----------OUTPUT PARAMETER VALUES FOR NEW PEAKS----------
      WRITE(6,104) K,KL,(B(I),I=1,K)
   30 CONTINUE
      IF(NADD+ND.EQ.0) GO TO 32
      CALL FUNCT(B,K,FF,N)
```

```
      DO 35 JT=1,N
   35 R(JT)=AX(JT)-FF(JT)
      CALL PEAKFT(N,AX)
   32 IF(INPLT.EQ.0) GO TO 31
      ITEM=2*ICALL
      WRITE(6,108) ICALL,ITEM
   31 ICALL=ICALL+1
      RETURN
      END
```

```
      SUBROUTINE INPUT(N,TIME,TMAG)
C     THIX SUBROUTINE READS TIMES AND TOTAL FIELD MAGNETIC VALUES FROM CARDS
C     (BARTLETT FORMAT) AND CONVERTS TIME INTO HOURS
      DIMENSION TIME(1),TMAG(1)
  100 FORMAT (3I6,F8.1,F2.0,F10.0)
  200 FORMAT(///,' DATA SUPPLIED BY THE USER:')
  201 FORMAT(/,16X,'YEAR  DAY   TIME',/,' START PROFILE:',I4,I6,I5,
     $F4.1,/,' END PROFILE:',2I6,I5,F4.1)
      TIME(1)=0.
      N=1
      READ(5,100) IY1,ID1,IH1,AM1,AS1,TMAG(1)
      AM1=AM1+AS1/60.
    1 READ(5,100,END=10) IY,ID,IH,AM,AS,TOTMAG
      AM=AM+AS/60.
      TEMP   =TMDF(IY1,ID1,IH1,AM1,IY,ID,IH,AM)/60.
      IF(TEMP.EQ.TIME(N)) GO TO 1
      N=N+1
      IF(N.GT.200) GO TO 10
      TMAG(N)=TOTMAG
      TIME(N)=TEMP
      GO TO 1
   10 WRITE(6,200)
      WRITE(6,201) IY1,ID1,IH1,AM1,IY,ID,IH,AM
      RETURN
      ENTRY PRINT(SHR,EHR)
      TEM=SHR+IH1
      DO 11 I=1,2
      IHRI=TEM
      AMI=(TEM-IHRI)*60.+AM1
      IDI=ID1
      IF(AMI.LT.60.) GO TO 12
      AMI=AMI-60.
      IHRI=IHRI+1
   12 IF(IHRI.LT.24) GO TO 13
      IHRI=IHRI-24
      IDI=IDI+1
   13 IF(I.EQ.2) GO TO 11
      IDS=IDI
      IHS=IHRI
      AMS=AMI
      TEM=EHR+IH1
   11 CONTINUE
      WRITE(6,201) IY1,IDS,IHS,AMS,IY1,IDI,IHRI,AMI
      Return
      END
```

```
LEVEL  20                TMDF                DATE = 73332         17/28/02

      FUNCTION TMDF(IY1,ID1,IH1,AM1,IY2,ID2,IH2,AM2)
C     THIS FUNCTION FINDS THE TIME DIFFERENCE IN MINUTES
      TMDF = 1440*(ID2-ID1)+60*(IH2-IH1)+AM2-AM1
      J=IY2-IY1
      IF(J.EQ.0) RETURN
      TMDF=TMDF+525600*J+1440*((IY2-1)/4-(IY1-1)/4)
      RETURN
      END
```