CUC THE INFORMATION SYSTEMS COMPANY

NASW-417

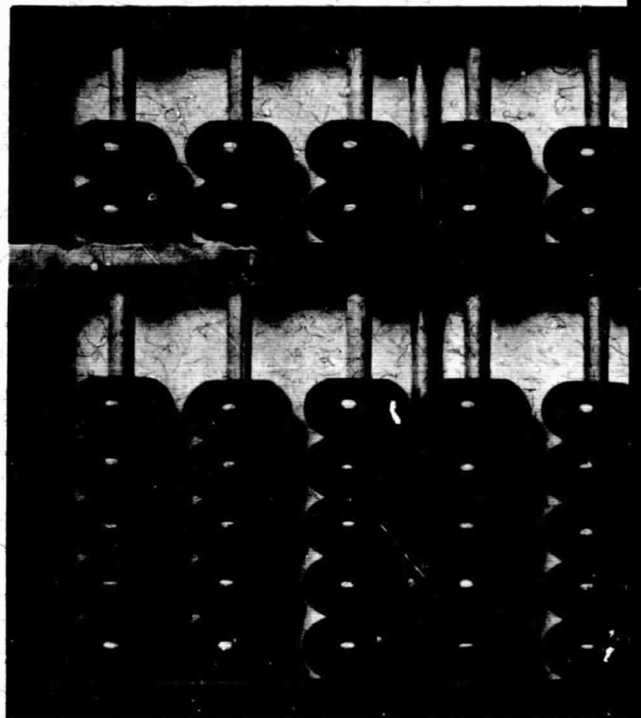
VOLUME 2 - MAINTENANCE MANUAL

A COMPUTER PROGRAM FOR POWER SPECTRAL ANALYSIS OF UNEQUALLY SPACED POINTS

By

Matthew Lybanon

PDN 70-502
January 30, 1970



VOLUME 2 - MAINTENANCE MANUAL

A COMPUTER PROGRAM FOR POWER SPECTRAL ANALYSIS OF UNEQUALLY SPACED POINTS

By

Matthew Lybanon

PDN 70-502
January 30, 1970

ACKNOWLEDGMENT

The work described in this report was performed for the Guidance and Navigation Department of Bellcomm, Inc. under Contract No. 10646 (CUC Contract 3645-220). Special thanks are given to Mr. B. G. Niedfeldt for several helpful discussions and suggestions.

ABSTRACT

The computer program whose mathematical method is derived in Volume 1 is discussed in some detail. The discussion and tables cover the contents of COMMON blocks, major calculations, subroutine argument lists, and program logic. Appendix 1 gives a listing of the program system, and Appendix 2 is an overall logic flowchart.

INTRODUCTION

This Maintenance Manual gives detailed information about the power spectral density computer program whose mathematical method and general form are discussed in Volume 1 of this report, the User's Manual. All equation numbers given in this volume refer to equations in the User's Manual.

This manual contains discussions of the functions of individual program elements, and information on the contents of COMMON blocks and argument lists. Appendix 1 is a listing of the entire program. Appendix 2, for convenience, shows the same overall logic flowchart contained in Appendix 1 of the User's Manual.

The entire program is written in FORTRAN V for the UNIVAC 1108. Each subprogram's source and relocatable elements, the MAP source element, and the absolute element, are in file LSQFIT*MLTSAN. in Bellcomm's computer system. The name of the absolute element is LSQFIT*MLTSAN.APSD.

In the subsequent discussion, sequence numbers refer to the program listings in Appendix 1. Equation numbers are those of equations in Volume 1, the User's Manual.

PROGRAM DESCRIPTION

Contents of COMMON Blocks

Blank COMMON:

PSPECT (1000) - array of spectral powers
Y (2000) - array of observations (d. p.)*
T (2000) - array of times (independent variable) (d. p.)
NAME (3) - identifying name of data
A (1000) - array of sine coefficients (d. p.)
B (1000) - array of cosine coefficients (d. p.)
PERCNT (1000) - array of percent powers

NOPEAK:

KNO (30) - array of subscripts of false peaks
NOPEAK - number of false peaks found since last removal
MOPEAK - number of false peaks from latest search

MORN:

NORM - norm of inverse times matrix, with 1 subtracted from diagonal elements (d. p.)

CUCPSA

Important Quantities Calculated

Quantity	Sequence Number	Equation	Comments
OMEGAZ	135	56	$2\pi f$ (d. p.)
KMAX	140	57	Index of highest frequency term
A(KK)	185-186	47	Sine coefficients (d. p.)
B(KK)	187-188	46	Cosine coefficients (d. p.)
PSPECT(KK)	192-194	54	Spectral power
PERCNT(KK)	195		Percent power
Y(I)	408-409		Prewhitened signal (d. p.)
YBAR	418		Mean of data (or residuals) (d. p.)
YYBAR2	424-425		Total power (variance) (d. p.)

* "d. p. " denotes a double precision variable.

Subroutine Calls

Subroutine	Arguments	Comments
PRNTCN		System routine – interface to print-page controls.
	2,'M,66,5,3.'	Changes margins to give 66-line page with 5-line top and 3-line bottom margins.
READ		Tape read subroutine for input data.
	N	Number of points.
	IY	Position of observation in record.
	IT	Position of time in record.
	TPC	Time scale factor.
RPLOT		Plots spectrum on one page.
	ITAPE	Scratch unit containing plot information.
GRAPH		Plots expanded spectrum.
	PERCNT	Array of percent powers.
	PSPECT	Array of spectral powers.
	KLIMIT	N/2 (N=number of points).
	NAME	Identifying name of data.
	HEAD3A	'P VS K'
	PRD	Fundamental period of data.
	RS	Product of range and spacing factors.
CAROL3		Solves for dominant terms.
	AREMOV	Array of sine coefficients (output) (d. p.)
	BREMOV	Array of cosine coefficients (output) (d. p.)
	PREMOV	Array of spectral powers (output)
	OREMOV	Array of angular frequencies (output) (d. p.)
	N	Number of points.
	KREMOV	Array of indices of peaks.
	OMEGAZ	Fundamental angular frequency ($2\pi f$) (d. p.)
	NTERMS	Number of peaks found in search.

CUCPSA is the main program in the system. Its main functions are to calculate the power spectrum, search the spectrum for peaks, call the subroutine which improves the estimates of the peaks, remove the peaks, and calculate the spectrum of the residuals (prewhitened signal). It also takes care of input and output.

Sequence numbers 144-197 are the portion of the program which calculates the coefficients, spectral powers, and percent powers. The search for peaks is handled by 322-384. The call to CAROL3 (which improves the estimates of coefficients and frequencies of peaks) is sequence number 394. The removal is done by 400-411. After each removal, the spectrum of the residuals is calculated by the same section of coding which calculates the original spectrum.

Sequence numbers 31-34, 88-105, 207-225, and 293-313 are the spectrum-smoothing option described in the User's Manual. These statements are currently in the program as comments. To activate this feature, the "C" in column 1 of each FORTRAN statement should be removed.

The spectrum search section searches the unsmoothed spectrum for peaks. However, there is also a provision for smoothing the spectrum as the search for peaks is made. To activate this feature the "C" in column 1 of each FORTRAN statement in sequence numbers 333-344 must be removed, and sequence numbers 345-351 should be deleted.

There are two flags (other than the print, plot, and removal flags) which control the flow of logic through the program. IFLAG is set equal to 0 at the beginning of each case. Its value is changed to 1 the first time CAROL3 successfully solves for peaks. There are a number of branch points at which the value of IFLAG determines what is done next. The logical variable LFLG is given the value .TRUE. at the beginning of the first search for peaks. When no more peaks can be found, TOL is cut in half and LFLG becomes .FALSE., which prevents TOL from being diminished again.

A modification which would permit a "magnified" view of one or more portions of the spectrum might be useful. This feature could be implemented in addition to or in place of the usual spectrum calculation; it could be triggered by input or by logic based on the result of some calculation. For example, it might be helpful to see the fine structure of regions of the spectrum where the search-and-removal process has trouble.

The variables KTOTAL, FMESH, and FBEGIN (sequence numbers 144-146) lend themselves to this application. For example, setting KTOTAL=101, FMESH=.02, FBEGIN=39, would cause the spectrum calculation to begin at k=39 and proceed to k=41, bracketing k=40 in 100 steps. If this modification is not made, the program could be optimized slightly by eliminating these variables and rewriting the beginning of the spectrum calculation.

CAROL3

Important Quantities Calculated

Quantity	Sequence Number	Equation	Comments
AREMOV(M)	130	55	a_k (d.p.)
BREMOV(M)	131		b_k (d.p.)
OREMOV(M)	132		$2\pi f_k$ (d.p.)
PREMOV(M)	142	54	Powers

Subroutine Calls

Subroutine	Arguments	Comments
BIORTH		Matrix inversion routine
	AMTRX	Matrix to be inverted (d.p.)
	AI	(Left) inverse of AMTRX (d.p.)
	NT3	Rank of AMTRX

CAROL3 is the subroutine which calculates the frequencies and coefficients of dominant terms in the spectrum. It does so by an iterative differential correction least squares technique. Such a method fits a model $\hat{\phi}(p_1, \ldots, p_m)$

to an observed variable ϕ by adjusting the parameters $p_1, \ldots, p_m$ to minimize the mean square error. The values of the parameters are adjusted iteratively according to:

$$p^{(k)} = p^{(k-1)} + \Delta p^{(k)}$$

$$\Delta p^{(k)} = \left(A^T W A\right)^{-1} A^T W \left[\phi - \hat{\phi}\left(p^{(k-1)}\right)\right]$$

where p denotes the vector whose components are $p_1, \ldots, p_m$, A is the matrix with elements $\frac{\partial \hat{\phi}_i (p)}{\partial p_j}$, i=1, ... , N (the number of observations) and j=1, ..., m (the number of parameters), W is a (diagonal) weighting matrix, and superscripts in parentheses label the stage of iteration. (In calculating $\Delta p^{(k)}$, A must be evaluated using $p^{(k-1)}$.)

In CAROL3 the parameters which are adjusted are the amplitudes, phases, and frequencies of spectral peaks. (This was found to lead to faster convergence than the two coefficients and frequency formulation of equation (55).) Frequency is the most critical parameter, so it is handled in perturbation form. The f_k of (55) is represented by $(k+\alpha)f$, where f is defined by (56), kf is the frequency found in the spectrum search, and α, $|\alpha| \leq .5$, is the quantity solved for.

The parameters p_j, j=1, ..., m (NT3 stands for m in the program) are: amplitude, phase, α, amplitude, phase, α, ... , one group of three parameters for each spectral peak. The identity matrix is used as the weighting matrix W in CAROL3. The square matrix $A^T A$ is called AMTRX; $A^T \left[\phi - \hat{\phi}\left(p^{(k-1)}\right)\right]$ is called BMTRX. They are set up by sequence numbers 56-67. Sequence number 69 is the call to BIORTH, which computes the inverse of AMTRX. The vector $p^{(k-1)}$ is called AKOLD, $\Delta p^{(k)}$ is called DA, and $p^{(k)}$ is called AKNEW. The section of coding which does the differential corrections is sequence numbers 72-77.

It is assumed that if the frequency corrections are too large, something is wrong (e.g. perhaps an apparent peak is only a sidelobe of a large peak). So the values of AKNEW(3), AKNEW(6), ... - that is, α for each peak - are checked. If any of them exceed 0.5, the corresponding peaks are considered "bad" peaks. They are "tagged" and a return is made to the main program to search the spectrum for the MTERMS largest peaks, not including the tagged ones, which exceed TOL. This may happen several times before CAROL3 converges on a solution./ After a successful solution (and removal) the tags are eliminated from the bad peaks – which may or may not be peaks in the spectrum of the residuals.

This process is done by keeping track of the k-values of the bad peaks. KNO is the array of bad k-values. NOPEAK is the total number of bad peaks found. MOPEAK counts the number found on the latest call to CAROL3. Before the return to the main program, MOPEAK is set equal to the subscript (of KNO) which indexes the first bad peak found in the current call to CAROL3. This portion of the program consists of sequence numbers 78-89.

Convergence is checked by observing the change in the sum of squares of the residuals (data-fit). The current value of the sum of squares is called DELTYN (calculated in sequence numbers 96-106); the value from the preceding iteration is called DELTYO. The quantity checked is TESTIT=DELTYO/DELTYN - 1.0. The differential correction process is permitted to go through three iterations before TESTIT is checked. Then it is checked after each iteration. TESTIT< 0 implies divergence. In this event there is a return to the main program with the values of the parameters from the preceding iteration chosen as the "best" values. If the number of iterations exceeds NITER (set equal to 50 in a DATA statement) before convergence, an error message is printed and execution is terminated. TESTIT < EPSLN (set equal to 10^{-8} in a DATA statement) is the condition for convergence. Sequence numbers 109-119 are the convergence check calculations.

The coefficients and frequencies found in the coarse search are converted to the fit parameters (amplitudes, phases, frequency correction factors) in sequence numbers 26-33. After convergence, the transformation back to coefficients and frequencies, along with the computation of spectral powers, are handled by 128-143.

Other Subroutines

The remaining subroutines in the system are straightforward. The following notes illiminate the principal features of the programs:

READ - The data tape is assumed to be a binary tape consisting of N (the number of points) records, each containing 7 (or more) single-precision words. One of the words is time, identified by IT; there is no day variable. TPC is the time scale factor. IY labels the data word. The tape is assumed to be on logical unit 3.

RPLOT - Automatic scale selection is done. The entire plot appears on one printer page, with each scale running from 0 to the maximum value of power or k. The scaling information is computed in CUCPSA and passed to RPLOT, along with the other plotting information, on scratch unit ITAPE. ITAPE (set equal to 19 in CUCPSA) is assigned by the operating system; no ASG statement is needed.

GRAPH - Automatic scale selection is done for the percent power scale. Three scales are available, with maxima of 20%, 40%, or 100%.

BIORTH - This subroutine is internally documented by means of embedded comments. It should be noted that BIORTH calls two internal subroutines, DOT and SCLR.

PROGRAM FILE

The entire program is contained on FASTRAND file LSQFIT*MLTSAN. The name of the MAP source element is LSQFIT*MLTSAN. PSAMAP. The absolute element is LSQFIT*MLTSAN. APSD.

The file has a write key, JULIE. Therefore, the control card:

```
@   ASG, A   LSQFIT*MLTSAN//JULIE.
```

must be used if it is necessary to write on the file. Tape number 1015 (permanent) is a save tape of LSQFIT*MLTSAN.

APPENDIX I

Program Listing

```
LSQFIT*MLTSAN.CUCPSA
      C
      C      POWER SPECTRUM ANALYSIS PROGRAM
      C
      C ..........................................................................
      C
      C
             DOUBLE PRECISION T,Y,YYBAR2,YBAR,A,B,SUMYS,SUMYC,SUMS2,
            . SUMC2,SUMSC,DELTA,XSIN,XCOS,AREMOV(10),BREMOV(10),
            . TWOPI,DUR,OMEGAZ,OMEGA,STEP,THETA,OREMOV(10)
             COMMON PSPECT(1000),Y(2000),T(2000),NAME(3),
            1       A(1000),B(1000),PERCNT(1000)
              COMMON /NOPEAK/ KNO(30),NOPEAK,MOPEAK
             LOGICAL LFLG
      C
             DATA XEND/4HEND /,VTITLE/6H POWER/,HTITLE/6H    K  /,
            1     IVERT/1/,IHORZ/2/,ISHADE/0/,HEAD3A/6HP VS K/
             DATA IBLANK /6H      /
             EQUIVALENCE (FMT(1),IFMT1)
      C
             DIMENSION FMT(14)
             DOUBLE PRECISION V(3)
             DIMENSION KREMOV(10),PREMOV(10),FREMOV(10),PDREMV(10)
      C
             CALL PRNTCN (2,'M,66,5,3,')
         100 CONTINUE                       @     START A NEW CASE.
             ICOUNT = 0
      C
      C      READ DATA SET AND INITIALIZE CONSTANTS
      C
             IFLAG=0
      C      FILTER(1) = 0.25
      C      FILTER(2) = 0.50
      C      FILTER(3) = 0.25
      C      LMAX = 3
             TWOPI = 6.2831853076D0
             ITAPE = 10
             NTAPE=0
      C
      C      FMT IS FORMAT FOR READING OBSERVATION CARDS
      C      IY IS OBSERVATION FIELD NUMBER (ON OBSERVATION CARDS)
      C      IT IS TIME FIELD NUMBER
      C      ID IS DAY FIELD NUMBER
      C      TPC IS TIME UNITS PER COMPUTATIONAL UNIT (SEE TIME CALC.)
      C      CPD IS COMPUTATIONAL UNITS PER DAY (SEE TIME CALC.)
      C
             READ (5,1005) FMT,IY,IT,ID,TPC,CPD
        1005 FORMAT (13A6,A2/3I5,2F10.5)
              IF (IY .EQ. -999) STOP        @     END EXECUTION.
              IF (IFMT1 .EQ. IBLANK)        @     DESIGNATES
            .  NTAPE = 1                    @      TAPE INPUT
      C
      C      NAME IDENTIFIES DATA
      C      N IS NUMBER OF OBSERVATIONS (LIMITED TO 2000)
      C      S IS SPACING CONSTANT FOR OMEGAZ
      C      R IS RANGE CONSTANT FOR KMAX
      C      IPO = 0 PLOTS ROUGH (AND SMOOTH) POWER SPECTRUM
      C             = 1 PLOTS ROUGH POWER SPECTRUM ONLY
      C             = 2 (PLOTS SMOOTH POWER SPECTRUM ONLY)
      C             = 3 NO PLOT
      C        JF = 0 REMOVE FREQUENCIES WHOSE PERCENT POWERS EXCEED TOL
      C           = 1 DO NOT REMOVE DOMINANT FREQUENCIES
      C        TOL IS REMOVAL TOLERANCE DESCRIBED ABOVE
      C        JS = 0 PRINT SPECTRUM
      C           = 1 DO NOT PRINT SPECTRUM
      C
               READ (5,1000) NAME,N,S,R,JPO,JF,TOL,JS
               IF (NTAPE .EQ. 1) CALL READ (N,IY,IT,TPC)
               IF (N.GT.2000) N = 2000
      C **     TOL MAY BE EITHER AN INPUT OR A CALCULATED QUANTITY.
                IF (TOL .LE. 0)                 @     SET TOL TO 95% SIGNIFICANCE
             .  TOL = 10. / FLOAT(N)            @      LEVEL UNLESS GIVEN AS INPUT.
               READ (5,1001) MTERMS             @     MAXIMUM NUMBER OF TERMS IN FINE FIT
               IF (MTERMS .GT. 10) MTERMS = 10
               WRITE (6,2000) NAME,N,S,R,JPO,JF,TOL,JS,MTERMS
        1000 FORMAT (3A6,I5,2F10.5,2I1, F10.5,I1)
        1001 FORMAT (I5)
        2000 FORMAT (1H1/  5X,13HSET NAME     = 3A6
            1                 /5X,13HNO. POINTS = I5
            2                 /5X,13HSPACING    = G13.9
            3                 /5X,13HRANGE      = G13.9
            4                 /5X,13HPLOT FLAG  = I5
            5                 /5X,13HREMOVAL    = I5
            6                 /5X,13HTOLERANCE  = F10.5
            7                 /5X,13HPRINT FLAG = I5
            8                 /5X,13HMTERMS     = I5)
                IF (NTAPE .EQ. 1) GO TO 4321
      C
      C        IF (S.EQ. 1.0) GO TO 110
      C        READ 1001,LEND,(FILTER(I),I=1,LEND)
      C1001 FORMAT (I5,7F10.5)
      C        LMAX = 2 * LEND -1
      C        LF= LEND
      C        LM= LMAX
      C 105 FILTER(LM) = FILTER(LF)
      C        LF = LF-1
      C        LM = LM-1
      C        IF (LE.GT.0) GO TO 105
      C        LY = LM
      C 107 FILTER(LM) = FILTER(LY)
      C        LY = LY + 1
      C        LM = LM-1
      C        IF (LM.GT.0) GO TO 107
      C 110 PRINT 2001,(FILTER(I),I=1,LMAX)
      C2001 FORMAT ( 5X,12HFILTER       =    7F11.5/17X,6F11.5 )
      C        LCENTR = (LMAX+1)/2
      C
      C        READ FIRST OBSERVATION CARD
               READ (5,FMT) V
      C
      C        COMPUTE INITIAL VALUES
               T(1) = V(IT)/TPC
               Y(1) = V(IY)
               DAYO = V(ID)
            DO 130 I=2,N
            READ (5,FMT) V
            Y(I) = V(IY)
      C     TIME CALCULATION
            T(I) = V(IT)/TPC + (V(ID)-DAYO)*CPD
        130 CONTINUE
       4321 CONTINUE
      C
            DUR = T(N) - T(1)
            WRITE (6,1004) FMT,IY,IT,ID,TPC,CPD,DUR
       1004 FORMAT ( /5X,13HFORMAT      = 13A6,A2,
          1          /5X,13HY FIELD     = I5,
          2          /5X,13HT FIELD     = I5,
          3          /5X,13HDAY FIELD   = I5,
          4          /5X,13HTIME SCALE = G14.9,34H TIME UNITS PER COMPUTATIONA
          4L UNIT  ,
          5          /5X,13HDAY SCALE   = G14.9,28H COMPUTATIONAL UNITS PER DAY
          6        / /5X,13HDURATION    = F10.4,20H COMPUTATIONAL UNITS   )
      C
            DUR = S * DUR * FLOAT(N) / FLOAT(N-1)
            PRD=DUR
            OMEGAZ = TWOPI/DUR
            KLIMIT = N/2
            RS = R*S
            RSMAX = 2000./FLOAT(N)
            IF (RS.GT.RSMAX) RS = RSMAX
            KMAX = RS*FLOAT(N)/2.
      C
            GO TO 950
      C
        400 KTOTAL = KMAX
            FMESH = 1.0
            FBEGIN = 1.
      C
      C ..........................................................................
      C ....... BEGIN OMEGA LOOP ..................................................
      C ..........................................................................
      C
        300   OMEGA = OMEGAZ *              R     THIS SECTION CALCULATES THE
           .          (FBEGIN - FMESH)      R      COEFFICIENTS, POWER, AND
              STEP = OMEGAZ * FMESH         R      * POWER OF SPECTRAL TERMS.
      C
            DO 600 KK=1,KTOTAL
      C
            OMEGA = OMEGA + STEP
      C
      C     SUMMATIONS
      C
            SUMS2 = 0.0
            SUMC2 = 0.0
            SUMYS = 0.0
            SUMYC = 0.0
            SUMSC = 0.0
      C
            DO 500 I=1,N
      C
            THETA = OMEGA*T(I)
          XSIN  = SIN(THETA)
          XCOS  = COS(THETA)
          SUMYS = SUMYS + Y(I)*XSIN
          SUMYC = SUMYC + Y(I)*XCOS
          SUMS2 = SUMS2 + XSIN*XSIN
          SUMC2 = SUMC2 + XCOS*XCOS
          SUMSC = SUMSC + XSIN*XCOS
C
  500   CONTINUE
C
          DELTA = SUMS2*SUMC2 - SUMSC*SUMSC
C
C         FORM SOLUTION VECTOR
C
           A(KK) = (SUMC2 * SUMYS  -    C     SINE
         .  SUMYC * SUMSC) / DELTA      C      COEFFICIENT
           B(KK) = (SUMS2 * SUMYC  -    C     COSINE
         .  SUMYS * SUMSC) / DELTA      C      COEFFICIENT
C
C         FORM POWER SPECTRUM
C
           PSPECT(KK) = 2. * (A(KK) *   C     POWER
         .  SUMYS  +  B(KK) * SUMYC)    C      SPECTRAL
         .  / FLOAT(N)                  C      DENSITY
          PERCNT(KK) = PSPECT(KK)/YYBAR2
C
   600 CONTINUE
C
C
C ..............................................................................
C ....... END OMEGA LOOP .......................................................
C
C ..............................................................................
C
C         COMPUTE SMOOTH POWER SPECTRUM
C
C 601 DO 700 K=1,KMAX
C
C         PSMOTH(K) = 0.0
C         WEIGHT    = 0.0
C
C         DO 650 L=1,LMAX
C
C         LK = K+L-LCENTR
C         IF ( (LK.LE.1) .OR. (LK.GE.KMAX) ) GO TO 620
C         PSMOTH(K) = PSMOTH(K) + FILTER(L)*PSPECT(LK)
C         GO TO 650
C 620 WEIGHT = WEIGHT + FILTER(L)
C         IF (LK.GT.KMAX) LK = KMAX
C         IF ( (LK.EQ.1) .OR. (L.EQ.LMAX) ) PSMOTH(K) = PSMOTH(K)
C    1                                                + WEIGHT*PSPECT(LK)
C
C 650 CONTINUE
C
C 700 CONTINUE
C
C         PRINT POWER SPECTRUM
```

```
C
         IF (IFLAG .EQ. 1)                 C     SEARCH FOR MORE PEAKS
      .       GO TO 799                    C      UNLESS FIRST TIME THROUGH.
        WRITE (6,3000) NAME,YYBAR2
 3000 FORMAT (1H1,3X,30HINITIAL DATA SPECTRUM FOR SET ,3A6,3X,
     1 15H(TOTAL POWER = ,E10.5,1H) /4X,21H------- ---- --------
     2 //)
        IF (JS.EQ.1) GO TO 759
        GO TO 730
C
  720   IF (LFLG) GO TO 980                C     REDUCE TOL AND SEARCH AGAIN.
         IF (IFLAG .NE. 1)                 C     GO TO NEXT CASE IF NO
      .       GO TO 100                    C      PEAKS WERE FOUND.
  725   IF (JS .EQ. 1) GO TO 759
        WRITE (6,3001) NAME,YYBAR2
 3001 FORMAT (1H1,3X,30H  FINAL DATA SPECTRUM FOR SET ,3A6,3X,
     1 15H(TOTAL POWER = ,E10.5,1H) /4X,21H ----- ---- --------
     2  //)
C
  730 IFEED = 0
        WRITE (6,3002) IFEED,YBAR
        IFEED = 1
C
        DO 750 K=1,KMAX
        OMEGA = OMEGAZ * FLOAT(K)
        PERD = TWOPI/OMEGA
        IF (MOD(K,52) .EQ. 51) WRITE (6,3002) IFEED,YBAR
        WRITE (6,3010) K,PERD,OMEGA,A(K),B(K),PSPECT(K),PERCNT(K)
  750 CONTINUE
C
 3002 FORMAT (I1,53X,25H   REGRESSION      PERCENT               /4X,1HK
     1 6X,6HPERIOD,7X,5HOMEGA,8X,1HA,11X,1HB,11X,5HPOWER,
     2 7X,5HPOWER,5X4HC = ,F10.5/)
 3010 FORMAT (2XI3,2XF10.4,2XF11.6,2(2XE10.5),2XG11.6,2XF10.6)
C
C       PLOT POWER SPECTRUM
C
  759   PNT =  5.*YYBAR2/FLOAT(KLIMIT)
        IF ( (JPO.NE.0).AND. (JPO.NE.1) ) GO TO 770
        VERMAX  = 0.0
        DO 760 K=1,KMAX
        IF (PSPECT(K).GT.VERMAX) VERMAX=PSPECT(K)
  760 CONTINUE
C
        HORMAX = KMAX
        VERINT  = VERMAX/50.
        HORINT  = HORMAX/120.
        REWIND ITAPE
        WRITE (ITAPE) KMAX,        VTITLE,HTITLE,NAME,HEAD3A,IVERT,IHORZ,
     1                  ISHADE,VERMAX,VERINT,HORMAX,HORINT,PNT
        DO 765 K=1,KMAX
        XK = K
        WRITE (ITAPE) PSPECT(K),XK
  765 CONTINUE
        WRITE (ITAPE) XEND
        REWIND ITAPE
C
```

```
        CALL RPLOT (ITAPE)             C    PLOT SPECTRUM ON ONE PAGE.
        CALL GRAPH (PERCNT,PSPECT,     C    PLOT SPECTRUM ON EXPANDED
     .   KLIMIT,NAME(1),NAME(2),       C     GRAPH, ONE SPECTRAL
     .   NAME(3),HEAD3A,PRD,PS)        C     TERM PER LINE.
C
C     PLOT SMOOTH POWER SPECTRUM
C
 770  CONTINUE
C 770 IF ( (JPO.NE.0).AND. (JPO.NE.2) ) GO TO 790
C     REWIND ITAPE
C     VERMAX = 0.0
C     DO 775 K=1,KMAX
C     IF (PSMOTH(K).GT.VERMAX) VERMAX=PSMOTH(K)
C 775 CONTINUE
C     HORMAX = KMAX
C     VERINT  = VERMAX/50.
C     HORINT = HORMAX/120.
C
C **  HEAD3P MUST BE ADDED TO DATA STATEMENT
C     WRITE (ITAPE) KMAX,       VTITLE,HTITLE,NAME,HEAD3P,IVERT,IHORZ,
C    1                ISHADE,VERMAX,VERINT,HORMAX,HORINT,PNT
C     DO 780 K=1,KMAX
C     XK = K
C     WRITE (ITAPE) PSMOTH(K),XK
C 780 CONTINUE
C     WRITE (ITAPE) XEND
C     REWIND ITAPE
C
C     CALL RPLOT (ITAPE)
C
  790 CONTINUE
C
        IF (IFLAG .EQ. 1)              C    FINAL SPECTRUM HAS BEEN
     .       GO TO 100                 C     PLOTTED AND PRINTED.
C
C     TEST FOR FREQUENCY REMOVAL
C
        NOPEAK = 0                              C    FALSE PEAK COUNTER
        LFLG = .TRUE.                           C    TOL HAS ORIGINAL VALUE.
        IF (JF .NE. 0) GO TO 100
        IF (IFLAG .EQ. 0) WRITE (6,3030) NAME,YYBAR2
  799   IF (ICOUNT .GT. KMAX) GO TO 725
C **    THIS SECTION FINDS PEAKS IN DESCENDING ORDER OF SIZE.
      NTERMS=0
      PEAK = YYBAR2
 7995 CONTINUE
      KK = -1
      PR = TOL
C     PT2 = .75 * PSPECT(1)  +  .25 * PSPECT(2)
C     PT1 = PT2
C     DO 819 K=1,KMAX                   C    BEGIN SEARCH FOR PEAK.
C **  SMOOTH SPECTRUM AS SEARCH FOR PEAK IS MADE.
C     K1 = K + 1
C     K2 = K + 2
C     IF (K1 - KMAX) 813,812,811   C    COLLAPSE FILTER AT END OF SPECTRUM.
C 811 K1 = K
C 812 K2 = K1
```

```
C 813 PT3 = .25 * (PSPECT(K) +      C     APPLY 3-POINT SMOOTHING
C     . PSPECT(K2)) + .5 *          C      FILTER TO DATA ABOUT
C     . PSPECT(K1)                  C      CURRENT K VALUE.
        PT2 = PSPECT(1)             C     SET INITIAL VALUES FOR
        PT1 = PT2                   C      LOCAL MAXIMUM SEARCH.
        DO 819 K = 1,KMAX           C     BEGIN SEARCH FOR PEAK.
        K1 = K + 1
        IF (K1 - KMAX) 813,813,811  C     ADJUST VALUES AT END OF SPECTRUM.
  811 K1 = K
  813 PT3 = PSPECT(K1)
        IF (PSPECT(K) .LT. PR) GO TO 800
        IF (PT2 .LT. PT1  .OR.      C     TEST FOR
      . PT2 .LT. PT3) GO TO 800     C      LOCAL MAXIMUM
        IF (NOPEAK .LE. 0) GO TO 815
        DO 814 NOP=1,NOPEAK         C     DO NOT CHOOSE BAD PEAK.
        IF (K - KNO(NOP)) 814,800,814
  814 CONTINUE
  815 CONTINUE
C **    TEST LOCAL MAXIMUM FOR PRESENCE IN PEAK TABLE
C **     SO AS TO AVOID DUPLICATION.
        KTRY = K
        IF (PSPECT(K) - PEAK) 824,1822,800
 1822 DO 823 L=1,NTERMS             C     VERIFY THAT WE HAVE A NEW PEAK.
        IF (IABS(KREMOV(L) - KTRY)  C        IF PEAK IS EXACTLY BETWEEN 2 K
      .     .LE. 1) GO TO 800       C         VALUES, ONLY PICK 1 OF THEM.
  823 CONTINUE
  824 KK = KTRY                     C     CONDITIONALLY SELECT THIS PEAK.
        PR = PSPECT(K)
  800 PT1 = PT2                     C     PREPARE TO TEST NEXT POINT.
        PT2 = PT3
  819 CONTINUE
C
         IF (KK .LE. 0)             C     KK .LE. 0 MEANS NO
      .        GO TO 810            C      (MORE) PEAKS FOUND.
        NTERMS=NTERMS+1             C     COUNT NUMBER OF PEAKS FOUND SO FAR.
        KREMOV(NTERMS)=KK           C     SUBSCRIPT OF THIS PEAK
        PEAK = PR                   C     MAGNITUDE OF PEAK JUST FOUND
        IF (NTERMS .LT. MTERMS)     C     GO SEARCH FOR ANOTHER PEAK UNLESS
      . GO TO 7995                  C      MTERMS PEAKS HAVE ALREADY BEEN FOUND.
  810 CONTINUE
C
        IF (NTERMS .LE. 0)          C     EXIT SEARCH ROUTINE IF
      . GO TO 720                   C      NO PEAKS WERE FOUND.
C
C       FIND EXACT REMOVAL FREQUENCY
C
 3030 FORMAT (1H1,28X,22HREMOVAL TABLE FOR SET 3A6/20X,
      1 13H------- -----          // 76X,8HSPECTRUM,5X,9HREMAINING/
      2  6X,6HPERIOD,4X,9HFREQUENCY,5X,5HOMEGA,8X,1HA,11X,1HB,11X,1HC,
      310X,5HPOWER,5X,11HTOTAL POWER//1X,6(4X,8H--------),4X,9H---------
      4 ,2X,F11.6)
C
        CALL CAROL3 (AREMOV,BREMOV,PREMOV,OREMOV,N,KREMOV,OMEGAZ,NTERMS)
C **     NOPEAK COUNTS FALSE PEAKS FOUND ON THIS PASS.
        IF (NOPEAK .GT. 0) GO TO 799 C    SEARCH AGAIN, IGNORING BAD PEAKS.
         IFLAG = 1                  C      AT LEAST 1 PEAK FOUND
        NOPEAK = 0
      C
            DO 840 M=1,NTERMS
            FREMOV(M)=OREMOV(M)/TWOPI
            PDREMV(M)=1./FREMOV(M)
      C
      C     REMOVE FREQUENCY
      C
            ICOUNT = ICOUNT + 5
            DO 850 I=1,N
            Y(I) = Y(I) - (  AREMOV(M)*SIN(OREMOV(M)*T(I))
           .                + BREMOV(M)*COS(OREMOV(M)*T(I)))
        850 CONTINUE
        840 CONTINUE
      C
      C     COMPUTE AVERAGE Y
      C
        950 YBAR = 0.0
            DO 200 I=1,N
        200 YBAR = YBAR + Y(I)
             YBAR = YBAR / FLOAT(N)        C     MEAN OF DATA
            YYBAR2 = 0.0
            DO 250 I=1,N
            Y(I) = Y(I) - YBAR
            YYBAR2 = Y(I) * Y(I)   + YYBAR2
        250 CONTINUE
             YYBAR2 = YYBAR2 * 2. /        C     TOTAL
           .             FLOAT(N)          C     A. C. POWER
      C **  TOL IS USED AS A FRACTION OF INITIAL TOTAL POWER.
            IF (IFLAG .EQ. 0) TOL=TOL*YYBAR2
      C
            IF (IFLAG .NE. 1) GO TO 400
      C
      C     PRINT REMOVAL DATA
      C
            NTERMS=NTERMS-1
            IF (NTERMS .EQ. 0) GO TO 970
            DO 960 M=1,NTERMS
            WRITE (6,3031) PDREMV(M),FREMOV(M),OREMOV(M),AREMOV(M),BREMOV(M),
           .                  YBAR,PREMOV(M)
        960 CONTINUE
       3031 FORMAT (2X,F11.4,2X,F10.6,2X,F10.6,3(2X,E10.5),2X,F11.6,2X,F11.6)
        970 CONTINUE
            M=NTERMS+1
            WRITE (6,3031) PDREMV(M),FREMOV(M),OREMOV(M),AREMOV(M),BREMOV(M),
           .                  YBAR,PREMOV(M),YYBAR2
      C
            GO TO 400
      C
      C **   CUT TOL IN HALF, THEN SEARCH FOR MORE PEAKS.
        980 CONTINUE
            LFLG = .FALSE.
            TOL = TOL / 2.
            WRITE (6,4000)
       4000 FORMAT (42H0SEARCH WILL NOW BE MADE WITH SMALLER TOL.)
            GO TO 799                      C     NOW SEARCH FOR PEAKS WITH SMALLER TOL.
      C
            END
```

```
LSQFIT*MLTSAN.READ
      SUBROUTINE READ (N,IY,IT,TPC)
      DOUBLE PRECISION T,Y
      COMMON PSPECT(1000),Y(2000),T(2000)
      DIMENSION V(7)
       DO 10 I=1,N
      READ (3) V
       T(I) = V(IT)/TPC
       Y(I) = V(IY)
   10 CONTINUE
      RETURN
      END

@     PRT     MLTSAN.RPLOT
```

LSQFIT*MLTSAN.RPLOT

```
      SUBROUTINE RPLOT (ITAPE)
C
      DIMENSION ALPHA(121),DATA(2,2000),ARRAY(120)
      DATA IEND /4HEND /, IMORE /4HMORE/, ASTER /1H*/
      DATA BLANK/1H /,DASH/1H-/,DOT/1H./
C
C     READ PLOT DATA
C
    7 READ (ITAPE) NBRCDS,VTITLE,HTITLE,HEAD1,HEAD2,HEAD3,HEAD4,IVERT,
     1               IHORZ,ISHADE,VERMAX,VERINT,HORMAX,HORINT,PNT
C
      DO 11 I=1,NBRCDS
   11 READ (ITAPE)   (DATA(J,I),J=1,2)
C
      DO 20 I=1,120
   20 ALPHA(I) = BLANK
C
      WRITE (6,955) HEAD1,HEAD2,HEAD3,HEAD4,VTITLE
  955 FORMAT (1H1,45X3A6,1XA6/1XA6)
C
      PMARK = PNT + VERINT/2.
      VERMAX = VERMAX + VERINT
      DO 150 MM = 1,51
      VERMAX = VERMAX - VERINT
C
      DO 100 NN = 1,NBRCDS
      HMAX = HORMAX - (121. * HORINT)
      IF(DATA(IVERT,NN).GE.VERMAX.AND.DATA(IVERT,NN).LT.VERMAX+VERINT)
     1   GO TO 50
      GO TO 100
C
   50 DO 75 JJ = 1,120
      HMAX = HMAX + HORINT
      IF(DATA(IHORZ,NN).LT.HMAX.OR.DATA(IHORZ,NN).GE.HMAX+HORINT)
     1   GO TO 75
      ALPHA(JJ) = ASTER
      GO TO 100
C
   75 CONTINUE
  100 CONTINUE
C
   80 WRITE (6,960) VERMAX,(ALPHA(I),I=1,120)
  960 FORMAT (1XG7.3,2H I,120A1)
C
  110  IF (ISHADE.EQ.1) GO TO 150
C
      FILL = BLANK
      IF (VERMAX.GE.PMARK .AND. VERMAX.LT.(PMARK+VERINT) ) FILL = DOT
      DO 125 I=1,120
  125  ALPHA(I) = FILL
  150 CONTINUE
C
      WRITE (6,965) (DASH,I=1,121)
  965 FORMAT(8X,2H I,121A1 )
C
C
```

```
          XYZ = 120.
          DO 210 I = 1,12
          XYZ = XYZ - 10.
      210 ARRAY(I) = HORMAX - (XYZ * HORINT)
C
C
          WRITE (6,975) (ARRAY(I),I=1,12),HTITLE
      975 FORMAT (11X,12(9X,1HV)/11X,12(4X,F6.1)/60X,A6)
C
C
          READ (ITAPE) KCODE
C
          IF (KCODE .EQ. IEND) RETURN
          IF (KCODE .EQ. IMORE) GO TO 7
          END
```

```
@    PRT    MLTSAN.GRAPH
```

```
LSOFIT*MLTSAN.GRAPH
            SUBROUTINE GRAPH (PCNT,PWR,KMAX,TITLE1,TITLE2,TITLE3,TITLE4,PRD,R)
      C
            DIMENSION IARY(102),PCNT(1),PWR(1)
            DATA IBLANK/1H /,IASTER/1H*/,IDOT/1H./
      C
            WRITE (6,1000) TITLE1,TITLE2,TITLE3,TITLE4
       1000 FORMAT (1H1,1X,3A6,1X,A6,30X,13HPERCENT POWER)
      C
            PMAX = 0.
            KPLOT = FLOAT(KMAX)*R
            DO 5 I =1,KPLOT
            IF(PCNT(I).GT.PMAX) PMAX = PCNT(I)
          5 CONTINUE
      C
            ISCALE = 2
            IF (PMAX.LE. 0.2) ISCALE = 1
            IF (PMAX.GT..4) ISCALE = 5
            PMAX = 500./FLOAT(ISCALE)
            OFSET = 1.0 + .005*PMAX
            MARK = ( 5./FLOAT(KMAX)) * PMAX + OFSET
            IF (MARK.GT.101) MARK = 102
      C
            DO 10 I=1,21
         10 IARY(I) = (I-1)*ISCALE
      C
            WRITE (6,1010) (IARY(I),I=1,21)
       1010 FORMAT (/26X,21(2X,I3))
      C
        200 WRITE (6,2020)
       2020 FORMAT (5X,5HPOWER,5X,9HFREQUENCY,3X,1HK,2X,1HV,20(4X,1HV))           **FREQ**
      C
            WRITE (6,1030)
       1030 FORMAT (29X,2HI-,20(5H----) )
      C
            DO 20 I=1,102
         20 IARY(I) = IBLANK
      C
            DO 100 I=1,KPLOT
      C
            IARY(MARK) = IDOT
            JPOINT = PCNT(I)*PMAX + OFSET
            IF (JPOINT.GT.101) JPOINT = 101
            IARY(JPOINT) = IASTER
            PLENG=FLOAT(I)/PRD                                                     **FREQ**
      C
            WRITE (6,2000) PWR(I),PLENG,I,IARY
       2000 FORMAT (1XG11.7,2XG10.6,I4,2H I,102A1)
      C
        100 IARY(JPOINT) = IBLANK
      C
            RETURN
      C
            END
```

```
LSQFIT*MLTSAN.CAROL3
            SUBROUTINE CAROL3 (AREMOV,BREMOV,PREMOV,OREMOV,N,KREMOV,
         .                 OMEGAZ,NTERMS)
      C **  CALCULATES FREQUENCIES AND COEFFICIENTS OF DOMINANT TERMS IN SPECTRUM
            DOUBLE PRECISION AREMOV(NTERMS),BREMOV(NTERMS),OREMOV(NTERMS)
            DIMENSION KREMOV(NTERMS),PREMOV(NTERMS),FK(10),XK(10)
            DOUBLE PRECISION T,Y,A,B
            DOUBLE PRECISION F,DF(30),DY,DELTYO,DELTYN,DA(30)
            DOUBLE PRECISION THETA,XSIN,XCOS,SUMYS,SUMYC,OMEGAZ
            DOUBLE PRECISION AMTRX(30,30),BMTRX(30),AKOLD(30),AKNEW(30),
         .                 AI(30,30)
            DOUBLE PRECISION DSIN,DCOS,DSQRT,DATAN2,WOT
            DOUBLE PRECISION NORM
            COMMON /MORN/ NORM
            COMMON PSPECT(1000),Y(2000),T(2000),NAME(3),
         1        A(1000),B(1000),PERCNT(1000)
             COMMON /NOPEAK/ KNO(30),NOPEAK,MOPEAK
             DIMENSION FNO(30)
            DATA EPSLN /1.E-08/            @     EPSILON FOR CONVERGENCE TEST
             DATA NITER /50/               @     MAXIMUM NUMBER OF ITERATIONS
       1000 FORMAT (39H1MAXIMUM NUMBER OF ITERATIONS PERFORMED//
         . 16X3HOLD,15X3HNEW/14X6HVALUES,12X6HVALUES/
         . 3H AK,4X2D18.10/(7X2D18.10))
       1002 FORMAT (7H DELTY ,2D18.10//12H RUN ABORTED)
            MOPEAK = 0
            NT3=3*NTERMS
            DO 5 M=1,NTERMS
            M3=3*M
            KRM=KREMOV(M)
            XK(M)=KRM
            AKNEW(M3-2)=DSQRT(A(KRM)*A(KRM)+B(KRM)*B(KRM))
            AKNEW(M3-1)=DATAN2(B(KRM),A(KRM))
            AKNEW(M3)=0.D0
          5 CONTINUE
            ITER=0                         @     ITER WILL COUNT ITERATIONS.
            GO TO 400
         20 CONTINUE
            DO 10 J=1,NT3
            BMTRX(J)=0.D0
            DA(J)=0.D0
            DO 10 I=1,NT3
            AMTRX(I,J)=0.D0
         10 CONTINUE
            DO 100 I=1,N                   @     BEGIN LOOP ON TIMES.
            F=0.D0
            WOT=OMEGAZ*T(I)
            DO 50 M=1,NTERMS
            M3=3*M
            THETA=OREMOV(M)*T(I)+AKOLD(M3-1)
            XSIN=DSIN(THETA)
            XCOS=DCOS(THETA)
            F=F+AKOLD(M3-2)*XSIN
            DF(M3-2)=XSIN
            DF(M3-1)=XCOS*AKOLD(M3-2)
            DF(M3)=DF(M3-1)*WOT
         50 CONTINUE
            DY=Y(I)-F                      @     DY = RESIDUAL AT T(I)
      DO 30 J=1,NT3
      BMTRX(J)=BMTRX(J)+DF(J)*DY
      DO 30 K=J,NT3
      AMTRX(K,J)=AMTRX(K,J)+DF(K)*DF(J)
   30 CONTINUE
  100 CONTINUE                                C    END LOOP ON TIMES.
      DO 40 I=2,NT3
      IJ=I-1
      DO 40 J=1,IJ
      AMTRX(J,I)=AMTRX(I,J)
   40 CONTINUE
C
      CALL BTORTH (AMTRX,AI,NT3)              C    PUTS LEFT INVERSE OF AMTRX IN AI
C
C **  CORRECT THE AK'S
      DO 250 I=1,NT3
      DO 200 J=1,NT3
      DA(I)=DA(I)+AI(I,J)*BMTRX(J)
  200 CONTINUE
      AKNEW(I)=AKOLD(I)+DA(I)
  250 CONTINUE
      DO 325 I=3,NT3,3                        C    CHECK FREQUENCY CORRECTIONS.
      IF (ABS(AKNEW(I)) .LE. 0.5)             C    CONTINUE PROCESSING IF FREQUENCY
     . GO TO 325                              C      CORRECTIONS ARE IN BOUNDS.
      NOPEAK = NOPEAK + 1                     C    COUNT FALSE PEAKS.
      I3 = I / 3
      KNO(NOPEAK) = KREMOV(I3)                C    SUBSCRIPTS OF FALSE PEAKS
      FNO(NOPEAK) = FLOAT(KNO(NOPEAK)) * OMEGAZ / 6.2831853076
      MOPEAK = MOPEAK + 1                     C    COUNT BAD PEAKS THIS TIME THROUGH.
  325 CONTINUE
      IF (MOPEAK .LE. 0) GO TO 400
      MOPEAK = NOPEAK - MOPEAK + 1
      RETURN
  400 CONTINUE
      DO 25 M=1,NTERMS
      OREMOV(M)=(XK(M)+AKNEW(3*M))*OMEGAZ
      IF (ITER .GT. 0) GO TO 25
      FK(M)=OREMOV(M)/6.2831853076
   25 CONTINUE
      DELTYN=0.D0
      DO 300 I=1,N                            C    CALCULATE DELTY USING CURRENT
      F=0.D0                                  C      VALUES OF PARAMETERS.
      DO 350 M=1,NTERMS
      M3=3*M
      THETA=OREMOV(M)*T(I)+AKNEW(M3-1)
      F=F+AKNEW(M3-2)*DSIN(THETA)
  350 CONTINUE
      DY=Y(I)-F
      DELTYN=DELTYN+DY*DY
  300 CONTINUE
      IF (ITER .LT. 1) GO TO 375
C **  TEST CONVERGENCE.
      TESTIT = DELTYO / DELTYN  - 1.0
      IF (TESTIT .LT. 0.)                     C    CUT OFF IN CASE
     . GO TO 500                              C    OF DIVERGENCE.
      IF (ITER .EQ. NITER) GO TO 475
  375 CONTINUE
```

```
            DO 450 I=1,NT3
            AKOLD(I)=AKNEW(I)
        450 CONTINUE
            IF (ITER .LT. 3) GO TO 465
            IF (TESTIT .LT. EPSLN)          C     CUT OFF IF CONVERGENCE
           . GO TO 500                      C      IS SUFFICIENTLY SLOW.
        465 CONTINUE
            DELTYO=DELTYN
            ITER=ITER+1                     C     INCREMENT ITERATION COUNTER.
            GO TO 20                        C     PERFORM ANOTHER ITERATION.
        475 WRITE (6,1000) (AKOLD(I),AKNEW(I),I=1,NT3)
            WRITE (6,1002) DELTYO,DELTYN
            STOP
        500 CONTINUE
            DO 550 M=1,NTERMS
            M3=3*M
            AREMOV(M)=AKOLD(M3-2)*DCOS(AKOLD(M3-1))
            BREMOV(M)=AKOLD(M3-2)*DSIN(AKOLD(M3-1))
            OREMOV(M)=(XK(M)+AKOLD(M3))*OMEGA2
      C
      C **  COMPUTE POWER.
            SUMYS=0.D0
            SUMYC=0.D0
            DO 600 I=1,N
            THETA=OREMOV(M)*T(I)
            SUMYS=SUMYS+Y(I)*DSIN(THETA)
            SUMYC=SUMYC+Y(I)*DCOS(THETA)
        600 CONTINUE
            PREMOV(M)=2.*(AREMOV(M)*SUMYS+BREMOV(M)*SUMYC)/FLOAT(N)
        550 CONTINUE
            RETURN
            END

@       PRT       MLTSAN.BIORTH
```

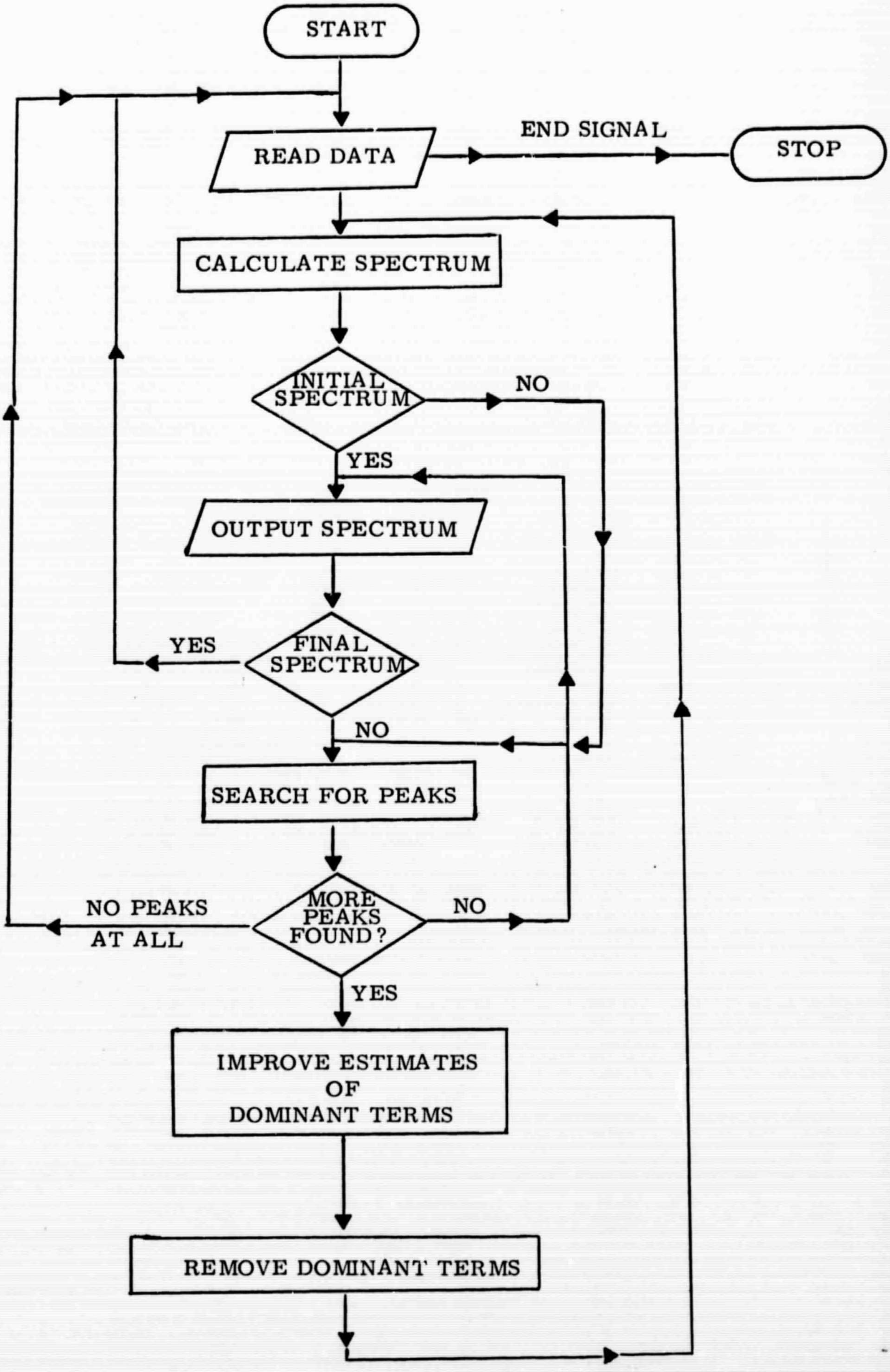
```
LSQFIT*MLTSAN.BIORTH
C
          SUBROUTINE BIORTH(A,B,N)       @
C
C         WHERE:  (1) A = AN N-BY-N DIMENSIONAL MATRIX
C                 (2) B = THE LEFT INVERSE OF THE MATRIX A
C                 (3) N = ROW- AND COLUMN-DIMENSION OF THE MATRICES
C
          DOUBLE PRECISION A(30,30),B(30,30),DOT,NORM,DSQRT,EPS,CJK
          DOUBLE PRECISION NSA
C
          DATA EPS/1D-9/                 @
C
          COMMON/MORN/NORM               @
C
C
          NIT=0                          @ INITIALIZE ITERATION COUNTER
          DO 100 I=1,N                   @ AS A 1-ST ''GUESS'', SET THE IN-
          DO 100 J=1,N                   @ VERSE MATRIX, *B*, EQUAL TO THE
          B(I,J)=A(I,J)                  @ INPUT MATRIX, *A* (THE B-MATRIX IS
  100     CONTINUE                       @ BEING COMPUTED AS ITS TRANSPOSE)
C
  200     CONTINUE                       @
          NIT=NIT+1                      @ INCREMENT ITERATION COUNTER
          DO 300 K=1,N                   @ THERE ARE *N* CYCLES
          CALL SCLP((1./DOT(B(1,K),      @ COMPUTE THE K-TH ROW (STORED AS A
     .         A(1,K))),B(1,K),          @ COLUMN) OF THE INVERSE MATRIX
     .         B(1,K))                   @
          DO 300 J=1,N                   @ DO FOR THE REMAINING ROWS
          IF(XOR(J,K).EQ.0)GO TO 300     @ BUT ONLY IF *J* NOT EQUALS *K*
          CJK=DOT(B(1,J),A(1,K))         @ COMPUTE THE SCALER, C(J,K)
          DO 301 I=1,N                   @ DO FOR EACH OF THE *N* COMPONENTS
          B(I,J)=B(I,J)-CJK*B(I,K)       @ COMPUTE THE (I,J)-TH ELEMENT OF THE
C                                        @ INVERSE FOR THE K-TH CYCLE
  301     CONTINUE                       @
  300     CONTINUE                       @
          NORM=0.0                       @ INITIALIZE THE NORM OF THE MATRIX
          DO 400 I=1,N                   @ THE ''NORM'' OF THE PRODUCT OF THE
          DO 400 J=1,N                   @ LEFT-INVERSE TIMES THE MATRIX IS
          NORM=NORM+(DOT(B(1,I),         @ DEFINED AS THE SQUARE-ROOT OF THE
     .         A(1,J))-AMIN0(I/J,        @ SUM OF THE TERMS, WHERE '1' IS SUB-
     .         J/I))**2                  @ TRACTED FROM THE DIAGONAL TERMS
  400     CONTINUE                       @
          NORM=DSQRT(NORM)               @ FINALLY, TAKE THE SQUARE-ROOT
          IF(NORM.GT.((2**NIT)*EPS))     @ GO DO AN ADDITIONAL ITERATION IF
     .         GO TO 200                 @ THE CONVERGENCE CRITERIA WAS NOT
C                                          MET
          DO 500 I=1,N                   @ TRANSPOSE THE B-MATRIX, SINCE IT
          DO 500 J=I,N                   @ WAS BEING COMPUTED AS COLUMN VEC-
          NSA=B(I,J)
          B(I,J)=B(J,I)                  @ OF ROW VECTORS
          B(J,I)=NSA
  500     CONTINUE                       @
C
          RETURN
C
          FUNCTION DOT(X,Y)              @ DOT-PRODUCT FUNCTION
```

```
                DOUBLE PRECISION X,Y           C
                DIMENSION X(1),Y(1)            C
                DOT=0.0                        C  INITIALIZE *DOT*
                DO 600 K=1,N                   C  N-DIMENSIONAL VECTORS
                DOT=DOT+X(K)*Y(K)              C
          600   CONTINUE                       C
                RETURN                         C
       C
                SUBROUTINE SCLR(Z,X,Y)         C  SCALER-PRODUCT SUBROUTINE
                DOUBLE PRECISION X,Y,Z         C
                DIMENSION X(1),Y(1)            C
                DO 700 K=1,N                   C  N-DIMENSIONAL VECTOR
                Y(K)=Z*X(K)                    C
          700   CONTINUE                       C
                RETURN                         C
                END                            C
```

```
@ FIN


RUNID:  MXL        ACCOUNT: D02               PROJECT: LSQFIT

TIME: 00:00:01.381     IN: 10      OUT: 0         PAGES:    20

INITIATION  TIME:    18:42:07-DEC 15,1969

TERMINATION TIME:    18:42:20-DEC 15,1969

CORE-SECONDS: 9

IO COUNT:      41

CHARGE:        0.331
```

APPENDIX II

Overall Logic Flowchart


START
READ DATA
END SIGNAL
STOP
CALCULATE SPECTRUM
INITIAL SPECTRUM
NO
YES
OUTPUT SPECTRUM
FINAL SPECTRUM
YES
NO
SEARCH FOR PEAKS
MORE PEAKS FOUND?
NO PEAKS AT ALL
NO
YES
IMPROVE ESTIMATES OF DOMINANT TERMS
REMOVE DOMINANT TERMS

END

DATE

FILMED

JUL 16 1970