PARTMENT OF ENERGY, MINES AND RESOURCES
MARINE SCIENCES BRANCH
TÈRE DE L'ÉNERGIE DE MINES ET DES RESSOURCES
DIRECTION DES SCIENCES DE LA MER

CANADA

ATLANTIC OCEANOGRAPHIC LABORATORY
BEDFORD INSTITUTE

LABORATOIRE OCEANOGRAPHIQUE DE L'ATLANTIQUE
INSTITUT de BEDFORD

Dartmouth, Nova Scotia
Canada

A SET OF PROGRAMS FOR THE POWER SPECTRUM ANALYSIS OF GRAVITY CROSS COUPLING DATA

by

R. T. HAWORTH

B.I. COMPUTER NOTE 68-6-C

JULY, 1968

PROGRAMMED BY

THE CANADIAN COMMITTEE OF OCEANOGRAPHY

65653

ATLANTIC OCEANOGRAPHIC LABORATORY
BEDFORD INSTITUTE

DARTMOUTH, N. S. - CANADA

Although this program was tested by the Author prior to submission, no warranty, expressed or implied, is made by the author or the Bedford Institute as to the accuracy and functioning of the program. No responsibility is assumed by the author or the Bedford Institue in connection therewith.

A SET OF PROGRAMS FOR THE POWER SPECTRUM ANALYSIS OF GRAVITY CROSS COUPLING DATA

by

R. T. HAWORTH

B.I. COMPUTER NOTE 68-6-C

JULY, 1968

CONTENTS

PAGE

Plotting Programs:

INTRODUCTION

The programs contained in this note are those used to analyze cross coupling data collected by the author during cruise BI HUDSON 19-66. The programs are mainly subroutines from which are fabricated the programs used to process time series data written in a standard format. The format is that devised by Dr. R.G. Stevens and most of the programs contained herein were either written by him or are adaptations of such programs.

Because of the adaptability of these programs to analysis of any time series data, the data format used is most important. Hence data collection will be dealt with more fully than is usual in a computer note.

Only a brief introduction is given to the mathematics of power spectrum analysis since it seems far better that the reader should consult an authoritative text on the subject rather than be taught by a fellow pupil.

An introduction is given to the sequence of actions followed in the processing of cross coupling data and examples are given of the computer output produced at various stages of processing since this tends to act as a guide during future usage of these programs.

This note is one of a series. The first, on the Construction and Use of a Cross Coupling Computer, is available as Internal Note 67-5-I. The results of the analysis using the programs described in the present note are available in Data Report 1967-6-D, entitled Gravity Cross Coupling Power Spectra. A further note on the interpretation of the analysis will be available in due course.

MULTIVARIABLE DATA PREPARATION

INPUT DATA FORMAT

The format used is that which is the standard format of paper tapes output from the B.I. F.M. Digitizing System described by R.H. Loucks in B.I. Internal Note 66-2. The input for this particular set of programs are data of gravity cross coupling errors and the gathering of these data is described in B.I. Internal Note 67-5-I.

It may be summarized that in the calculation and investigation of these errors, four quantities are of interest:

surge acceleration
gravimeter beam motion
instantaneous cross coupling error
filtered cross coupling error

Two analog computers were used and comparison between the two was one objective of the experiment. Each of the eight electrical signals representing the above quantities was used to modulate a voltage controlled oscillator (VCO), the frequency output of which has a deviation from its center frequency dependent upon the magnitude and polarity of the input signal. The VCO outputs were recorded on magnetic tape and on return to the laboratory, the signal levels were recovered and punched on paper tape using the F.M. Digitizing System previously mentioned. Because the system was, at that time, (Dec. 1966) capable of handling a maximum of 5 channels, the two sets of computer data were recovered separately. The paper tape output is in the format of 4 sets of three characters followed by a carriage return symbol. Each character group is a three digit number, with the sign included in code, representing the recorded signal level of one of the four variables. If the number is positive, the + sign is understood. If negative, the first character is alphabetic with J = -0, A = -1, B = -2 and so on. For example, we might have the data group:

$\text{A63216B37036}^{c.}_{r.}$

This will be interpreted as:

The first variable on tape has a value of -163

The second variable on tape has a value +216

The third variable on tape has a value -237 and

The fourth variable on tape has a value + 36.

In the Multi-Variable Data Compensation program the variables are identified, their values are standardized and the data are output on computer magnetic tape in the format required by subsequent processing programs.

MVD (Multi-Variable Data) COMPENSATION PROGRAM

I. PURPOSE:

1) To read in paper tape containing multi-variable data.
2) To standardize the values of the variables using standardization data supplied separately on paper tape.
3) To write out the data in a standard format on magnetic tape together with a heading containing the means and variances of the multi-variable and standardization data, and other parameters of the run.

II. OPERATION:

The paper tape output from the F.M. Digitizing System is in the format of 4 sets of three characters followed by a carriage return. Each three character group is a three digit number with the sign included in code. If the number is positive, the + sign is understood. If the number is negative, the first character is alphabetic with J = -0, A = -1, B = -2, and so on. The data is assumed to be good from the first character and the program expects the 13th character it reads to be a carriage return. The last character on tape must be a carriage return. If necessary, the data tapes must be modified to ensure that this is so.

The program also requires a paper tape containing standardization data, consisting of the minimum, zero and maximum values of the variable. The format for this standardization data is identical to that for the data. At least 100 samples of each of these quantities must be available, or an error statement is printed. In practice it should be conventional to begin collection of data with punching of these standardization data. Before any measurements are made, a recording should be made of Low, Centre and High and this can then be used prior to the data for processing.

In addition the following cards are necessary:

1) Tape Specification Card:

col.	123	4 to 29	30	31 to 59	60
	NEW	Blank	VTAPE	Blank	SCRATCH

In column 30 is the output tape handler number.

In column 60 is the scratch tape handler number.

e.g.

```
                                   30                              60
NEW OUTPUT TAPE ON HANDLER            SCRATCH TAPE ON HANDLER
```

"NEW" is only specified if the output tape is a new one in which case a new tape label is written.

1a) Tape Label Cards:

The tape label information is given on several cards punched in columns 1 to 80 terminated by an end of file card punched:

```
Col. 1        10
     8        END OF TAPE LABEL
```

In column 1, figures 7 and 8 are both punched. If the word NEW appears on the tape specification card the tape label cards must be included. If the word NEW does not appear, i.e. the output tape already has information on it, the tape label cards <u>must not</u> be included in the data card deck.

2) Run Identification Card:

Cols 1 to 8: Blank

Cols 9 to 40: Run name (32 characters maximum). In this analysis, examples of run name are:

```
CAMBRIDGE T3 0740-0800 258 66
DOMINION  A1 1510-1535 257 66

Computer Run    Time    Day Year
```

Cols 41 to 50: Sampling Interval (δ t) secs. This value is written in format F9.0 e.g. 0.500.

3) Variable Identification Cards:

One for each of the variables to be examined, the cards being in the same order as the variables on tape.

Cols 1 to 5: "CHAN"

6 to 7: Channel Number (2 digits) e.g. "1", "2", . . . "11", "12". Channel number refers to the IRIG channel number used i.e. 4,5,6,7,9,10,11,12, in this case.

8: " ". i.e. blank

9 to 32: Variable Name (24 characters)

$$\left.\begin{array}{r}\text{DOMOBS}\\ \text{CAMBRIDGE}\end{array}\right\}\left\{\begin{array}{l}\text{HORIZ ACCEL}\\ \text{BEAM MOTION}\\ \text{ICC}\\ \text{FCC}\end{array}\right.$$

33 to 40: "   ". i.e. 8 blanks.

41 to 50: "1.0   ". i.e. 1.0 and 7 blanks.

51 to 60: The number of practical units to which the range of (High reading - Low reading) in the standardization data is equivalent. For Cambridge Beam Motion +46-(-489) is equivalent to 77 gals, specified as 77.0 in cols 51 to 60.

61 to 80: Blank.

4) End of File:

Cols 1 to 5:	"$EOF"
6 to 7:	"13"
8:	Blank
9 to 24:	"END OF DATA FILE".

Cards 2 to 4 inclusive should be repeated if desired. If you are running a series of runs, the supply of cards 2 to 4 can be kept up until the end of the magnetic tape on which the data is written is reached. When no more data is to be read in, the last card should be blank.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

Any format error or short record will cause an error message to be printed, and the program halts at an unidentified pause. SS 1 and SS 2 should be set at this stage according to what is to be done to the data.

SS1 ON Accept more data by re-reading the frame at which the pause occurred and then continuing.

SS2 ON Abandon run and start new run by reading in a new set of Run and Variable identification cards. This might be necessary if only a short record is obtained before an error is reached which would tend to invalidate the data analysis.

SS1 and SS2 OFF Accept the short record read in before the PAUSE was reached.

SS1 and SS2 ON Same as with SS1 ON.

IV. USAGE:

Compile program.

Set up tapes according to the scheme of the tape specification card e.g. dial scratch tape on handler 2
dial output tape on handler 4
Load paper tape containing standardization data.
GO
PAUSE 2222_8

Set up paper tape experimental data
GO
PAUSE (unidentified) when all the data has been read. Set up cards for next run. Load standardization data.
GO to process next run as before.

Terminate operations by causing a card with a blank run name to be read. This causes the output tape to be rewound, the scratch tape having been rewound earlier in the program. "END OF TAPE" is always written at the end of each data group on the output tape, but is then written over during the transfer of data to the output tape during the following run. Hence, if the program is destroyed before the transfer of data, the end of the output tape data can be located by loading the program again and not specifying NEW on the Tape Specification Card.

The Output Tape Format

The first information block on the tape is the Tape Label followed by an end of file mark. The label is only written once, thereafter there are heading files and data files which alternate, one set for each variable analyzed. The heading lists all the run parameters in two 'records', the first of which contains information which is likely to be used in processing and the other which contains information such as the standardization data which is purely incidental. This heading file is referred to in most program descriptions as the RUNFILE or Run Heading File. After the runfile comes the data file which contains the MVD data written in blocks of 13 (one value for each variable and the frame number), so that on tape the information is in the sequence:

Tape Label
E.O.F.
Runfile for first run
E.O.F.
Datafile for first run
E.O.F.
Runfile for second run
E.O.F.
Datafile for second run
etc.

The basic subroutines OPENFILE and RUNFILE of the data analysis programs expect that the order of the files is as above and count the E.O.F. marks to find out whether the tape is positioned to read either run parameters or data. If for some reason the continuity is lost, ingenuity in operation of the program can sometimes restore order.

V. STORAGE REQUIREMENTS:

4677_8 2495_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

EOFCK COMPARE SELECT DIGDAT EJECT
READB WRITEB

```
00000  C     MULTI-VARIABLE DATA COMPENSATION
00000  C     3 APRIL
00000  C
00000  C     MODIFIED WITH PATCHES FOR EARLY BUZZARDS BAY AND PANAMA CITY DATA.
00000  C     PATCHES ARE AFTER =1101= AND AFTER =2239=
00000  C                 29 DECEMBER 1966
00000  C
00000  C      ----------------------------------------------------------------
00000  C      COMMON STORAGE
00000  C
00004        COMMON RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004  C     -----------------------------------------------------------------
00004  C     OUTPUT FILES
00004  C
00004  C                  VTAPE FILE 1 TAPE LABEL
00004  C
00004  C     WRITE OUTPUT TAPE VTAPE,7,(SUM(I),I=1,10)
00004  C7    FORMAT(10A8)
00004  C
00004  C                  VTAPE FILE 2 RECORD1   (     FP WORDS)
00004  C     WRITE TAPE VTAPE,RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004        DIMENSION RNM(8),CHNO(13),VNM(3,13),MEANV(13),FACT(13)
00004        INTEGER RNM,CHNO                   N.B. above format
00004        REAL MEANV
00004  C
00004  C                  VTAPE FILE 2 RECORD2   (     FP WORDS)
00004  C
00004  C     WRITE TAPE VTAPE,A,VARV,MEANLO,MEANCTR,MEANHI,VARLO,VARCTR,VARHI
00004        DIMENSION A(4,13),VARV(12),MEANLO(12),MEANCTR(12),MEANHI(12),
00004       1          VARLO(12),VARCTR(12),VARHI(12)
00004        REAL MEANLO,MEANCTR,MEANHI
00004  C
00004  C                  VTAPE FILE 2 RECORD3   (     FP WORDS)
00004  C
00004  C     WRITE TAPE VTAPE,RL,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004        DIMENSION RL(8)
00004        INTEGER RL
00004  C
00004  C                  VTAPE FILE 3 RECORD1   (     FP WORDS)
00004  C
00004  C     CALL WRITEB(VTAPE,V,V(14,10))
00004        DIMENSION V(13,10)
00004  C     -----------------------------------------------------------------
00004  C                  WORKING STORAGE
00004  C
00004        DIMENSION BUF(12),SUM(12),SUMSQ(12)
00004        INTEGER VTAPE,SCRATCH,EOF,CHNN,BLANK
00004  C
00004  C     PROGRAM CONSTANTS
00004  C
00004        NEW=152950
00006        RL(1)=5657904
00011        RL(2)=-6722508
00015        RL(3)=4617584
00020        RL(4)=-3994575
00024        RL(5)=-3994575
00030        RL(6)=-3994575
00034        RL(7)=-3994575
00040        RL(8)=-3994575
00044        BLANK=-3994575
00047        PAUSE 4095
```

```
00051          DO 15000 I=1,6
00053          IF(SENSE SWITCH 1)15000,15002
00057   15000 CONTINUE
00064          CALL LEADER
00065          CALL DUMP
00066   15002 CONTINUE
00066   C
00066   C       INITIALIZE
00066   C
00066          PRINT 1
00071   1      FORMAT(1H1)
00074   2      FORMAT(//////)
00077   C      ----------------------------------------------------------------
00077   C      READ TAPE CONTROL CARD AND WRITE TAPE LABEL IF -NEW- IS SPECIFIED,
00077   C      OTHERWISE POSITION OUTPUT TAPE TO END OF PREVIOUS DATA RECORDS
00077   C
00077          READ 20,KONTROL,VTAPE,SCRATCH
00110   20     FORMAT(R3,26X,I1,29X,I1)
00116          REWIND VTAPE
00120          REWIND SCRATCH
00122          IF(KONTROL-NEW)3,6,3
00127   C
00127   C      READ AND PRINT TAPE LABEL
00127   C
00127   3      READ INPUT TAPE VTAPE,7,(SUM(I),I=1,10)
00151          PRINT 30,(SUM(I),I=1,10)
00172   30     FORMAT(1X,10A8)
00176          CALL EOFCK(VTAPE,LITE)
00201          GO TO(31,3),LITE
00206   C
00206   C      SEARCH FOR =END OF TAPE= RECORD, AND LIST RUN NAMES  ENCOUNTERED
00206   C
00206   31     READ TAPE VTAPE,RNM
00214          PRINT 32,(RNM(I),I=1,8)
00233   32     FORMAT(21X,8A4)
00237          CALL COMPARE (RNM,RL,8,LITE)
00244          GO TO (5,4),LITE
00251   4      CALL SELECT (VTAPE,11)
00254          CALL SELECT (VTAPE,11)
00257          GO TO 31
00260   C
00260   C      POSITION TAPE TO WRITE OVER =END OF TAPE= RECORD
00260   C
00260   5      CALL SELECT(VTAPE,12)
00263          CALL SELECT(VTAPE,11)
00266          PRINT 1
00271          GO TO 1000
00272   C
00272   C      READ NEW LABEL FROM CARDS, WRITE ON TAPE AND PRINT
00272   C
00272   6      READ 7,(SUM(I),I=1,10)
00313   7      FORMAT(10A8)
00316          CALL EOFCK(1536,LITE)
00321          PRINT 30,(SUM(I),I=1,10)
00342          GO TO (9,8),LITE
00347   8      WRITE OUTPUT TAPE VTAPE,7,(SUM(I),I=1,10)
00371          GO TO 6
00372   9      ENDFILE VTAPE
00374          GO TO 1000
00375   C      ----------------------------------------------------------------
00375   C      READ RUN IDENTIFICATION FROM CARDS
00375   C
```

```
00375  1000  READ 100,(RNM(I),I=1,8),DELT
00416  100   FORMAT(8X,8A4,F9.0)
00423        IF (RNM(1)-BLANK)1100,9000,1100
00431  C
00431  C     READ VARIABLE IDENTIFICATION FROM CARDS
00431  C
00431  1100  J=1
00433        PRINT 101,(RNM(I),I=1,8)
00452  101   FORMAT(21X,40HMULTI-CHANNEL DATA COMPENSATION FOR RUN  8A4,/, 75H
00452       1        VARIABLE NAME        CHAN  SENS        A1          A2          A3
00452       2        A4/)
00515  1101  READ 110,CHNN,(VNM(N,CHNN),N=1,3),ISEN,(A(I,CHNN),I=1,4)
00570  110   FORMAT(5X,I2,1X,3A8,6X,I1,4(1X,F9.0))
00601        CALL EOFCK(1536,LITE)
00604        GO TO(2000,1102),LITE
00611  1102  IF(CHNN-13)1103,1103,1104
00616  1103  CHNO(J)=CHNN
00621        JBIG=J
00623        J=J+1
00626        PRINT 111,(VNM(N,CHNN),N=1,3),CHNN,ISEN,A(1,CHNN),A(2,CHNN),
00626       1A(3,CHNN),A(4,CHNN)
00733  111   FORMAT(1X,3A8,3X,I2,5X,I1,4X,F9.4,1X,F9.4,1X,F9.5,1X,F9.6)
00751        GO TO 1101
00752  1104  PRINT 114
00755  114   FORMAT(5X,45HCHANNEL NUMBER EXCEEDS 13. PROGRAM TERMINATED)
00774        GO TO 9000
00775  C     -----------------------------------------------------------------
00775  C     READ LOW STANDARDIZATION FROM PAPER TAPE CALC MEANS AND VARIANCES
00775  C
00775  2000  DO 2009 J=1,12
00777           SUM(J)=0
01005           SUMSQ(J)=0
01013  2009  CONTINUE
01020        PRINT 2
01023  2010  DO 2019 I=1,100
01025  2011     CALL DIGDAT (BUF,JBIG,EOF,KOUNT)
01032        GO TO(2012,2090,2080),EOF
01040  2012  DO 2019 JJ=1,JBIG
01042              J=CHNO(JJ)
01045              SUM(J)=BUF(JJ)+SUM(J)
01056              SUMSQ(J)=BUF(JJ)*BUF(JJ)+SUMSQ(J)
01070  2019  CONTINUE
01102  2020  CALL DIGDAT(BUF,JBIG,EOF,KLUNK)
01107        GO TO (2020,2030,2020),EOF
01115  2030  DO 2039 JJ=1,JBIG
01117           J=CHNO(JJ)
01122           MEANLO(J)=SUM(J)/KOUNT
01133           VARLO(J)=SUMSQ(J)/KOUNT-MEANLO(J)*MEANLO(J)
01152  2039  CONTINUE
01157  2099  GO TO 2100
01160  C
01160  C     FORMAT ERROR ON LOW STANDARDIZATION
01160  C
01160  2080  PRINT 208,KOUNT
01165  208   FORMAT(1X,27H   FORMAT ERROR ON FRAME = I3,16H LOW STANDARDIZE)
01205        CALL EJECT
01206        KOUNT=KOUNT-1
01211        PAUSE
01213        IF(SENSE SWITCH 1) 2011,2081
01217  2081  IF(SENSE SWITCH 2) 1000,2020
01223  C
01223  C     SHORT RECORD ON LOW STANDARDIZATION
```

```
01223  C
01223  2090  PRINT 209,KOUNT
01230  209   FORMAT(4X,30HLOW STANDARDIZE CONTAINS ONLY I3,8H SAMPLES)
01246        CALL EJECT
01247        PAUSE
01251        IF(SENSE SWITCH 1) 2011,2091
01255  2091  IF(SENSE SWITCH 2)1000,2030
01261  C     ------------------------------------------------------------------
01261  C     READ CENTER STANDARDIZATION FROM PAPER TAPE CALC MEANS AND VAR
01261  C
01261  2100  DO 2109 J=1,12
01263           SUM(J)=0
01271           SUMSQ(J)=0
01277  2109  CONTINUE
01304  2110  DO 2119 I=1,100
01306  2111     CALL DIGDAT (BUF,JBIG,EOF,KOUNT)
01313        GO TO(2112,2190,2180),EOF
01321  2112     DO 2119 JJ=1,JBIG
01323              J=CHNO(JJ)
01326              SUM(J)=BUF(JJ)+SUM(J)
01337              SUMSQ(J)=BUF(JJ)*BUF(JJ)+SUMSQ(J)
01351  2119  CONTINUE
01363  2120  CALL DIGDAT(BUF,JBIG,EOF,KLUNK)
01370        GO TO (2120,2130,2120),EOF
01376  2130  DO 2139 JJ=1,JBIG
01400           J=CHNO(JJ)
01403           MEANCTR(J)=SUM(J)/KOUNT
01414           VARCTR(J)=SUMSQ(J)/KOUNT-MEANCTR(J)*MEANCTR(J)
01433  2139  CONTINUE
01440  2199  GO TO 2200
01441  C
01441  C     FORMAT ERROR ON CENTER STANDARDIZATION
01441  C
01441  2180  PRINT 218,KOUNT
01446  218   FORMAT(1X,27H   FORMAT ERROR ON FRAME = I3,19H CENTER STANDARDIZE)
01466        KOUNT=KOUNT-1
01471        CALL EJECT
01472        PAUSE
01474        IF(SENSE SWITCH 1) 2111,2181
01500  2181  IF(SENSE SWITCH 2) 1000,2120
01504  C
01504  C     SHORT RECORD ON CENTER STANDARDIZATION
01504  C
01504  2190  PRINT 219,KOUNT
01511  219   FORMAT(4X,33HCENTER STANDARDIZE CONTAINS ONLY I3,8H SAMPLES)
01530        CALL EJECT
01531        PAUSE
01533        IF(SENSE SWITCH 1) 2111,2191
01537  2191  IF(SENSE SWITCH 2) 1000,2130
01543  C     ------------------------------------------------------------------
01543  C     READ HIGH STANDARDIZATION FROM PAPER TAPE CALC MEANS AND VARIANCES
01543  C
01543  2200  DO 2209 J=1,12
01545           SUM(J)=0
01553           SUMSQ(J)=0
01561  2209  CONTINUE
01566  2210  DO 2219 I=1,100
01570  2211     CALL DIGDAT (BUF,JBIG,EOF,KOUNT)
01575        GO TO(2212,2290,2280),EOF
01603  2212     DO 2219 JJ=1,JBIG
01605              J=CHNO(JJ)
01610              SUM(J)=BUF(JJ)+SUM(J)
```

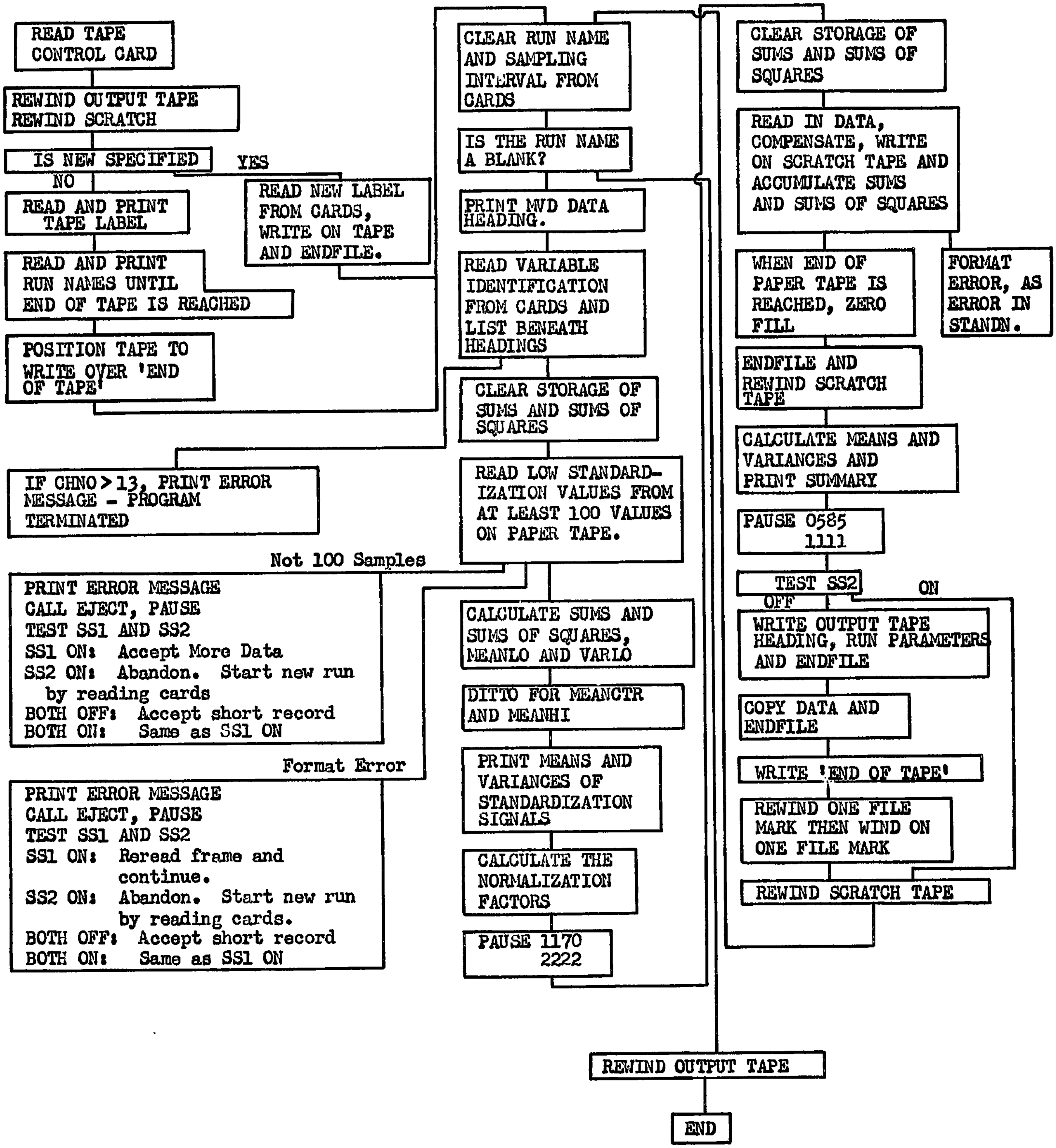
```
01621               SUMSQ(J)=BUF(JJ)*BUF(JJ)+SUMSQ(J)
01633  2219   CONTINUE
01645  2220   CALL DIGDAT(BUF,JBIG,EOF,KLUNK)
01652         GO TO (2220,2230,2220),EOF
01660  2230   DO 2239 JJ=1,JBIG
01662            J=CHNO(JJ)
01665            MEANHI(J)=SUM(J)/KOUNT
01676         VARHI(J)=SUMSQ(J)/KOUNT-MEANHI(J)*MEANHI(J)
01715  2239   CONTINUE
01722  2299   GO TO 2300
01723  C
01723  C      FORMAT ERROR ON HIGH STANDARDIZATION
01723  C
01723  2280   PRINT 228,KOUNT
01730  228    FORMAT(1X,27H   FORMAT ERROR ON FRAME = I3,17H HIGH STANDARDIZE)
01750         KOUNT=KOUNT-1
01753         CALL EJECT
01754         PAUSE
01756         IF(SENSE SWITCH 1) 2211,2281
01762  2281   IF(SENSE SWITCH 2) 1000,2220
01766  C
01766  C      SHORT RECORD ON HIGH STANDARDIZATION
01766  C
01766  2290   PRINT 229,KOUNT
01773  229    FORMAT(4X,31HHIGH STANDARDIZE CONTAINS ONLY I3,8H SAMPLES)
02011         CALL EJECT
02012         PAUSE
02014         IF(SENSE SWITCH 1) 2211,2291
02020  2291   IF(SENSE SWITCH 2) 1000,2230
02024  C      ------------------------------------------------------------
02024  C      PRINT MEANS AND VARIANCES OF STANDARDIZATION SIGNALS
02024  C
02024  2300   PRINT 230,(RNM(I),I=1,8)
02043  230    FORMAT(21X,33HSTANDARDIZATION SUMMARY FOR RUN   ,8A4,/)
02060         PRINT 231,DELT
02065  231    FORMAT(35X,18HSAMPLING INTERVAL ,F9.3,8H SECONDS/)
02101         PRINT 232
02104  232    FORMAT(7X,102HVARIABLE NAME        CHAN   MEAN-LOW   MEAN-CTR    ME
02104        1AN-HI   VARIANCE-LOW   VARIANCE-CTR    VARIANCE-HI/)
02141         DO 2309 JJ=1,JBIG
02143         J=CHNO(JJ)
02146         PRINT 233,(VNM(I,J), I=1,3),CHNO(JJ),MEANLO(J),MEANCTR(J),
02146        1         MEANHI(J),VARLO(J),VARCTR(J),VARHI(J)
02252  233    FORMAT(1X,3A8,3X,I2,1X,3(2X,F9.4),3(6X,F9.5))
02265  2309   CONTINUE
02272         PRINT 234
02275  234    FORMAT(1X,//////)
02301  C      ------------------------------------------------------------
02301  C      CALCULATE NORMALIZATION FACTORS
02301  C
02301  2400   DO 2409 JJ=1,JBIG
02303            J=CHNO(JJ)
02306         FACT(J)=1.0/(MEANHI(J)-MEANLO(J))
02317  2409   CONTINUE
02324  C
02324  C      CLEAR SUMMING AREAS FOR DATA PASS
02324  C
02324         PAUSE 1170
02326  2900   DO 2909 J=1,12
02330         SUM(J)=0
02336         SUMSQ(J)=0
02344  2909   CONTINUE
```

```
02351  C       ------------------------------------------------------------
02351  C        READ DATA FROM PAPER TAPE, APPLY COMPENSATION AND WRITE
02351  C       ON SCRATCH TAPE.
02351  C       ACCUMULATE SUMS AND SUMS OF SQUARES.
02351  C
02351  3000    DO 3019 K=1,10
02353  3001       CALL DIGDAT (BUF,JBIG,EOF,KOUNT)
02360             GO TO (3002,3900,3004),EOF
02366  3002       DO 3009 JJ=1,JBIG
02370               J=CHNO(JJ)
02373               VN=(BUF(JJ)-MEANCTR(J)*A(1,J))*FACT(J)
02422               V(J,K)=((VN*A(4,J)+A(3,J))*VN+A(2,J))*VN
02465               SUM(J)=V(J,K)+SUM(J)
02502               SUMSQ(J)=V(J,K)*V(J,K)+SUMSQ(J)
02527  3009       CONTINUE
02534             V(13,K)=KOUNT
02546  3019    CONTINUE
02553          CALL WRITEB(SCRATCH,V,V(14,10))
02570          GO TO 3000
02571  C
02571  C       FORMAT ERROR ON DATA TAPE
02571  C
02571  3004    PRINT 300,KOUNT
02576  300     FORMAT(4X,24HFORMAT ERROR ON FRAME = I5,13H OF DATA TAPE//////)
02616          KOUNT=KOUNT-1
02621          CALL EJECT
02622          PAUSE
02624          IF(SENSE SWITCH 1) 3001,3003
02630  3003    IF(SENSE SWITCH 2)1000,3005
02634  3005    CALL DIGDAT(BUF,JBIG,EOF,KLUNK)
02641          GO TO (3005,3900,3005),EOF
02647  C
02647  C       END OF PAPER TAPE
02647  C       FILL LAST DATA BLOCK WITH ZEROES
02647  C       LEAVING KOUNT AT MAXIMUM VALUE
02647  C
02647  3900    DO 3909 N=K,10
02651             DO 3919 JJ=1,12
02653                V(JJ,K)=0.0
02664  3919       CONTINUE
02671             KM1=K-1
02674             V(13,K)=V(13,KM1) +1
02716  3909    CONTINUE
02723          CALL WRITEB(SCRATCH,V,V(14,10))
02740          ENDFILE SCRATCH
02742          REWIND SCRATCH
02744          GO TO 4000
02745  C       ------------------------------------------------------------
02745  C       CALCULATE AND PRINT SUMMARY OF DATA
02745  C
02745  4000    PRINT 400,(RNM(I),I=1,8),KOUNT,JBIG
02770  400     FORMAT(21X,25HSUMMARY OF DATA FOR RUN ,8A4,//,31X,
02770       1        13HTOTAL SAMPLESI6,/,25X,19HNUMBER OF VARIABLESI4,/)
03027          PRINT 401
03032  401     FORMAT(7X,50HVARIABLE NAME       CHAN  DATA MEAN  DATA VARIANCE/)
03052          DO 4009 JJ=1,JBIG
03054             J=CHNO(JJ)
03057             MEANV(J)=SUM(J)/KOUNT
03070             VARV(J)=SUMSQ(J)/KOUNT-MEANV(J)*MEANV(J)
03107             PRINT 402,(VNM(I,J), I=1,3),CHNO(JJ),MEANV(J),VARV(J)
03157  402     FORMAT(1X,3A8,3X,I2,3X,F9.4,5X,F10.5)
03170  4009    CONTINUE
```

```
03175           PRINT 1
03200   C       -----------------------------------------------------------------
03200   C       PAUSE SHOWING =1111=
03200   C       =GO= CAUSES RUN TO BE COPIED ONTO OUTPUT TAPE.
03200   C        =SENSE SWITCH 2 +GO=  CAUSES NEW RUN TO START WITHOUT COPYING
03200   C       PRESENT RUN ONTO OUTPUT TAPE
03200           PAUSE 0585
03202           IF(SENSE SWITCH 2)4029,4010
03206   C       -----------------------------------------------------------------
03206   C       WRITE OUTPUT TAPE HEADING WITH RUN PARAMETERS, CALIBRATION AND
03206   C       DATA SUMMARY
03206   C
03206   4010    WRITE TAPE VTAPE,RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
03236           WRITE TAPE VTAPE,A,VARV,MEANLO,MEANCTR,MEANHI,VARLO,VARCTR,VARHI
03271           ENDFILE VTAPE
03273   C       -----------------------------------------------------------------
03273   C       COPY DATA FROM SCRATCH TAPE ONTO SMOOTH OUTPUT TAPE
03273   C
03273   4020    CALL READB(SCRATCH,V,V(14,10))
03310           CALL EOFCK(SCRATCH,LITE)
03313           GO TO(4022,4021),LITE
03320   4021    CALL WRITEB(VTAPE,V,V(14,10))
03335           GO TO 4020
03336   4022    ENDFILE VTAPE
03340   C
03340   C       WRITE =END OF TAPE= RECORD AND BACK UP SO THAT THIS RECORD WILL BE
03340   C       WRITTEN OVER DURING THE NEXT RUN.  IF THIS IS THE LAST RUN, THE
03340   C       =END OF TAPE= RECORD WILL REMAIN IN PLACE TO MARK THE END OF DATA
03340   C
03340           WRITE TAPE VTAPE,RL,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
03370           WRITE TAPE VTAPE,A,VARV,MEANLO,MEANCTR,MEANHI,VARLO,VARCTR,VARHI
03423           CALL SELECT(VTAPE,12)
03426           CALL SELECT(VTAPE,11)
03431   4029    REWIND SCRATCH
03433           GO TO 1000
03434   C       -----------------------------------------------------------------
03434   C       TERMINATE PROGRAM
03434   C       -----------------------------------------------------------------
03434   9000    REWIND VTAPE
03436           PRINT 1
03441           PRINT 1
03444           END


 SUBPROGRAMS

   LEADER      DUMP        EOFCK       COMPARE     SELECT      DIGDAT
   EJECT       WRITEB      READB

 PROGRAM ALLOCATION

  03511    NEW        03512    BLANK      03513    I          03514    KONTROL
  03515    VTAPE      03516    SCRATCH    03517    LITE       03520    J
  03521    CHNN       03522    N          03523    ISEN       03524    EOF
  03525    JJ         03526    KLUNK      03527    K          03530    VN
  03532    KMI

  03533    A          03703    VARV       03733    MEANLO     03763    MEANCTR
  04013    MEANHI     04043    VARLO      04073    VARCTR     04123    VARHI
  04153    RL         04163    V          04567    BUF        04617    SUM
  04647    SUMSQ
```

```
COMMON ALLOCATION

 00000     RNM         00010     CHNO        00025     VNM         00143     MEANV
 00175     FACT        00227     JBIG        00230     KOUNT       00231     DELT

PROGRAM END
```

MULTI VARIABLE DATA COMPENSATION PROGRAM


READ TAPE CONTROL CARD
REWIND OUTPUT TAPE REWIND SCRATCH
IS NEW SPECIFIED
YES
NO
READ NEW LABEL FROM CARDS, WRITE ON TAPE AND ENDFILE.
READ AND PRINT TAPE LABEL
READ AND PRINT RUN NAMES UNTIL END OF TAPE IS REACHED
POSITION TAPE TO WRITE OVER 'END OF TAPE'
IF CHNO > 13, PRINT ERROR MESSAGE - PROGRAM TERMINATED
Not 100 Samples
PRINT ERROR MESSAGE
CALL EJECT, PAUSE
TEST SS1 AND SS2
SS1 ON: Accept More Data
SS2 ON: Abandon. Start new run by reading cards
BOTH OFF: Accept short record
BOTH ON: Same as SS1 ON
Format Error
PRINT ERROR MESSAGE
CALL EJECT, PAUSE
TEST SS1 AND SS2
SS1 ON: Reread frame and continue.
SS2 ON: Abandon. Start new run by reading cards.
BOTH OFF: Accept short record
BOTH ON: Same as SS1 ON
CLEAR RUN NAME AND SAMPLING INTERVAL FROM CARDS
IS THE RUN NAME A BLANK?
PRINT MVD DATA HEADING.
READ VARIABLE IDENTIFICATION FROM CARDS AND LIST BENEATH HEADINGS
CLEAR STORAGE OF SUMS AND SUMS OF SQUARES
READ LOW STANDARDIZATION VALUES FROM AT LEAST 100 VALUES ON PAPER TAPE.
CALCULATE SUMS AND SUMS OF SQUARES, MEANLO AND VARLO
DITTO FOR MEANCTR AND MEANHI
PRINT MEANS AND VARIANCES OF STANDARDIZATION SIGNALS
CALCULATE THE NORMALIZATION FACTORS
PAUSE 1170
2222
CLEAR STORAGE OF SUMS AND SUMS OF SQUARES
READ IN DATA, COMPENSATE, WRITE ON SCRATCH TAPE AND ACCUMULATE SUMS AND SUMS OF SQUARES
WHEN END OF PAPER TAPE IS REACHED, ZERO FILL
FORMAT ERROR, AS ERROR IN STANDN.
ENDFILE AND REWIND SCRATCH TAPE
CALCULATE MEANS AND VARIANCES AND PRINT SUMMARY
PAUSE 0585
1111
TEST SS2
ON
OFF
WRITE OUTPUT TAPE HEADING, RUN PARAMETERS AND ENDFILE
COPY DATA AND ENDFILE
WRITE 'END OF TAPE'
REWIND ONE FILE MARK THEN WIND ON ONE FILE MARK
REWIND SCRATCH TAPE
REWIND OUTPUT TAPE
END

EDIT MVD TAPES

I. PURPOSE:

When the initial MVD tapes were produced, an error in the program resulted in the count of the frame numbers (i.e. the count of the complete sets of variable values) beginning at 100. This meant that all the calculated means and variances were in error. This program edits the invalid MVD tape and produces a corrected tape. The program also allows for ordering data runs on tape and eliminating useless data.

II. OPERATION:

Using subroutines OPENFILE and RUNFILE the program searches for a given run and when found prints a message to this effect and pauses at 7771_8. SS1 ON will cause the run found on the input tape to be ignored (e.g. it may be such a short run that the data are worthless for spectrum analysis) and the next run on tape will be examined and its run name printed.

With SS1 OFF, the program expects to read the run name and old value of KOUNT from a card, and then proceeds to write out the runfile on the output tape with the value of KOUNT reduced by 100. After writing an end of file mark, the program reads in a data block, inserts the correct value of data frame number as the 13th value in each of the 10 frames read and rewrites the corrected data block on the output tape. This carries on until the number of corrected data sets reaches the corrected value of KOUNT inserted in the new runfile. As a check, with SS 5 OFF, the data frame number and KOUNT will be printed at the end of processing each data block of 10 frames.

When KOUNT frames have been read, the tape is endfiled and END OF TAPE is written as the next run name. The tape is then positioned to write over this heading and the program pauses at 7772_8.

If we have come to the end of the input tape, or the next run we want is on another tape, SS2 is put ON which causes the input tape to be rewound before a pause 7773_8 is reached.

With SS2 OFF, pressing GO causes the next runfile on the input tape to be read in and the program recycles.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

PAUSE 7771_8 / 4089_{10} Do we want to rewrite on the output tape, as next in sequence, the run which has just been printed?

YES: Load reader with card containing the run name and the old value of KOUNT in format (8X,8A4,10X,I10), set SS1 OFF and GO.

NO: Set SS1 ON and GO. Next run on tape will be examined.

PAUSE 7772_8 4090_{10} Have we come to the end of the tape? Is the next run we wish to edit on another tape or earlier on the tape we are using?

NO: Set SS2 OFF and GO. Next run in sequence will be examined.

YES: Set SS2 ON and GO. Input tape will be rewound, and program pauses 7773_8.

PAUSE 7773_8 4091_{10} Prepare next input tape and load reader with the run name of the next run we wish to examine. When this is found, we revert to pause 7771_8.

PAUSE 7777_8 4095_{10} Initial pause on loading program to allow setting up of data tapes and cards.

IV. USAGE:

Compile program.
PAUSE 7777_8.
Set up data tapes with Tape Specification Card
Tape Label Cards (if necessary)
as required by OPENFILE and a
Run Identification Card
for the first run to be examined as required by subroutine RUNFILE.
GO
PAUSE 7771_8 when desired run is found.
Then continue using program according to the instructions given in Section III.

V. STORAGE REQUIREMENTS:

1656_8 942_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

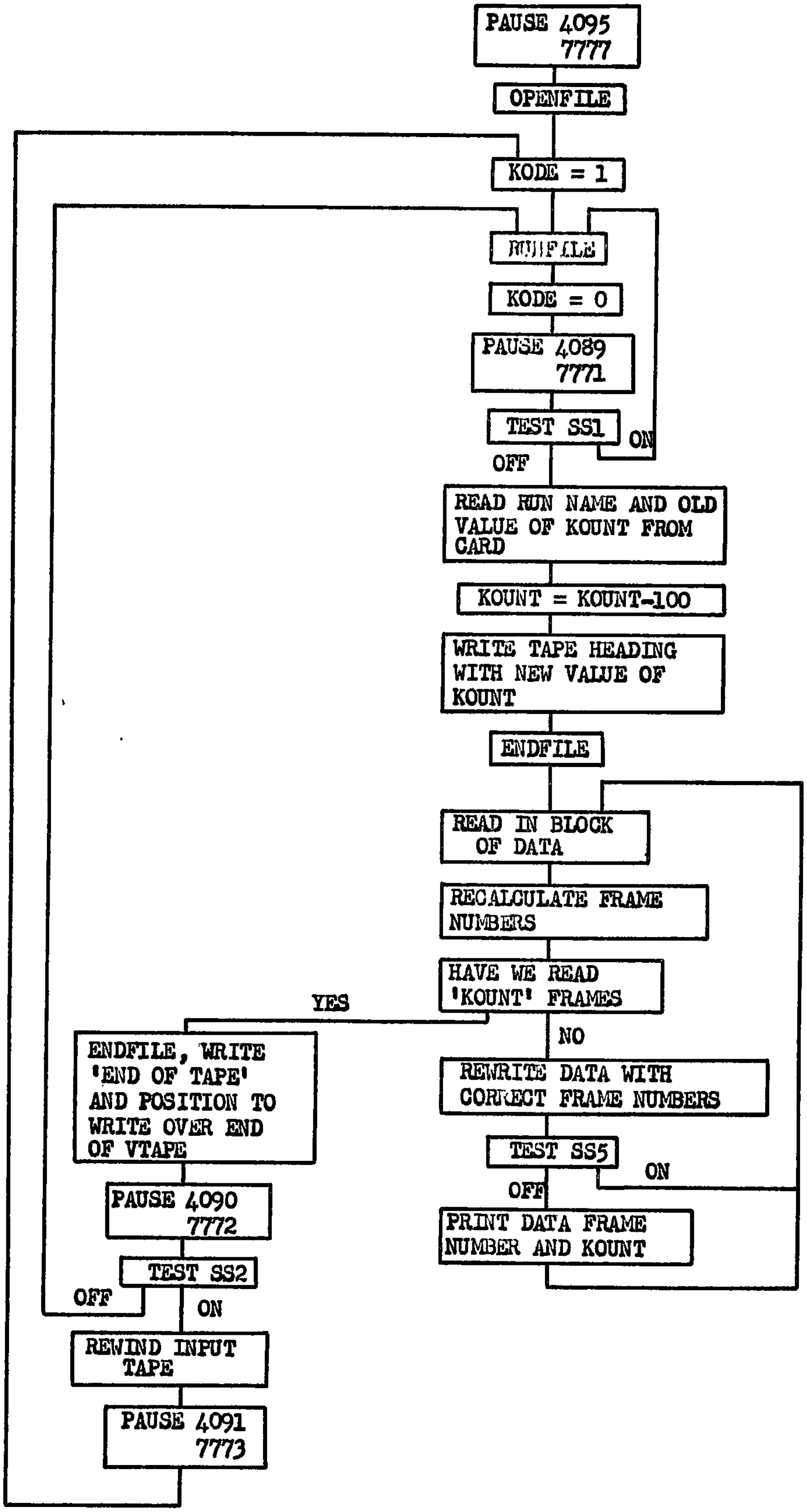
OPENFILE RUNFILE EJECT
SELECT READB WRITEB

```
00000  C     PROGRAM TO EDIT MVD TAPES
00004        COMMON RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004        DIMENSION RNM(8),CHNO(13),VNM(3,13),MEANV(13),FACT(13)
00004        INTEGER RNM,CHNO
00004        REAL MEANV
00004  C
00004  C     ------------------------------------------------------------------
00004  C     OUTPUT FILES
00004  C
00004  C                       VTAPE FILE 1 TAPE LABEL
00004  C
00004  C     WRITE OUTPUT TAPE VTAPE,7,(SUM(I),I=1,10)
00004  C7    FORMAT(10A8)
00004  C
00004  C                       VTAPE FILE 2 RECORD1  (     FP WORDS)
00004  C     WRITE TAPE VTAPE,RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004        INTEGER RNM,CHNO
00004        REAL MEANV
00004  C
00004  C                       VTAPE FILE 2 RECORD2  (     FP WORDS)
00004  C
00004  C     WRITE TAPE VTAPE,A,VARV,MEANLO,MEANCTR,MEANHI,VARLO,VARCTR,VARHI
00004        DIMENSION A(4,13),VARV(12),MEANLO(12),MEANCTR(12),MEANHI(12),
00004       1         VARLO(12),VARCTR(12),VARHI(12)
00004        REAL MEANLO,MEANCTR,MEANHI
00004  C
00004  C                       VTAPE FILE 2 RECORD3  (     FP WORDS)
00004  C
00004  C     WRITE TAPE VTAPE,RL,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004        DIMENSION RL(8)
00004        INTEGER RL
00004        DIMENSION NL(8)
00004        EQUIVALENCE (NL(1),RL(1))
00004        RL(1)=5657904
00007        RL(2)=-6722508
00013        RL(3)=4617584
00016        RL(4)=-3994575
00022        RL(5)=-3994575
00026        RL(6)=-3994575
00032        RL(7)=-3994575
00036        RL(8)=-3994575
00042  C
00042  C                       VTAPE FILE 3 RECORD1  (     FP WORDS)
00042  C
00042  C     CALL WRITEH(VTAPE,V,V(14,10))
00042        DIMENSION V(13,10)
00042        INTEGER VTAPE
00042  C     ------------------------------------------------------------------
00042  C                       WORKING STORAGE
00042  C
00042        DIMENSION BUF(12),SUM(12),SUMSQ(12)
00042        INTEGER VTAPE,SCRATCH,EOF,CHNN,BLANK
00042        INTEGER OUT
00042  1000  PAUSE 4095
00044        CALL OPENFILE(OUT,IN,KEY)
00050  1001  KODE=1
00052  1100  CALL RUNFILE(IN,KEY,KODE,LAMP)
00057        KODE=0
00061        CALL EJECT
00062        PAUSE 4089
00064        IF(SENSE SWITCH 1)1100,1101
```

```
00070   1101   READ 110,(RNM(I),I=1,8),KOUNT
00111   110    FORMAT(8X,8A4,10X,I10)
00116          KOUNT=KOUNT-100
00121   C
00121   C      COPY RUNFILE
00121   C
00121   2000   WRITE TAPE OUT,RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00151          READ TAPE IN,A,VARV,MEANLO,MEANCTR,MEANHI,VARLO,VARCTR,VARHI
00204          WRITE TAPE OUT,A,VARV,MEANLO,MEANCTR,MEANHI,VARLO,VARCTR,VARHI
00237          ENDFILE OUT
00241          CALL SELECT(IN,11)
00244          CNT=0.
00246   2001   CALL READB(IN,V,V(14,10))
00263          DO 2009 M=1,10
00265             CNT=CNT+1.
00270             V(13,M)=CNT
00301             IF(KOUNT-CNT) 3000,3000,2009      (AMEND TO 3000,2009,3000)
00307   2009   CONTINUE
00314          CALL WRITEB(OUT,V,V(14,10))
00331          IF(SENSE SWITCH 5) 2011,2010
00335   2010   PRINT 200, KOUNT,V(13,10)
00355   200    FORMAT(11X,I10,10X,F9.0)
00363   2011   GO TO 2001
00364   C
00364   C      END OF COPY, WRITE END OF TAPE AND POSITION
00364   C      TO WRITE OVER END OF TAPE MARK
00364   C
00364   3000   ENDFILE OUT
00366          KEY=2
00370          WRITE TAPE OUT,RL,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00420          CALL SELECT(OUT,12)
00423          CALL SELECT(OUT,11)
00426          PAUSE 4090
00430          IF(SENSE SWITCH 2)9000,1100
00434   9000   REWIND IN
00436          CALL EJECT
00437          PAUSE 4091
00441          GO TO 1001
00442          END

 SUBPROGRAMS

    OPENFILE   RUNFILE    EJECT      SELECT     READB      WRITEB

 PROGRAM ALLOCATION

  00512    OUT       00513    IN        00514    KEY       00515    KODE
  00516    LAMP      00517    I         00520    CNT       00522    M

  00523    A         00673    VARV      00723    MEANLO    00753    MEANCTR
  01003    MEANHI    01033    VARLO     01063    VARCTR    01113    VARHI
  00502    RL        00502    NL        01143    V         01547    BUF
  01577    SUM       01627    SUMSQ

 COMMON ALLOCATION

  00000    RNM       00010    CHNO      00025    VNM       00143    MEANV
  00175    FACT      00227    JBIG      00230    KOUNT     00231    DELT
```

PROGRAM END.

EDIT MVD TAPES


PAUSE 4095 7777
OPENFILE
KODE = 1
RUNFILE
KODE = 0
PAUSE 4089 7771
TEST SS1
ON
OFF
READ RUN NAME AND OLD VALUE OF KOUNT FROM CARD
KOUNT = KOUNT-100
WRITE TAPE HEADING WITH NEW VALUE OF KOUNT
ENDFILE
READ IN BLOCK OF DATA
RECALCULATE FRAME NUMBERS
HAVE WE READ 'KOUNT' FRAMES
YES
NO
ENDFILE, WRITE 'END OF TAPE' AND POSITION TO WRITE OVER END OF VTAPE
REWRITE DATA WITH CORRECT FRAME NUMBERS
TEST SS5
ON
OFF
PRINT DATA FRAME NUMBER AND KOUNT
PAUSE 4090 7772
TEST SS2
OFF
ON
REWIND INPUT TAPE
PAUSE 4091 7773

PRINTOUT OF DATA MAGNETIC TAPE AFTER PROCESSING BY EDIT PROGRAM

This printout is included to give an indication of the format in which the data are written on the computer magnetic tape. The program used to obtain this printout is a BI library program for use with the CDC 3100 which infers the code used for writing the information and prints it out, record by record. The print used as an example is Cambridge Tape No. 2 (edited).

As described in the introductions to the Edit and Modify Format programs, some confusion arose in the writing of the computer tapes due to a revision of the format used in the MVD program and associated subroutines. When the original data was transferred from paper tape to computer magnetic tape, the constants MEANV and FACT were dimensioned MEANV(12) and FACT(12) in the MVD program. The revised dimensions are MEANV(13) and FACT(13).

This original MVD program was found later to have another fault in that it began the count of frame numbers of data at 100. Hence each value of KOUNT written in the Runfiles on tape is an over-estimate. The purpose of the Edit program was therefore to amend the old value of KOUNT and insert the correct value in the correct position for referencing by subroutines dimensioned MEANV(13) and FACT(13) since such subroutines previously could not recover the value of KOUNT.

The printout of information on the tape indicates the format produced by the original program and the revision introduced by the Edit program.

Records 1 to 4 comprise the title of the tape as read from cards and written on the tape by the MVD program. These records are then followed by a file mark.

Record 5 marks the beginning of the Runfile for the first run. The contents of subsequent records are tabulated below. Each record contains 34 words, the first 32 of which contain the information and the last two of which are used for a parity check.

RNM(8)	Record	5	word	1	to record	5	word	8,	8	words
CHNO(13)		5		9		5		21,	13	
VNM(3,13)		5		22		8		3,	78	
MEANV(12)		8		4		8		27,	24	
FACT(12)		8		28		9		19,	24	
JBIG		9		20					1	
KOUNT		9		21					1	
DELT		9		22		9		23,	2	

The remainder of record 9 is blank (denoted by -3994575) except for the figure 2390 as word 25 in that record. On the original MVD tape this would have been blank, but the edit program has written the new value of KOUNT in the space referenced by subroutines dimensioned MEANV(13) and FACT(13) i.e. 4 words in advance of where it was written by the original program.

The variables given above complete the first half of the Runfile which contains information likely to be used in processing the data. The second half of the Runfile is written as follows:

A(4,13)	Record 10	word 1	to record	13	word 8,	104 words
VARV(12)	13	9		13	32,	24
MEANLO(12)	14	1		14	24,	24
MEANCTR(12)	14	25		15	16,	24
MEANHI(12)	15	17		16	8,	24
VARLO(12)	16	9		16	32,	24
VARCTR(12)	17	1		17	24,	24
VARHI(12)	17	25		18	16,	24

The remainder of record 18 is blank. This completes the Runfile and a file mark is written.

The values of run constants and sampled values of the variables are referenced on the data tape by means of the I.R.I.G. channel number of the V.C.O. on which each variable's data were recorded. These channel numbers are entered in CHNO for referencing by the subroutines and the data pertinent to each number are to be found at that same numbered location in each set of constants. For example, the factors MEANV and FACT of the variable recorded on I.R.I.G. channel number 6 are MEANV(6) and FACT(6). The name of the variable recorded on channel 6 is written in the space VNM(3,6). Similarly, in the data file, the data for this variable is written in the locations (6,N) where N is the sample number.

The Datafile begins on record 19. This data is written in blocks of (13,10) each data block being one record.

Sample 1	of CHNO 1	occupies	record 19	words 1	and 2	no value
1	6		19	11	12	
1	10		19	19	20	
1	11		19	21	22	
1	12		19	23	24	
Cumulative KOUNT (sample number)			19	25	26	
2	6		19	37	38	
2	10		19	45	46	
2	11		19	47	48	
2	12		19	49	50	

Although the entire printout is not included, the end of the data file is reached after 256 records i.e. 238 from the beginning of the data file. With 10 samples per record this yields 2380 samples which agrees with the corrected value of KOUNT written in the Runfile.

It is apparent that although KOUNT has been amended and put in its correct position for reference by the revised subroutines, the latter will not reference MEANV and FACT in the correct way. It is necessary therefore to completely modify the format by reading in the constants from the edited tape and rewriting them in the correct format on a new tape. This is the purpose of the MODIFY FORMAT program.

```
 RECORD NO.     1 (BCD) CONTAINING    136 CHARACTERS
THIS TAPE CONTAINS CAMBRIDGE DATA TAPE NUMBER TWO


 RECORD NO.     2 (BCD) CONTAINING    136 CHARACTERS
EDITED


 RECORD NO.     3 (BCD) CONTAINING    136 CHARACTERS
BELONGING TO RICHARD HAWORTH


 RECORD NO.     4 (BCD) CONTAINING    136 CHARACTERS
TWO


 END OF FILE
 THERE ARE      4 RECORDS CONTAINING     136 WORDS BEFORE FILEMARK NO.   1
```

Print of Cambridge Tape 2 (Edited version)

In this case the value of KOUNT has been read from one place and put in another so that JBIG and DELT were not referenced by MEANV(13), FACT(13). To get the modified version, this tape was read as MEANV(12) FACT(12) and written as MEANV(13) FACT(13) to make it consistent with the subprograms.

```
                                                                   RNM
RECORD NO.     5 (BINARY)  CONTAINING    34 WORDS
 5052690  -5925608   5704961  -4190143    131137    527362   1346566   1772592        10        12        11         6        12
       0         0         0         0         0         0         0         0   CHNO  0         0         0         0        24
       0         0  VNM(3,1)0        0         0         0         0         0   4419596         0

RECORD NO.     6 (BINARY)  CONTAINING    34 WORDS
VNM(3,2)0        0         0         0         0         0  VNM(3,3)0        0         0         0         0         0        12
VNM(3,4)0        0         0         0         0         0  VNM(3,5)0  5052690  -5925608   5703059   5180464  -3994575        24
-3994575         0         0         0         0         0  VNM(3,7)0        0   2021455         0
VNM(3,6)

RECORD NO.     7 (BINARY)  CONTAINING    34 WORDS
       0         0         0         0  VNM(3,8)0        0         0         0         0         0  VNM(3,9)0  5052690        12
-5925608   5702805   4607012  -8605209  -6878159   5052690  -5925608   5703251   5180464  -3994575  -3994575   5052690        24
                                        VNM(3,10)                                                  VNM(3,11)
-5925608   5703206  -5923215   4535509  -7402447  -4105899  -4037199   5315793   7965223         0
                                        VNM(3,12)

RECORD NO.     8 (BINARY)  CONTAINING    34 WORDS
                    VNM(3,13)                                                                     MEANV(10)   MEANV(11)
-4102556   5704752  -3994575         0         0         0         0         0         0         0         0         0        12
       0  -4198364  -6849747         0         0         0         0         0         0   4185013   5152606  -4205466        24
                    MEANV(6)
-7422942  -4204869  -2852719         0         0         0         0         0  -6011652         0
                    MEANV(12)

RECORD NO.     9 (BINARY)  CONTAINING    34 WORDS          FACT(6)
       0         0         0         0         0   4160101   6046611         0         0         0         0         0        12
          FACT(10)            FACT(11)            FACT(12)                             DELT
       0   4157263  -1505245   4155466  -6316927   4156781   -770101  JBIG -> 4 KOUNT->2490 4196352        0  -3994575        24
    2390  -3994575  -3994575  -3994575  -3994575  -3994575  -3994575  -3994575   3105800         5

RECORD NO.    10 (BINARY)  CONTAINING    34 WORDS
       0         0         0         0         0         0         0         0         0         0         0         0        12
       0         0         0         0         0         0         0         0         0         0         0         0        24
       0         0         0         0         0         0         0         0         0         0

RECORD NO.    11 (BINARY)  CONTAINING    34 WORDS
       0         0         0         0         0         0         0         0   4200448         0   4205363   3355443        12
       0         0         0         0         0         0         0         0         0         0         0         0        24
       0         0         0         0         0         0         0         0  -5015961         0

RECORD NO.    12 (BINARY)  CONTAINING    34 WORDS
       0         0         0         0         0         0         0         0   4200448         0   4225440         0        12
       0         0         0         0   4200448         0   4234032         0         0         0         0         0        24
 4200448         0   4226976         0         0         0         0         0  -8266638         0

RECORD NO.    13 (BINARY)  CONTAINING    34 WORDS
                                                                     VARV(6)
       0         0         0         0         0         0         0         0         0         0         0         0        12
       0         0         0         0         0         0   4180975  -5855157         0         0         0         0        24
                    VARV(10)            VARV(11)                    VARV(12)
       0         0   4225123  -2112901   4214531   4164087   4222168  -7308853   5729973         0

RECORD NO.    14 (BINARY)  CONTAINING    34 WORDS
       0         0         0         0         0         0         0         0         0        -0  -4230720        -0        12
       0         0         0         0         0         0  -4235080        -0  -4234856        -0  -4225664        -0        24
       0         0         0         0         0         0         0         0   -149105         0

RECORD NO.    15 (BINARY)  CONTAINING    34 WORDS
```

```
        0        0 -4218752       -0        0        0        0        0        0        0 -4230608       -0         12
  4220992        0  4230608        0        0        0        0        0        0        0        0        0         24
        0        0  4229728        0        0        0        0        0  4231968        0

  RECORD NO.   16 (BINARY)  CONTAINING   34 WORDS
        0        0  4221824        0  4237372        0  4237368        0        0        0        0        0         12
        0        0        0        0        0        0       -0       -0        0        0        0        0         24
        0        0       -0       -0       -0       -0       -0       -0 -4080651        0

  RECORD NO.   17 (BINARY)  CONTAINING   34 WORDS
        0        0        0        0        0        0        0        0        0        0       -0       -0         12
        0        0        0        0        0        0       -0       -0       -0       -0       -0       -0         24
        0        0        0        0        0        0        0        0        0        0

  RECORD NO.   18 (BINARY)  CONTAINING   34 WORDS
        0        0       -0       -0        0        0        0        0        0        0       -0       -0         12
       -0       -0       -0       -0 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575         24
 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575  3195660        9

  END OF FILE
  THERE ARE   18 RECORDS CONTAINING    612 WORDS BEFORE FILEMARK NO.  2
```

```
RECORD NO.  19 (BINARY)  CONTAINING  260 WORDS
   3932160         0         0         0         0         0         0         0         0         0  -4198088  -2043924          12
         0         0         0         0         0         0   4214556  -4860688   4209374  -2784882   4210473   -550071          24
   4200448         0         0         0         0         0         0         0         0         0         0         0          36
  -4198379  -3491705         0         0         0         0         0         0   4214630   6710885   4200626   4482981          48
   4209949   6325835   4204544         0         0         0         0         0         0         0         0         0          60
         0         0  -4198233   5620793         0         0         0         0         0         0   4213156  -6616809          72
  -4209374   2784882  -4205433   4950652   4205568         0         0         0         0         0         0         0          84
         0         0         0         0  -4198059   -221425         0         0         0         0         0         0          96
   4197251  -6899037  -4209003   3532045  -4212839   3025400   4208840         0         0         0         0         0         108
         0         0         0         0         0         0  -4198175  -7511422         0         0         0         0         120
         0         0  -4210571  -4108066  -4189080  -5917312  -4213625   4950654   4209152         0         0         0         132
         0         0         0         0         0         0         0         0  -4198233   5620793         0         0         144
         0         0         0         0  -4214188  -4380299   4212914   4482980  -4217039   4400582   4209664         0         156
         0         0         0         0         0         0         0         0         0         0  -4198291   1975794         168
         0         0         0         0         0         0  -4214593   7463509   4208818   4482980  -4216882  -8050798         180
   4210176         0         0         0         0         0         0         0         0         0         0         0         192
  -4198379  -3491705         0         0         0         0         0         0  -4213746  -2069712  -4206577  -8218797         204
  -4209319   2200289   4212736         0         0         0         0         0         0         0         0         0         216
         0         0  -4198320    153294         0         0         0         0         0         0  -4209465   1693400         228
  -4209003   3532045  -4205643   7701015   4212992         0         0         0         0         0         0         0         240
         0         0         0         0  -4198320    153294         0         0         0         0         0         0         252
   4204559   5237000  -4205093  -5230144   4206482  -1925253   4213248         0

END OF FILE
THERE ARE   256 RECORDS CONTAINING  62492 WORDS BEFORE FILEMARK NO.  3
```

PROGRAM TO MODIFY FORMAT OF EDITED TAPES

I. PURPOSE:

In the program used to produce the initial MVD tapes, the dimensions MEANV(12) and FACT(12) were used. Since the data analysis programs have been written assuming the dimensions are MEANV(13) and FACT(13) as in the output from the present version of the MVD Compensation Program, this program carries out the necessary format modification. The program also recalculates the means and variances for correction of the run headings.

II. OPERATION:

The OPENFILE and RUNFILE subroutines used in this program are written with dimensions MEANV(12) and FACT(12) whereas the standard version has both dimensioned 13. The program reads the old runfile and advances one file mark. Then the data is read in and the means and variances of these data calculated. These figures are contained in the runfile, and the revised version is written on the output tape with dummy variables NONE and NILL used to fill up the 13th "dimension" of MEANV and FACT so that the latter can be referenced on the output by MEANV(13) and FACT(13).

The input tape is rewound to the beginning of the data file when calculation of the means and variances is completed. After rewriting the headings on the output tape, the data is directly transferred from input to output tape. The output tape is endfiled, END OF TAPE is written and the tape backspaced over the end of file mark.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

PAUSE 7777_8 4095_{10}	Initial pause on loading program to allow setting up of data tapes and cards.
SS 2 ON	if the whole tape is to be modified
OFF	a pause 7772_8 is reached after each run. This does not allow for any editing of the tape, the pause is just there to give some idea of what is being modified.
PAUSE 7772_8 4090_{10}	is the pause reached after each run is modified if SS 2 is left OFF.

IV. USAGE:

Load program.
PAUSE 7777_8.
Load input edited tape on handler 1.
Load output tape on handler 2.
Load card reader with

```
col: 123                             30                 60
     NEW   INPUT TAPE ON HANDLER 1     OUTPUT TAPE ON HANDLER 2
```

Tape heading cards as required by OPENFILE.
Card specifying first run to be modified as required by RUNFILE.

Set SS 5 ON to suppress printing of frame numbers.
Set SS 2 ON unless the complete tape is not to be modified.
GO.

V. STORAGE REQUIREMENTS:

1775_8 1021_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

OPENFILE RUNFILE SELECT EJECT READB
EOFCK BACKUP WRITEB

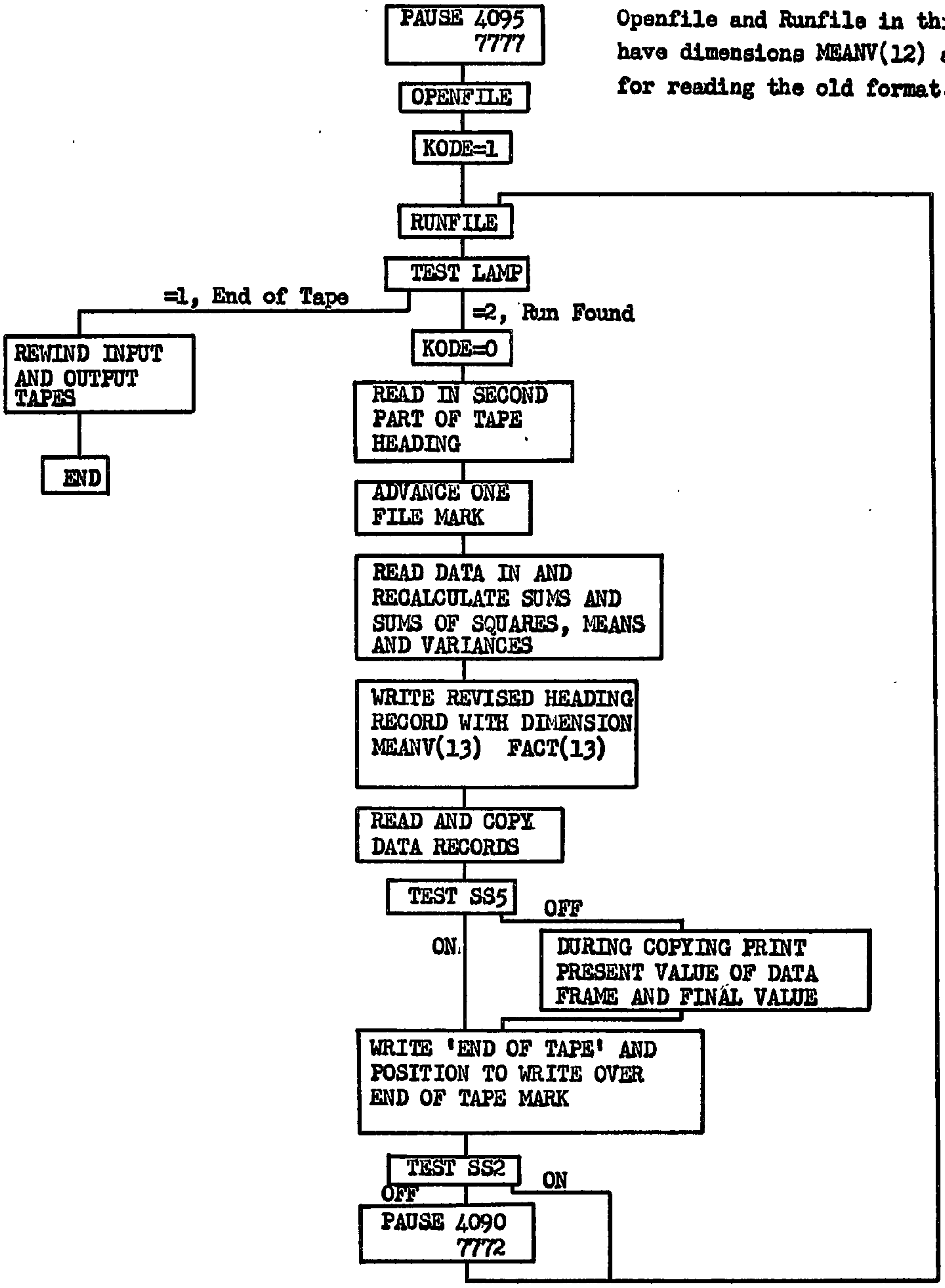
```
00000  C    PROGRAM TO MODIFY FORMAT OF EDITED TAPES
00000  C     ----------------------------------------------------------------
00000  C     COMM ON STORAGE
00000  C
00004        COMMON RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004        DIMENSION RNM(8),CHNO(13),VNM(3,13),MEANV(12),FACT(12)
00004        INTEGER RNM,CHNO
00004        REAL MEANV
00004  C
00004  C     ----------------------------------------------------------------
00004  C     OUTPUT FILES
00004  C
00004  C                        VTAPE FILE 1 TAPE LABEL
00004  C
00004  C     CALL OPENFILE (OUT,IN,KEY)
00004        INTEGER OUT
00004  C
00004  C                        VTAPE FILE 2 RECORD1  (     FP WORDS)
00004  C     WRITE TAPE VTAPE,RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004  C
00004  C                        VTAPE FILE 2 RECORD2  (     FP WORDS)
00004  C
00004  C     WRITE TAPE VTAPE,A,VARV,MEANLO,MEANCTR,MEANHI,VARLO,VARCTR,VARHI
00004        DIMENSION A(4,13),VARV(12),MEANLO(12),MEANCTR(12),MEANHI(12),
00004       1              VARLO(12),VARCTR(12),VARHI(12)
00004        REAL MEANLO,MEANCTR,MEANHI
00004  C
00004  C                        VTAPE FILE 2 RECORD3  (     FP WORDS)
00004  C
00004  C     WRITE TAPE VTAPE,RL,CHNO,VNM,MEANV,FACT,JBIG,KOUNI,DELT
00004        DIMENSION RL(8)
00004        INTEGER RL
00004        EQUIVALENCE (NL(1),RL(1))
00004        DIMENSION NL(8)
00004        RL(1)=5657904
00007        RL(2)=-6722508
00013        RL(3)=4617584
00016        RL(4)=-3994575
00022        RL(5)=-3994575
00026        RL(6)=-3994575
00032        RL(7)=-3994575
00036        RL(8)=-3994575
00042  C
00042  C                        VTAPE FILE 3 RECORD1  (     FP WORDS)
00042  C     CALL WRITEH(VTAPE,V,V(14,10))
00042  C
00042        DIMENSION V(13,10)
00042        INTEGER VTAPE
00042  C     ----------------------------------------------------------------
00042  C                        WORKING STORAGE
00042  C
00042        DIMENSION SUM(12),SUMSQ(12)
00042        REAL NILL,NONE
00042        NONE=0
00045        NILL=0
00050  C
00050  C     ----------------------------------------------------------------
00050  1000  PAUSE 4095
00052        CALL OPENFILE(OUT,IN,KEY)
00056  1001  KODE=1
00060  1100  CALL RUNFILE(IN,KEY,KODE,LAMP)
```

```
00065          GO TO(9000,1101),LAMP
00072   1101   KODE=0
00074   C
00074   C      READ SECOND HEADING RECORD AND ADVANCE TAPE TO DATA RECORD
00074   C
00074          READ TAPE IN,A,VARV,MEANLO,MEANCTR,MEANHI,VARLO,VARCTR,VARHI
00127          CALL SELECT(IN,11)
00132          CALL EJECT
00133   C
00133   C       READ DATA IN AND RECALCULATE SUMS AND SUMS SQUARED OF DATA
00133   C
00133   1200   DO 1209 J=1,12
00135             SUM(J)=0.
00142             SUMSQ(J)=0.
00147   1209   CONTINUE
00154   1210   CALL READB(IN,V,V(14,10))
00171          CALL EOFCK(IN,LITE)
00174          KEY=3
00176          GO TO (1300,1211),LITE
00203   1211   DO 1219 K=1,10
00205             DO 1219 J=1,12
00207                SUM(J)=V(J,K)+SUM(J)
00224                SUMSQ(J)=V(J,K)*V(J,K)+SUMSQ(J)
00251   1219   CONTINUE
00263          KOUNT=V(13,10)
00275          GO TO 1210
00276   C
00276   C       CALCULATE NEW MEANS AND VARIANCES
00276   C
00276   1300   CALL BACKUP(IN,KEY)
00301          CALL SELECT(IN,11)
00304          DO 1309 J=1,12
00306             MEANV(J)=SUM(J)/KOUNT
00317             VARV(J)=SUMSQ(J)/KOUNT-MEANV(J)*MEANV(J)
00336   1309   CONTINUE
00343          GO TO 2000
00344   C
00344   C       WRITE REVISED HEADING RECORDS
00344   C
00344   2000   WRITE TAPE OUT,RNM,CHNO,VNM,MEANV,NONE,FACT,NILL,JBIG,KOUNT,DELT
00400          WRITE TAPE OUT,A,VARV,MEANLO,MEANCTR,MEANHI,VARLO,VARCTR,VARHI
00433          ENDFILE OUT
00435   C
00435   C      READ AND COPY DATA RECORDS
00435   C
00435   2001   CALL READB(IN,V,V(14,10))
00452          CALL EOFCK(IN,LITE)
00455          GO TO (3000,2002),LITE
00462   2002   CALL WRITEB(OUT,V,V(14,10))
00477          IF(SENSE SWITCH 5) 2011,2010
00503   2010   PRINT 200, KOUNT,V(13,10)
00523   200    FORMAT(11X,I10,10X,F9.0)
00531   2011   GO TO 2001
00532   C
00532   C      END OF COPY, WRITE END OF TAPE AND POSITION
00532   C      TO WRITE OVER END OF TAPE MARK
00532   C
00532   3000   ENDFILE OUT
00534          KEY=3
00536          WRITE TAPE OUT,RL,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00566          CALL SELECT(OUT,12)
00571          CALL SELECT(OUT,11)
```

```
 00574          IF(SENSE SWITCH 2)1100,3001
 00600   3001   PAUSE 4090
 00602          GO TO 1100
 00603   9000   REWIND IN
 00605          REWIND OUT
 00607          CALL EJECT
 00610          END


 SUBPROGRAMS

   OPENFILE   RUNFILE    SELECT     EJECT      READB      EOFCK
   BACKUP     WRITEB

 PROGRAM ALLOCATION

  00656    NONE      00660    NILL      00662    OUT       00663    IN
  00664    KEY       00665    KODE      00666    LAMP      00667    J
  00670    LITE      00671    K

  00672    A         01042    VARV      01072    MEANLO    01122    MEANCTR
  01152    MEANHI    01202    VARLO     01232    VARCTR    01262    VARHI
  00646    RL        00646    NL        01312    V         01716    SUM
  01746    SUMSQ

 COMMON ALLOCATION

  00000    RNM       00010    CHNO      00025    VNM       00143    MEANV
  00173    FACT      00223    JBIG      00224    KOUNT     00225    DELT


 PROGRAM END
```

MODIFY FORMAT

Openfile and Runfile in this program have dimensions MEANV(12) and FACT(12) for reading the old format.


PAUSE 4095
7777
OPENFILE
KODE=1
RUNFILE
TEST LAMP
=1, End of Tape
=2, Run Found
REWIND INPUT AND OUTPUT TAPES
END
KODE=0
READ IN SECOND PART OF TAPE HEADING
ADVANCE ONE FILE MARK
READ DATA IN AND RECALCULATE SUMS AND SUMS OF SQUARES, MEANS AND VARIANCES
WRITE REVISED HEADING RECORD WITH DIMENSION MEANV(13) FACT(13)
READ AND COPY DATA RECORDS
TEST SS5
OFF
ON
DURING COPYING PRINT PRESENT VALUE OF DATA FRAME AND FINAL VALUE
WRITE 'END OF TAPE' AND POSITION TO WRITE OVER END OF TAPE MARK
TEST SS2
ON
OFF
PAUSE 4090
7772

PRINTOUT OF MODIFIED EDITED VERSION OF DATA MAGNETIC TAPE

The printout provided as an example of the revised tape format is that of Dominion Tape No. 3 (revised). This tape has an error on it which cannot be removed. There is a reported parity error in the first record of the Runfile for run Dominion A1. To overcome this, set BPI 03521 at the first pause. The computer will halt before trying to backspace. Set P 03516, turn off BPI and press GO.

The contents of the records on tape are much as before revision and they will not be listed again. The main difference in this particular example is that the constants for all variables have been put on one tape during repeated transfer of data from one tape to another. The Cambridge cross coupling computer constants cannot be referenced by the subroutines since CHNO on this tape contains only the I.R.I.G. channel numbers of the Dominion computer data.

On examination it will be found that NONE occupies record 7 words 28 and 29 and NILL occupies record 8 words 22 and 23. These two dummy variables defined in the Modify Format program, fill up the '13th dimension' of MEANV and FACT so that the latter can be referenced by MEANV(13) and FACT(13). JBIG, KOUNT, and DELT are all advanced 4 words in the last record of the first half of the Runfile. Since the remainder of this record is blank, subsequent records are unaffected by this change in format.

The data tape is now in the correct format for processing by programs employing the revised format.

```
 RECORD NO.     1 (BCD) CONTAINING    136 CHARACTERS
THIS TAPE CONTAINS DOMINION DATA TAPE NUMBER THREE RUNS A1 TO R3


 RECORD NO.     2 (BCD) CONTAINING    136 CHARACTERS
REVISED BY PROGRAM TO MODIFY FORMAT OF EDITED TAPES AS AT 18 AUGUST ≠67


 RECORD NO.     3 (BCD) CONTAINING    136 CHARACTERS
BELONGING TO RICHARD HAWORTH


 END OF FILE
 THERE ARE      3 RECORDS CONTAINING     102 WORDS BEFORE FILEMARK NO.  1
```

19 August, 1967

Print of Dominion Tape Number 3 (Modified edited version)

Runs A1 and A2.

Parity error observed in first record of the Leading File for Run A1.

To overcome this set BPI 03521 at 1st pause

Computer will halt before trying to backspace.

Set P = 03516

Turn off BPI

GO.

```
UNRESOLVED TAPE PARITY ERROR ASSOCIATED WITH THE FOLLOWING RECORD. BINARY MODE CHOSEN ARBITRARILY      A1 Heading File

RECORD NO.    4 (BINARY)  CONTAINING   34 WORDS
  5400857 -6973018 -3998606 -4189886   131141   809986  1342470  1772592        7        9        5        4     12
        0        0        0        0        0        0        0        0        0        0        0        0     24
        0        0        0        0        0        0        0        0 -5702439        0

RECORD NO.    5 (BINARY)  CONTAINING   34 WORDS
        0        0        0        0        0        0        0  5400870  4926486  5061680 -3994575 -3994575     12
 -3994575  5400870  4926489  5061680 -3994575 -3994575 -3994575  5052690 -5925608  5703059  5180464 -3994575     24
 -3994575  5400870  4926488 -6646150 -4123436  5651504 -3994575        0 -6730434        0

RECORD NO.    6 (BINARY)  CONTAINING   34 WORDS
        0        0        0        0        0  5400870  4926482  5577008 -7181094 -6661071 -3994575  5052690     12
 -5925608  5702805  4607012 -6805209 -6878159  5052690 -5925608  5703251  5180464 -3994575 -3994575  5052690     24
 -5925608  5703206 -5923215  4535509 -7402447 -4105899 -4037199  5315793  6032843        0

RECORD NO.    7 (BINARY)  CONTAINING   34 WORDS
 -4102556  5704752 -3994575        0        0        0        0        0        0 -4197113 -7179595  4202487     12
  2211773 -4180085  5335105 -4206371  1111998        0        0 -4189583   958111 -4196367 -5236504 -4204907     24
  3532447  4201463 -6601457        0        0        0        0        0 -8253762        0

RECORD NO.    8 (BINARY)  CONTAINING   34 WORDS
        0        0        0  4156631  1209642  4156289  7351814  4160101  6046611  4160074 -6491686        0     12
        0  4155565 -7393349  4157263 -1505245  4155466 -6316927  4156781  -770101        0        0        4     24
     3010  4196352        0 -3994575 -3994575 -3994575 -3994575 -3994575 -7181795        5

RECORD NO.    9 (BINARY)  CONTAINING   34 WORDS
        0        0        0        0        0        0        0        0        0        0        0        0     12
        0        0        0        0        0        0        0        0        0        0        0        0     24
  4200448        0  4213248        0        0        0        0        0 -8363519        0

RECORD NO.   10 (BINARY)  CONTAINING   34 WORDS
  4200448        0  4213248        0        0        0        0        0  4200448        0  4205363  3355443     12
        0        0        0        0  4200448        0  4218163  3355443        0        0        0        0     24
        0        0        0        0        0        0        0        0 -1605426        0

RECORD NO.   11 (BINARY)  CONTAINING   34 WORDS
  4200448        0  4222259  3355443        0        0        0        0  4200448        0  4225440        0     12
        0        0        0        0  4200448        0  4234032        0        0        0        0        0     24
  4200448        0  4226976        0        0        0        0        0  3511512        0

RECORD NO.   12 (BINARY)  CONTAINING   34 WORDS
        0        0        0        0        0        0        0        0       -0       -0       -0       -0     12
       -0       -0  4188925  4332905  4196961  -398466  4148376   400386  4222159  1042932       -0       -0     24
  4227014  3574797  4217405  6014851  4197019 -2754742  4210234  4138577  -372312        0

RECORD NO.   13 (BINARY)  CONTAINING   34 WORDS
        0        0        0        0        0        0 -4233600       -0 -4233736       -0 -4230720       -0     12
 -4229904       -0        0        0 -4234792       -0 -4235080       -0 -4234856       -0 -4225664       -0     24
        0        0        0        0        0        0  4214528        0  3910606        0
```

```
RECORD NO.   14 (BINARY)  CONTAINING   34 WORDS
  4221120        0 -4218752       -0  4217600        0        0        0  4217216        0 -4230608       -0          12
  4220992        0  4230608        0        0        0        0        0        0        0  4233840        0          24
  4234296        0  4229728        0  4230608        0        0        0 -3967782        0

RECORD NO.   15 (BINARY)  CONTAINING   34 WORDS
  4235096        0  4221824        0  4237372        0  4237368        0        0        0        0        0          12
        0        0       -0       -0       -0       -0       -0       -0       -0       -0        0        0          24
       -0       -0       -0       -0       -0       -0       -0       -0   154445        0

RECORD NO.   16 (BINARY)  CONTAINING   34 WORDS
        0        0        0        0        0        0       -0       -0       -0       -0       -0       -0          12
       -0       -0        0        0       -0       -0       -0       -0       -0       -0       -0       -0          24
        0        0        0        0        0        0       -0       -0        0        0

RECORD NO.   17 (BINARY)  CONTAINING   34 WORDS
       -0       -0       -0       -0       -0       -0        0        0       -0       -0       -0       -0          12
       -0       -0       -0       -0 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575          24
 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575  3195880        9

END OF FILE
THERE ARE    17 RECORDS CONTAINING    578 WORDS BEFORE FILEMARK NO.  2
```

```
END OF FILE
THERE ARE   318 RECORDS CONTAINING  78838 WORDS BEFORE FILEMARK NO.  3
```

```
RECORD NO.  319 (BINARY)  CONTAINING   34 WORDS
  5400857 -6973018 -3998605 -4190138  1441859   527362  1580038  1772592        7        9        5        4     12
        0        0        0        0        0        0        0        0        0        0        0        0     24
        0        0        0        0        0        0        0        0 -4437029        0

RECORD NO.  320 (BINARY)  CONTAINING   34 WORDS
        0        0        0        0        0        0        0  5400870  4926486  5061680 -3994575 -3994575     12
 -3994575  5400870  4926489  5061680 -3994575 -3994575 -3994575  5052690 -5925608  5703059  5180464 -3994575     24
 -3994575  5400870  4926488 -6646150 -4123436  5651504 -3994575        0 -6730434        0

RECORD NO.  321 (BINARY)  CONTAINING   34 WORDS
        0        0        0        0        0  5400870  4926482  5577008 -7181094 -6661071 -3994575  5052690     12
 -5925608  5702805  4607012 -6605209 -6878159  5052690 -5925608  5703251  5180464 -3994575 -3994575  5052690     24
 -5925608  5703206 -5923215  4535509 -7402447 -4105899 -4037199  5315793  6032843        0

RECORD NO.  322 (BINARY)  CONTAINING   34 WORDS
 -4102556  5704752 -3994575        0        0        0        0        0        0  4196862  5180258  4205058     12
 -7142302 -4180085  5335102 -4205563 -6917620        0        0  4196368 -1963566 -4196367 -5236507 -4204907     24
  3532443  4201463 -6801452        0        0        0        0        0   584019        0

RECORD NO.  323 (BINARY)  CONTAINING   34 WORDS
        0        0        0  4156631  1209642  4156289  7351814  4160101  6046611  4160074 -6491686        0     12
        0  4155565 -7393349  4157263 -1505245  4155466 -6316927  4156781  -770101        0        0        4     24
     2980  4196352        0 -3994575 -3994575 -3994575 -3994575 -3994575 -7161825        5

RECORD NO.  324 (BINARY)  CONTAINING   34 WORDS
        0        0        0        0        0        0        0        0        0        0        0        0     12
        0        0        0        0        0        0        0        0        0        0        0        0     24
  4200448        0  4213248        0        0        0        0        0 -8363519        0

RECORD NO.  325 (BINARY)  CONTAINING   34 WORDS
  4200448        0  4213248        0        0        0        0        0  4200448        0  4205363  3355443     12
        0        0        0        0  4200448        0  4218163  3355443        0        0        0        0     24
        0        0        0        0        0        0        0        0 -1605426        0

RECORD NO.  326 (BINARY)  CONTAINING   34 WORDS
  4200448        0  4222259  3355443        0        0        0        0  4200448        0  4225440        0     12
        0        0        0        0  4200448        0  4234032        0        0        0        0        0     24
  4200448        0  4226976        0        0        0        0        0  3511512        0

RECORD NO.  327 (BINARY)  CONTAINING   34 WORDS
        0        0        0        0        0        0        0        0       -0       -0       -0       -0     12
       -0       -0  4196970  4335682  4188643 -7167565  4148376   400306  4217167  4819846       -0       -0     24
  4209353 -1470319  4217405  6014853  4197019 -2754734  4210234  4138579 -4429830        0

RECORD NO.  328 (BINARY)  CONTAINING   34 WORDS
        0        0        0        0        0        0 -4233600       -0 -4233736       -0 -4230720       -0     12
 -4229904       -0        0        0 -4234792       -0 -4235080       -0 -4234856       -0 -4225664       -0     24
        0        0        0        0        0        0  4214528        0  3910606        0

RECORD NO.  329 (BINARY)  CONTAINING   34 WORDS
```

```
  4221120        0 -4218752       -0  4217600        0        0        0  4217216        0 -4230608       -0         12
  4220992        0  4230608        0        0        0        0        0        0        0  4233840        0         24
  4234296        0  4229728        0  4230604        0        0        0 -3967782        0

 RECORD NO.  330 (BINARY)  CONTAINING   34 WORDS
  4235096        0  4221824        0  4237372        0  4237368        0        0        0        0        0         12
        0        0       -0       -0       -0       -0       -0       -0       -0       -0        0        0         24
       -0       -0       -0       -0       -0       -0       -0       -0   154445        0

 RECORD NO.  331 (BINARY)  CONTAINING   34 WORDS
        0        0        0        0        0        0       -0       -0       -0       -0       -0       -0         12
       -0       -0        0        0       -0       -0       -0       -0       -0       -0       -0       -0         24
        0        0        0        0        0        0       -0       -0        0        0

 RECORD NO.  332 (BINARY)  CONTAINING   34 WORDS
       -0       -0       -0       -0       -0       -0        0        0       -0       -0       -0       -0         12
       -0       -0       -0       -0 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575         24
 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575  3195660        9

 END OF FILE
 THERE ARE   332 RECORDS CONTAINING  79314 WORDS BEFORE FILEMARK NO.  4
```

LIST SUMMARIES OF MVD TAPES

I. PURPOSE:

To print a summary of the parameters of the variables in each run on a Multi-Variable Data Tape as it exists after editing and modifying the format.

II. OPERATION:

The program reads in the Run Heading File and lists the required parameters in a standard format, the specification of which comprises the majority of the program. Each listing fills a computer line printer page when four variables are concerned. A specimen summary is included with this description.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

PAUSE 7777
4095 — Initial pause after compilation to allow setting up of data tape.

PAUSE 7773
4091 — Pause reached after input tape summaries have all been listed. When new tape is mounted, pressing GO will initiate listing of summaries from new tape.

IV. USAGE:

Compile program.
Pause 7777
Load input tape on handler 1.   Col. 60
Load card: INPUT TAPE ON HANDLER   1
GO.

V. STORAGE REQUIREMENTS:

1471_8 825_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

OPENFILE   RUNFILE   SELECT

```
00000  C      LIST SUMMARIES OF MVD TAPES
00004         DIMENSION RNM(8),CHNO(13),VNM(3,13),MEANV(13),FACT(13),VARV(12),
00004        1A(4,13),MEANLO(12),MEANCTR(12),MEANHI(12),VARLO(12),VARCTR(12),
00004        2VARHI(12)
00004         COMMON RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004         INTEGER RNM,CHNO
00004         REAL MEANV
00004         REAL MEANLO,MEANCTR,MEANHI
00004         PAUSE 4095
00006         DO 15000 I=1,6
00010         IF(SENSE SWITCH 1)15000,15002
00014  15000 CONTINUE
00021         CALL LEADER
00022         CALL DUMP
00023  15002 CONTINUE
00023         CALL OPENFILE(0,IN,KEY)
00027         KODE=0
00031  100    CALL RUNFILE(IN,KEY,KODE,LAMP)
00036         GO TO (9999,1000),LAMP
00043  1000   READ TAPE IN,A,VARV,MEANLO,MEANCTR,MEANHI,VARLO,VARCTR,VARHI
00076         CALL SELECT(3072,4)
00101  2      FORMAT(//////)
00104         PRINT200,(RNM(I),I=1,8),KOUNT,JBIG,DELT
00131  200    FORMAT(21X,25HSUMMARY OF DATA FOR RUN   ,8A4,//,31X,20HTOTAL NO. OF
00131        1 SAMPLES I7,/,31X,19HNUMBER OF VARIABLES,I8,/,31X,20HSAMPLING INTE
00131        2RVAL    ,F7.3,8H SECONDS/)
00200         PRINT 210
00203  210    FORMAT(7X,50HVARIABLE NAME      , CHAN  DATA MEAN  DATA VARIANCE/)
00223         DO 3000 JJ=1,JBIG
00225         J=CHNO(JJ)
00230         PRINT 310,(VNM(I,J),I=1,3),CHNO(JJ),MEANV(J),VARV(J)
00300  310    FORMAT(1X,3A8,3X,I2,3X,F9.4,5X,F10.5)
00311  3000   CONTINUE
00316         PRINT 2
00321         PRINT 400,(RNM(I),I=1,8)
00340  400    FORMAT(21X,40HMULTI-CHANNEL DATA COMPENSATION FOR RUN  8A4,/, 75H
00340        1          VARIABLE NAME          CHAN                A1          A2          A3
00340        2       A4/)
00403         DO 4000 JJ=1,JBIG
00405         J=CHNO(JJ)
00410         PRINT 410,(VNM(I,J),I=1,3),CHNO(JJ),A(1,J),A(2,J),A(3,J),A(4,J)
00516  410    FORMAT(1X,3A8,3X,I2,10X,F9.4,1X,F9.4,1X,F9.5,1X,F9.6)
00533  4000   CONTINUE
00540         PRINT 2
00543         PRINT 500,(RNM(I),I=1,8)
00562  500    FORMAT(21X,33HSTANDARDIZATION SUMMARY FOR RUN   ,8A4,/)
00577         PRINT 510
00602  510    FORMAT(7X,112HVARIABLE NAME         CHAN     MEAN-LOW    MEAN-ZRO    ME
00602        1AN-HI   VARIANCE-LOW   VARIANCE-ZRO    VARIANCE-HI  NORMFACT/)
00642         DO 6000 JJ=1,JBIG
00644         J=CHNO(JJ)
00647         PRINT 520,(VNM(I,J), I=1,3),CHNO(JJ),MEANLO(J),MEANCTR(J),
00647        1         MEANHI(J),VARLO(J),VARCTR(J),VARHI(J),FACT(J)
00762  520    FORMAT(1X,3A8,3X,I2,1X,3(2X,F9.4),3(6X,F9.5),2X,F7.5)
00777  6000   CONTINUE
01004         PRINT 2
01007         GO TO 100
01010  9999   CALL SELECT(3072,4)
01013         REWIND IN
01015         PAUSE 4091
01017         GO TO 100
```

```
 01020         END

 SUBPROGRAMS

   LEADER      DUMP        OPENFILE   RUNFILE    SELECT

 PROGRAM ALLOCATION

 01043    I         01044    IN         01045    KEY        01046    KODE
 01047    LAMP      01050    JJ         01051    J

 01052    VARV      01102    A          01252    MEANLO     01302    MEANCTR
 01332    MEANHI    01362    VARLO      01412    VARCTR     01442    VARHI

 COMMON ALLOCATION

 00000    RNM       00010    CHNO       00025    VNM        00143    MEANV
 00175    FACT      00227    JBIG       00230    KOUNT      00231    DELT

 PROGRAM END
```

```
            SUMMARY OF DATA FOR RUN  CAMBRIDGE A1 1510-1535 257 66

                        TOTAL NO. OF SAMPLES    3020
                        NUMBER OF VARIABLES        4
                        SAMPLING INTERVAL      0.500 SECONDS

       VARIABLE NAME        CHAN  DATA MEAN  DATA VARIANCE

CAMBRIDGE BEAM MOTION        10     -0.0663     113.16093
CAMBRIDGE HORIZ ACCEL        12     -2.5681      48.53989
CAMBRIDGE ICC                11     -5.3160      21.39071
CAMBRIDGE FCC                 6      0.2753       0.10954



            MULTI-CHANNEL DATA COMPENSATION FOR RUN CAMBRIDGE A1 1510-1535 257 66
       VARIABLE NAME        CHAN                A1        A2         A3         A4

CAMBRIDGE BEAM MOTION        10               1.0000   77.0000   0.00000   0.000000
CAMBRIDGE HORIZ ACCEL        12               1.0000  125.0000   0.00000   0.000000
CAMBRIDGE ICC                11               1.0000  358.0000   0.00000   0.000000
CAMBRIDGE FCC                 6               1.0000    2.8000   0.00000   0.000000



            STANDARDIZATION SUMMARY FOR RUN  CAMBRIDGE A1 1510-1535 257 66

       VARIABLE NAME        CHAN   MEAN-LOW   MEAN-ZRO    MEAN-HI   VARIANCE-LOW   VARIANCE-ZRO     VARIANCE-HI  NORMFAC

CAMBRIDGE BEAM MOTION        10  -489.0000  -221.0000    46.0000      -0.00000       -0.00000        -0.00000  0.00187
CAMBRIDGE HORIZ ACCEL        12   -84.0000   221.0000   526.0000      -0.00000       -0.00000        -0.00000  0.00164
CAMBRIDGE ICC                11  -461.0000    33.0000   527.0000      -0.00000       -0.00000        -0.00000  0.00101
CAMBRIDGE FCC                 6  -228.0000   -31.0000   166.0000      -0.00000       -0.00000        -0.00000  0.00254



            CAMBRIDGE A2 1155-1220 261 66
   FOUND RUN CAMBRIDGE A2 1155-1220 261 66       2980
```

POWER SPECTRUM ANALYSIS

A stationary Gaussian random process with zero time average has all its relevant statistical properties defined by its autocovariance function or power spectrum. In many practical cases in which we are interested, the signals representing a process are approximately stationary (are unaffected by translations in the time origin), but are not Gaussian and are often usefully studied in terms of autocovariance functions and power spectra. To provide an exact estimate of the spectrum requires a perfectly measured, infinitely long piece of a random function. This is of course impractical since we can only make discrete measurements over a small sample of the process. Techniques have been developed so that such measurements can provide a smoothed estimate of the true power spectrum.

X(t) is one of an ensemble of functions generated by a random process, whose value at any time is a random variable with a probability distribution induced by the ensemble. The random process is assumed to be Gaussian so that the values of the function at a set of points $t = t_i$, $i = 1$ to n, have an n-dimensional joint probability function which is completely determined by the ensemble averages

$$\bar{X}(t_i) = \text{ave} \left\{ X(t_i) \right\}$$

and by the covariances $C_{ij} = \text{cov} \left\{ X(t_i), X(t_j) \right\}$

$$= \text{ave} \left\{ \left[X(t_i) - \bar{X}(t_i) \right] \cdot \left[X(t_j) - \bar{X}(t_j) \right] \right\}$$

Restricting ourselves to processes with zero averages and which are stationary, a complete specification of the distribution is provided by the covariances C_{ij} which now depend only on the time separation $(t_i - t_j)$. $C_{ij} = C(t_i - t_j)$ where C(0) is the variance of X(t).

It is more usual in practical situations to work with a single time function of infinite extent rather than an ensemble of finite pieces of such functions. However, in the statistically random case, the averages across the ensemble and single functions are equivalent.

In single function terms, the covariance at lag K is :

$$C(K) = \lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} X(t) . X(t+K) \, dt$$

and C(K) is called the autocovariance function. The autocovariance function and the power spectrum P(f) are Fourier transforms of each other.

$$C(K) = \int_{-\infty}^{+\infty} P(f).e^{i2\pi fK}. df$$

$$\text{where } P(f) = \lim_{T\to\infty} \frac{1}{T}\left[\int_{-T/2}^{T/2} X(t).e^{-i2\pi ft}. dt\right]^2$$

P(f)df represents the contribution to the variance from frequencies between f and (f+df).

The practical difficulties involved in making the estimate of this quantity are discussed in Blackman and Tukey ('The Measurement of Power Spectra' (Dover)) and only the steps involved in the calculation are included here, with reference to the subroutines which perform these calculations.

In the present analysis prewhitening of the data was not included. The purpose of such treatment is to smooth out the spectrum so that any dominant frequencies which may exist in the spectrum do not contaminate the remainder of the spectrum due to the side lobes of the spectral windows used.

The lagged products R(K) are summed for all lags, K,

$$R(K) = \sum_{i=1}^{IBIG-(K-1)} X_i.X_{i+(K-1)}$$ where IBIG is the number of samples available, in

the subroutine PWRSPECT, and a further passage through this subroutine yields the autocovariance function

$$C(K) = \frac{R(K)}{IBIG-(K-1)}$$

The autocovariance function has then to be modified by a filter before Fourier transforming to produce the power spectrum.

The functions calculated above are generally referred to as the apparent autocovariance functions because we are not using an infinite record as required for a true specification of the function. By multiplying the apparent function by a function of K, a modified apparent function is obtained, which, although possibly not a good estimate of the true autocovariance function, has a transform which is a good estimate of the smoothed value of the true power spectrum. Three of these 'filters' are provided in PWRSPECT and the relative merits of these are discussed in Blackman and Tukey. In analyzing the cross coupling data, the Hamming filter was used.

The cosine transform to yield the spectrum is applied by the subroutine COSTR. It is this output for three lag values which comprises the analysis of cross coupling data presented in Data Report 1967-6-D.

DICROSCOPES (DIcks CROSs COupling PowEr Spectra)

I. PURPOSE:

To provide a power spectrum analysis of time series data written in a specified format.

II. OPERATION:

The program first searches for a given run with the operation of subroutines OPENFILE and RUNFILE and the run parameter file is written on the output tape. The program then proceeds to calculate the power spectra and autocovariance functions for all the variables of the specified run using the set of subroutines described in this volume. When each spectrum has been written on tape, an end of file mark is written and the tape backspaced over it in readiness for writing the next spectrum. Only when all the spectra for a given run have been written is this end of file mark allowed to remain on tape, so that a searching subroutine will be able to identify its position on tape.

The program runs in the sequential mode (i.e. with KODE = 0 as the key for RUNFILE) so that successive runs will be analyzed without the need for control cards. When the input tape is exhausted, both the input and output tapes are rewound and the program is terminated. Hence the program may be left unattended.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

PAUSE 7777_8 4095_{10}	Initial pause after compiling program to allow setting up of tapes.
SS 5 ON	Suppresses printing of the power spectrum and autocovariance functions.
SS 6 ON	If starting the program in the middle of a run and it is required to restart with the analysis of a particular variable. The program must start by calculating the sums of products for up to 200 lags, which is then used to calculate the autocovariance and power spectra for all other lag values. Hence only the variable can be specified, we cannot pick out a particular lag value.
SS 6 OFF	Causes the program to examine all the variables of the current run.
PAUSE 7	Is reached with SS 6 ON to allow input of a value of L indicating the first variable of the current run to be examined. The L value is written in the 30th column of a card which is read by the program upon pressing GO.

```
                        Col 30
"INITIAL VALUE OF L IS    L
L = 1 starts with the 1st variable
L = 2 starts with the 2nd variable, etc.
```

IV. USAGE:

Compile program.
Pause 7777.
Load input tape on handler 1, output tape on handler 2.
Load reader with cards as required by OPENFILE and RUNFILE including heading cards for the output tape if it is a new one, and the name of the first run to be found by RUNFILE.
Set SS 6 and SS 5 as required (see above).
GO.
If SS 6 is OFF the program will then continue unattended.

The program requires a little tailoring to the particular usage because of the use of subroutine VARBLE. Statements 1110, 1120, 1130, and 1140 are followed by unnumbered FORMAT statements which require the names of the variables on tape in the order in which they appear on tape. The FORMAT statements should be included as described in Section IV USAGE (a) to (c) of VARBLE.

There is no limit to the number of variables which the program can handle, but the GO TO statement prior to 1110 should be amended accordingly and statement 2001 must have (L-N) where N is the total number of variables.

V. STORAGE REQUIREMENTS:

3626_8 1942_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

OPENFILE	RUNFILE	WRITEANS	SELECT	VARBLE
PWRSPECT	DATAIN	PRINTPSP	BACKUP	PRINTACV

VII. RUNNING TIMES:

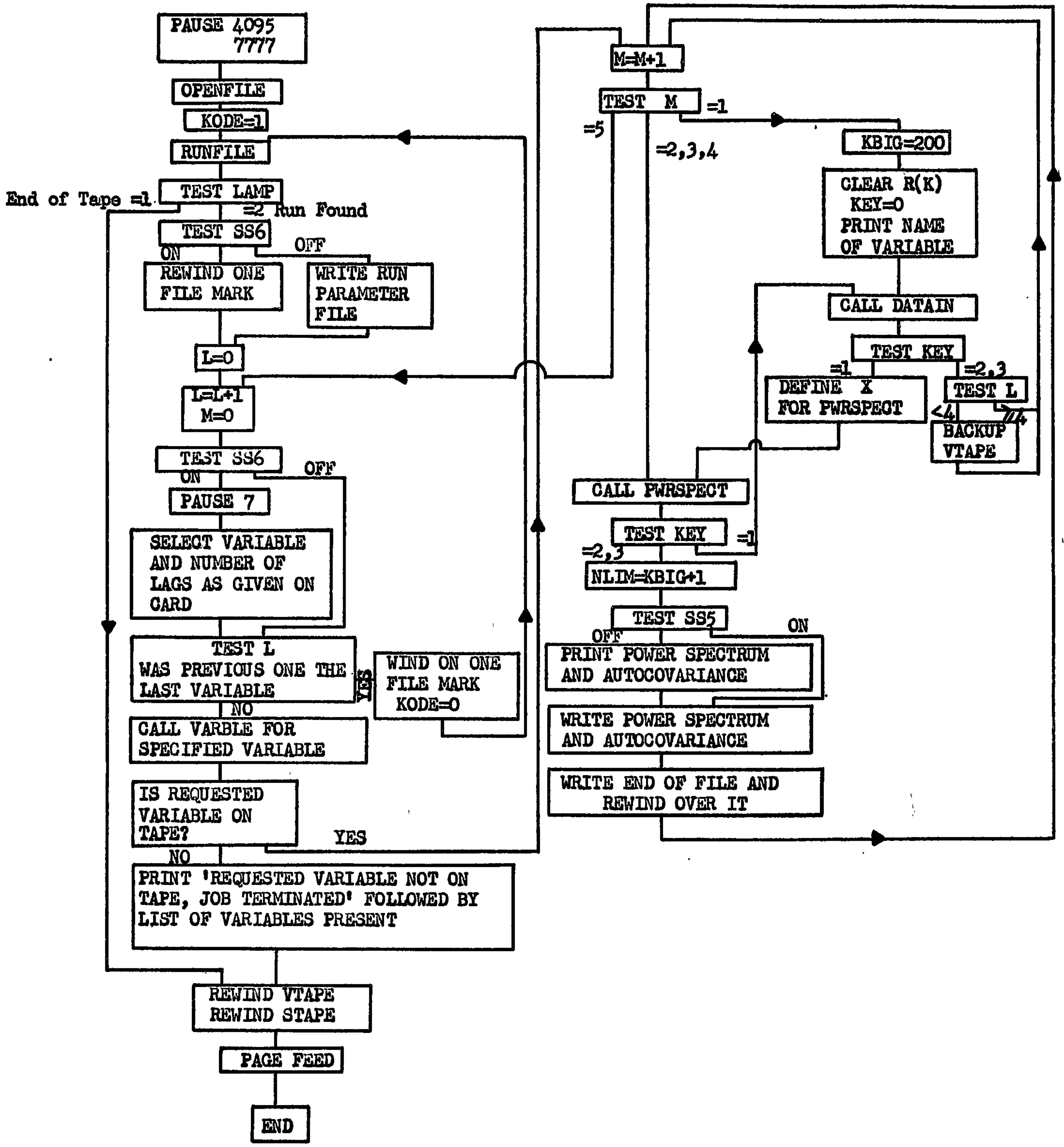
Calculation and writing of spectra and autocovariance on tape (without print-out) is 1 hour for 3300 samples of 4 variables and 3 lags.

```
00000  C      DICROSCOPES
00000  C      POWER SPECTRUM ANALYSIS OF HAWORTH CROSS COUPLING DATA
00000  C      AUGUST 22, 1967
00004         COMMON RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DFLT
00004         DIMENSION RNM(8),CHNO(13),VNM(3,13),MEANV(13),FACT(13)
00004         INTEGER RNM,CHNO
00004         REAL MEANV
00004  C
00004         DIMENSION R(251),C(251),S(251),FRAME(13)
00004         INTEGER VTAPE,STAPE
00004         PAUSE 4095
00006         DO 15000 I=1,6
00010         IF(SENSE SWITCH 1)15000,15002
00014  15000 CONTINUE
00021         CALL LEADER
00022         CALL DUMP
00023  15002 CONTINUE
00023         CALL OPENFILE(STAPE,VTAPF,KEY)
00027         KODE = 1
00031  1000   CALL RUNFILE(VTAPE,KEY,KODE,LAMP)
00036         GO TO (9000,1010),LAMP
00043  1010   IF(SENSE SWITCH 6) 1012,1011
00047  CSUPPRESSES WRITING OF RUN PARAMETER FILE WHEN STARTING IN THE MIDDLE OF
00047  C A RUN
00047  C
00047  C      OPEN NEW ANSWER FILE FOR THIS RUNFILE
00047  C
00047  1011   CALL WRITEANS(STAPE,S,0,0,0,KOUNT)
00056         GO TO 1013
00057  1012   CALL SELECT(STAPE,12)
00062  1013   L=0
00064  1100   L=L+1
00067  1101   M=0
00071  C
00071  C
00071  C      ALLOWS SELECTION OF VARIABLE(L) AND NUMBER OF LAGS(M) WHEN
00071  C      RESTARTING IN THE MIDDLE OF A RUN
00071         IF (SENSE SWITCH 6) 1102,1109
00075  1102   PAUSE 7
00077  1103   READ 1106,L,M
00106  1106   FORMAT(29X,I1,29X,I1)
00113         M=M-1
00116  C
00116  1109   CONTINUE
00116  C      SELECT VARIABLE.
00116  C
00116         GO TO (1110,1120,1130,1140,1170),L
00126  1110   CALL VARBLE(X,VNM,IX)
00132         FORMAT(24HDOMOBS HORIZ ACCEL         )
00143            IF(IX) 1200,9100,1200
00147  1120   CALL VARBLE(X,VNM,IX)
00153         FORMAT(24HDOMOBS BEAM MOTION         )
00164            IF(IX) 1200,9100,1200
00170  1130   CALL VARBLE(X,VNM,IX)
00174         FORMAT(24HDOMOBS ICC                 )
00205            IF(IX) 1200,9100,1200
00211  1140   CALL VARBLE(X,VNM,IX)
00215         FORMAT(24HDOMOBS FCC                 )
00226            IF(IX) 1200,9100,1200
00232  C      END OF RUN FILE. CLOSE ANS FILE ON OUTPUT TAPE
00232  1170   CALL WRITEANS(STAPE,S,5,0,0,0)
```

```
00241               KODE=0
00243         GO TO 1000
00244   C
00244   C SELECT NUMBER OF LAGS
00244   C
00244   1200        M=M+1
00247   C ALLOWS SELECTION OF KBIG WHEN STARTING IN THE MIDDLE OF A RUN.
00247         IF(SENSE SWITCH 6) 1201,1202          SUPERFLUOUS
00253   1201  PAUSE 77
00255   1202        GO TO(1203,1210,1220,1230,1100),M
00265   1203           KBIG=200
00267                  GO TO 1900
00270   1210           KBIG=40
00272                  GO TO 2004
00273   1220           KBIG=80
00275                  GO TO 2004
00276   1230           KBIG=200
00300                  GO TO 2004
00301   C
00301   CCLEAR SUMMING AREAS IN POWER SPECTRUM SUBROUTINE
00301   C
00301   1900           CALL PWRSPECT(X,R,C,S,KBIG,0,IBIG,0)
00312                  KEY=0
00314   C     PRINT NAME OF CURRENT VARIABLE
00314                  PRINT 190, (VNM(I,IX),I=1,3),KBIG
00343   190            FORMAT(55X,3A8,I8)
00347                  GO TO 2000
00350   C     ------------------------------------------------------------
00350   C READ DATA AND CALCULATE POWER SPECTRUM
00350   C
00350   2000  CALL DATAIN(VTAPE,FRAME,1,KOUNT,1,KEY)
00357         GO TO(2003,2001,2001),KEY
00365   C RESET TAPE TO BEGINNING OF DATA EXCEPT ON LAST PASS
00365   2001  IF(L-4)2002,1200,1200
00372   2002           CALL BACKUP(VTAPE,KEY)
00375                  GO TO 1200
00376   2003  CONTINUE
00376         X=FRAME(IX)-MEANV(IX)
00404   2004  CALL PWRSPECT(X,R,C,S,KBIG,3,IBIG,KEY)
00415         GO TO(2000,3000,3000),KEY
00423   3000  NLIM=KBIG+1
00426         IF(SENSE SWITCH 5)3020,3010
00432   3010  CALL PRINTPSP(S,IX,KBIG,IBIG)
00437         CALL PRINTACV(C,IX,KBIG,IBIG)
00444   3020  CALL WRITEANS(STAPE,S,1,IX,KBIG,IBIG)
00453         CALL WRITEANS(STAPE,C,2,IX,KBIG,IBIG)
00462         GO TO 1200
00463   C     ------------------------------------------------------------
00463   9100  PRINT 910
00466   910   FORMAT(1X,30HREQUESTED VARIABLE NOT ON TAPE /1X,14HJOB TERMINATED)
00506   9110  DO 9109 J=1,JBIG
00510            JJ=CHNO(J)
00513            PRINT 911,(VNM(I,JJ),I=1,3),CHNO(J)
00545   911      FORMAT(1X,3A8,2X,I2)
00552   9109  CONTINUE
00557   C     ------------------------------------------------------------
00557   9000  REWIND VTAPE
00561         REWIND STAPE
00563         CALL SELECT(3072,4)
00566         CALL SELECT(3072,4)
00571         END
```

```
SUBPROGRAMS

  LEADER       DUMP         OPENFILE     RUNFILE      WRITEANS     SELECT
  VARBLE       PWRSPECT     DATAIN       BACKUP       PRINTPSP     PRINTACV

PROGRAM ALLOCATION

  00613    I        00614    STAPE    00615    VTAPE    00616    KEY
  00617    KODE     00620    LAMP     00621    L        00622    M
  00623    X        00625    IX       00626    KBIG     00627    IBIG
  00630    NLIM     00631    J        00632    JJ

  00633    R        01621    C        02607    S        03575    FRAME

COMMON ALLOCATION

  00000    RNM      00010    CHNO     00025    VNM      00143    MEANV
  00175    FACT     00227    JBIG     00230    KOUNT    00231    DELT

PROGRAM END
```

DICROSCOPES


PAUSE 4095 7777
OPENFILE
KODE=1
RUNFILE
TEST LAMP
End of Tape =1
=2 Run Found
TEST SS6
ON
OFF
REWIND ONE FILE MARK
WRITE RUN PARAMETER FILE
L=0
L=L+1 M=0
TEST SS6
ON
OFF
PAUSE 7
SELECT VARIABLE AND NUMBER OF LAGS AS GIVEN ON CARD
TEST L WAS PREVIOUS ONE THE LAST VARIABLE
YES
NO
WIND ON ONE FILE MARK KODE=0
CALL VARBLE FOR SPECIFIED VARIABLE
IS REQUESTED VARIABLE ON TAPE?
YES
NO
PRINT 'REQUESTED VARIABLE NOT ON TAPE, JOB TERMINATED' FOLLOWED BY LIST OF VARIABLES PRESENT
REWIND VTAPE REWIND STAPE
PAGE FEED
END
M=M+1
TEST M
=1
=5
=2,3,4
KBIG=200
CLEAR R(K) KEY=0 PRINT NAME OF VARIABLE
CALL DATAIN
TEST KEY
=1
=2,3
DEFINE X FOR PWRSPECT
TEST L
<4
=4
BACKUP VTAPE
CALL PWRSPECT
TEST KEY
=1
=2,3
NLIM=KBIG+1
TEST SS5
OFF
ON
PRINT POWER SPECTRUM AND AUTOCOVARIANCE
WRITE POWER SPECTRUM AND AUTOCOVARIANCE
WRITE END OF FILE AND REWIND OVER IT

PRINTER OUTPUT DURING PROCESSING BY 'DICROSCOPES'

The example given is that which was output when processing was to be resumed with the ICC variable for run Cambridge T2.

The setting up of the program is described in the DICROSCOPES section. In order to start with the ICC variable, SS 6 was put ON and a card loaded in the reader specifying the initial value of 'L', in this case 3 (corresponding to the ICC variable).

The first page of the printout is a listing of the output tape label. This is produced by the subroutine OPENFILE. The input tape label and all the runs on tape up to and including Cambridge T2 are then listed by subroutine RUNFILE. Control is then returned to the main program. The name of the current variable is written together with the initial number of lags used for calculation of the spectra. Having SS 5 ON has suppressed printing of the power spectrum and autocovariance functions. SS 5 was, at one stage, switched OFF and the autocovariance of the ICC variable for Cambridge T2 using 80 lags was printed out. When SS 5 was again switched ON (this was probably done during the printing of the 80 lag autocovariance because printing is only suppressed after the complete function has been printed out), the program continued to print the run names and the variables as they were reached during processing. The end of the data tape was reached after run Cambridge T3, and END OF TAPE was printed.

```
    OUTPUT TAPE LABEL ON HANDLER   2

 POWER SPECTRUM ANALYSIS OF HAWORTH CROSS COUPLING DATA
  END OF TAPE LABEL
```

```
        INPUT TAPE LABEL ON HANDLER    1

THIS TAPE CONTAINS CAMBRIDGE DATA TAPE NUMBER ONE RUNS A1 TO T3
REVISED BY PROGRAM TO MODIFY FORMAT OF EDITED TAPES AS AT 18 AUGUST ≠67
BELONGING TO RICHARD HAWORTH
 0
                     CAMBRIDGE A1 1510-1535 257 66
                     CAMBRIDGE A2 1155-1220 261 66
                     CAMBRIDGE B1 2020-2050 258 66
                     CAMBRIDGE B2 1705-1735 260 66
                     CAMBRIDGE C1 0310-0830 261 66
                     CAMBRIDGE D1 1125-1210 241 66
                     CAMBRIDGE D2 0710-0723 242 66
                     CAMBRIDGE E1 0315-0340 239 66
                     CAMBRIDGE E2 0305-0340 242 66
                     CAMBRIDGE F1 1610-1630 241 66
                     CAMBRIDGE F2 1925-1945 241 66
                     CAMBRIDGE F3 0650-0710 239 66
                     CAMBRIDGE G1 0045-0145 243 66
                     CAMBRIDGE G2 0410-0430 243 66
                     CAMBRIDGE H1 1900-1935 243 66
                     CAMBRIDGE H2 1105-1130 245 66
                     CAMBRIDGE J1 0815-0840 246 66
                     CAMBRIDGE J2 1700-1730 243 66
                     CAMBRIDGE K1 0310-0325 245 66
                     CAMBRIDGE K2 0810-0827 245 66
                     CAMBRIDGE K3 0305-0330 248 66
                     CAMBRIDGE L1 1635-1655 247 66
                     CAMBRIDGE L2 0750-0810 253 66
                     CAMBRIDGE L3 0345-0405 257 66
                     CAMBRIDGE M1 0635-0700 248 66
                     CAMBRIDGE M2 0330-0355 253 66
                     CAMBRIDGE M3 1000-1010 257 66
                     CAMBRIDGE N1 2345-0006 242 66
                     CAMBRIDGE N2 1920-1940 257 66
                     CAMBRIDGE N3 2215-2235 260 66
                     CAMBRIDGE P1 2250-2315 239 66
                     CAMBRIDGE P2 0240-0311 240 66
                     CAMBRIDGE Q1 1000-1030 244 66
                     CAMBRIDGE R1 0625-0650 240 66
                     CAMBRIDGE R2 0120-0155 241 66
                     CAMBRIDGE S1 0420-0445 247 66
                     CAMBRIDGE S2 1225-1300 247 66
                     CAMBRIDGE T1 0030-0050 258 66
                     CAMBRIDGE T2 0335-0355 258 66
           FOUND RUN CAMBRIDGE T2 0335-0355 258 66      2390
```

```
     AUTOCOVARIANCE FOR CAMBRIDGE T2 0335-0355 258 66
                          CAMBRIDGE ICC                              PAGE  1

RECORD DURATION    =1194.50 SECONDS            TOTAL SAMPLES      = 2389
AVERAGING INTERVAL=  40.00 SECONDS             MAXIMUM LAGS       =   80
SAMPLING INTERVAL = 0.5000 SECONDS             DEGREES OF FREEDOM=    59


     INTERVAL        TIME    AUTOCOVAR

            0      -0.000     16.88161
            1       0.500     12.21482
            2       1.000      3.79273
            3       1.500     -0.62068
            4       2.000      0.69827
            5       2.500      3.87764
            6       3.000      5.15223
            7       3.500      4.28829
            8       4.000      2.95985
            9       4.500      1.76081
           10       5.000      0.19362
           11       5.500     -1.55774
           12       6.000     -2.20537
           13       6.500     -0.96111
           14       7.000      1.18661
           15       7.500      2.26239
           16       8.000      1.36164
           17       8.500     -0.45650
           18       9.000     -1.45378
           19       9.500     -0.98275
           20      10.000      0.07129
           21      10.500      0.52524
           22      11.000      0.10565
           23      11.500     -0.45300
           24      12.000     -0.49873
           25      12.500     -0.10448
           26      13.000      0.23037
           27      13.500      0.11881
           28      14.000     -0.27181
           29      14.500     -0.46916
           30      15.000     -0.30679
           31      15.500     -0.02307
           32      16.000      0.10010
           33      16.500      0.00150
           34      17.000     -0.11627
           35      17.500     -0.04292
           36      18.000      0.23161
           37      18.500      0.42965
           38      19.000      0.37701
           39      19.500      0.15742
           40      20.000     -0.01283
           41      20.500      0.00213
           42      21.000      0.12690
           43      21.500      0.21932
           44      22.000      0.22043
           45      22.500      0.15016
           46      23.000      0.02322
           47      23.500     -0.16713
           48      24.000     -0.37141
           49      24.500     -0.42993
           50      25.000     -0.27733
```

```
AUTOCOVARIANCE FOR CAMBRIDGE T2 0335-0355 258 66
                   CAMBRIDGE ICC                        PAGE  2


INTERVAL        TIME    AUTOCOVAR

       51       25.500        -0.06035
       52       26.000        -0.00219
       53       26.500        -0.11789
       54       27.000        -0.24817
       55       27.500        -0.28539
       56       28.000        -0.23329
       57       28.500        -0.15066
       58       29.000        -0.06765
       59       29.500         0.00097
       60       30.000         0.01577
       61       30.500        -0.02767
       62       31.000        -0.06753
       63       31.500        -0.07730
       64       32.000        -0.06602
       65       32.500        -0.05648
       66       33.000        -0.04572
       67       33.500        -0.03704
       68       34.000        -0.03177
       69       34.500        -0.03925
       70       35.000        -0.05922
       71       35.500        -0.07431
       72       36.000        -0.06378
       73       36.500        -0.02907
       74       37.000        -0.00047
       75       37.500        -0.00595
       76       38.000        -0.03268
       77       38.500        -0.04602
       78       39.000        -0.03303
       79       39.500        -0.01740
       80       40.000        -0.02265
                                                  CAMBRIDGE FCC           200
            CAMBRIDGE T3 0740-0800 258 66
FOUND RUN CAMBRIDGE T3 0740-0800 258 66             2390
                                                  CAMBRIDGE BEAM MOTION  200
                                                  CAMBRIDGE HORIZ ACCEL  200
                                                  CAMBRIDGE ICC           200
                                                  CAMBRIDGE FCC           200
            END OF TAPE
```

Subroutine: COMPARE(X,Y,LBIG,LITE)

I. PURPOSE:

To compare two alphanumeric fields X and Y of lengths LBIG. If the fields are equal LITE = 1 upon return to the main program and LITE = 2 otherwise.

II. OPERATION:

The program reads in each element of the two fields to be compared and compares corresponding pairs of them. Usually one field is specified in the program or on cards read by the program and the other is specified on the tape being read by the program. In this way a specified field can be recognized on a data tape.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

None.

IV. USAGE:

```
        READ 100,(RUNAME(N),N=1,8)
  100   FORMAT(8X,8A4)
  200   READ TAPE VTAPE,RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT,
        CALL COMPARE(RNM,RUNAME,8,LITE)
        GO TO(2000,1000),LITE
 C      RUN NAMES DO NOT CORRESPOND TRY NEXT RUN
  1000  CALL SELECT(VTAPE,11)
        CALL SELECT(VTAPE,11)
        GO TO 200
 C      RUN NAMES CORRESPOND, PROCEED
  2000  PRINT 2100,(RUNAME(N),N=1,8)
  2100  FORMAT(11X,10HFOUND RUN 8A4)
        etc.
```

V. STORAGE REQUIREMENTS:

61_8 49_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

None

```
 00004          SUBROUTINE COMPARE(X,Y,LBIG,LITE)
 00004   C      SUBROUTINE TO COMPARE TWO ALPHA FIELDS OF LENGTH =LBIG=.
 00004   C      SUBROUTINE RETURNS THE VALUE =1= IN =LITE= IF THE FIELDS ARE EQUAL
 00004          INTEGER X,Y
 00004          DIMENSION X(1),Y(1)
 00004          M=0
 00006          DO 9 L=1,LBIG
 00010   9      M=X(L)-Y(L)+M
 00032          IF(M)10,11,10
 00036   10     LITE=2
 00040          GO TO 9999
 00041   11     LITE=1
 00043   9999   RETURN
 00045          END

 PROGRAM ALLOCATION

 00056     COMPARE    00060      M           00061      L

 PROGRAM END
```

<u>Subroutine: OPENFILE(STAPE,VTAPE,KEY)</u>

I. PURPOSE:

1. To assign logical tape handler numbers to input and output tapes.

2. To read and list input tape label.

3. To write tape label on new output tape, or if already labelled, to read and list output tape label and position tape to begin writing after last runfile already on tape.

II. OPERATION:

1. <u>Tape Specification Card</u> (as in MVD Compensation)

Subroutine reads a card with a numeral (1 to 8) in columns 30 and 60. The numeral appearing in column 30 is assigned to STAPE (the output tape) and the numeral in column 60 is assigned to VTAPE (the input tape).

2. <u>Tape Label Cards</u>

If the word NEW appears in card columns 1, 2, 3 of the Tape Specification Card, the program assumes that a new output tape is mounted and writes a tape label on it. The label is contained on punched cards immediately following the Tape Specification Card, and consists of any number of 80 column cards terminated by an end of file card (7/8 in column 1).

3. If the word NEW does not appear in columns 1, 2, and 3 of the Tape Specification Card, the program will list the label on the output tape and then advance the tape until a runfile with the name END OF TAPE is found. The tape will then be positioned so that the next runfile (that is the heading and run parameters) will be written over the END OF TAPE record.

4. After the output tape has been positioned the program lists the tape label appearing on the input tape.

5. The subroutine then sets KEY = 3 and returns control to the main program sequence.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

None

IV. USAGE:

CALL OPENFILE(OUT,IN,KEY)

Load the reader with cards as follows:

NEW OUTPUT TAPE ON HANDLER 2 INPUT TAPE ON HANDLER 1

(Tape label cards)

TAPE NUMBER ONE

BELONGING TO MARINE GEOPHYSICS

CONTAINS CROSS COUPLING INFORMATION

$ END OF TAPE LABEL

If NEW appears on the Tape Specification Card, tape label cards must be included.

If NEW does not appear on the Tape Specification Card, tape label cards must not be included.

V. STORAGE REQUIREMENTS:

554_8 364_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

SELECT EOFCK COMPARE EJECT

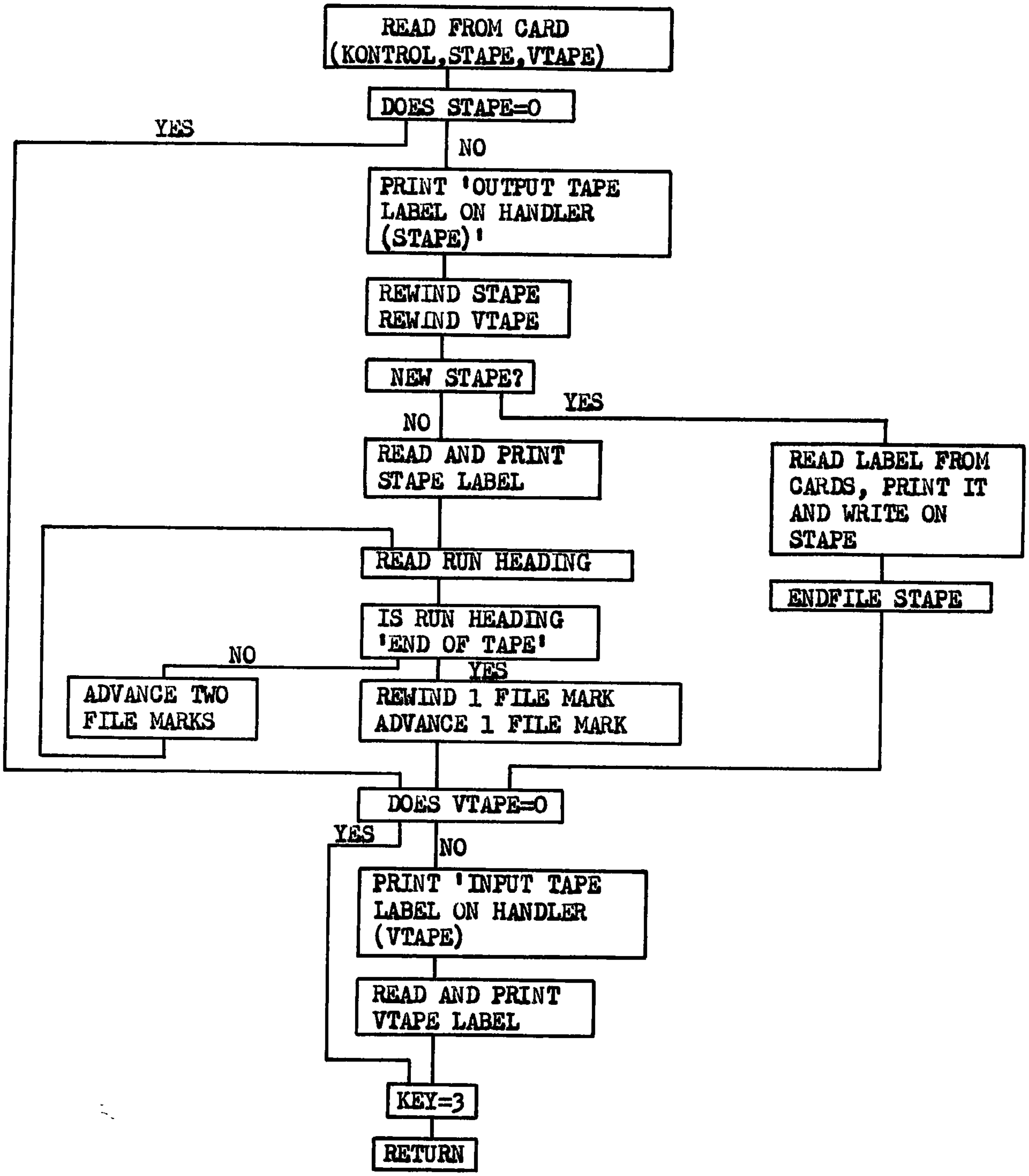
```
00004          SUBROUTINE OPENFILE(STAPE,VTAPE,KEY)
00004   C      SUBROUTINE TO READ TAPE ASSIGNMENT CARD AND TO OPEN THE TAPE
00004   C      FILES. IF =NEW= IS SPECIFIED ON ASSIGNMENT CARD THE TAPE LABEL
00004   C      WILL BE READ FROM CARDS, PRINTED AND WRITTEN ON THE OUTPUT TAPE,
00004   C      OTHERWISE THE TAPE LABEL WILL BE READ FROM THE OLD OUTPUT TAPE,
00004   C      PRINTED AND THE TAPE ADVANCED SO THAT WRITING WILL BEGIN AFTER
00004   C      THE LAST RECORD ALREADY WRITTEN ON THE TAPE.
00004   C      THE INPUT TAPE LABEL WILL BE READ FROM THE TAPE AND PRINTED.
00004   C      ----------------------------------------------------------------
00004   C      COMMON STORAGE
00004   C
00004          COMMON RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004          DIMENSION RNM(8),CHNO(13),VNM(3,13),MEANV(12),FACT(12)
00004          INTEGER RNM,CHNO                        N.B. ABOVE FORMAT
00004          REAL MEANV
00004   C
00004   C      ----------------------------------------------------------------
00004   C      OUTPUT FILES
00004   C                        STAPE FILE 1 TAPE LABEL
00004   C      WRITE OUTPUT TAPE STAPE,300,(LBL(I),I=1,20)
00004   C300   FORMAT(20A4)
00004          INTEGER STAPE
00004          DIMENSION LBL(20)
00004          EQUIVALENCE (LBL(1),FACT(1))
00004   C      ----------------------------------------------------------------
00004   C      INPUT FILES
00004   C
00004   C                        CARD FILE RECORD 1
00004   C
00004   C      READ 100,KONTROL,STAPE,VTAPE
00004          INTEGER VTAPE
00004          INTEGER STAPE
00004   C
00004   C                        CARD FILE RECORD 2
00004   C      READ 300,(LBL(I),I=1,20)
00004   C
00004   C                        STAPE FILE 1 TAPE LABEL
00004   C
00004   C      READ INPUT TAPE STAPE,200,(LBL(I),I=1,20)
00004   C200   FORMAT(20A4)
00004   C
00004   C                        STAPE FILE 2 RECORD1  (     FP WORDS)
00004   C
00004   C       READ TAPE STAPE,RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004          INTEGER TAG,CHNN
00004          DIMENSION IDUM(11)
00004          EQUIVALENCE (CHNO(1),TAG),(CHNO(2),CHNN),(CHNO(3),IDUM(1)),
00004         1(JBIG,NBIG),(KOUNT,IBIG)
00004   C
00004   C                        VTAPE FILE 1 TAPE LABEL
00004   C
00004   C      READ INPUT TAPE VTAPE,111, (LBL(I),I=1,20)
00004   C111   FORMAT(10A8)
00004   C
00004   C      ----------------------------------------------------------------
00004   C      WORKING STORAGE
00004   C
00004          DIMENSION NL(8)
00004   C
00004   C      ALPHANUMERIC CONSTANTS
00004   C
```

```
00004          NEW=152950
00006          NL(1)=5657904
00011          NL(2)=-6722508
00015          NL(3)=4617584
00020          NL(4)=-3994575
00024          NL(5)=-3994575
00030          NL(6)=-3994575
00034          NL(7)=-3994575
00040          NL(8)=-3994575
00044   C
00044   C      ----------------------------------------------------------------
00044   C
00044          READ 100,KONTROL,STAPE,VTAPE
00061   100    FORMAT(R3,26X,I1,29X,I1)
00067   1000   IF(STAPE) 1001,4000,1001
00073   1001   CALL SELECT(3072,4)
00076          PRINT 101, STAPE
00105   101    FORMAT(10X, 29HOUTPUT TAPE LABEL ON HANDLER I3,//)
00121          REWIND STAPE
00123          REWIND VTAPE
00125          IF(KONTROL-NEW) 2000,3000,2000
00132   C
00132   C      NOT A NEW TAPE. READ AND PRINT LABEL FROM TAPE.
00132   C
00132   2000   READ INPUT TAPE STAPE,200,(LBL(I),I=1,20)
00152   200    FORMAT(20A4)
00155          CALL EOFCK(STAPE,LITE)
00162          PRINT 201,(LBL(I),I=1,20)
00201   201    FORMAT(1X,20A4)
00205          GO TO (2101,2000),LITE
00212   2101   READ TAPE STAPE,RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00242          CALL COMPARE(RNM,NL,8,LITE)
00247          GO TO (2103,2102),LITE
00254   C
00254   C      ADVANCE TAPE TO READ NEXT RUN IDENTIFICATION.
00254   C
00254   2102   CALL SELECT(STAPE,11)
00261          CALL SELECT(STAPE,11)
00266          GO TO 2101
00267   C
00267   C      POSITION TAPE TO  WRITE OVER =END OF TAPE= RECORD.
00267   C
00267   2103   CALL SELECT(STAPE,12)
00274          CALL SELECT(STAPE,11)
00301          GO TO 4000
00302   C
00302   C      NEW TAPE.  READ AND PRINT LABEL FROM CARDS AND WRITE ON TAPE
00302   C
00302   3000   READ 200,(LBL(I),I=1,20)
00321          CALL EOFCK(1536,LITE)
00324          PRINT 301,(LBL(I),I=1,20)
00343   301    FORMAT(1X,20A4)
00347          GO TO (3002,3001),LITE
00354   3001   WRITE OUTPUT TAPE STAPE,200,(LBL(I),I=1,20)
00374          GO TO 3000
00375   3002   ENDFILE STAPE
00377          GO TO 4000
00400   C
00400   C      OPEN INPUT TAPE. READ AND PRINT LABEL.
00400   C
00400   4000   IF(VTAPE) 4001,9999,4001
00404   4001   CALL EJECT
```

```
00405          PRINT 400, VTAPE
00414   400    FORMAT(10X, 28HINPUT TAPE LABEL ON HANDLER I3,//)
00430   4002   READ INPUT TAPE VTAPE,200,(LBL(I),I=1,20)
00450          CALL EOFCK(VTAPE,LITE)
00455          PRINT 201,(LBL(I),I=1,20)
00474          GO TO (9999,4002),LITE
00501   C
00501   9999   KEY=3
00503          CALL EJECT
00504          RETURN
00506          END

SUBPROGRAMS

  SELECT     EOFCK      COMPARE    EJECT

PROGRAM ALLOCATION

00537    OPENFILE  00541     NEW        00542     KONTROL   00543     I
00544    LITE

00545    NL

COMMON ALLOCATION

00000    RNM       00010     CHNO       00025     VNM       00143     MEANV
00173    FACT      00223     JBIG       00224     KOUNT     00225     DELT


00173    LBL       00010     TAG        00011     CHNN      00012     IDUM
00223    NBIG      00224     IBIG

PROGRAM END
```

OPENFILE

OPENFILE(STAPE,VTAPE,KEY)


READ FROM CARD
(KONTROL,STAPE,VTAPE)
DOES STAPE=0
YES
NO
PRINT 'OUTPUT TAPE LABEL ON HANDLER (STAPE)'
REWIND STAPE
REWIND VTAPE
NEW STAPE?
YES
NO
READ AND PRINT STAPE LABEL
READ LABEL FROM CARDS, PRINT IT AND WRITE ON STAPE
READ RUN HEADING
ENDFILE STAPE
IS RUN HEADING 'END OF TAPE'
NO
YES
ADVANCE TWO FILE MARKS
REWIND 1 FILE MARK
ADVANCE 1 FILE MARK
DOES VTAPE=0
YES
NO
PRINT 'INPUT TAPE LABEL ON HANDLER (VTAPE)
READ AND PRINT VTAPE LABEL
KEY=3
RETURN

Subroutine: RUNFILE(VTAPE,KEY,KODE,LAMP)

I. PURPOSE:

1. To search a data tape or answer tape for a specified runfile and to position the tape at the end of that runfile.

2. To list the run names of the data contained on the tape until the desired runfile is reached.

II. OPERATION:

1. The function of the subroutine depends on the value of KODE.

   (a) If KODE = 0, no run name is furnished and subroutine assumes that the next runfile in sequence is desired. This is the SEQUENTIAL MODE when every run is examined.

   (b) If KODE ≠ 0 or 8 (it is usually given the value of 1) the subroutine reads a card on which is specified a run name, up to 32 characters in length beginning in column 9. This is the same format as the Run Identification Card in the MVD Compensation Program so that the program can try to match it with what is written on tape.

   (c) If KODE = 8, the subroutine reads a card image identical to the above, from a paper tape reader.

2. For KODE ≠ 0 the subroutine searches the tape mounted on handler VTAPE until the specified run name matches the RNM written in the heading section (record 1) of the runfile. If no match is found before encountering an RNM named END OF TAPE, the tape is rewound, and searched from the beginning once only. This provision ensures that the entire tape will be searched completely. If the end of tape is reached a second time without successfully finding the specified run name, the message REQUESTED RUN (32 character RUN NAME) NOT ON THIS TAPE is printed. The VTAPE is rewound again and the subroutine pauses showing 3744_8. The next run name appearing on card or paper tape may be searched for simply by pressing GO.

3. For an entry to the subroutine with KODE = 0, the subroutine will advance the tape to the next runfile in sequence.

4. When the requested runfile has been located the subroutine has read into common storage all the parameters appearing in record 1 of the runfile preamble and is positioned to read record 2. In normal processing this record is not usually used but it may be read by the main program upon exit from the subroutine or advanced to the next file mark after which the data are written.

5. Upon entry to the subroutine, the value of KEY is tested and the tape is positioned accordingly. If KEY = 1 it indicates that the last runfile had not been completely read and the program pauses showing 1010_8. This is a warning that there is a possible program malfunction, and operation assuming that one file mark has to be passed before reading the next runfile preamble, is resumed by pressing GO.

6. Upon exit from RUNFILE, KEY is set to 0 indicating that we are in the runfile preamble and that one filemark must be passed before the data are reached. LAMP is set to 2.

7. Exit from the subroutine with LAMP = 1 indicates that either the run name specified by card or paper tape consists of all blanks or, in the sequential mode (KODE = 0), the END OF TAPE run name is reached. This may be used to terminate the main program.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

PAUSE 1010_8 0520_{10}	Indicates that entry to subroutine RUNFILE was attempted before completion of the reading of the DATA FILE OR ANSWER FILE. Press GO to continue.
PAUSE 3744_8 2020_{10}	Indicates that a specified run cannot be found on the data tape and an error message is printed. Press GO to search the tape for the next run name supplied on card or paper tape.

IV. USAGE:

```
     KODE = 1
1000 CALL RUNFILE(INTAPE,KEY,KODE,LAMP)
     GO TO(999,1001),LAMP
1001 CALL DATAIN(INTAPE,FRAME,1,KOUNT,1,KEY)
     - - - - - - - - - - - - - - - -
     Continue processing data
     - - - - - - - - - - - - - - - -
C    RETURN TO PROCESS NEXT RUNFILE IN SEQUENCE
     KODE = 0
     GO TO 1000
9999 REWIND INTAPE
     END
```

V. STORAGE REQUIREMENTS:

552_8 362_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

SELECT COMPARE EJECT

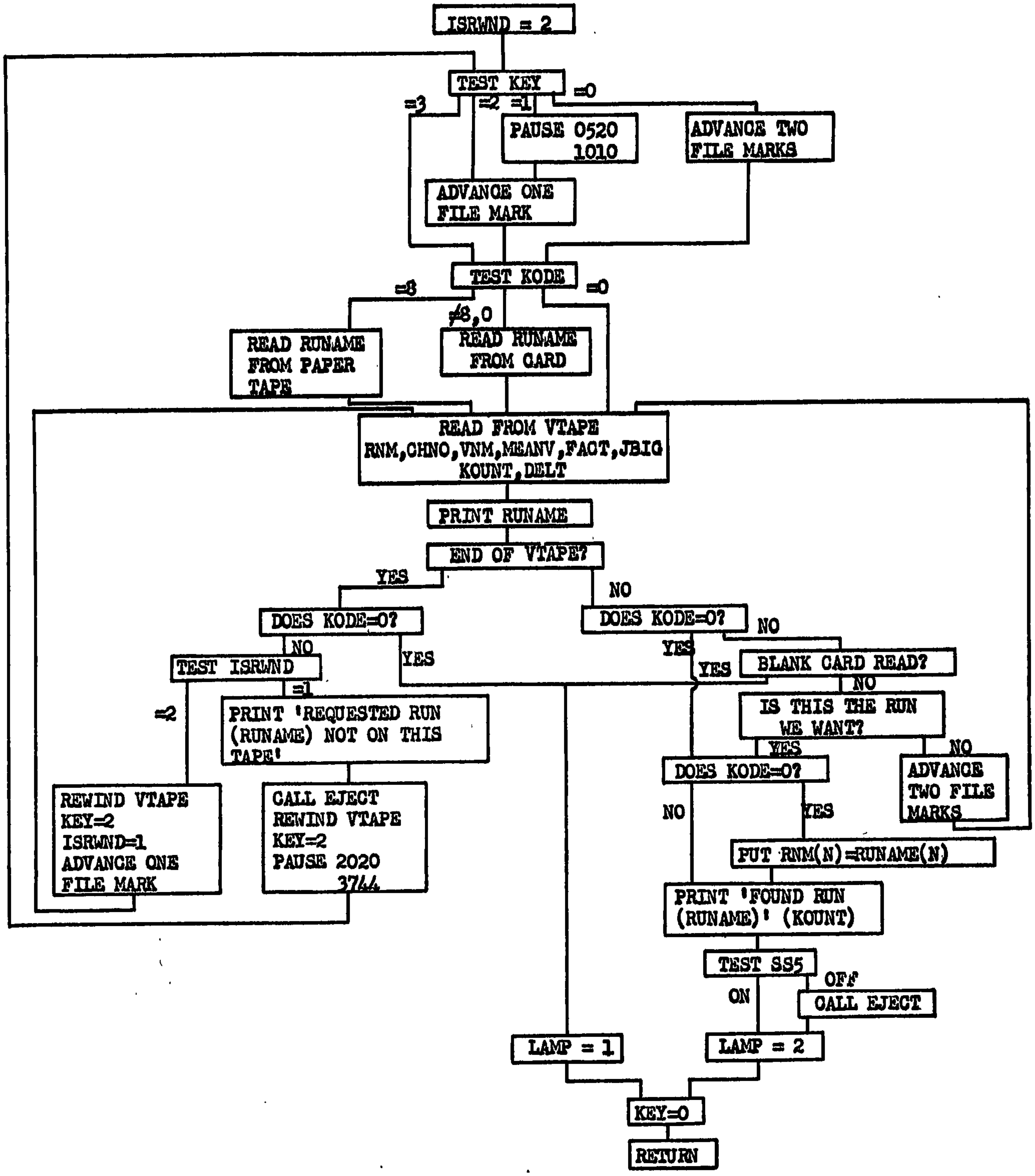
```
00004           SUBROUTINE RUNFILE(VTAPE,KEY,KODE,LAMP)
00004   C       SUBROUTINE TO SEARCH TAPE FOR SPECIFIED  RUNFILE.
00004   C       RUN NAME IS READ FROM A CARD OR FROM PAPER TAPE BY THIS ROUTINE.
00004   C       ADVANCE PAST END OF FILE DEPENDING ON =KEY=
00004   C       KEY=0  INDICATES THAT TAPE IS STILL WITHIN THE RUN HEADING FILE.
00004   C       TAPE MUST BE ADVANCED PAST TWO FILE MARKS IN ORDER TO READ THE
00004   C       NEXT RUN HEADING.
00004   C       KEY=1  INDICATES THAT DATA READING WAS NOT COMPLETE IN SUBROUTINE
00004   C       DATAIN.  PROGRAM HALTS SHOWING =~~2020~~=.  PRESS =GO= TO ADVANCE TO
                                                1010
00004   C       NEXT RUNFILE.
00004   C       KEY =2  INDICATES THAT DATA READING WAS COMPLETE IN SUBROUTINE
00004   C       DATAIN BUT TAPE HAS NOT ADVANCED TO END OF FILE..
00004   C       KEY=3.  INDICATES THAT TAPE IS CORRECTLY POSITIONED AT THE
00004   C       BEGINNING OF A RUN HEADING FILE.
00004   C       NORMAL EXIT FROM SUBROUTINE =RUNFILE= LEAVES TAPE IN POSITION TO
00004   C       READ NEXT RECORD IN FILE 1 OF RUNFILE- IF ANY-, BUT TAPE MUST
00004   C       BE ADVANCED PAST END OF FILE MARK BEFORE DATA READING CAN COMMENCE
00004   C       THIS OPERATION IS PERFORMED BY SUBROUTINE =DATAIN=.
00004   C
00004   C       ----------------------------------------------------------------
00004   C       COMMON STORAGE
00004   C
00004           COMMON RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004   C       ----------------------------------------------------------------
00004   C       INPUT FILES
00004   C
00004   C                        CARD FILE
00004   C
00004   C       READ 100,KONTROL,(RUNAME(N),N=1,8)
00004   C100    FORMAT(R3,5X,8A4)
00004   C100    FORMAT(R3,5X,8A4)
00004           DIMENSION RUNAME(8)
00004   C       READ 100,KONTROL,(RUNAME(N),N=1,8)
00004           INTEGER RUNAME
00004   C
00004   C                        VTAPE FILE 2 RECORD1  (     FP WORDS)
00004   C
00004   C        READ TAPE VTAPE,RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004           DIMENSION RNM(8),CHNO(13),VNM(3,13),MEANV(12),FACT(12)
00004           INTEGER VTAPE                               N.B. above format
00004           INTEGER RNM,CHNO
00004           REAL MEANV
00004   C       ----------------------------------------------------------------
00004   C       WORKING STORAGE
00004   C
00004           NEW=152950
00006           DIMENSION NL(8)
00006           NL(1)=5657904
00011           NL(2)=-6722508
00015           NL(3)=4617584
00020           NL(4)=-3994575
00024           NL(5)=-3994575
00030           NL(6)=-3994575
00034           NL(7)=-3994575
00040           NL(8)=-3994575
00044           BLANK=-3994575
00051              ISRWND=2
00053   C
00053   C       ----------------------------------------------------------------
00053   C
00053            IF(KEY)1000,1002,1000
```

```
00057  1000  GO TO (1001,1003,2000),KEY
00065  1001  PAUSE 0520
00067        GO TO 1003
00070  1002  CALL SELECT(VTAPE,11)
00075  1003  CALL SELECT(VTAPE,11)
00102        GO TO 2000
00103  C
00103  C     SELECT SEQUENTIAL OR SEARCH MODE AND INPUT UNIT
00103  C     DEPENDING ON = KODE =     KODE=0   SEQUENTIAL MODE,
00103  C     KODE=8   SEARCH FOR RUN SPECIFIED ON PAPER TAPE, OTHERWISE
00103  C     SEARCH FOR RUN SPECIFIED ON CARD.
00103  C
00103  2000  IF(KODE)2001,2300,2001
00107  2001  IF(KODE-8)2100,2200,2100
00114  2100  READ 210,(RUNAME(N),N=1,8)
00133  210   FORMAT(8X,8A4)
00136        GO TO 2300
00137  2200  READ INPUT TAPE 8,210,(RUNAME(N),N=1,8)
00157        GO TO 2300
00160  C
00160  C     READ RECORD 1 FROM RUN FILE AND PRINT
00160  C
00160  2300  READ TAPE VTAPE,RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00210        PRINT 230,(RNM(N),N=1,8)
00227  230   FORMAT(21X,8A4)
00233  C     TEST FOR END OF TAPE
00233        CALL COMPARE(NL,RNM,8,LITE)
00240        GO TO(3000,2310),LITE
00245  C     SUCCESSFUL SEARCH IN SEQUENTIAL MODE IF NOT END OF TAPE
00245  2310  IF(KODE)2400,9000,2400
00251  C     SET SIGNAL TO TERMINATE PROGRAM WHEN BLANK CARD IS FOUND.
00251  2400  CALL COMPARE(NL(4),RUNAME,2,LITE)
00261        GO TO(9100,2500),LITE
00266  C
00266  C     IN SEARCH MODE TEST TO SEE IF RUN NAMES CORRESPOND.  IF NOT
00266  C     DESIRED RUN, ADVANCE TO BEGINNING OF NEXT RUNFILE AND CONTINUE.
00266  C
00266  2500  CALL COMPARE(RNM,RUNAME,8,LITE)
00273        GO TO(9000,2600),LITE
00300  2600  CALL SELECT(VTAPE,11)
00305        CALL SELECT(VTAPE,11)
00312        GO TO 2300
00313  C
00313  C     TERMINATE PROGRAM AT END OF DATA IN SEQUENTIAL MODE.  IN SEARCH
00313  C     MODE, REWIND TAPE ONCE AND CONTINUE SEARCH UNTIL TAPE HAS BEEN
00313  C     COMPLETLY SEARCHED ONCE.  THEN PRINT ERROR MESSAGE AND HALT.
00313  C     PRESS =GO= AFTER HALT TO SEARCH FOR NEXT RUN ON CARDS OR PAPER TPE
00313  C
00313  3000  IF(KODE)3100,9100,3100
00317  3100  GO TO(3200,3300),ISRWND
00324  3200  PRINT 320,(RUNAME(N),N=1,8)
00343  320   FORMAT(7X,14HREQUESTED RUN 8A4,16HNOT ON THIS TAPE)
00360        CALL EJECT
00361        REWIND VTAPE
00363        KEY=2
00365        PAUSE 2020
00367  C     PRESS=GO= TO ADVANCE TAPE PAST TAPE LABEL AND SEARCH NEXT RUN
00367        GO TO 1000
00370  C     REWIND TAPE AND SEARCH ONCE FROM BEGINNING
00370  3300  REWIND VTAPE
00372        KEY=2
00374        ISRWND=1
```

```
00376          CALL SELECT(VTAPE,II)
00403          GO TO 2300
00404    C     SUCCESSFUL SEARCH, READY WITH NEXT RUN
00404   9000   IF(KODE)9010,9001,9010
00410   9001   DO 9009 N=1,8
00412             RUNAME(N)=RNM(N)
00415   9009   CONTINUE
00422   9010   PRINT 901,(RUNAME(N),N=1,8),KOUNT
00443   901    FORMAT(11X,10HFOUND RUN 8A4,I8)
00453          IF(SENSE SWITCH 5)9012,9011
00457   9011   CALL EJECT
00460   9012   LAMP=2
00462          GO TO 9999
00463    C     JOB TERMINATED BY BLANK CARD OR END OF TAPE IN
00463    C     SEQUENTIAL MODE
00463   9100   LAMP=1
00465   9999   KEY=0
00467          RETURN
00471          END

SUBPROGRAMS

  SELECT     COMPARE    EJECT

PROGRAM ALLOCATION

 00523    RUNFILE    00525    NEW        00526    BLANK      00530    ISRWND
 00531    N          00532    LITE

 00533    RUNAME     00543    NL

COMMON ALLOCATION

 00000    RNM        00010    CHNO       00025    VNM        00143    MEANV
 00173    EACT       00223    JBIG       00224    KOUNT      00225    DELT

PROGRAM END
```

RUNFILE

RUNFILE(VTAPE,KEY,KODE,LAMP)



Subroutine: WRITEANS(NTAPE,S,TAG,CHNN,NBIG,IBIG)

I. PURPOSE:

To write on magnetic tape the power spectrum, autocovariance, cross spectrum or cross covariance.

II. OPERATION:

The operation of the subroutine is determined by the keyword TAG.

TAG = 0 causes the output tape to be prepared for the writing of spectra. The constants referring to the particular run (those carried as the first record of the run heading file on the data tape) are written as a heading, and an end of file mark added.

TAG = 1 or 2 causes the power spectrum or the autocovariance function respectively to be written comprising a heading followed by the function data written in a single data block.

TAG = 3 or 4 causes the cross spectrum or cross covariance function to be written.

TAG = 1,2,3,4. After writing the spectra, the subroutine end files the tape, writes END OF TAPE and the last set of run constants and then positions the tape to write over the end of file mark.

TAG = 5 causes the output tape file to be closed by advancing over the end of file mark written during the passes with TAG = 1,2,3, or 4.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

None

IV. USAGE:

```
 1000 CALL DATAIN(VTAPE,FRAME,1,KOUNT,1,KEY)
C     DEFINE VALUE OF VARIABLE
      - - - - - - - - - - - - - - -
      X = FRAME(IX) - MEANV(IX)
 2000 CALL PWRSPECT(X,R,C,S,KBIG,3,IBIG,KEY)
      GO TO(1000,3000,3000),KEY
C     WRITE POWER SPECTRUM ON MAGNETIC TAPE
 3000 CALL WRITEANS(NTAPE,S,1,IX,NBIG,IBIG)
C     WRITE AUTOCOVARIANCE ON MAGNETIC TAPE
      CALL WRITEANS(NTAPE,C,2,IX,NBIG,IBIG)
C     DEFINE NEXT VARIABLE TO BE PROCESSED
      - - - - - - - - - - - - - - -
      - - - - - - - - - - - - - - -
      GO TO 1000
```

V. STORAGE REQUIREMENTS:

346_8 230_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

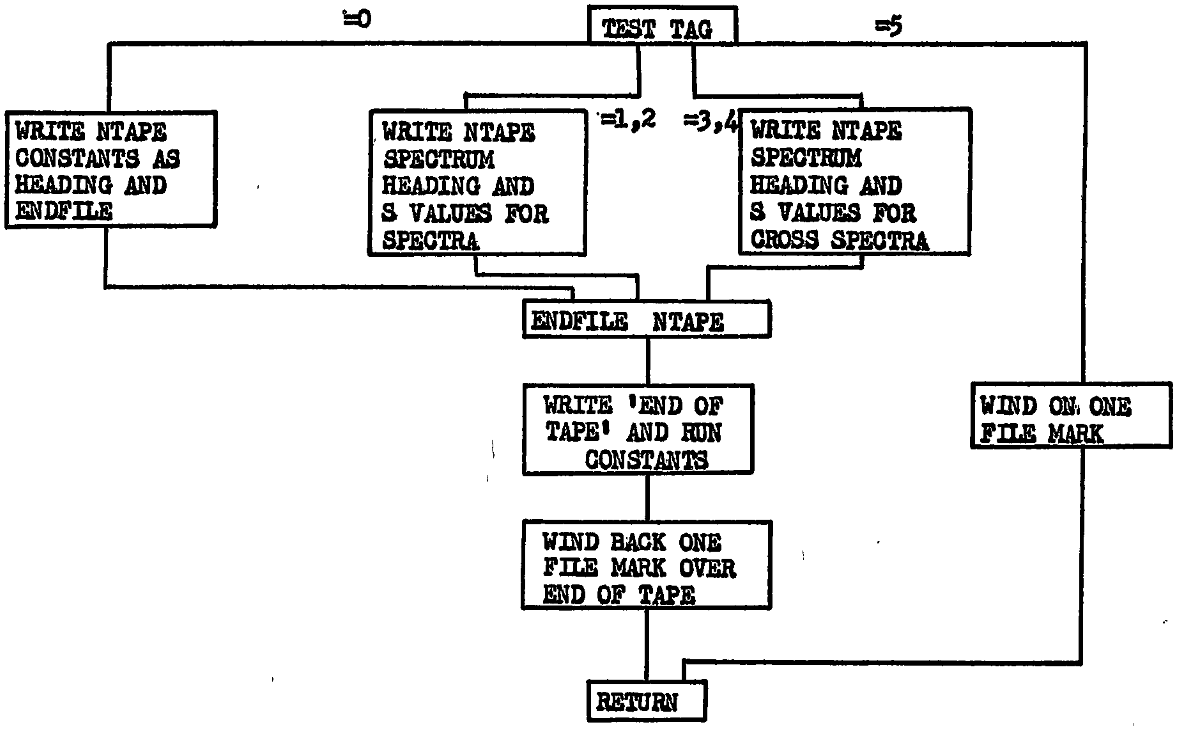
WRITEB SELECT

```
00004          SUBROUTINE WRITEANS(NTAPE,S,TAG,CHNN,NBIG,IBIG)
00004   C      SUBROUTINE TO WRITE POWER SPECTRUM (TAG=1), AUTOCOVARIANCE (TAG=2)
00004   C      CROSS SPECTRUM (TAG=3), OR CROSS COVARIANCE (TAG=4),
00004   C      OPEN ANSWERFILE(TAG=0), CLOSF ANS FILE (TAG=5)
00004   C      ----------------------------------------------------------------
00004   C       COMMON STORAGE
00004   C
00004          COMMON RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004          DIMENSION RNM(8),CHNO(13),VNM(3,13),MEANV(13),FACT(13)
00004          INTEGER RNM,CHNO
00004          REAL MEANV
00004   C      ----------------------------------------------------------------
00004   C      OUTPUT FILE
00004   C
00004   C                      NTAPE FILE 1 TAPE LABEL
00004   C      CALL OPENFILE(NTAPE,0,KEY)
00004          INTEGER NTAPE
00004   C
00004   C                     NTAPE FILE 2 RECORD 1
00004   C       WRITE TAPE NTAPE,RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004   C
00004   C                     NTAPE FILE 2 RECORD 2
00004   C      WRITE TAPE NTAPE,NL,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004             DIMENSION NL(8)
00004          NL(1)=5657904
00007          NL(2)=-6722508
00013          NL(3)=4617584
00016          NL(4)=-3994575
00022          NL(5)=-3994575
00026          NL(6)=-3994575
00032          NL(7)=-3994575
00036          NL(8)=-3994575
00042   C
00042   C                     NTAPE FILE 3 RECORD 1
00042   C      WRITE TAPE NTAPE,TAG,CHNN,VNM,NBIG,IBIG
00042          INTEGER TAG,CHNN
00042   C
00042   C                     NTAPE FILE 3 RECORD 2
00042   C      CALL WRITEB(NTAPE,S,S(NBIG+2))
00042   C
00042   C                     NTAPE FILE 3 RECORD 3
00042   C      CALL WRITEB(NTAPE,S,S(4*(NBIG+1)+1))
00042          DIMENSION S(1)
00042   C
00042   C      ----------------------------------------------------------------
00042   C      WORKING STORAGE
00042   C
00042   C      ----------------------------------------------------------------
00042   C
00042   1000   KEE=TAG+1
00045          GO TO (1001,1002,1002,1003,1003,1009),KEE
00056   C                      WRITE FILE 2 RECORD 1
00056   1001    WRITE TAPE NTAPE,RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00106          ENDFILE NTAPE
00110          GO TO 1004
00111   C                     WRITE FILE 3 RECORD 1
00111   1002   WRITE TAPE NTAPE,TAG,CHNN,VNM,NBIG,IBIG
00137   C                     WRITE FILE 3 RECORD 2
00137          CALL WRITEB(NTAPE,S,S(NBIG+2))
00157          GO TO 1004
00160   C                     WRITE FILE 3 RECORD 1
```

```
00160   1003   WRITE TAPE NTAPE,TAG,CHNN,VNM,NBIG,IBIG
00206   C                    WRITE FILE 3 RECORD 3
00206          LIM=4*(NBIG+1)+1
00213          CALL WRITEB(NTAPE,S,S(LIM))
00233          GO TO 1004
00234   1004   ENDFILE NTAPE
00236   C                    WRITE FILE 1 RECORD 2
00236          WRITE TAPE NTAPE,NL,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00266          CALL SELECT(NTAPE,12)
00273          GO TO 9999
00274   1009   CALL SELECT(NTAPE,11)
00301   9999   RETURN
00303          END

SUBPROGRAMS

  WRITEB      SELECT

PROGRAM ALLOCATION

 00333    WRITEANS  00335    KEE        00336    LIM

 00337    NL

COMMON ALLOCATION

 00000    RNM       00010    CHNO       00025    VNM        00143    MEANV
 00175    FACT      00227    JBIG       00230    KOUNT      00231    DELT

PROGRAM END
```

WRITEANS

WRITEANS(NTAPE,S,TAG,CHNN,NBIG,IBIG)


TEST TAG
=0
=5
=1,2
=3,4
WRITE NTAPE CONSTANTS AS HEADING AND ENDFILE
WRITE NTAPE SPECTRUM HEADING AND S VALUES FOR SPECTRA
WRITE NTAPE SPECTRUM HEADING AND S VALUES FOR CROSS SPECTRA
ENDFILE NTAPE
WRITE 'END OF TAPE' AND RUN CONSTANTS
WIND BACK ONE FILE MARK OVER END OF TAPE
WIND ON ONE FILE MARK
RETURN

TAG = 1	signifies	POWER SPECTRUM
TAG = 2	signifies	AUTOCOVARIANCE
TAG = 3	signifies	CROSS SPECTRUM
TAG = 4	signifies	CROSS COVARIANCE
TAG = 0	signifies	OPEN ANSWERFILE
TAG = 5	signifies	CLOSE ANSWERFILE

Subroutine: VARBLE(X,VNM,IX)

I. PURPOSE:

To establish the relation between the Fortran variable names "X" and the variable name appearing in the FORMAT statement which has to accompany the CALL of this subroutine, and also between the variables named on the tape (array VNM) and the sequence number of that variable in the sampled data record (as given in array CHNO).

II. OPERATION:

The subroutine searches the array VNM (which defines the names of variables contained on the data tape) for an exact match with the 24 characters appearing in the FORMAT statement which follows the CALL VARBLE instruction. When a match is found, the corresponding sample sequence number (appearing in array CHNO) is inserted into the word IX. Thus the word IX contains the subscript value necessary to retrieve sample values of the named variable. The named variable will subsequently be referred to by its Fortran name X. The name X appears in the subroutine entry vector for purposes of documentation only.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

In the event that no match between the FORMAT name and the variables defined in VNM is found, the value of IX will be set to zero. This can be used in the main program to generate an error halt.

IV. USAGE:

It should be noted in the following that:

(a) The FORMAT statement is unnumbered.

(b) No space is allowed between the T of FORMAT and the left parenthesis.

(c) Exactly 24 characters are required between H and the right parenthesis.

(d) The subroutine is written in BAP.

```
1000 CALL VARBLE (X,VNM,IY)
     FORMAT(24HCAMBRIDGE BEAM MOTION    )
     IF(IY) 2000,1001,2000
C    ERROR CONDITION.  NAMES DO NOT MATCH.
1001 PRINT 101
101  FORMAT(10X,30HREQUESTED VARIABLE NOT ON TAPE)
     DO 1009 J=1,JBIG
     JJ=CHNO(J)
     PRINT 102 (VNM(I,JJ),I=1,3),CHNO(J)
```

```
 102  FORMAT(12X,3A8,2X,I2)
 1009 CONTINUE
      CALL EJECT
      GO TO 9999
C     VARIABLE FOUND READ IN DATA
 2000 CALL DATAIN(INTAPE,FRAME,1,1000,1,KEY)
      Y=FRAME(IY)-MEAN(IY)
      etc.
```

V. STORAGE REQUIREMENTS:

63_8 51_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

None

VII. EXAMPLE OF OPERATION OF PROGRAM:

Since the program is written in BAP and no printout is available, an example of the operation of the program is instructive.

Let there be seven variables defined on the data tape, their names being simply VARIABLE ONE, VARIABLE TWO and so on. Suppose that the sample values of the variables are in the following sequence on the data tape.

VARIABLE NAME (VNM)	CHANNEL NUMBER (CHNN)
VARIABLE ONE	12
VARIABLE TWO	1
VARIABLE THREE	2
VARIABLE FOUR	3
VARIABLE FIVE	7
VARIABLE SIX	8
VARIABLE SEVEN	10

On another data tape the same seven variables might be organized in a different sequence, depending upon the telemetry channels assigned on a particular day during digitizing. Thus we might have:

VARIABLE ONE	1
VARIABLE TWO	2
VARIABLE THREE	3
VARIABLE FOUR	7
VARIABLE FIVE	8
VARIABLE SIX	9
VARIABLE SEVEN	10

In the programs we would like to process the variables by name regardless of the sequential location of the sample values in the data or the order of naming the variables in the array VNM. The subroutine VARBLE is intended to provide this linkage.

While any number of variables may be defined in a given program, suppose that the analysis is concerned with only two variables and we wish to call them by the Fortran names of simply X and Y. The instructions for doing this are:

```
CALL VARBLE(X,VNM,IX)
FORMAT(24HVARIABLE ONE            )
CALL VARBLE(Y,VNM,IY)
FORMAT(24HVARIABLE TWO            )
```

When a name match has been found, the subroutine places the corresponding channel sequence number (from array CHNO) into the location IX or IY. If no match is found then 0 is entered, and may be used to generate an error halt in the main program.

The data are read by DATAIN one sample at a time for all possible variables. The sample values are placed in an array FRAME. In order to process the named variables VARIABLE ONE and VARIABLE TWO the programmer writes:

```
X=FRAME(IX)
Y=FRAME(IY)
```

Consequently, even though the order of the variable samples varies from one data tape to the next, the program will always process VARIABLE ONE and VARIABLE TWO.

Subroutine: PWRSPECT(X,R,C,S,KBIG,FILTER,IBIG,KEY)

I. PURPOSE:

To develop the power spectrum, S, of a set of data, X.

II. OPERATION:

The subroutine calculates the lagged products, R, from a series of data, X, and when all X have been read in formulates the autocovariance function, C. Various filters are available depending upon the value given to FILTER and the filtered autocovariance function is transformed by the subroutine COSTR to give the power spectrum, S.

Passage through the subroutine is controlled by the content of the word KEY. The first passage through the subroutine is with KEY set to 0 in the main program and this results in the previously stored values of Z(K) and R(K) being cleared and the definitions of program terms being restated. Subsequent passages are controlled by automatic adjustment of KEY in the main program or by a subroutine such as DATAIN. With KEY=1 data may be being read in by DATAIN and this data used by subroutine PWRSPECT to accumulate the sums of lagged products.

A single variable value is read in by DATAIN during each pass and the subroutine calculates a new set of R(K) and Z(K). When all the data for a single variable have been read in by DATAIN, the final R(K) represents the summation:

$$R(K) = \sum_{i=1}^{IBIG-(K-1)} X_i \cdot X_{i+(K-1)}$$

which are the sums of products for all lags, K.

Upon passage through the subroutine with KEY=2 or 3, usually respectively referring to the case where data reading within a data block has been completed as specified and to the case where the end of the data has been reached, the autocovariances are calculated for all lag values. These are then modified by the filters available, which are called by their name FILTER. The Bartlett, Hanning and Hamming filters are available but provision is made for the inclusion of further examples.

Having been modified, the function is then cosine transformed using the subroutine COSTR before returning to the main program.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

None

IV. USAGE:

```
C     CLEAR R(K) AND Z(K) STORAGE AREAS IN PWRSPECT SUBRT.
      CALL PWRSPECT(X,R,C,S,KBIG,0,IBIG,0)
      KEY=0
 1000 CALL DATAIN(INTAPE,FRAME,I1ST,ILAST,INCR,KEY)
      GO TO(1200,1400,1400),KEY
 1200 X=FRAME(IX)-MEANV(IX)
 1400 CALL PWRSPECT(X,R,C,S,KBIG,3,IBIG,KEY)
      GO TO(1000,9000,9000) KEY
C     PRINT OUT POWER SPECTRUM
 9000 CALL PRINTPSP(S,IX,KBIG,IBIG)
      etc.
```

V. STORAGE REQUIREMENTS:

1453_8 811_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

COS COSTR

Subroutine: PWRSPECT

EXPANSION OF SERIES USED TO CALCULATE SUMS OF PRODUCTS

Take as example K=KLIM=5 i.e. 4 lags
I=256 (or anything else)

DATAIN is taken to be reading in a series X_1, X_2, X_3, X_4, etc.
Working through the steps of the series we have:

J=4 $R(5) = Z(5).Z(1) + R(5)$
$= Z(5).X_1 = 0$ since R(K) and Z(K) have been cleared
$Z(5) = Z(4) = 0$ on a previous run through the 1st
$R(4) = Z(4).Z(1) + R(4)$ of the subprogram.
$= 0$
$Z(4) = Z(3) = 0$
$R(3) = Z(3).Z(1) + R(3) = 0$
$Z(3) = Z(2) = 0$
$R(2) = Z(2).Z(1) + R(2) = 0$
$Z(2) = Z(1) = X_1$
$R(1) = Z(1).Z(1) + R(1) = X_1^2$
$Z(1) = X_1$

On the second pass $Z(1) = X_2$

$R(5) = Z(5).Z(1) + R(5) = 0$ $Z(5) = Z(4) = 0$
$R(4) = Z(4).Z(1) + R(4) = 0$ $Z(4) = Z(3) = 0$
$R(3) = Z(3).Z(1) + R(3) = 0$ $Z(3) = Z(2) = X_1$
$R(2) = Z(2).Z(1) + R(2)$ $Z(2) = Z(1) = X_2$
$= X_1.X_2 + 0 = X_1.X_2$
$R(1) = Z(1).Z(1) + R(1)$ $Z(1) = X_2$
$= X_2^2 + X_1^2$

The third pass yields $Z(1) = X_3$

$R(3) = X_1.X_3$
$R(2) = X_1.X_2 + X_2.X_3$
$R(1) = X_1^2 + X_2^2 + X_3^2$

This will in general yield:

$$R(1) = \sum_{i=1}^{I} X_i^2 \qquad R(2) = \sum_{i=1}^{I} X_i \cdot X_{i-1}$$

$$R(3) = \sum_{i=1}^{I} X_i \cdot X_{i-2} \qquad R(4) = \sum_{i=1}^{I} X_i \cdot X_{i-3}$$

$$R(n) = \sum_{i=N}^{I} X_i \cdot X_{i-(N-1)}$$

OR

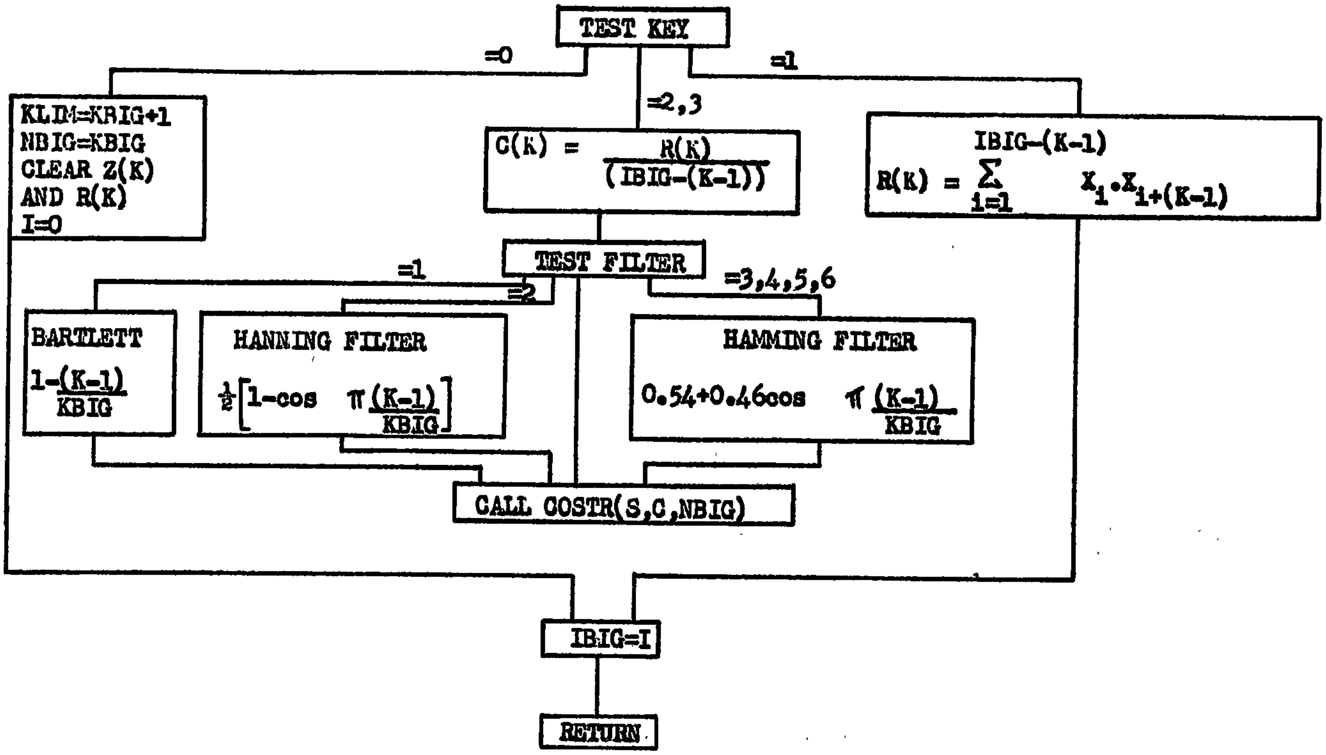
$$R(n) = \sum_{i=1}^{I-N+1} X_i \cdot X_{i+N-1}$$

```
00004          SUBROUTINE PWRSPECT(X,R,C,S,KBIG,FILTER,IBIG,KEY)
00004   C      SUBROUTINE TO CALCULATE POWER SPECTRUM
00004          COMMON RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004          DIMENSION R(1),C(1),S(1),Z(251)
00004          INTEGER FILTER
00004   C      ------------------------------------------------------------
00004          KEE=KEY+1
00007          GO TO (1000,1100,2000,2000),KEE
00016   1000   KLIM=KBIG+1
00021          NBIG=KBIG
00023          PI=3.1415927
00025          DO 1009 K=1,KLIM
00027             Z(K)=0
00035             R(K)=0
00046   1009   CONTINUE
00053          I=0
00055          GO TO 9999
00056   C      CALCULATE SUMS OF PRODUCTS
00056   C
00056   1100   K=KLIM
00060          I=I+1
00063          Z(I)=X
00070   1101   J=K-1
00073   1102      R(K)=Z(K)*Z(I)+R(K)
00116             Z(K)=Z(J)
00126          K=K-1
00131          IF(K-1)9999,1102,1101
00136   C      ------------------------------------------------------------
00136   C      CALCULATE COVARIANCES
00136   C
00136   2000   T=I
00141          KLIM=KBIG+1
00144          NBIG=KBIG
00146          DO 2019 K=1,KLIM
00150             C(K)=R(K)/T
00164             T=T-1
00172   2019   CONTINUE
00177          GO TO 3000
00200   C      ------------------------------------------------------------
00200   C      CALCULATE SMOOTHING CONVOLUTION
00200   C      ONLY FILTERS 0,1,2, AND 3 ARE IMPLEMENTED
00200   C
00200   3000   IF (FILTER) 3001,4000,3001
00204   3001   GO TO(3100,3200,3300,3400,3500,3600),FILTER
00215   3100   DO 3109 K=1,KLIM
00217             C(K)=(1-(K-1)/KBIG)*C(K)
00250   3109   CONTINUE
00255          GO TO 4000
00256   3200   DO 3209 K=1,KLIM
00260             C(K)=0.5*(1-COS(PI*(K-1)/KBIG))*C(K)
00322   3209   CONTINUE
00327          GO TO 4000
00330   3300   DO 3399 K=1,KLIM
00332             C(K)=(0.54+0.46*COS(PI*(K-1)/KBIG))*C(K)
00367   3399   CONTINUE
00374          GO TO 4000
00375   3400   GO TO 3300
00376   3500   GO TO 3300
00377   3600   GO TO 3300
00400   C      ------------------------------------------------------------
00400   C      CALCULATE POWER SPECTRUM AND ADJUST END POINTS
```

```
 00400   C
 00400   4000   CALL COSTR(S,C,NBIG)
 00410          IBIG=1
 00412   C
 00412   9999   IBIG=1
 00414          RETURN
 00416          END

 SUBPROGRAMS

   COS          COSTR

 PROGRAM ALLOCATION

  00452      PWRSPECT   00454      KEE        00455      KLIM       00456      NBIG
  00457      PI         00461      K          00462      I          00463      J
  00464      T

  00466      Z

 COMMON ALLOCATION

  00000      RNM        00002      CHNO       00004      VNM        00006      MEANV
  00007      FACT       00011      JBIG       00012      KOUNT      00013      DELT

 PROGRAM END
```

PWRSPECT

PWRSPECT(X,R,C,S,KBIG,FILTER,IBIG,KEY)


TEST KEY
=0
=2,3
=1
KLIM=KBIG+1
NBIG=KBIG
CLEAR Z(K)
AND R(K)
I=0
$C(K) = \frac{R(K)}{(IBIG-(K-1))}$
$R(K) = \sum_{i=1}^{IBIG-(K-1)} X_i \cdot X_{i+(K-1)}$
TEST FILTER
=1
=2
=3,4,5,6
BARTLETT
$1-\frac{(K-1)}{KBIG}$
HANNING FILTER
$\frac{1}{2}\left[1-\cos \pi\frac{(K-1)}{KBIG}\right]$
HAMMING FILTER
$0.54+0.46\cos \pi\frac{(K-1)}{KBIG}$
CALL COSTR(S,C,NBIG)
IBIG=I
RETURN

Subroutine: COSTR(Y,X,M)

I. PURPOSE:

To calculate the cosine transform,Y, of the input function, X, where M is the number of lags for which each is specified.

II. OPERATION:

The calculation is performed according to an algorithm.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

None

IV. USAGE:

```
C     DEFINE C
      - - - - -
      - - - - -
C     NO OF LAGS IS NBIG
      CALL COSTR(S,C,NBIG)
C     RETURN TO MAIN PROGRAM WITH S AS COSINE
C     TRANSFORM OF C.
      RETURN
```

V. STORAGE REQUIREMENTS:

245_8 165_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

COS

```
00004          SUBROUTINE COSTR(Y,X,M)
00004   C      SUBROUTINE TO CALCULATE COSINE TRANSFORM
00004          DIMENSION Y(1), X(1)
00004          MO=M+1
00007          FM=M
00012          DO 1 K=1, MO
00014          FK=K-1
00020          S=X(1)
00030          VO=0.
00032          V1=1.
00034             A=COS((FK*3.1415927)/FM)
00043          B=2.*A
00046           DO2L=2,M
00050          V2=B*V1-VO
00054          CO =A*V1-VO
00060          VO=V1
00062          V1=V2
00064     2    S=S+(X(L)*2.)*CO
00106          S=S+X(MO)*COS (FK*3.1415927)
00124       1  Y(K)=S/FM
00142          Y(1)=Y(1)*.5
00156          Y(MO)=Y(MO)*.5
00172          RETURN
00174          END

SUBPROGRAMS

  COS

PROGRAM ALLOCATION

00217   COSTR     00221   MO        00222   FM        00224   K
00225   FK        00227   S         00231   VO        00233   V1
00235   A         00237   B         00241   L         00242   V2
00244   CO

PROGRAM END
```

Subroutine: DATAIN(JACK,FRAME,I1ST,ILAST,INCR,KEY)

I. PURPOSE:

To load the sample values of up to 12 variables into an array FRAME where the initial and final frame numbers to be processed, and the processing interval are specified by the subroutine.

II. OPERATION:

A single passage through the subroutine results in one data block being read into the array FRAME. It is used in a DO loop in the main program to read in a sequence of data blocks.

JACK specifies the number of the tape handler which holds the data tape. It is assumed that the data tape has been positioned to the beginning of the sampled data by use of the subroutine RUNFILE and subsequent SELECT(JACK,11), and that the correspondence between the variables named in the tape heading and the program variables has been established by use of the subroutine VARBLE.

FRAME is the name of an array FRAME(J),J=1,13 which will be loaded by the subroutine DATAIN to contain sample values of the 12 variables plus the sequence or frame number of the sample.

I1ST and ILAST specify the first and last frame numbers respectively for which sampled values are desired.

INCR specifies the increment in frame number between samples presented for processing by the program.

KEY is a control word which is under control of the subroutine and which has the value 1 upon return to the main program sequence so long as the frame number of the sample processed lies within the interval specified by I1ST and ILAST. When the frame number is larger than ILAST, KEY is set equal to 2 indicating that the end of the specified data has been reached. When an end of file mark is encountered before frame number ILAST is reached KEY is set to 3. The value of KEY may be used in the main program or subroutine PWRSPECT to indicate when all data have been read in, and that calculation of spectra should be started.

The frame number is made available upon return from the subroutine DATAIN and may be used for control purposes. The frame number is specified as FRAME(13) and setting KOUNT=FRAME(13) in the main program retrieves the frame number.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

PAUSE 2222_8 1170_{10} is non-recoverable and is reached if the subroutine is entered with KEY=3 i.e. the complete data file has been read in and we are at the beginning of the next runfile.

IV. USAGE:

```
     100  N=0
          CALL VARBLE(Y,VNM,IX)
          FORMAT(24HWAVE HEIGHT AT MIDDAY    )
          IF(IX) 1000.110.1000
   C      ERROR CONDITION FOR NON CORRESPONDENCE OF NAMES
     110  - - - - - - - - - - - - - - - -
          PRINT ERROR CONDITION
          - - - - - - - - - - - - - - - -
    1000  CALL DATAIN(JACK,FRAME,I1ST,ILAST,INCR,KEY)
          GO TO(1100,1200,2000) KEY
    1100  X=FRAME(IX)
          KOUNT=FRAME(13)
          N=N+1
          SUMX=SUMX+X
          GO TO 1000
   C      REACHED LAST FRAME OF SPECIFIED DATA SECTION
    1200  CALL SELECT(JACK,11)
          GO TO 2300
   C      END OF DATA FILE REACHED BEFORE ILAST
    2000  PRINT 2100
    2100  FORMAT(10X,31HSPECIFIED DATA BLOCK INCOMPLETE)
          PRINT 2200,KOUNT
    2200  FORMAT(10X,40HFRAME NUMBER OF LAST SAMPLE PROCESSED IS,
          15I)
    2300  MEANX=SUMX/N
          CALL SELECT(JACK,11)
   C      CONTINUE PROCESSING FOR MEAN OF NEXT DATA BLOCK
          GO TO 100
          etc.
```

V. STORAGE REQUIREMENTS:

672_8 442_{10}

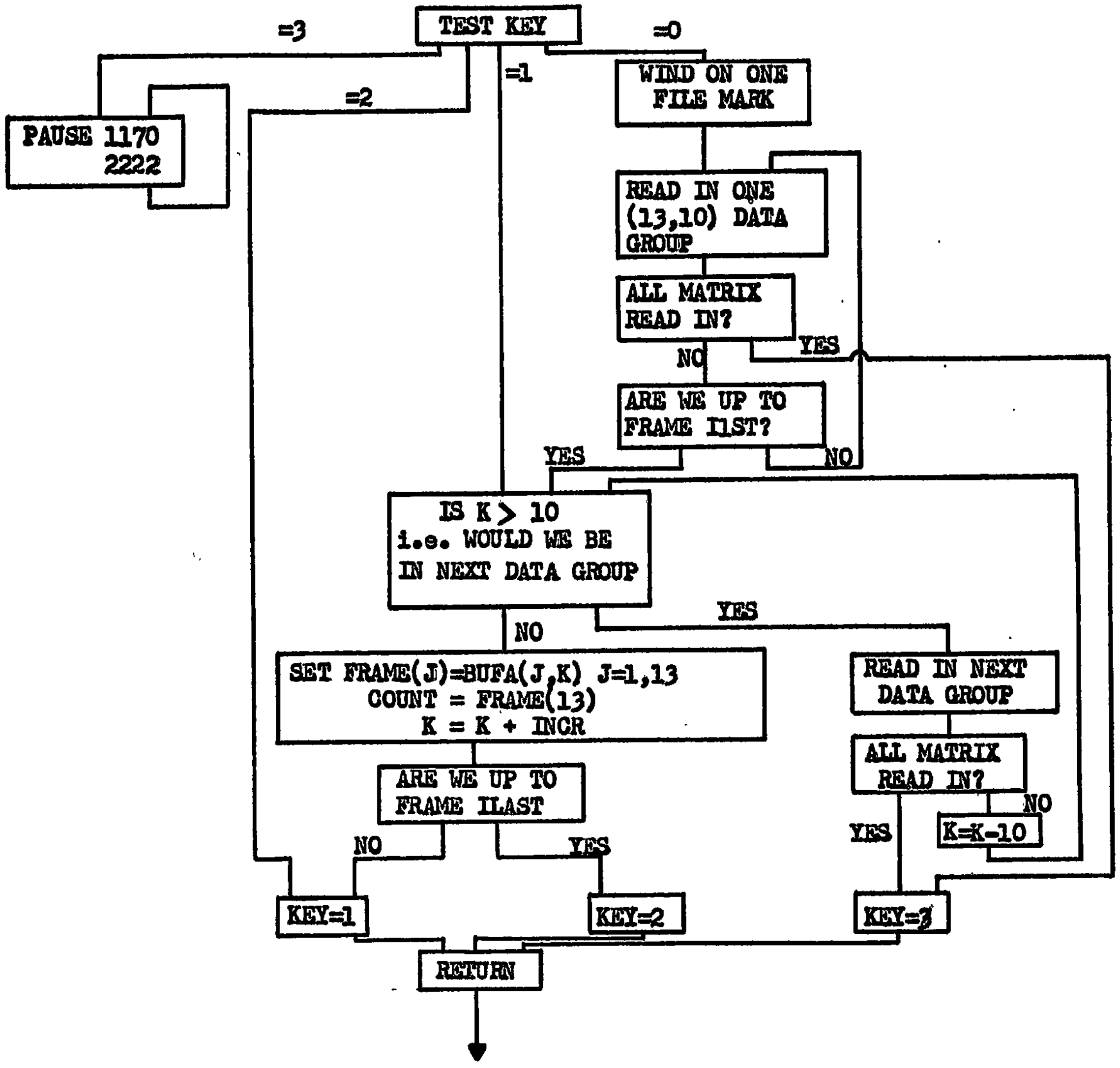
VI. SUBROUTINES AND FUNCTIONS CALLED:

SELECT READB EOFCK

```
 00004            SUBROUTINE DATAIN(JACK,FRAME,IIST,ILAST,INCR,KEY)
 00004  C         SUBROUTINE TO READ TIME SERIES DATA
 00004  C         27 MARCH 1967
 00004            DIMENSION FRAME(13),BUFA(13,10)
 00004  C         ----------------------------------------------------------
 00004            KEE=KEY+1
 00007            GO TO (1000,2000,9100,9400),KEE
 00016  C         ----------------------------------------------------------
 00016  1000      CALL SELECT(JACK,II)
 00023  1001      CALL READB(JACK,BUFA,BUFA(14,10))
 00042            CALL EOFCK (JACK,LITE)
 00047            GO TO (9300,1002),LITE
 00054  1002      DO 1009 K=1,10
 00056            COUNT=BUFA(13,K)
 00067               IF(COUNT-IIST) 1009,2000,2000
 00077  1009      CONTINUE
 00104            GO TO 1001
 00105  C         ----------------------------------------------------------
 00105  2000      IF(K-10) 2100,2100,2001
 00112  2001      CALL READB(JACK,BUFA,BUFA(14,10))
 00131            CALL EOFCK(JACK,LITE)
 00136            GO TO (9300,2002),LITE
 00143  2002      K=K-10
 00146            GO TO 2000
 00147  C         ----------------------------------------------------------
 00147  2100      DO 2109 J=1,13
 00151               FRAME(J)=BUFA(J,K)
 00170  2109      CONTINUE
 00175            COUNT=FRAME(13)
 00205            K=K+INCR
 00210            IF(COUNT-ILAST) 9100,9200,9200
 00220  C         ----------------------------------------------------------
 00220  9100      KEY=1
 00222            GO TO 9999
 00223  9200      KEY=2
 00225            GO TO 9999
 00226  9300      KEY=3
 00230            GO TO 9999
 00231  9400      PAUSE 1170
 00233            GO TO 9400
 00234  9999      RETURN
 00236            END

 SUBPROGRAMS

   SELECT     READB      EOFCK

 PROGRAM ALLOCATION

 00257    DATAIN    00261    KEE      00262    LITE      00263    K
 00264    COUNT     00266    J

 00267    BUFA

 PROGRAM END
```

DATAIN

DATAIN(JACK,FRAME,I1ST,ILAST,INCR,KEY)



Subroutine: PRINTACV(R,CHNN,KBIG,IBIG)
PRINTPSP(S,CHNN,KBIG,IBIG)

I. PURPOSE:

To print the values for the autocovariance function as calculated by the program DICROSCOPES. PRINTPSP is a similar program for printing the power spectra.

II. OPERATION:

The values R(N) are calculated by DICROSCOPES at times TIME(N) and, after a heading has been written, these values are tabulated. 50 values are written per page and the pages are numbered sequentially, reverting to 1 each time a new function is to be written.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

None

IV. USAGE:

```
C       CALCULATE R(N) WITH THE PROGRAM DICROSCOPES
        - - - - - - - - - - - - - - - -
        CALL PRINTACV(R,CHNN,KBIG,IBIG)
C       CONTINUE
```

V. STORAGE REQUIREMENTS:

535_8 349_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

SELECT

```
00004           SUBROUTINE PRINTACV(R,CHNN,KBIG,IBIG)
00004  C        SUBROUTINE TO PRINT AUTO-COVARIANCE FUNCTION
00004  C
00004           COMMON RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004           DIMENSION RNM(8),CHNO(13),VNM(3,13),MEANV(13),FACT(13)
00004           INTEGER RNM,CHNO
00004           REAL MEANV
00004           DIMENSION R(1)
00004           INTEGER CHNN,DF
00004           NLIM=KBIG+1
00007           T=IBIG
00012           TI=DELT*T
00015           TN=DELT*KBIG
00021           DF=(2*T-(KBIG/3))/KBIG
00043           IPG=1
00045           CALL SELECT(3072,4)
00050   1000    PRINT 100,(RNM(I),I=1,8),(VNM(N,CHNN),N=1,3),IPG
00113   100         FORMAT(10X,19HAUTOCOVARIANCE FOR 8A4,/,29X,3A8,13X,5HPAGE I2,
00113          1    /)
00134           PRINT 101,TI,IBIG
00145   101         FORMAT(5X,19HRECORD DURATION   =,F7.2,8H SECONDS10X,
00145          1    19HTOTAL SAMPLES     =,I5)
00174           PRINT 102,TN,KBIG
00205   102         FORMAT(5X,19HAVERAGING INTERVAL=,F7.2,8H SECONDS10X,
00205          1    19HMAXIMUM LAGS      =,I5)
00234           PRINT 103,DELT,DF
00243   103         FORMAT(5X,19HSAMPLING INTERVAL =,F7.4,8H SECONDS10X,
00243          1    19HDEGREES OF FREEDOM=I5//)
00272           PRINT 104
00275   104     FORMAT(10X,30HINTERVAL       TIME   AUTOCOVAR/)
00310           LCNT=0
00312   2000    DO 2009 N=1,NLIM
00314               IF(LCNT-50) 2002,2002,2001
00321   2001            IPG=IPG+1
00324                   CALL SELECT(3072,4)
00327                   PRINT 100,(RNM(I),I=1,8),(VNM(K,CHNN),K=1,3),IPG
00372                   CALL SELECT(3072,2 )
00375                   CALL SELECT(3072,2 )
00400                   CALL SELECT(3072,1 )
00403                   PRINT 104
00406                   LCNT=1
00410   2002        NM=N-1
00413               TIME=(N-1)*DELT
00425               PRINT 200,NM,TIME,R(N)
00446   200         FORMAT(15X,I3,4X,F7.3,2X,F11.5)
00456               LCNT=LCNT+1
00461   2009    CONTINUE
00466   9999    RETURN
00470           END


 SUBPROGRAMS

   SELECT

 PROGRAM ALLOCATION

 00514    PRINTACV   00516    NLIM     00517    T        00521    TI
 00523    TN         00525    DF       00526    IPG      00527    I
 00530    N          00531    LCNT     00532    K        00533    NM
 00534    TIME
```

```
COMMON ALLOCATION

 00000     RNM         00010     CHNO        00025     VNM         00143     MEANV
 00175     FACT        00227     JBIG        00230     KOUNT       00231     DELT

PROGRAM END
```

Subroutine: BACKUP(VTAPE,KEY)

I. PURPOSE:

To rewind a magnetic tape as determined by the value of KEY.

II. OPERATION:

Upon entry to the subroutine, the value of KEY is tested.

KEY = 1 causes a pause 2020_8 to be reached, indicating that the tape is positioned in the middle of the data file and that possibly the data reading is incomplete. On pressing GO, the tape is rewound across one file mark to position it at the end of the runfile heading. Upon entry to DATAIN, the tape is wound over a file mark before data reading is commenced.

KEY = 2 is the normal indicator that data reading is complete, and the tape is positioned within the data file. The tape is rewound over one file mark to position it ready for DATAIN.

KEY = 3 is the output value from subroutine DATAIN when the end of the data file is reached before the completion of data reading. Thus the tape is positioned in the next runfile. The rewind effected by this subroutine over two file marks, positions the tape ready for DATAIN.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

PAUSE 2020_8 1040_{10} is indicated when backing up the tape is attempted during data reading. Pressing GO causes a rewind to the beginning of the data as indicated in II (KEY=1) above.

IV. USAGE:

```
 1000  CALL DATAIN(VTAPE,FRAME,1,KOUNT,1,KEY)
       GO TO(2000,3000,3000),KEY
 2000  - - - - - - - - - - - - - - -
C      PROCESS DATA
       - - - - - - - - - - - - - - -
 3000  CALL BACKUP(VTAPE,KEY)
 4000  - - - - - - - - - - - - - - - - - - - - - - - -
C      DEFINE NEXT VARIABLE TO BE SAMPLED BY DATAIN
       - - - - - - - - - - - - - - - - - - - - - - - -
       GO TO 1000
```

V. STORAGE REQUIREMENTS:

37_8 31_{10}

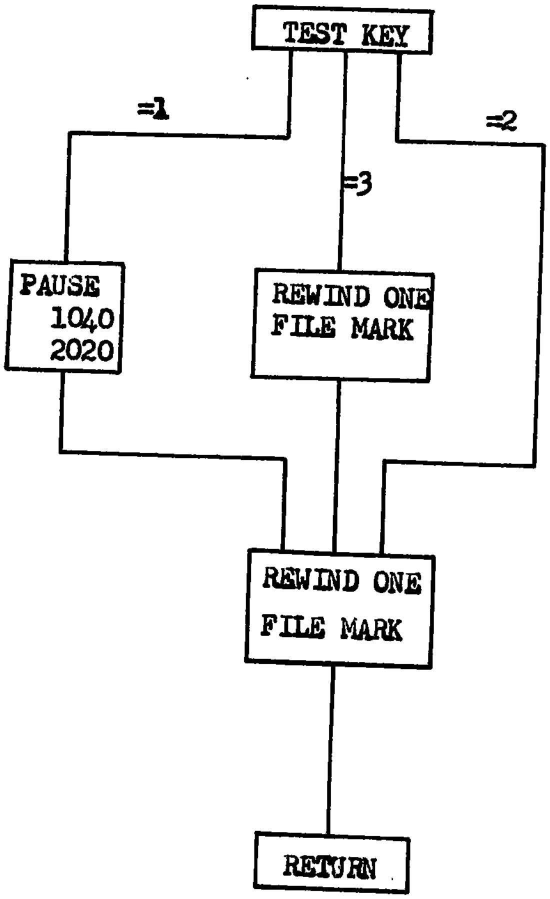
VI. SUBROUTINES AND FUNCTIONS CALLED:

SELECT

```
00004          SUBROUTINE BACKUP(VTAPE,KEY)
00004   C      SUBROUTINE TO BACKSPACE INPUT TAPE TO BEGINNING OF DATA IN CURRENT
00004   C      RUN FILE.  PAUSE 2020 INDICATES THAT BACKSPACING HAS BEEN
00004   C      ATTEMPTED BEFORE COMPLETING DATA READING ON PRESENT PASS)
00004   C      PRESS =GO= TO ACCOMPLISH BACKSPACE.
00004   C
00004          GO TO(1001,1003,1002),KEY
00012   1001   PAUSE 1040
00014          GO TO 1003
00015   1002   CALL SELECT(VTAPE,12)
00022   C-     CAUSES BACKSPACE OVER FILE MARK AT END OF DATA IF THIS POINT HAS
00022   C-     BEEN PASSED
00022   1003   CALL SELECT(VTAPE,12)
00027   C-     CAUSES TAPE TO SEARCH BACKWARD AND STOP AHEAD OF FILE MARK AT THE
00027   C-     BEGINNING OF DATA FILE.
00027   C      SUBROUTINE DATAIN ASSUMES FILE MARK MUST BE PASSED BEFORE READING
00027   9999   RETURN
00031          END

  SUBPROGRAMS

     SELECT

  PROGRAM ALLOCATION

   00037     BACKUP

  PROGRAM END
```

BACKUP

BACKUP(VTAPE,KEY)



Program: PRINTOUT

I. PURPOSE:

This program is really obsolete since its searching function is provided by ANSWERIN, but it is appended here for completeness. The program prints the spectrum or autocovariance for a variable, run, and number of lags as specified on cards.

II. OPERATION:

Runfile searches the input tape for a run specified as in the instructions for RUNFILE. Having found the run, the program reads a card specifying the variable, number of lags and whether the power spectrum or autocovariance is required, and prints the required information before reaching a pause 7775_8. On pressing GO, the input tape is rewound over two file marks i.e. to the beginning of the present run, and the next run as specified on cards is searched for.

When the input tape is exhausted, it is rewound to pause on 7773_8.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

PAUSE 7777_8 4095_{10}	Initial halt after compiling program to allow for loading of input tape and cards.
PAUSE 7775_8 4093_{10}	Pause after printing spectrum to allow setting up of cards for next spectrum required. GO instigates backspacing prior to search.
PAUSE 7773_8 4091_{10}	Input tape exhausted and rewound. Load new input tape and cards.

IV. USAGE:

Compile program
Pause 7777_8

Load input tape on handler 1
Load card reader with

col 30
INPUT TAPE ON HANDLER 1
Card specifying run name of spectrum to be examined, as required by RUNFILE.
Cards specifying spectrum of variable required in the format:

(CAMBRIDGE)	BEAM MOTION	1 0	TAG:	
	HORIZ ACCEL	1 2	POWERSPECTRUM 1	NO of LAGS 4 0
	ICC	1 1	AUTOCOVARIANCE 2	or 8 0
	FCC	6		or 2 0 0
(DOMINION)	BEAM MOTION	9		
	HORIZ ACCEL	7		
	ICC	5		
	FCC	4		

Press GO
Pause 7775 after spectrum printed. Repeat above.

V. STORAGE REQUIREMENTS:

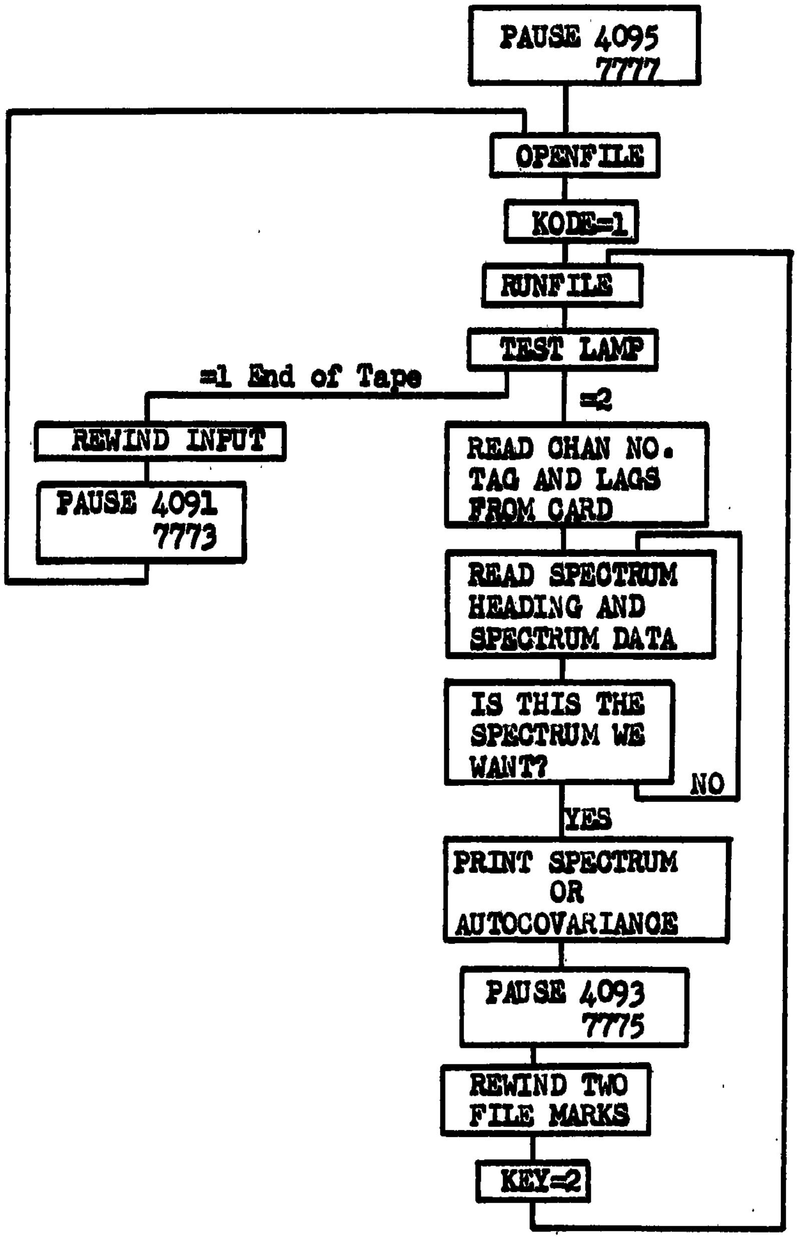
2206_8 1158_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

OPENFILE RUNFILE SELECT READB PRINTPSP PRINTACV

```
00000  C      PROGRAM PRINTOUT
00000  C      LIST POWER SPECTRA AND AUTOCOVARIANCE FROM MAGNETIC TAPE
00000  C      AUGUST 23 1967
00004         DIMENSION RNM(3),CHNO(13),VNM(3,13),MEANV(13),FACT(13),S(251),
00004        1C(251)
00004         COMMON RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004         INTEGER RNM,CHNO,CHNN,DF,TAG,TAGWANT
00004         REAL MEANV
00004         PAUSE 4095
00006  10     CALL OPENFILE(0,IN,KEY)
00012         KODE=1
00014  100    CALL RUNFILE(IN,KEY,KODE,LAMP)
00021         GO TO(9999,200),LAMP
00026  200    CALL SELECT(1,11)
00031         READ 400,IXWANT,TAGWANT,KWANT
00042  400    FORMAT(18X,I2,19X,I1,17X,I3)
00051  1000   READ TAPE IN,TAG,IX,VNM,KBIG,IBIG
00067         GO TO(2000,3000),TAG
00074  2000   CALL READB(1,S,S(KBIG+2))
00105         GO TO 4000
00106  3000   LIM=4*(KBIG+1)+1
00113         CALL READB(1,C,C(LIM))
00124  4000   IF(IX-IXWANT)1000,4200,1000
00131  4200   IF(KBIG-KWANT)1000,4400,1000
00136  4400   IF(TAG-TAGWANT)1000,4600,1000
00143  4600   GO TO(5000,6000),TAG
00150  5000   CALL PRINTPSP(S,IX,KBIG,IBIG)
              Go To 7000
00155  6000   CALL PRINTACV(C,IX,KBIG,IBIG)
00162  7000   PAUSE 4093
00164         CALL SELECT(1,12)
00167         CALL SELECT(1,12)
00172         KEY=2
00174         GO TO 100
00175  9999   REWIND IN
00177         PAUSE 4091
00201         GO TO 10
00202         END

SUBPROGRAMS

 OPENFILE   RUNFILE    SELECT     READB      PRINTPSP   PRINTACV

PROGRAM ALLOCATION

 00216   IN        00217   KEY       00220   KODE      00221   LAMP
 00222   IXWANT    00223   TAGWANT   00224   KWANT     00225   TAG
 00226   IX        00227   KBIG      00230   IBIG      00231   LIM

 00232   S         01220   C

COMMON ALLOCATION

 00000   RNM       00010   CHNO      00025   VNM       00143   MEANV
 00175   FACT      00227   JBIG      00230   KOUNT     00231   DELT

PROGRAM END
```

PRINTOUT PROGRAM



PRINTOUT OF POWER SPECTRUM TAPE

The tape to be examined is the magnetic tape, output by the program DICROSCOPES, which is used as the input tape for the program DICROSCOPIC PLOT.

Record 1 contains the tape label, followed by a file mark.

Records 2 to 6 contain the constants comprising the first half of the Runfile for this particular run (Cambridge A1). These records have precisely the same format as in the revised edited MVD tape. The second half of the Runfile is not included.

Records 7 to 9 contain the constants referring to the first function written on tape i.e. the power spectrum for 40 lags for the first variable:

TAG	Record 7	word 1		1 word
CHNN	7	2		1
VNM(3,13)	7	3	to record 9 word 16	78
KBIG	9	17		1
IBIG	9	18		1

The remainder of record 9 is blank.

Record 10 contains the spectrum data in 80 words (plus two for the parity check).

All the data for this first function are thus contained in four records, the first three classifying the function and the last containing the data for that function. All the functions are similarly described in four records, the length of the last record of each group of four depending upon the amount of data and hence upon the number of lags e.g. record 18 contains the data for a function with 80 lags, so that it contains 162 words.

The order of writing the functions on tape is:

1st variable	Power spectrum	40 lags	4 records	184 words
	Autocovariance	40	4	184
	Power spectrum	80	4	264
	Autocovariance	80	4	264
	Power spectrum	200	4	504
	Autocovariance	200	4	504
2nd variable	Power spectrum	40	4	184
	Autocovariance	40	4	184
	Power spectrum	etc.		

When all the functions of all variables have been written, an end of file mark terminates the data for the run.

```
RECORD NO.      1 (BCD) CONTAINING    136 CHARACTERS
       POWER SPECTRUM ANALYSIS OF HAWORTH CROSS COUPLING DATA


END OF FILE
THERE ARE       1 RECORDS CONTAINING       34 WORDS BEFORE FILEMARK NO.
```

```
 RECORD NO.     2 (BINARY)  CONTAINING    34 WORDS
 5052890  -5925608   5702721  -4189985    131141    809986   1342470   1772592        10        12        11         6      12
       0         0         0         0         0         0         0         0         0         0         0         0      24
       0         0         0         0         0         0         0         0   4696145         0

 RECORD NO.     3 (BINARY)  CONTAINING    34 WORDS
       0         0         0         0         0         0         0         0         0         0         0         0      12
       0         0         0         0         0         0         0   5052890  -5925608   5703059   5180464  -3994575      24
-3994575         0         0         0         0         0         0         0   2021455         0

 RECORD NO.     4 (BINARY)  CONTAINING    34 WORDS
       0         0         0         0         0         0         0         0         0         0         0   5052690      12
-5925608   5702805   4607012  -6605209  -5878159   5052690  -5925608   5703251   5180464  -3994575  -3994575   5052690      24
-5925608   5703206  -5923215   4535509  -7402447  -4105899  -4037199   5315793   7985223         0

 RECORD NO.     5 (BINARY)  CONTAINING    34 WORDS
-4102556   5704752  -3994575         0         0         0         0         0         0         0         0         0      12
       0   4188367  -5890879         0         0         0         0         0         0  -4180093   3390868  -4209313      24
 3836115  -4205125   4025856         0         0         0         0         0  -5436583         0

 RECORD NO.     6 (BINARY)  CONTAINING    34 WORDS
       0         0         0         0         0         0         0   4160101   6046811         0         0         0      12
       0         0         0   4157263  -1505245   4155466  -6316927   4156781   -770101         0         0         4      24
    3020   4196352         0  -3994575  -3994575  -3994575  -3994575  -3994575  -1689550         5
   KOUNT      DELT                                                                                          JBIG

 END OF FILE
 THERE ARE     6 RECORDS CONTAINING    204 WORDS BEFORE FILEMARK NO.  2
```

```
RECORD NO.    7 (BINARY)  CONTAINING   3
       1       10        0        0                         0        0        0        0        0        0
       0        0        0        0                         0        0        0        0        0        0
       0        0        0        0                         0        0       11        0

RECORD NO.    8 (BINARY)  CONTAINING   34 WORDS
 5052690 -5925608  5703059  5180464 -3994575 -3994575        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
 5052690 -5925608  5702805  4807012 -6605209 -6878159  5052690 -5925608 -2897932        0

RECORD NO.    9 (BINARY)  CONTAINING   34 WORDS
 5703251  5180464 -3994575 -3994575  5052690 -5925608  5703206 -5923215  4535509 -7402447 -4105899 -4037199    12
 5315793 -4102556  5704752 -3994575       40     3019 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575    24
-3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575  4902885        3

RECORD NO.   10 (BINARY)  CONTAINING   82 WORDS
 4173732  5073104  4172544  1174911  4188935  2788858  4197470  7024728  4205564   706397  4218841  1644888    12
 4221239 -1723823  4221162 -6007187  4214647  5044968  4206198 -1222727  4196996  3756679  4184448 -8210199    24
 4176673 -5328864  4177084 -4903954  4169695 -6411055  4173788 -2079425  4168636  4998611  4173290 -2969428    36
 4168302 -7694196  4172588  8318388  4168612  3921072  4172730  6552458  4165471  -671174  4169457  5095869    48
 4165081  -848329  4171802 -4830439  4164832  8360995  4169083  5250298  4164589 -3321011  4168923 -1402418    60
 4164689  3525129  4169655  4627763  4168014  5094351  4163237  6040689  4165426 -5284806  4168803 -2646556    72
 4187920 -1494809  4169345 -3741602  4164931  3138240  4169505 -1735918  4164704  2939288

RECORD NO.   11 (BINARY)  CONTAINING   34 WORDS
       2       10        0        0        0        0        0        0        0        0        0        0    12
       0        0        0        0        0        0        0        0        0        0        0        0    24
       0        0        0        0        0        0        0        0       12        0

RECORD NO.   12 (BINARY)  CONTAINING   34 WORDS
 5052690 -5925608  5703059  5180464 -3994575 -3994575        0        0        0        0        0        0    12
       0        0        0        0        0        0        0        0        0        0        0        0    24
 5052690 -5925608  5702805  4607012 -6605209 -6878159  5052690 -5925608 -2897932        0

RECORD NO.   13 (BINARY)  CONTAINING   34 WORDS
 5703251  5180464 -3994575 -3994575  5052690 -5925608  5703206 -5923215  4535509 -7402447 -4105899 -4037199    12
 5315793 -4102556  5704752 -3994575       40     3019 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575    24
-3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575  4902885        3

RECORD NO.   14 (BINARY)  CONTAINING   82 WORDS
 4226595  6161612  4226108    35587  4222585 -8083144  4208864  7129074 -4221913 -7236998 -4225658  4118638    12
-4226055 -1387446 -4225641  8276488 -4222142 -7355757 -4213221  5584176  4218397 -6799561  4222332  2081618    24
 4222880   456864  4222362 -9300759  4221091 -2163734  4213266 -2856827 -4213777  5547755 -4218218  5546523    36
-4220933 -4064321 -4218450  3492067 -4217259 -7070009 -4210615  6267251  4205090 -3454680  4213137  2446633    48
 4213875 -8250826  4213666  2533331  4212839  1448908  4208758 -2566727  4189829  -220388 -4204694  -665425    60
-4205877  6313703 -4205799  -911154 -4204933  4955846 -4200965 -4470328 -4188173  3132042  4189538   202755    72
 4197195  1730093  4197111 -5946876  4190064  6287928  4185412  4629747 -4176776  3256628

RECORD NO.   15 (BINARY)  CONTAINING   34 WORDS
       1       10        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0       11        0
 RECORD NO.   16 (BINARY)  CONTAINING   34 WORDS
  5052690 -5925608  5703059  5180464 -3994575 -3994575        0        0        0        0        0        0      12
        0        0        0        0        0        0        0        0        0        0        0        0      24
  5052590 -5925608  5702805  4607012 -6605209 -6878159  5052690 -5925608 -2897932        0

 RECORD NO.   17 (BINARY)  CONTAINING   34 WORDS
  5703251  5180464 -3994575 -3994575  5052690 -5925608  5703206 -5923215  4535509 -7402447 -4105899 -4037199      12
  5315793 -4102556  5704752 -3994575       80     3019 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575      24
 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575  4902925        3

 RECORD NO.   18 (BINARY)  CONTAINING  162 WORDS
  4172685  3438460 -4152314  4032376  4173659 -4347063 -4160890  8089746  4181117  3680089  4188360 -4144802      12
  4196519  3331535  4189886 -2328852  4197113 -1346275  4202011 -2433537  4209627 -1867879  4214224  2337488      24
  4217541   680821  4217903  7727792  4217422  4334087  4213773  2939326  4209459    -8238  4205344 -5344398      36
  4200989  8317720  4196373  2663128  4184778 -1124079  4181024 -1555971  4177049  1892565  4176524 -7966064      48
  4171986 -8384567  4173244 -5174721  4169260  2135033  4172483 -7670008  4155068  4599651  4169599  7762766      60
  4165193  -806943  4169316 -6836398  4164674 -1560095  4169017  3692673  4165360 -2912100  4169128  -505233      72
  4163964 -2845064  4168256  4465293  4164532  2289285  4168367 -4230115  4163895  1566677  4168905  7768110      84
  4168036  4880653  4168006  8012877  4160373  7514898  4165370 -7759671  4160692 -3630242  4164776  4643690      96
  4163643  5744252  4967726 -1608447  4164023  7770385  4165251  7033061  4160578  7585376  4164887 -1841465     108
  4160961    55949  4164684 -1736728  4160771  7027061  4164737  4009901  4161070 -4759492  4164574 -1618711     120
  4160300 -4500309  4164368 -3486490  4164635 -1898245  4168032   555604  4163892  5493615  4164635  7639813     132
  4161524  -984042  4165334  3574969  4163893 -2266792  4164466  3112004  4160183  5097767  4164482 -4077256     144
  4164197 -7314563  4167885 -8150708  4164059 -1172367  4164379 -1366904  4160296  6965325  4164517 -5594739     156
  4164007   -44213  4167895  8020374  4161328 -7812585

 RECORD NO.   19 (BINARY)  CONTAINING   34 WORDS
        2       10        0        0        0        0        0        0        0        0        0        0      12
        0        0        0        0        0        0        0        0        0        0        0        0      24
        0        0        0        0        0        0        0        0       12        0

 RECORD NO.   20 (BINARY)  CONTAINING   34 WORDS
  5052690 -5925608  5703059  5180464 -3994575 -3994575        0        0        0        0        0        0      12
        0        0        0        0        0        0        0        0        0        0        0        0      24
  5052690 -5925608  5702805  4607012 -6605209 -6878159  5052690 -5925608 -2897932        0

 RECORD NO.   21 (BINARY)  CONTAINING   34 WORDS
  5703251  5180464 -3994575 -3994575  5052690 -5925608  5703206 -5923215  4535509 -7402447 -4105899 -4037199      12
  5315793 -4102556  5704752 -3994575       80     3019 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575      24
 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575 -3994575  4902925        3

 RECORD NO.   22 (BINARY)  CONTAINING  162 WORDS
  4226595  6161612  4226111  5660147  4222601  5687247  4208886  5703722 -4221965  4631630 -4225731  7915226      12
 -4226176 -6825751 -4225785   915469 -4222376 -2190978 -4213454  7707715  4218814   555648  4222821 -1591869      24
  4225320  4251755  4225076 -4363267  4221633 -7956118  4214010  5679428 -4216849  4914248 -4221270  4056838      36
 -4221865 -6787739 -4221673  7722012 -4218750 -8256091 -4214090  7896661  4208916  -236543  4217118 -5782984      48
  4218030  5070398  4218038 -2568910  4217318  6531047  4213453  1744577  4200981 -7181875 -4209929 -4141270      60
 -4213507 -1045103 -4213723 -7771998 -4213164  5108943 -4209477 -2976135 -4200964 -3279938  4204840 -3407868      72
  4208749  5267330  4208840 -2390219  4205819  8293191  4201309 -2254159 -4188603 -7213901 -4201207 -8317467      84
 -4201680  -824118 -4200725 -3725466 -4188626 -3347211  4190144  4239536  4198159  6404102  4197864  5137852      96
  4189320 -5570943 -4188382  5165710 -4197876 -3396194 -4200816 -8111801 -4200638  5648728 -4196922 -2108298     108
  4168443 -3377329  4195916  5235193  4200596  5430263  4200848  -584642  4198368 -6232131  4196598 -1199986     120
  4176412   444224 -4189463 -2989972 -4197065 -3015207 -4197201 -2996391 -4196496 -2096424 -4188193 -7151355     132
  4176289  4086086  4185984  6590592  4188683 -3481956  4188213 -1664049  4181393  7957671 -4180407 -4311791     144
 -4185479  5261110 -4188257  -921821 -4185444 -2947953 -4180274 -3643858  4181568  1541321  4188454 -3738658     156
```

Program: DICROSCOPIC PLOT

I. PURPOSE:

To coordinate the plotting subroutines so that a paper tape is output which when fed to the PDP8 computer with Calcomp plotter on line will produce plots of the required spectra in a specified configuration. In this description and those of the plotting subroutines, mention of plotting or spacing of the paper implies the punching on paper tape of instructions to perform these functions.

II. OPERATION:

OPENFILE and RUNFILE search for a specified run. ANSWERIN then reads in the spectra from the input tape. When each spectrum has been read, (the program does not accept autocovariance functions as it stands) the program positions the axes 1.8" to the right of the margin or the previous spectrum and plots the spectrum. When all three spectra (one for each lag value) have been plotted side by side the paper is spaced up 7" and PAUSE 7773_8 is reached. With SS 4 OFF, pressing GO will cause the spectra for the next variable on tape to be plotted. This continues until all the spectra of all variables of a particular run have been plotted when PAUSE 7775_8 is reached. A card is then read specifying the name of the next run to be found by RUNFILE for which spectra are required.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

PAUSE 7777_8 4095_{10} — Initial pause after compiling program to allow setting up of tapes.

SS 3 ON — The program will read from a card the values of TAG and NBIG for a specified variable (NVBL) and use ANSWERIN to find this spectrum and start the plotting routine from there. This would be used when restarting the program after interruption.

SS 3 OFF — The program will plot all spectra of the specified run.

SS 5 ON — Suppresses printing of the power spectrum and autocovariance function.

OFF — Both spectrum and autocovariance are printed.

PAUSE 7773_8 4091_{10} — is reached when all the spectra for a particular variable have been plotted. The pause allows setting of SS 4.

SS 4 OFF — allows the spectra for the next variable on tape to be plotted.

ON — causes the tape to be rewound to the beginning of the present run so that if required a particular spectrum from the run just examined could be selected.

PAUSE 7775_8 4093_{10} — is reached when all the spectra for a particular run have been plotted. The pause allows loading of a card naming the next run to be found by RUNFILE.

IV. USAGE:

Compile program.
PAUSE 7777
Load input tape on handler 1. No output tape needed.
Load reader with cards as required by OPENFILE and RUNFILE including the name of the first run to be processed.
Set SS 3 and SS 5 as required (see above).
GO
Program will cycle as described in OPERATION until one of the pauses is reached, the procedure being as described above.
The program does not cycle unattended. This is not possible because of the large amount of punched paper tape used by the program, the supply having to be continually replenished. A careful watch has to be kept over the punch during operation.
The program can be run at the same time as the actual plotting of the spectra. Each program needs attention, but it is possible to combine both operations.

V. STORAGE REQUIREMENTS:

1673_8 955_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

OPENFILE RUNFILE ANSWERIN AXISXY PLOTXY ENDPLOT
PLOTSPD SELECT PRINTPSP PRINTACV

VII. RUNNING TIMES:

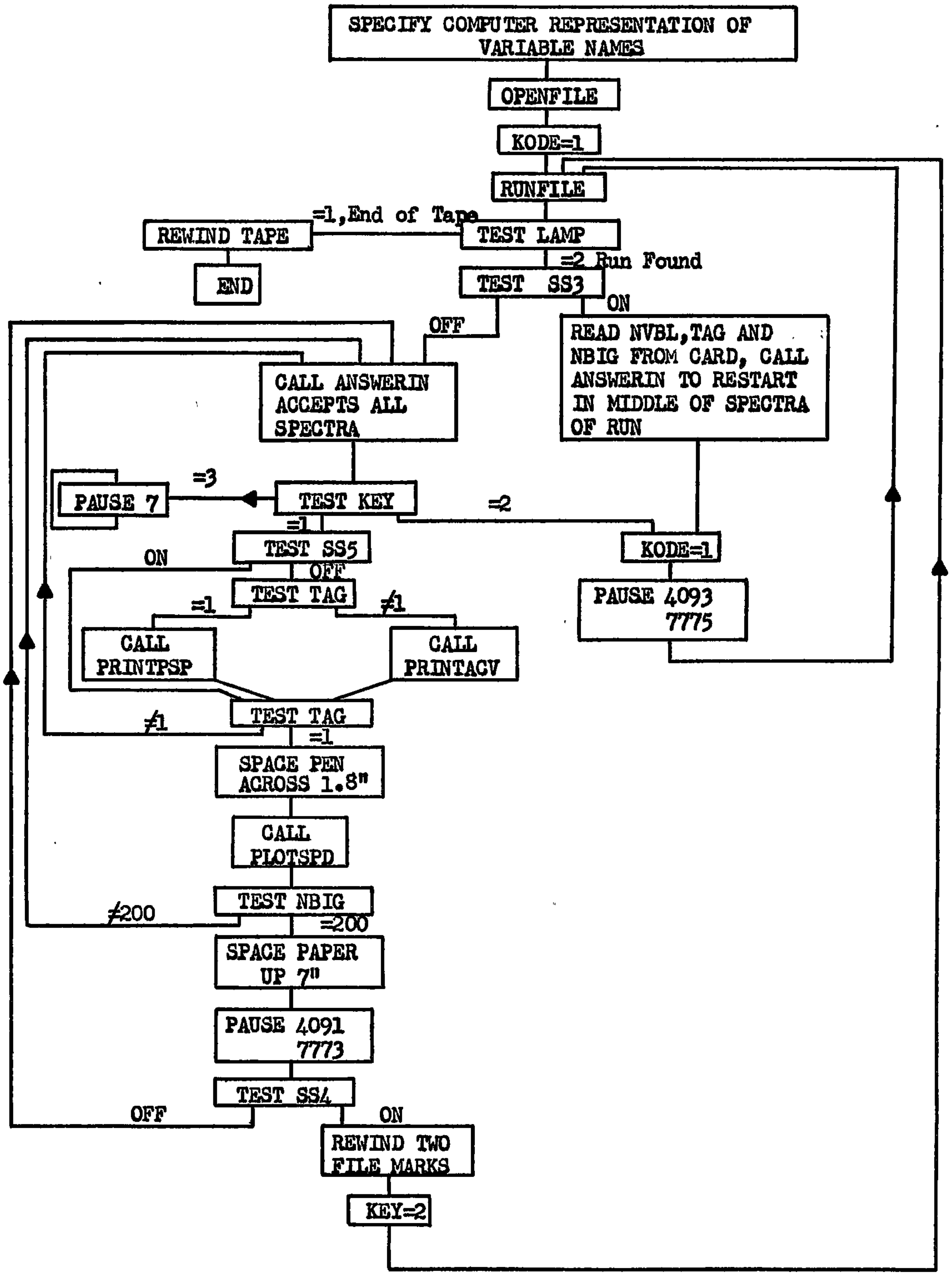
15 minutes for punching of 1 run (4 variables and 3 lags).

```
00000  C     DICROSCOPIC PLOT
00000  C     AUGUST 31 1967
00000  C     ------------------------------------------------------------
00000  C     COMMON STORAGE
00000  C
00004        COMMON RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004        DIMENSION RNM(8),CHNO(13),VNM(3,13),MEANV(13),FACT(13)
00004        INTEGER RNM,CHNO
00004        REAL MEANV
00004  C     ------------------------------------------------------------
00004  C     INPUT FILES
00004  C
00004  C                   CTAPE FILE 1 TAPE LABEL
00004  C     CALL OPENFILE(0,CTAPE,KEY)
00004        INTEGER CTAPE
00004  C
00004  C                   CTAPE FILE 2 RECORD1
00004  C     CALL RUNFILE( CTAPE,KEY,KODE,LAMP)
00004  C
00004  C                   CTAPE FILE 3 RECORD 1 , RECORD 2 AND RECORD 4
00004  C     CALL ANSWERIN(CTAPE,VBLNAME,TAG,NBIG,S,KEY)
00004        INTEGER TAG,CHNN
00004        DIMENSION IDUM(11)
00004        EQUIVALENCE (CHNO(1),TAG),(CHNO(2),CHNN),(CHNO(3),IDUM(1)),
00004       1(JBIG,NBIG),(KOUNT,IBIG)
00004  C
00004        DIMENSION S(251)
00004  C     ------------------------------------------------------------
00004  C     WORKING STORAGE
00004  C     BINARY SPECIFICATION OF VARIABLES AS FOLLOWS
00004        DIMENSION NCBM(6)
00004        NCBM(1)= 5052690
00007        NCBM(2)=-5925608
00013        NCBM(3)= 5702805
00016        NCBM(4)= 4607012
00021        NCBM(5)=-6605209
00025        NCBM(6)=-6878159
00031        DIMENSION NCHA(6)
00031        NCHA(1)= 5052690
00034        NCHA(2)=-5925608
00040        NCHA(3)= 5703206
00043        NCHA(4)=-5923215
00047        NCHA(5)= 4535509
00052        NCHA(6)=-7402447
00056        DIMENSION NCIC(6)
00056        NCIC(1)= 5052690
00061        NCIC(2)=-5925608
00065        NCIC(3)= 5703251
00070        NCIC(4)= 5180464
00073        NCIC(5)=-3994575
00077        NCIC(6)=-3994575
00103        DIMENSION NCFC(6)
00103        NCFC(1)= 5052690
00106        NCFC(2)=-5925608
00112        NCFC(3)= 5703059
00115        NCFC(4)= 5180464
00120        NCFC(5)=-3994575
00124        NCFC(6)=-3994575
00130        DIMENSION NDHA(6)
00130        NDHA(1)= 5400870
00133        NDHA(2)= 4926488
```

```
00136          NDHA(3)=-6646150
00142          NDHA(4)=-4123436
00146          NDHA(5)= 5651504
00151          NDHA(6)=-3994575
00155          DIMENSION NDBM(6)
00155          NDBM(1)= 5400870
00160          NDBM(2)= 4926482
00163          NDBM(3)= 5577008
00166          NDBM(4)=-7181094
00172          NDBM(5)=-6661071
00176          NDBM(6)=-3994575
00202          DIMENSION NDIC(5)
00202          NDIC(1)= 5400870
00205          NDIC(2)= 4926489
00210          NDIC(3)= 5061680
00213          NDIC(4)=-3994575
00217          NDIC(5)=-3994575
00223          NDIC(6)=-3994575
00227          DIMENSION NDFC(6)
00227          NDFC(1)= 5400870
00232          NDFC(2)= 4926486
00235          NDFC(3)= 5061680
00240          NDFC(4)=-3994575
00244          NDFC(5)=-3994575
00250          NDFC(6)=-3994575
00254    C
00254          PAUSE 4095
00256          DO 15000 I=1,6
00260          IF(SENSE SWITCH 1)15000,15002
00264    15000 CONTINUE
00271          CALL LEADER
00272          CALL DUMP
00273    15002 CONTINUE
00273          CALL OPENFILE(0,CTAPE,KEY)
00277          KODE =1
00301    1000  CALL RUNFILE(CTAPE,KEY,KODE,LAMP)
00306          GO TO(9000,1001),LAMP
00313    1001  IF(SENSE SWITCH 3) 1002,1010
00317    C------------------------------------------------------------------
00317    C     ALLOWS RESTART IN MIDDLE OF SPECTRA OF A RUN
00317    1002  READ 1003,NVBL,TAG,NBIG
00330    1003  FORMAT(16X,A4,19X,I1,17X,I3)
00337          CALL ANSWERIN(CTAPE,NVBL,TAG,NBIG,S,KEY)
00346          GO TO 1011
00347    C------------------------------------------------------------------
00347    1010  CALL ANSWERIN(CTAPE,0,0,0,S,KEY)
00356          GO TO (1020,1011,1013),KEY
00364    1011  KODE=1
00366          PAUSE 4093
00370          GO TO 1000
00371    1013  PAUSE 7
00373          GO TO 1013
00374    1020  IF (SENSE SWITCH 5) 2000,1100
00400    1100  IF(TAG-1) 1101,1102,1101
00405    1101  CALL PRINTACV(S,CHNN,NBIG,IBIG)
00412          GO TO 2000
00413    1102  CALL PRINTPSP(S,CHNN,NBIG,IBIG)
00420          GO TO 2000
00421    2000  CONTINUE
00421          IF(TAG-1)2100,2001,2100
00426    2001  CALL AXISXY(20,1.,1.8,0.,1.,1.8,0.,0.,0.,0.)
00441          CALL PLOTXY(0.,1.8,0,0)
```

```
 00446          CALL ENDPLOT(20)
 00450          CALL PLOTSPD(S,1.0,0,1,2)
 00456          CALL ENDPLOT(20)
 00460          IF(NBIG-200) 2003,2002,2003
 00465   2002   CALL AXISXY(20,7.,1.8,0.,1.,1.8,0.,0.,0.,0.)
 00500          CALL PLOTXY(7.,0.,0,0)
 00505          CALL ENDPLOT(20)
 00507          CALL LEADER
 00510          PAUSE 4091
 00512   2004   IF(SENSE SWITCH 4) 2005,2006
 00516   2005   CALL SELECT(CTAPE,12)
 00521          CALL SELECT(CTAPE,12)
 00524          KEY=2
 00526          GO TO 1000
 00527   2006   GO TO 2100
 00530   2003   CALL ENDPLOT(20)
 00532   2100   GO TO 1010
 00533   9000   REWIND CTAPE
 00535          END

SUBPROGRAMS

  LEADER      DUMP        OPENFILE    RUNFILE     ANSWERIN    PRINTACV
  PRINTPSP    AXISXY      PLOTXY      ENDPLOT     PLOTSPD     SELECT

PROGRAM ALLOCATION

 00617    I         00620    CTAPE     00621    KEY       00622    KODE
 00623    LAMP      00624    NVBL

 00625    S         01613    NCBM      01621    NCHA      01627    NCIC
 01635    NCFC      01643    NDHA      01651    NDBM      01657    NDIC
 01665    NDFC

COMMON ALLOCATION

 00000    RNM       00010    CHNO      00025    VNM       00143    MEANV
 00175    FACT      00227    JBIG      00230    KOUNT     00231    DELT

 00010    TAG       00011    CHNN      00012    IDUM      00227    NBIG
 00230    IBIG

PROGRAM END
```

DICROSCOPIC PLOT



Subroutine: ANSWERIN(NTAPE,VBLNAME,ITYPE,NOINT,ANS,KEY)

I. PURPOSE:

To find a specified spectrum or autocovariance.

II. OPERATION:

VBLNAME is the name of the variable for which we require the spectrum or autocovariance. VBLNAME is specified in binary in the main program for comparison with the variable name read by the subroutine whilst searching through the spectrum headings.

ITYPE is the value of TAG which we require. It is 1 if the power spectrum is required and 2 if the autocovariance. Again this is specified in the main program and transferred to the subroutine through ITYPE.

NOINT is the number of lags of the function required.

Normal entry to the subroutine is with KEY = 0 when the tape is positioned within the run heading file. This is the case upon exit from the subroutine RUNFILE which normally precedes this subroutine in the main program. The subroutine then causes the tape to be wound over one file mark so that it is positioned at the beginning of the spectrum heading ready to search for the desired function.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

PAUSE 2222_8 1170_{10} The word IND is used as an error indicator. Its value is either 1 or 0, the former when the search is proceeding normally and the latter when the possibility of an error condition exists. If the program is trying to read spectrum headings in the wrong place exit from the subroutine is effected with KEY = 3 so that re-entry will be a non-recoverable pause 2222_8.

IV. USAGE:

```
      CALL OPENFILE(0,CTAPE,KEY)
      KODE = 1
 1000 CALL RUNFILE(CTAPE,KEY,KODE,LAMP)
      GO TO(9000,1010),LAMP
 2000 CALL ANSWERIN(CTAPE,VBLNAME,ITYPE,NOINT,ANS,KEY)
      GO TO(4000,3000,2000),KEY
 3000 KODE = 0
      GO TO 1000
C     PRINTOUT THE REQUIRED FUNCTION
 4000 CALL PRINTFN(PARAM)
      - - - - - - - - - - - -
      - - - - - - - - - - - -
      GO TO 2000
```

V. STORAGE REQUIREMENTS:

363_8 243_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

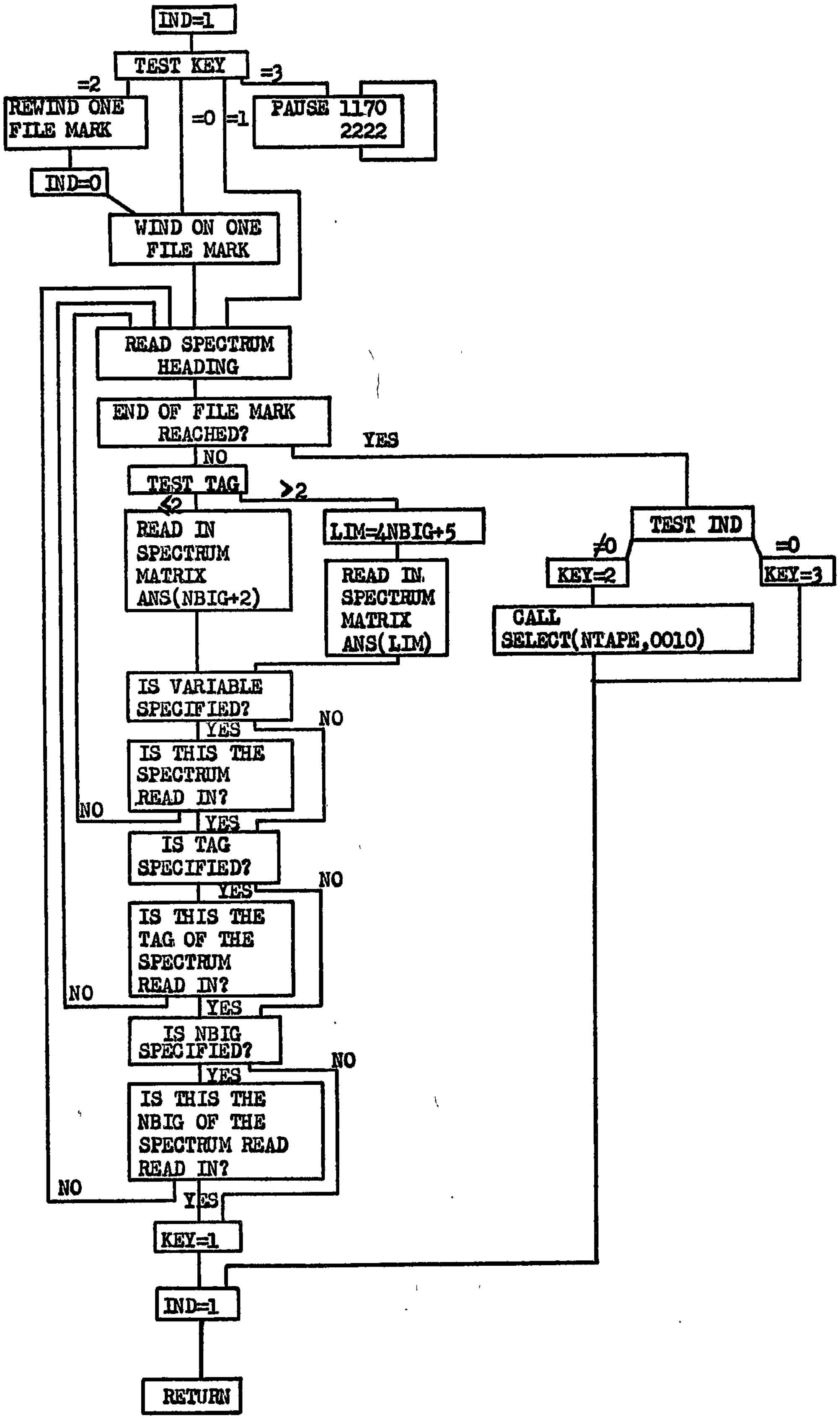
SELECT EOFCK READB COMPARE

```
00004          SUBROUTINE ANSWERIN(NTAPE,VBLNAME,ITYPE,NOINT,ANS,KEY)
00004    C     SUBROUTINE TO FIND SPECIFIED SPECTRUM OR AUTOCOVARIANCE
00004    C     4 APRIL 1967
00004    C     28 MARCH 1967
00004    C
00004    C      ENTRY OF SUBROUTINE WITH KEY =2 WILL CAUSE INPUT TAPE TO BE
00004    C     SEARCHED BACKWARD TO BEGINNING OF CURRENT RUN FILE AND A FORWARD
00004    C     SEARCH IS THEN RESUMED FOR THE DESIRED ANSWER.
00004    C
00004    C     NORMAL ENTRY IS WITH KEY=0 WHEN TAPE IS POSITIONED AHEAD OF
00004    C     FILE MARK ( AS IS THE CASE WHEN LEAVING =RUNFILE= SUBROUTINE).
00004    C     ENTRY WITH KEY=3 WILL CAUSE A NON-RECOVERABLE PAUSE SHOWING 2222.
00004    C
00004    C     ------------------------------------------------------------------
00004    C      COMMON STORAGE
00004    C
00004          COMMON RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004          DIMENSION RNM(8),CHNO(13),VNM(3,13),MEANV(13),FACT(13)
00004          INTEGER RNM,CHNO
00004          REAL MEANV
00004    C     ------------------------------------------------------------------
00004    C     INPUT FILE
00004    C
00004    C                    NTAPE FILE 1 TAPE LABEL
00004    C     CALL OPENFILE(0,NTAPE,KEY)
00004          INTEGER NTAPE
00004    C
00004    C                   NTAPE FILE 2 RECORD 1
00004    C     READ TAPE NTAPE,RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004    C
00004    C                   NTAPE FILE 2 RECORD 2
00004    C     READ TAPE NTAPE,NL,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004          DIMENSION NL(8)
00004          NL(1)=5657904
00007          NL(2)=-6722508
00013          NL(3)=4617584
00016          NL(4)=-3994575
00022          NL(5)=-3994575
00026          NL(6)=-3994575
00032          NL(7)=-3994575
00036          NL(8)=-3994575
00042    C
00042    C                   NTAPE FILE 3 RECORD 1
00042    C     READ TAPE NTAPE,TAG,CHNN,VNM,NBIG,IBIG
00042          INTEGER TAG,CHNN
00042          EQUIVALENCE (CHNO(1),TAG),(CHNO(2),CHNN),(JBIG,NBIG),(KOUNT,IBIG)
00042    C
00042    C                   NTAPE FILE 3 RECORD 2
00042    C     CALL READB(NTAPE,ANS,ANS(NBIG+2))
00042    C
00042    C                   NTAPE FILE 3 RECORD 3
00042    C     LIM=4*NBIG+5
00042    C     CALL READB(NTAPE,ANS,ANS(LIM))
00042          DIMENSION ANS(1)
00042    C
00042    C     WORKING STORAGE
00042    C
00042          DIMENSION VBLNAME(3)
00042    C     ------------------------------------------------------------------
00042          IND=1
00044    C
```

```
00044  1000  KEE=KEY+1
00047        GO TO (1100,2000,1120,1130),KEE
00056  1100  CALL SELECT(NTAPE,11)
00063        GO TO 2000
00064  1120  CALL SELECT(NTAPE,12)
00071        IND=0
00073        GO TO 1100
00074  1130  PAUSE 1170
00076        GO TO 1130
00077  C                READ FILE 3 RECORD 1
00077  2000  READ TAPE NTAPE,TAG,CHNN,VNM,NBIG,IBIG
00115        CALL EOFCK(NTAPE,LITE)
00122        GO TO (9200,2010),LITE
00127  C     TEST FOR  1 OR 2 DIMENSIONAL ANSWER
00127  2010  IF(TAG-2) 2020,2020,2030
00134  C                READ FILE 3 RECORD 2
00134  2020  CALL READB(NTAPE,ANS,ANS(NBIG+2))
00154        GO TO 3000
00155  C                READ FILE 3 RECORD 3
00155  2030  LIM=4*NBIG+5
00161        CALL READB(NTAPE,ANS,ANS(LIM))
00201        GO TO 3000
00202  3000  IF(VBLNAME(1))3001,3100,3001
00214  3001  CALL COMPARE(VBLNAME,VNM(1,CHNN),6,LAMP)
00234        GO TO(3100,2000),LAMP
00241  3100  IF(ITYPE)3101,3200,3101
00245  3101  IF(ITYPE-TAG)2000,3200,2000
00252  3200  IF(NOINT)3201,3300,3201
00256  3201  IF(NOINT-NBIG)2000,3300,2000
00263  3300  GO TO 9100
00264  9100  KEY=1
00266        GO TO 9999
00267  9200  IF (IND)9201,9300,9201
00273  9201  KEY=2
00275        CALL SELECT(NTAPE,0010)
00302        GO TO 9999
00303  9300  KEY=3
00305        GO TO 9999
00306  9999  IND=1
00310        RETURN
00312        END

SUBPROGRAMS

 SELECT     EOFCK      READB      COMPARE

PROGRAM ALLOCATION

 00345   ANSWERIN  00347   IND        00350   KEE        00351   LITE
 00352   LIM       00353   LAMP

 00354   NL

COMMON ALLOCATION

 00000   RNM       00010   CHNO       00025   VNM        00143   MEANV
 00175   FACT      00227   JBIG       00230   KOUNT      00231   DELT

 00010   TAG       00011   CHNN       00227   NBIG       00230   IBIG
PROGRAM END
```

ANSWERIN

ANSWERIN(NTAPE,VBLNAME,ITYPE,NOINT,ANS,KEY)



Subroutine: AXISXY(LUN,XL,YL,XTIC,XLNG,YLNG,XMIN,YMIN,XORG,YORG,YTIC,N)

I. PURPOSE:

To define the axes to be used by a plotting routine.

II. OPERATION:

The parameters specified in the CALL Command are:

LUN The tape number. In the case of the CDC 3100, the tape punch is number 20.
XL X axis length in inches. The X axis is the vertical axis i.e. it refers to paper or drum movement.
YL Y axis length in inches. The Y axis is the horizontal axis i.e. it refers to pen movement.
XTIC Logical interval between tick marks on the X axis.
XLNG Length of X axis in logical units.
YLNG Length of Y axis in logical units.
XMIN Initial point of X axis in logical units.
YMIN Initial point of Y axis in logical units.
XORG Origin of X axis.
YORG Origin of Y axis.
YTIC Logical interval between tick marks on Y axis.
N Is the number of ticks between grid lines.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

None

IV. USAGE:

Parameters XL to YTIC in the list of parameters above must be specified in the main program, when calling the subroutine, in floating point.

The last two parameters need not be specified in the CALL statement. They are purely optional.

A specific example of its usage will be found in PLOTSPD.

V. STORAGE REQUIREMENTS:

VI. SUBROUTINES AND FUNCTIONS CALLED:

None

Subroutine: PLOTXY(X,Y,IPOS,IMARK)

I. PURPOSE:

To move the plotter pen to a new position either by drawing a line to that point, or by moving to that point and then marking it.

II. OPERATION:

The parameters specified in the CALL command are:

X is the new paper position in logical units.
Y is the new pen position in logical units.
IPOS If the pen should be up during travel, IPOS = 0. If the pen should be down, as is the case if a line graph is to be drawn, IPOS = 1.
IMARK is the code number of the data mark, if any, to be made upon reaching the new position.

IMARK	Symbol
1-2	↑
3-4	→
5-6	↓
7-8	←
9-10	+
11-12	⊠
13-14	⧗
15-16	×
17-18	∧
19-20	>
21-22	┌
23-24	┘
25-26	┤
27-28	┬
29-30	┃
31-32	━

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

None

IV. USAGE:

The subroutine has been used in the subroutine PLOTSPD to both plot the spectrum and also to mark the axis where a bad reading has been input. This is done through the words IMARK and IPOS. Tests are made for bad readings and when one is found, instead of having the pen down during travel, with IPOS = 1, IPOS is set to 0 and IMARK is given a non negative value so that the bad reading is shown up.

This technique is shown in DICROSCOPIC PLOT.

V. STORAGE REQUIREMENTS:

VI. SUBROUTINES AND FUNCTIONS CALLED:

None

Subroutine: LABEL(INUM,ISIZ,IDIR,KARAY(J))

I. PURPOSE:

To draw a label of given size.

II. OPERATION:

The parameters specified in the CALL command are:

INUM Integer number of characters to be plotted.
ISIZ Integer size of characters to be plotted.
IDIR Direction of the character line
0 = +X 1 = -Y 2 = -X 3 = +Y
KARAY(J) Starting word address of alphanumeric data to be written.

In the subroutine PLOTSPD this subroutine is used to provide the labelling of the spectra. The value of ISIZ is passed through that subroutine into the main program where the scaling of the axes and lettering is chosen.

KARAY(J) can reference some alphanumeric data such as the run name RNM, and this will then be written out.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

None

IV. USAGE:

```
      CALL LABEL( 24,ISIZ,1,RNM)
C     24 CHARACTERS OF RNM WILL BE WRITTEN OUT
      etc.
```

V. STORAGE REQUIREMENTS:

VI. SUBROUTINES AND FUNCTIONS CALLED:

None

Subroutine: PLOTSPD(SPECT,ORDMAX,ISYMBOL,LABLE,ISIZ)

I. PURPOSE:

Subroutine to plot power spectra or power spectral densities.

II. OPERATION:

ISIZ Is a scale factor used for scaling the size of the spectrum axes and the lettering used in the titles, so that these fit conveniently on the Calcomp plotter. If ISIZ is greater than 6, a single graph is plotted across the width of the plotter. The letter size is proportionately scaled in each case. In present usage where 3 lag values are used, ISIZ is set to 2 so that three spectra, one for each lag value, are plotted side by side across the plotter.

SPECT Is the name of the array to be plotted. In its present usage for plotting normalized power spectra, the array S is referenced by the subroutine.

ORDMAX Is the length of the X axis in logical units and is transferred to the AXISXY and PLOTXY subroutines within this subroutine.

ISYMBOL Is the code for the data mark to be used by PLOTXY during plotting. The value of this is tested at the beginning of the subroutine, and, if zero, the graph is drawn as a continuous line with a separate data mark available for use at discontinuities e.g. if the spectrum has a calculated negative value at some point.

LABLE If LABLE ≠ 0, a title will be written on the plotted spectrum as follows:

```
POWER SPECTRUM FOR
( RUN NAME )
( VARIABLE NAME )
FREQUENCY INTERVAL IS ( δf ) HZ.
```

The subroutine initially calculates the scaling of the axes and lettering, and then draws the axes and titles the graph, returning the pen to the origin of the axes when this is complete. The maximum value of the array SPECT is found and the normalized values of SPECT calculated and plotted. If the power goes negative, a data mark is made where this occurs and the value plotted as zero. After plotting the graph, the pen is left at the end of the Y axis. The maximum spectral amplitude and number of lags are printed on the line printer and control is returned to the main program.

This subroutine is used as a subroutine of the program DICROSCOPIC PLOT. The end product of this program is a paper tape containing instructions for the plotting of spectra, this tape being fed into the PDP8 computer for plotting on the Calcomp plotter.

III. ERROR CONDITIONS AND PROGRAMMED HALTS:

None

IV. USAGE:

```
C      ESTABLISH ORIGIN FOR SPECTRUM PLOT
       CALL AXISXY(20,1.,1.8,0.,1.,1.8,0.,0.,0.,0.)
       CALL PLOTXY(0.,1.8,0,0)
C      CANCEL THE PREVIOUS ORIGIN SO THAT PLOTSPD CAN START
C      FROM WHERE PEN IS
       CALL ENDPLOT(20)
       CALL PLOTSPD(S,1.0,0,1,2)
       CALL ENDPLOT(20)
C      READY TO SHIFT PEN AND ORIGIN FOR NEXT SPECTRUM
```

V. STORAGE REQUIREMENTS:

1024_8 532_{10}

VI. SUBROUTINES AND FUNCTIONS CALLED:

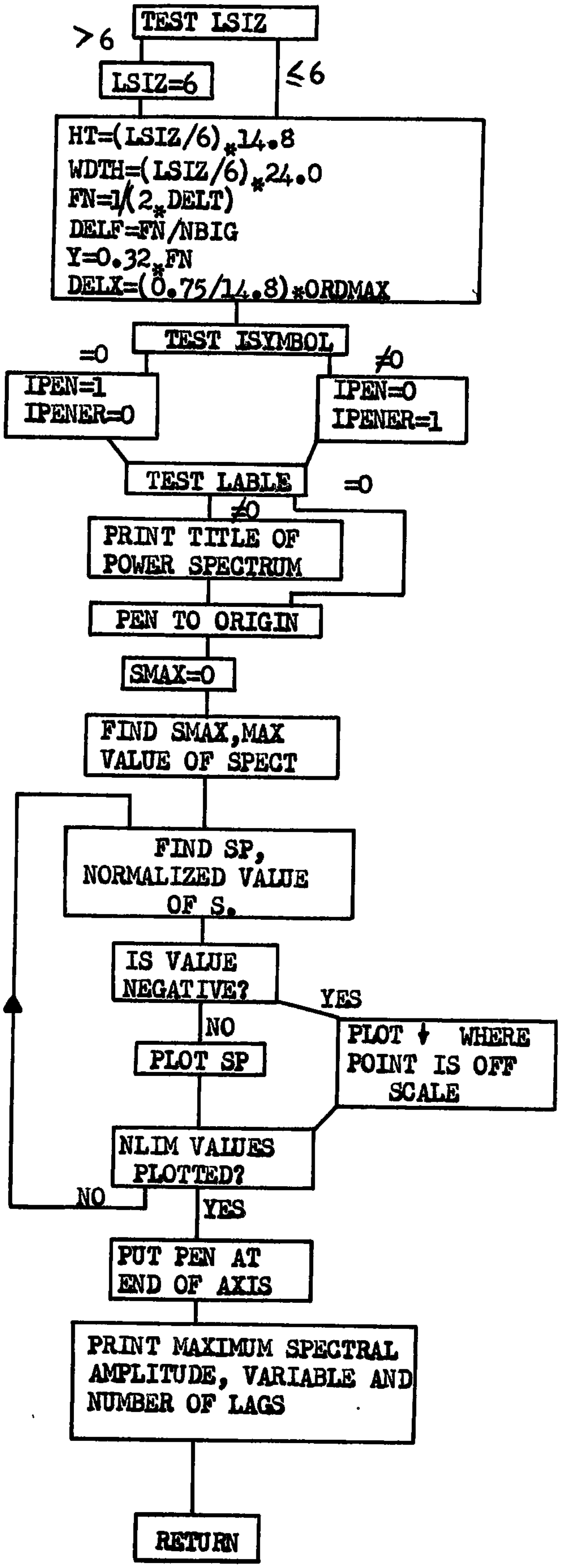
AXISXY PLOTXY LABEL ENCODE

```
00004          SUBROUTINE PLOTSPD(SPECT,ORDMAX,ISYMBOL,LABLE,LSIZ)
00004   C      SUBROUTINE TO PLOT POWER SPECTRAL DENSITIES
00004   C      AUGUST 31 1967
00004   C      JANUARY 9TH 1967
00004   C      ------------------------------------------------------
00004   C      COMMON STORAGE
00004   C
00004          COMMON RNM,CHNO,VNM,MEANV,FACT,JBIG,KOUNT,DELT
00004          DIMENSION RNM(8),CHNO(13),VNM(3,13),MEANV(13),FACT(13)
00004          INTEGER RNM,CHNO
00004          REAL MEANV
00004   C
00004          DIMENSION IDUM(11)
00004          EQUIVALENCE (IDUM(1),CHNO(3))
00004          EQUIVALENCE (NBIG,JBIG),(IBIG,KOUNT)
00004          INTEGER TAG,CHNN
00004          EQUIVALENCE (TAG,CHNO(1)),(CHNN,CHNO(2))
00004   C
00004   C      ------------------------------------------------------
00004   C      WORKING STORAGE
00004   C
00004          DIMENSION SPECT(1)
00004          DIMENSION FRINT(32)
00004          NLIM=NBIG+1
00007          DIMENSION NAME(5)
00007          NAME(1)=-6394474
00013          NAME(2)=-5829464
00017          NAME(3)=5586153
00022          NAME(4)=-2995177
00026          NAME(5)=-6644687
00032          DIMENSION INT(6)
00032          INT(1)=5936488
00035          INT(2)=-3057324
00041          INT(3)=-1898906
00045          INT(4)=-3319178
00051          INT(5)=4602905
00054          INT(6)=-3470287
00060          INTEGER FMT
00060          DIMENSION FMT(2)
00060          FMT(1)=-958116
00064          FMT(2)=904240
00067          NHZ=-4092303
00072   C      ------------------------------------------------------
00072   C
00072   1000   IF(LSIZ-6)1002,1002,1001
00077   1001   LSIZ=6
00101   1002   HT=(LSIZ/6)*14.8
00111          WDTH=(LSIZ/6)*24.0
00121          FN=1/(2*DELT)
00131          DELF=FN/NBIG
00137          FREQ=0.
00141          Y=0.32*FN
00144          DELX=(0.75/14.8)*ORDMAX
00150   1003   IF(ISYMBOL)1005,1004,1005
00154   1004   IPEN=1
00156          IPENER=0
00160          GO TO 1006
00161   1005   IPEN=0
00163          IPENER=1
00165   1006   IF(LABLE)1007,2000,1007
00171   1007   CALL AXISXY(20,HT,WDTH,ORDMAX/10.,ORDMAX,FN,0.,0.,0.,0.,FN/10)
```

```
 00220          CALL PLOTXY(ORDMAX-2.*DELX,Y,0,0)
 00233          CALL LABEL(18,LSIZ,I,NAME)
 00242          CALL PLOTXY(ORDMAX-3.*DELX,Y,0,0)
 00255          CALL LABEL(32,LSIZ,I,RNM)
 00264          CALL PLOTXY(ORDMAX-4.*DELX,Y,0,0)
 00277          CALL LABEL(24,LSIZ,I,VNM(I,CHNN))
 00317          CALL PLOTXY(ORDMAX-5.*DELX,Y,0,0)
 00332          CALL LABEL(23,LSIZ,I,INT)
 00341          CALL ENCODE (I,FMT,FRINT,DELF)
 00346          CALL LABEL(5,LSIZ,I,FRINT)
 00355          CALL LABEL(3,LSIZ,I,NHZ)
 00364   2000   CALL PLOTXY(0.,0.,0,0)
 00371   2010   SMAX=0.
 00373          DO 2019 N=1,NLIM
 00375             IF(SPECT(N)-SMAX)2019,2011,2011
 00410   2011      SMAX=SPECT(N)
 00420   2019   CONTINUE
 00425          DO 2009 N=1,NLIM
 00427          SP=SPECT(N)/SMAX
 00440             IF(SP)2001,2002,2002
 00444   2001          CALL PLOTXY(0.0,FREQ,IPENER,7)
 00451                 GO TO 2003
 00452   2002      CALL PLOTXY(SP,FREQ,IPEN,ISYMBOL)
 00461   2003      FREQ=FREQ+DELF
 00464   2009   CONTINUE
 00471          CALL PLOTXY(0.,FREQ-DELF,0,0)
 00501          PRINT 2020,(VNM(I,CHNN),I=1,3),SMAX,NBIG
 00532   2020   FORMAT(11X,31HMAXIMUM SPECTRAL AMPLITUDE FOR 3A8,2H =F9.3,7H  WITH
 00532         1 I3,6H LAGS.)
 00556   9999   RETURN
 00560          END

SUBPROGRAMS

  AXISXY      PLOTXY      LABEL       ENCODE

PROGRAM ALLOCATION

 00655    PLOTSPD    00657    NLIM      00660    NHZ      00661    HT
 00663    WDTH       00665    FN        00667    DELF     00671    FREQ
 00673    Y          00675    DELX      00677    IPEN     00700    IPENER
 00701    SMAX       00703    N         00704    SP       00706    I

 00707    FRINT      01007    NAME      01014    INT      01022    FMT

COMMON ALLOCATION

 00000    RNM        00010    CHNO      00025    VNM      00143    MEANV
 00175    FACT       00227    JBIG      00230    KOUNT    00231    DELT

 00012    IDUM       00227    NBIG      00230    IBIG     00010    TAG
 00011    CHNN

PROGRAM END
```

PLOTSPD

PLOTSPD(SPECT,ORDMAX,ISYMBOL,LABLE,LSIZ)



INSTRUCTIONS FOR USE OF PDP 8 IN PLOTTING ROUTINES

High speed RIM LOADER

```
                     LOAD 7756 and press L.A. (load address)
                     LOAD 6014 and press DEP  (deposit)
Address 7757         LOAD 6011 and DEP
        7760              5357
        7761              6016
        7762              7106
        7763              7006
        7764              7510
        7765              5374
        7766              7006
        7767              6011
        7770              5367
        7771              6016
        7772              7420
        7773              3776
        7774              3376
        7775              5357
Address 7776         LOAD  0000 and DEP to give an address of 7777.
```

```
BINARY LOADER:       Insert BINARY LOADER tape in reader.
                     LOAD 7756 - START.

PROGRAM:             Insert the plotting program tape in reader.
                     LOAD 7777 - L.A.
                     LOAD 3777 - START.
```

The program is then loaded and awaits the data tape. Make sure that pen and paper are ready for operation; that the pen will clear the paper when the PEN UP command is given, and that it will write properly when touching the paper. Position the pen at the left hand side of the plotter drum with PEN UP before loading the data.

```
Load reader with input tape (i.e. the output from
    DICROSCOPIC PLOT)
LOAD 200 - L.A.
           START
           CONTINUE
```

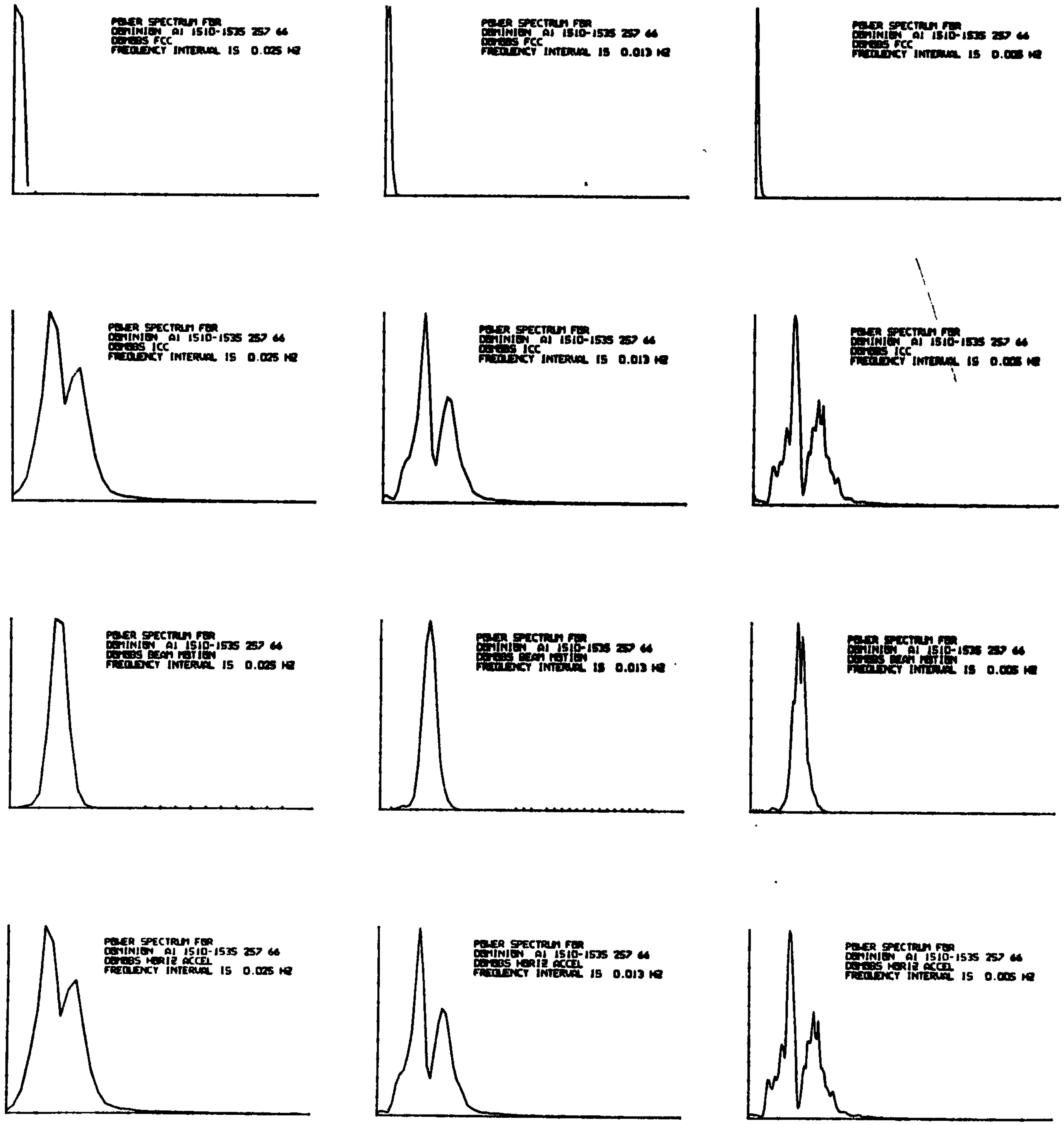
The plotting of three spectra will be executed across the width of the Calcomp. During the plotting of the last graph LOAD 4000. This will cause plotting to cease after the pen has lifted and the drum rotated to start the next set of graphs 7" above the previous set. On this pause ensure that the pen is UP and return pen to the left hand side of the plotter. Then continue:

```
LOAD 200 - L.A.
           START
           CONTINUE
```

Since any break in the tape produced by DICROSCOPIC PLOT will occur at the end of a set of 3 graphs for a particular variable the pause at 4000 is also used to allow loading a new input tape.

If the plotter halts in the middle of plotting a graph it is very difficult to restart it to plot the graph in the same position, although this can be done at times with practice. The easiest way is to revert to the beginning of a set of three graphs and reload the program. A halt may be encountered because of too much tension on the input tape in the reader, unclean punching, a dirty photocell, or some other cause.

EXAMPLE OF POWER SPECTRA PLOTTED FOR EACH RUN



ACKNOWLEDGEMENTS

I would like to acknowledge the assistance during this project of many members of Bedford Institute.

I thank the Director of the Institute for the facilities made available to me during my visits, and for the financial assistance given to me by the Canadian Government.

My thanks go to Dr. R.G. Stevens for his advice and guidance throughout the analysis of my data. As mentioned earlier, most of the programs in this note were devised by him, and without their availability my task would have been gargantuan.

The opportunity to visit B.I. and undertake collection of the cross coupling data was provided by Dr. B.D. Loncarevic whose energy and enthusiasm has encouraged me throughout the project.

The computing facilities made available to me at the Institute were second to none, and I would like to thank R. Richards and J. Wilson and the staff of the Computing Centre for their tolerance shown and assistance given to a novice in their midst.

The Marine Geophysics Group provided a stimulating atmosphere in which to work. My thanks go to all members of the Group for their hospitality and for their tidying-up of innumerable loose ends left after each of my visits.

None of this work would however have been possible without the continuous support given to me by my supervisor at Cambridge University, Mr. B.C. Browne, and I thank him for his encouragement at all times.