NRL Report 7863

UNCLASSIFIED

A Program for Plotting an Annotated Track

MARILYN L. BLODGETT

Research Computation Center
Office of Director of Research

and

JAMES V. MASSINGILL

Propagation Branch
Acoustics Division

March 31, 1975

NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited.

UNCLASSIFIED

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
NRL Report 7863

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
A PROGRAM FOR PLOTTING AN ANNOTATED TRACK

5. TYPE OF REPORT & PERIOD COVERED
Final report

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Marilyn L. Blodgett and James V. Massingill

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Research Laboratory
Washington, D.C. 20375

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
NRL Problem S01-47 ZF 52-552-01, Element 62755N, Arctic Submarine Lab. Proj. 4-0002 ARPA Order 1787

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Electronic Systems Command
Washington, D.C. 20360

12. REPORT DATE
March 31, 1975

13. NUMBER OF PAGES
47

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

GEODATA, Bathymetry, Magnetics, Oceanography, Geophysics

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A program has been written for plotting an annotated track on a polar stereographic projection. The program reads the date, navigation, bathymetry, and magnetics, from a magnetic tape in BCD form. The data is annotated every nth point.

Navigation is annotated with fix numbers, bathymetry with uncorrected fathoms, meters, and corrected meters, and the magnetics with the residual magnetic intensity. This program enables the user to plot oceanographic parameters in relationship to their geographical position. The polar stereographic projection was used because great circles appear as straight (continued)

(block 20 continued)

lines and because it has very little distortion in the high latitudes.

The program was written in Fortran IV for use on the CDC 3800; however the program can be converted to read on other systems with little difficulty.

UNCLASSIFIED

CONTENTS

1.0 IDENTIFICATION 1

2.0 PURPOSE 2

3.0 USAGE 3

4.0 METHOD OR ALGORITHM 5

5.0 FLOW CHART AND/OR SOURCE LANGUAGE LISTING . 5

6.0 COMPARISON 5

7.0 TEST METHOD AND RESULTS 6

8.0 REMARKS 6

APPENDIX A — Sample Input Data Records 7

APPENDIX B — Deck Assembly 10

APPENDIX C1 — Flow Chart 15

APPENDIX C2 — Source Language Listing 16

UNCLASSIFIED

A PROGRAM FOR PLOTTING AN ANNOTATED TRACK

1.0 IDENTIFICATION

1.1 Title

Program for Plotting Annotated Navigation, Bathymetric, or Magnetics Track on a Polar Stereographic Projection.

1.2 Identification Name

ANNOT.

1.3 Classification Code

None.

1.4 NRL Research Computation Center Identification Number

None.

1.5 Entry Points

ANNOT.

1.6 Programming Language

Language: CDC 3600/3800 Fortran.
Routine Type: Program
Operating System: Drum Scope 2.1.

1.7 Computer and Configuration

CDC 3800.

1.8 Contributor or Programmer

Marilyn L. Blodgett, Code 4223MB, Research Computation Center, written for Environmental Sciences Section, Acoustics Division.

1.9 Contributing Organization

NRL — Naval Research Laboratory, Washington, D.C. 20375.

NOTE: Manuscript submitted January 9, 1975.

1.10 Program Availability

If supplied with a magnetic tape, a copy of this program will be made available by the Environmental Sciences Section, Acoustics Division.

1.11 Verification

This program has been used and tested by the Environmental Sciences Section, Acoustics Division, for several months.

1.12 Date

September 1974.

2.0 PURPOSE

2.1 Description of the Routine

This program reads the data collected by an oceanographic or geophysical experiment from a magnetic tape and plots an annotated track. We use the format recommended by the National Research Council of the National Academy of Sciences with one slight modification for the input data tape. There is one logical record (of 80 characters) for each data point. The different types of data (navigation, bathymetry, and magnetics) are separated by an end-of-file mark with a double end-of-file mark at the end of all the data.

Before the program reads this input tape, it reads a series of input cards. These input cards define the actual data format on the input tape (the format varies for the three types of data), the number of files to be skipped over on the input tape, the physical height of the map to be drawn, the actual latitude and longitude values to be included in the grid, and the dates of the data to be considered.

With all the required parameters defined, the program starts to read the input tape one record at a time. Each record is checked to see that the fix falls on the defined grid and that it was taken on or between the two specified dates. Only those points which meet both requirements are stored in core. The program continues reading the input tape until it reads an end-of-file mark or a fix taken after the last specified date. When either situation occurs, the program stops reading the input tape and prepares to plot the annotated track.

The track is plotted on a Polar Stereographic Projection which is drawn exactly to scale. The grid may be blown up to any reasonable size. The largest grid we have defined is 1 degree equals 20 inches. The number of degrees of longitude included in the grid will depend on the scale of the entire grid and the specific area of interest. In the case of 1 degree equals 20 inches, no more than 10 degrees of longitude can be included in the grid. Since the projection is drawn exactly to scale, a mosaic can later be built of the entire area. Depending on the type of data read, the track is annotated with the fix number, uncorrected fathoms, corrected meters, uncorrected meters, or residual magnetic intensity.

UNCLASSIFIED

2.1.1 Navigation Data

The program reads the year, date (month and day), hour, minute, latitude, longitude, and fix number from the input tape according to the specified format. The southern latitudes and the western longitudes are preceded by a minus sign. Normally, the track is plotted in a continuous line with every nth fix marked with a square symbol and annotated with the fix number.

2.1.2 Bathymetry Data

The program reads the year, date (month and day), hour, minute, latitude, longitude, and uncorrected fathoms from the input tape according to the specified format. The southern latitudes and the western longitudes are preceded by a minus sign. The program can convert uncorrected fathoms to uncorrected meters. Each fix is plotted with a plus symbol (+) and annotated with the type of bathymetry wanted. (Uncorrected fathoms, meters, or corrected meters).

2.1.3 Magnetics Data

The program reads the year, date (month and day), hour, minute, latitude, longitude, and residual magnetic intensity from the input tape according to the specified format. The southern latitudes and the western longitudes are preceded by a minus sign. Each fix is plotted with a plus symbol (+) and annotated with the residual magnetic intensity.

2.2 Problem Background

This program eliminates the problem of plotting vast amounts of data on a geographical chart by hand. The polar stereographic projection is used because it has little distortion in the high latitudes.

3.0 USAGE

3.1 Calling Sequence or Operation Procedure

Not applicable.

3.2 Arguments, Parameters, and/or Initial Conditions

Not applicable.

3.3 Space Required (Decimal and Octal)

3.3.1 Unique Storage

10757 octal (4591 decimal) exclusive of system library functions.

3.3.2 Common Blocks

Blank common /1/, /3/, /5/, /7/, /8/, /11/, /12/.

3.3.3 Temporary Storage

None.

3.4 Messages and Instructions to the Operator

None.

3.5 Error Return, Messages, and Codes

Series contain no data.
Improper parameter-plot commands canceled.
Warning: No names allowed in last command.
No name supplied for series — data ignored.
No format established for data.
Warning: Data card out of sequence.

3.6 Informative Messages to the User

None.

3.7 Input

The actual format of the data on the input tape, the map parameters, the title, and the command words are read in via input cards. The data to be plotted and annotated are read in via magnetic tape on logical unit 15. Appendix A shows samples of our data formats on the input tape. Appendix B is a complete description of the input deck setup.

3.8 Output

The program prints on the standard printer (logical unit 61) the data format, map title, chart parameters, number of data points read in, and number of data points plotted on the map. It writes the plotting instructions on a magnetic tape (logical unit 40).

3.9 Formats

Appendix B, which shows the program deck structure, describes the formats.

3.10 External Routines and Symbols

ATAN2, SQRTF, SINF, COSF, ATANF, XMODF, BACKFILE, SKIPFILE,

UNCLASSIFIED

PLOTS, NUMBER, STOPPLOT, PLOT, SYMBOL.

3.11 Timing

The time required depends on the size of the grid and the number of data read and plotted.

3.12 Accuracy

The grid is reproduced exactly to scale.

3.13 Cautions to Users

None.

3.14 Program Deck Structure

See Appendix B.

3.15 References — Literature

The UCSD Hypermap Programs by R.L. Parker, University of California, San Diego.

GRENHY by M.J. Kertyzak and J.D. Phillips, Woods Hole Oceanographic Institute, Woods Hole, Massachusetts.

M.L. Blodgett and J.V. Massingill, "A Program for Storing Oceanographic Data on Magnetic Tape," NRL Report 7861, March 1975.

4.0 METHOD OR ALGORITHM

Not applicable.

5.0 FLOW CHART AND/OR SOURCE LANGUAGE LISTING

Flow chart and listing are given in Appendix C.

6.0 COMPARISON

There are no other known programs available for comparison.

7.0 TEST METHOD AND RESULTS

Used and tested successfully on a CALCOMP plotter by the Environmental Sciences Section of the Propagation Branch, NRL Acoustics Division

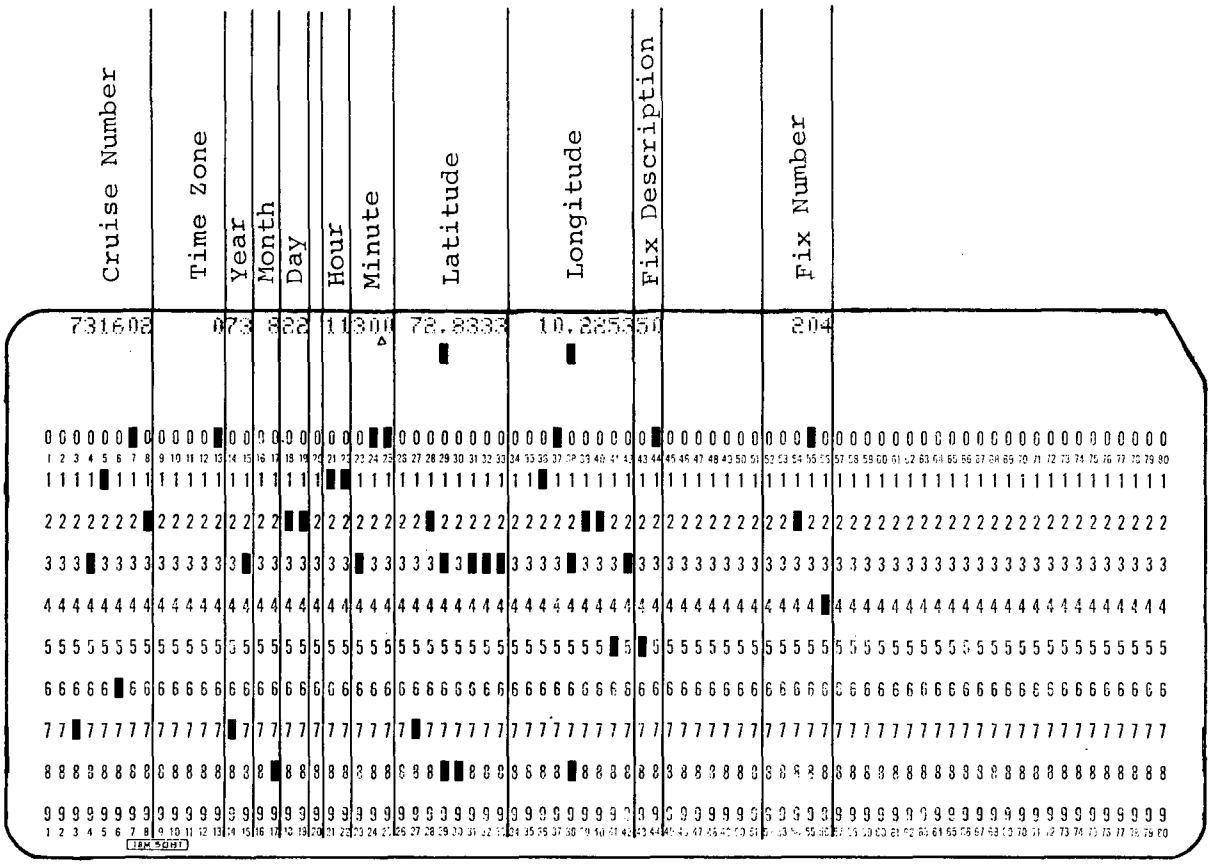
8.0 REMARKS

None.

UNCLASSIFIED

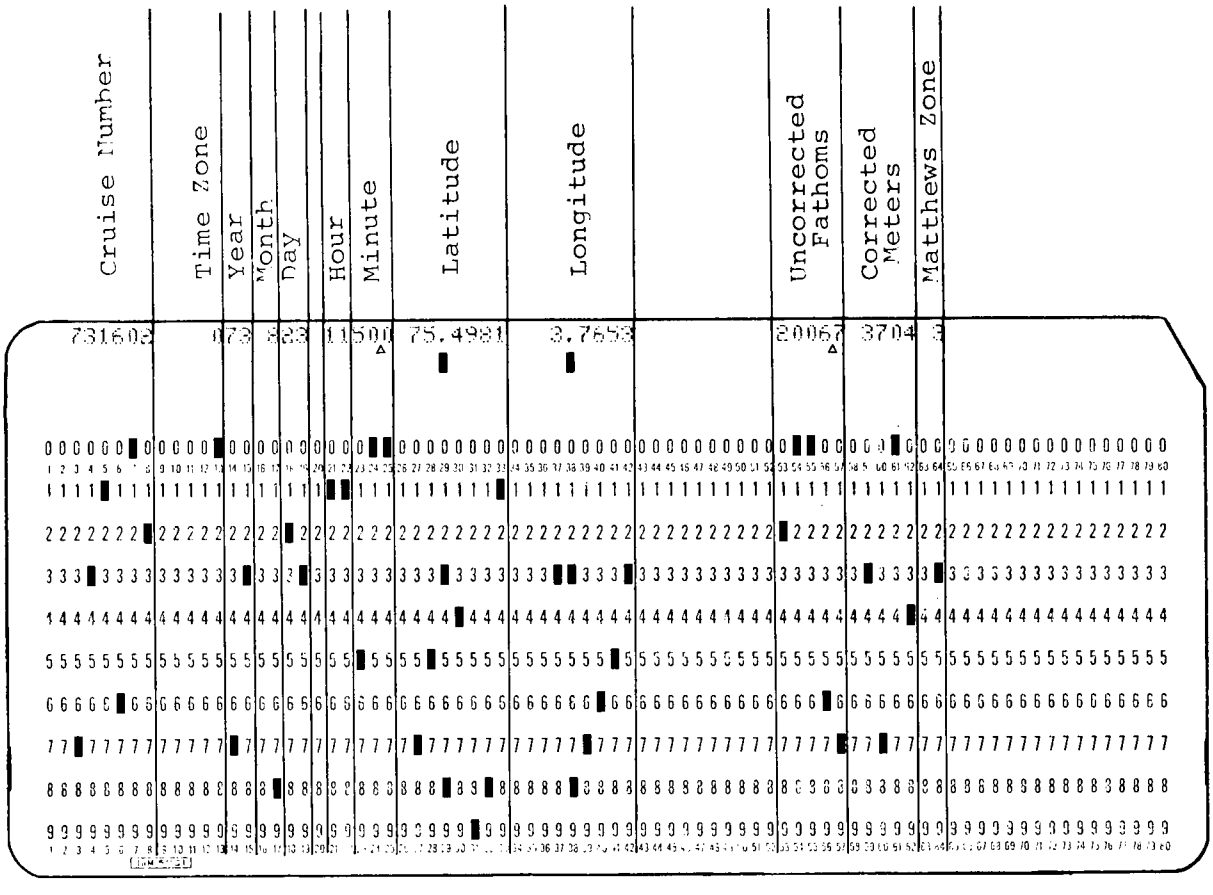
APPENDIX A
Sample Input Data Record

NAVIGATION RECORD



[Δ]Implies a decimal point.

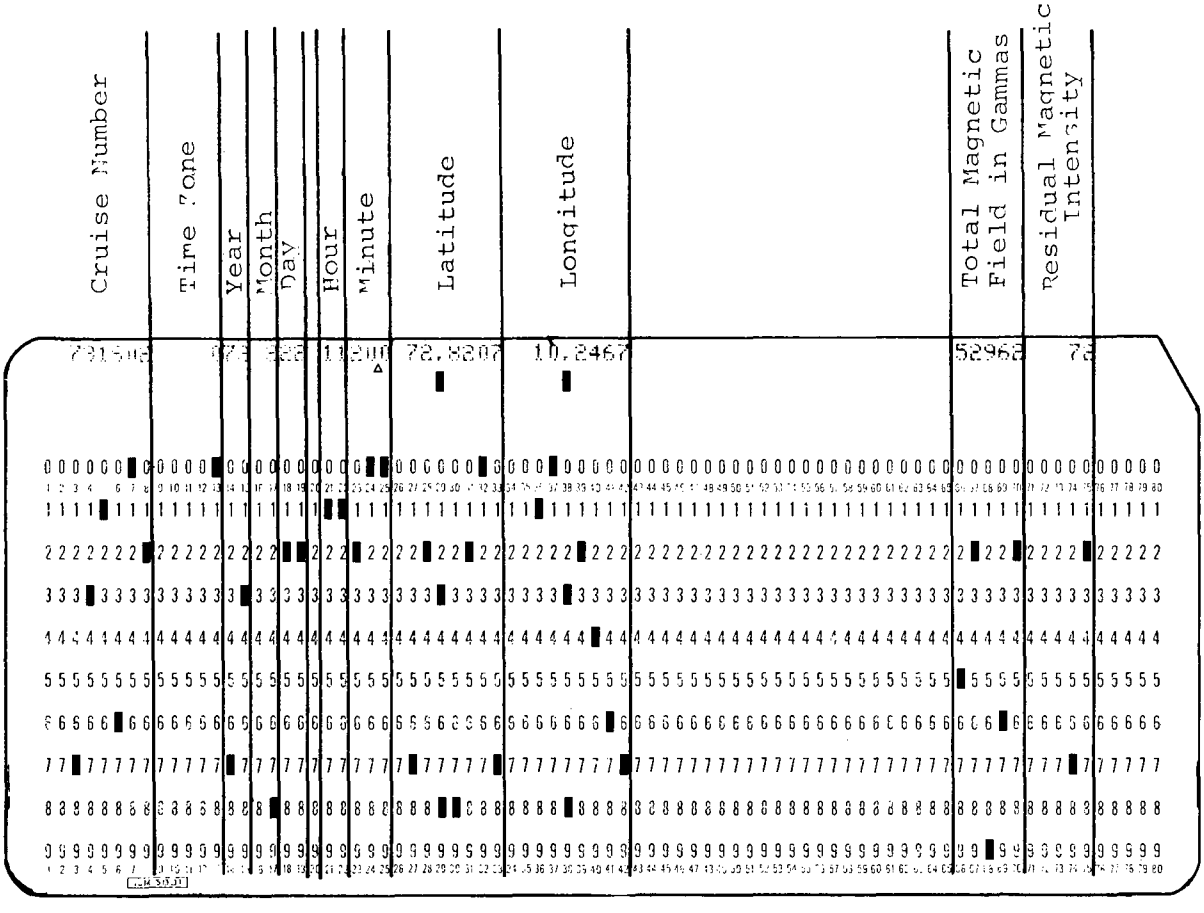
BATHYMETRY RECORD



[Δ]Implies a decimal point.

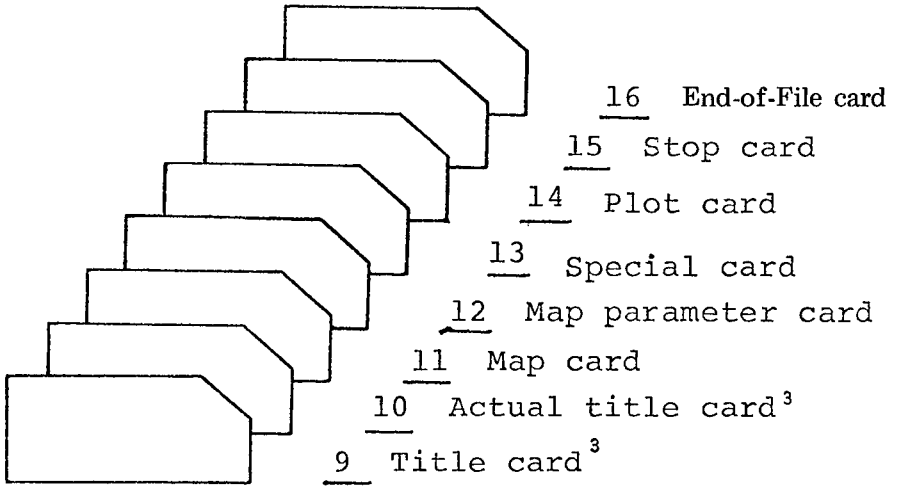
UNCLASSIFIED

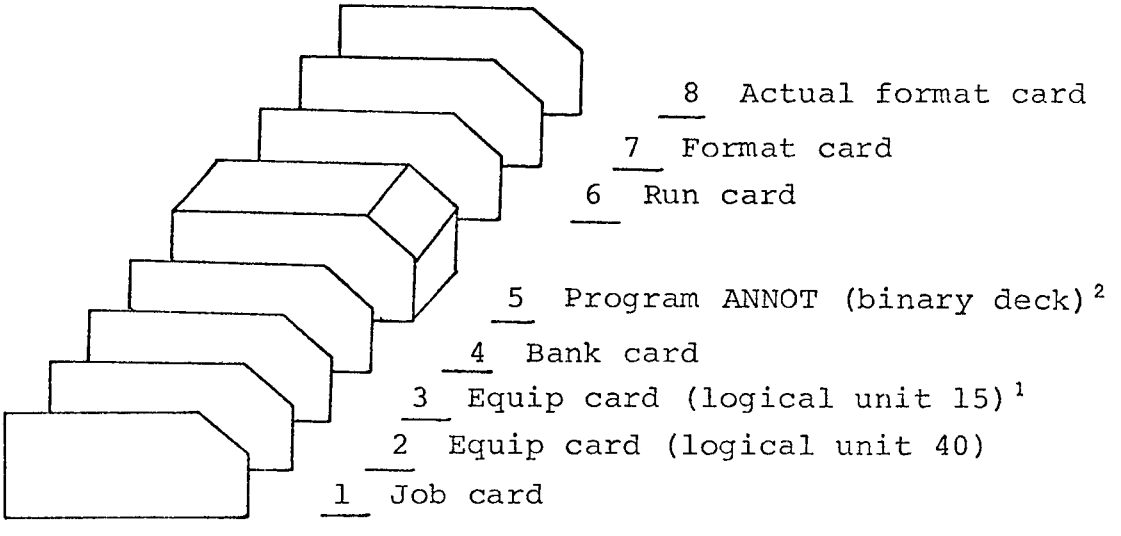
MAGNETICS RECORD



[Δ]Implies a decimal point.

APPENDIX B
Deck Assembly for Program Annot





[1]The program uses scratch tapes on logical units 20 and 06, but no Equip cards are required, since the drum is used.

[2]If the Fortran source deck is used instead of the binary deck, a Fortran card is required after the Bank card. In addition, a Scope card and Load card must follow the source deck.

[3]These two cards are not required by the program; both cards may be present or both omitted.

UNCLASSIFIED

Number	Card Title	Column Number	Description
1	Job	1-21	7/9 JOB, charge No., ID No., time. See page 2-2 of the 3600/3800 Computer System Drum Scope Manual.
2	Equip	1-18	7/9 EQUIP; 40 = **, WO, LO
3	Equip	1-18	7/9 EQUIP, 15 = **, RO, HI 40, 15 = logical unit number. RO = read only. WO = write only. LO = low density. HI = high density. See page 2-3 of the 3600/3800 Computer System Drum Scope Manual.
4	Bank		— /0/7/9 BANK, (0), /1/ See page 7-17 of the 3600/3800 Computer System Drum Scope Manual.
5	Program ANNOT	Deck of cards	This is the main program with associated subroutines. If the Fortran source deck is used instead of the binary deck, a Fortran card is required after the Bank card. The Fortran card reads 7/9 FTN, L, R, X. In addition, a Scope card with SCOPE starting in column 10 and a LOAD must follow the source deck.
6	Run	1-13	7/9 RUN, T, P, R, M, D T = time limit in minutes. P = maximum number of print or write operations. R, M, D may be left blank. See page 2-15 of the 3600/3800 Computer System Drum Scope Manual.
7	Format	1-6	FORMAT This command word establishes the format of the input data to be read in via magnetic tape on logical unit 15.
8	Actual Format	1-?	(13X, I2, I4, 1XI2, F3.1, F8.4, F9.4, 28XF5, 5X) This format should be replaced by the desired input format. The format must be enclosed in parentheses and left-justified. Via this format the program reads the year, date, hour, minute, latitude, longitude, value to be annotated: fix numbers for navigation, uncorrected fathoms, corrected meters (uncorrected meters, see card 12, column 2), and residual magnetic intensity.

Number	Card Title	Column Number	Description
			The formats for reading the three data types on our input tapes are: a. Navigation (13X, I2, I4, 1XI2, F3.1, F8.4, F9.4, 9XF5) b. Bathymetry — uncorrected fathoms (13XI2, I4, 1XI2, F3.1, F8.4, F9.4, 10XF5) c. Magnetics (13XI2, I4, 1XI2, F3.1, F8.4, F9.4, 28XF5).
9	Title	1-5	TITLE. This command allows the user to label the chart. This is a nonobligatory card.
10	Actual Title	1-80	Arctic Basin. . . . The title that is appropriate for the chart being plotted may be punched anywhere in the 80 columns. This is a nonobligatory card.
11	Map	1-3	MAP. This command word allows the user to define the chart.
12	Map Parameters	2	0 or 1 0 = the program will read the values to be annotated from the input tape (fix numbers for navigation, uncorrected fathoms, and residual magnetic intensity). 1 = the program will convert uncorrected fathoms to uncorrected meters and use this value to annotate the bathymetry track.
		4	0 or 1 0 = every fix will be plotted with a (+) symbol and annotated. 1 = the track will be plotted in a continuous line, and every nth fix (columns 63-70) will be marked with a small square and annotated.
		6	0, 1, or 2 Number of files to be skipped over on the input tape. There are a maximum of three files on our GEODATA tapes.

UNCLASSIFIED

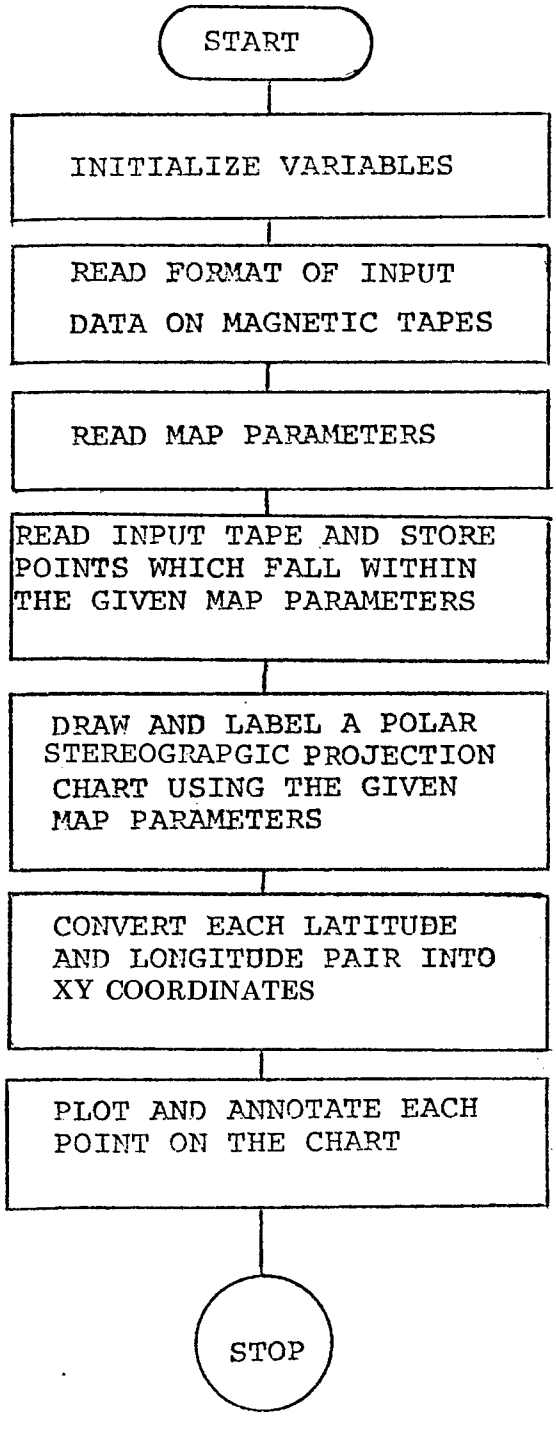
Card Number	Card Title	Colunm Number	Description
		17-20	20.0 The physical height of the chart to be drawn. To obtain this figure, measure the number of inches/degrees of latitude and multiply by the total number of degrees of latitude to be drawn.
		23-28	82.5 The degree of latitude at the base of the chart (the southernmost latitude). This value may be either a whole or a half degree. (A minus sign precedes southern latitudes).
		31-36	84.5. The northernmost degree of latitude. The difference between the degrees of latitude should be an integer. (A minus sign precedes southern latitudes).
		39-44	—15.0. The westernmost degree of longitude. (A minus sign precedes western longitudes).
		47-52	05.0. The easternmost degree of longitude. (A minus sign precedes western longitudes).
		53-60	12 (right justified). The n of the nth navigation fix to be annotated. This value is specified only if there is a 1 in column 8 of this card.
		61-68	02250330 The date and time of the first data point to be plotted. (The first two digits are for the month, the second two digits are for the day, the third two digits are for the hour and the fourth two digits are for the minutes).
		69-76	02280550 The date and time of the last data point to be plotted. (The first two digits are for the month, the second two digits are for the day, the third two digits are for the hour and the fourth two digits are for the minutes). All data on and between these dates and times will be plotted if they fall within the defined chart.

Number	Card Title	Column Number	Description
13	Special Values	1-14	SPECIAL VALUES This command allows the user to associate the name values with the series of data points read from the input tape on logical unit 15. The program will store only those data points which fall on the define chart and which fall on or between the two dates specified on the Map Parameter Card.
14	Plot	1-11	PLOT VALUES This command causes the named series to be plotted.
15	Stop	1-4	STOP This command terminates the program.
16	End-of-File		Terminates the run.

UNCLASSIFIED

APPENDIX C1
Flow Chart

PROGRAM ANNOT



APPENDIX C2
Source Language Listing

```
      PROGRAM ANNOT
      DIMENSION IBUF(254)
      DIMENSION NAME(6)
      COMMON DELAT,DELON,XPOLAT,XPOLON
      COMMON LX
      COMMON W,DIST,ANOMCK,CHANGE,NP,LKK,KNUM,INUM,GINCH
      COMMON ANOM(     2),III,JJJ,KKK,XLAST
      COMMON X(2)
      COMMON INP,IO,U,V,XLAT,XLON
      COMMON POLAT,POLONG,ROT,UMIN,UMAX,VMIN,VMAX,HEIGHT,NLAT,NLON,
     +IPROJ,IBOX,SCALE,ISYMB,ILINE
      COMMON NALL,ISTART(11),NAMES(10,6),LENG,LAST
      COMMON YES,TITLE(10),ICOL,IA(76)
      COMMON/1/IDATE(2) ,MEDN(2)
      DATA(KEY1=4HFORM),(KEY2=4HREAD),(KEY3=4HCOMP),(KEY4=4HSPEC),(KEY5=
     13HMAP),(KEY6=4HTITL),(KEY7=4HHUND),(KEY8=4HPLOT),(KEY9=4HPRIN),(KE
     2Y10=4HSTOP),(KEY11=4HERAS),(KEY12=4HSTOR),(KEY13=4HRECA),(KEY14=4H
     3OTHE),(KEY15=4HPOLE),(KEY16=4HTURN),(KEY17=4HFPLO)
      REWIND 15
      REWIND 20
      REWIND 30
      REWIND 05
      REWIND 06
      INP=60
      IO=61
      KNUM=0
      LKK=0
      INUM=0
      JJJ=0
      LX=0
      POLAT=0.0
      POLONG=0.0
      ROT=0.0
      UMIN=-3.1
      UMAX=3.1
      VMIN=-1.5
      VMAX=1.5
      HEIGHT=8.0
      XLAST=0
      IGRID=0
      IPROJ=3
      IBOX=1
      NALL=0
      LAST=0
      ISTART(1)=1
      YES=-100.0
      ICOL=100
C KKK IS A INDICATOR IF=0 WILL ONLY READ TRACK SERIES FROM CALCM
C  IF=1 WILL CALCULATE ANOMALY SERIES AS WELL
      KKK=1
      DIST=10.
      ANOMCK=1500.
      CHANGE=20.
C  NP==OF POINTS TO BE INPUTED
      NP=2800
      GINCH=2000.
```

UNCLASSIFIED

```
      W=1.
      DELAT=0.
      DELON=0.
      CALL PLOTS(IBUF,254,40,29)
 10   CALL NEXT(KEY,NAME)
      IF(KEY,EQ,KEY1) CALL INPUT(NAME,1)
      IF(KEY,EQ,KEY2) CALL INPUT(NAME,2)
      IF(KEY,EQ,KEY3) CALL INPUT(NAME,3)
      IF(KEY,EQ,KEY4) CALL INPUT(NAME,4)
      IF(KEY,EQ,KEY5) CALL INPUT(NAME,5)
      IF(KEY,EQ,KEY6) CALL INPUT(NAME,6)
      IF(KEY,EQ,KEY7) CALL INPUT(NAME,8)
      IF(KEY,EQ,KEY8) CALL OUTPUT(NAME,1)
      IF(KEY,EQ,KEY9) CALL OUTPUT(NAME,2)
      IF (KEY ,EQ. KEY10) GO TO 100
C
C  IF THE TAPE STORAGE AND RECOVERY PROCEDURES ARE NOT REQUIRED THE 2 FOLLOWING
C  CALLS MAY BE REMOVED TOGETHER WITH THE ROUTINE *STORE*
C     IF(KEY,EQ,KEY12) CALL STORE(NAME)
C     IF(KEY,EQ,KEY13) CALL RECALL(NAME)
C  ENTRY TO *OTHER* WITHOUT USING *INPUT* - THIS ALLOWS *OTHER* TO CALL *INPUT*
      IF(KEY,EQ,KEY14) CALL OTHER(X,X(LAST+1))
C     IF (KEY ,EQ.KEY15) CALL ROTATE (NAME,1)
C     IF (KEY ,EQ. KEY16) CALL ROTATE(NAME,2)
C     KEY 17 WAS ADDED 300473
C     KEY 17 IS A FAKE PLOT
      IF(KEY,EQ,KEY17)CALL OUTPUT(NAME,3)
      GO TO 10
 100  CALL PLOTS(0,0)
      CALL STOPPLOT
      END
```

```
      SUBROUTINE NEXT(INSTR,NAME)
C
C  THIS PROGRAM ATTEMPTS TO PROVIDE A MACHINE-INDEPENDENT ROUTINE FOR READING
C  CONTROL CARDS IN SUPERMAP, THE WORD LENGTH OF THE MACHINE IS REQUIRED TO BE
C  AT LEAST FOUR BCD CHARACTERS LONG, THIS IS MET BY ALL MACHINES LIKELY TO BE
C  ENCOUNTERED.
C
      DIMENSION NAME(6)
      COMMON DELAT,DELON,XPOLAT,XPOLON
      COMMON LX
      COMMON W,DIST,ANOMCK,CHANGE,NP,LKK,KNUM,INUM,GINCH
      COMMON ANOM(     2),III,JJJ,KKK,XLAST
      COMMON X(2)
      COMMON INP,IO,U,V,XLAT,XLON
      COMMON POLAT,POLONG,ROT,LMIN,LMAX,VMIN,VMAX,HEIGHT,NLAT,NLON,
     *IPROJ,IBOX,SCALE,ISYMB,ILINE
      COMMON NALL,ISTART(11),NAMES(10,6),LENG,LAST
      COMMON YES,TITLE(10),ICOL,IA(76)
      COMMON/1/IDATE(2) ,HEDN(2)
      DATA(IBLANK=1H ),(ICOMMA=1H,),(NULL=1)
10    DO 11 IWORD=1,6
11    NAME(IWORD)=IBLANK
      IWORD=1
15    IF (IWORD.GT.1) RETURN
20    ICOL=ICOL+1
      IF (ICOL.GT.76) GO TO 30
25    IAC=IA(ICOL)
      IF (IAC.EQ.IBLANK .OR. IAC.EQ.ICOMMA) GO TO 15
      IF (IWORD.LE.6) NAME(IWORD)=IA(ICOL)
      IWORD=IWORD+1
      NULL=1
      GO TO 20
30    IF (NULL.EQ.0) GO TO 35
      IF (IWORD.GT.1) RETURN
      READ (INP,1000) INSTR,IA
1000  FORMAT(A4,76A1)
      NULL=0
      NAME(1)=IBLANK
      DO 33 KCOL=1,76
C  KCOL IS USED BECAUSE THE COMPILER SEEMS UNABLE TO ACCEPT THE DO LOOP BELOW IF
C  ICOL IS USED THROUGHOUT.
      ICOL=KCOL
      IF (IA(KCOL).EQ.IBLANK.OR. IA(KCOL).EQ.ICOMMA) GO TO 10
33    CONTINUE
35    NULL=1
      RETURN
      END
```

UNCLASSIFIED

```
                                        IDENT      NEXT
   PROGRAM LENGTH                 00200
   ENTRY POINTS     NEXT          00012
   BLOCK NAMES
                                  00315
                    1             00004
   EXTERNAL SYMBOLS
                    THEND,
                    Q8CDICT,
                    TSH,
                    SLI,
                    QNSINGL,
      00156 SYMBOLS


      SUBROUTINE SEARCH(NAME,INDEX,N1,N2)
C
C   THIS ROUTINE SEARCHES THE TABLE OF NAMES IN COMMON /TABLES/ FOR THE 6-VECTOR
C   *NAME*, AND IF PRESENT RETURNS THE NUMBER ASSOCIATED WITH IT .
C   THE INTEGERS *N1*,*N2* ARE SET TO THE BEGINNING AND END OF THE ARRAY AS HELD
C   IN STORE, OR IF THE NAME WAS BLANK N1=1 AND N2=LAST.
C
C     REAL NAME,NAMES,IBLANK
      DIMENSION NAME(6)
      COMMON DELAT,DELON,XPOLAT,XPOLON
      COMMON LX
      COMMON W,DIST,ANOMCK,CHANGE,NP,LKK,KNUM,INUM,GINCH
      COMMON ANOM(     2),III,JJJ,KKK,XLAST
      COMMON X(2)
      COMMON INP,IO,U,V,XLAT,XLON
      COMMON POLAT,POLONG,ROT,UMIN,UMAX,VMIN,VMAX,HEIGHT,NLAT,NLON,
     *IPROJ,IBOX,SCALE,ISYMB,ILINE
      COMMON NALL,ISTART(11),NAMES(10,6),LENG,LAST
      COMMON YES,TITLE(10),ICOL,IA(76)
      COMMON/1/IDATE(2) ,HEDN(2)
      DATA(IBLANK=1H )
      IF (NAME(1).EQ.IBLANK) GO TO 30
      IF (NALL.LT.1) GO TO 16
      DO 15 I=1,NALL
      DO 10 J=1,6
      IF (NAMES(I,J).NE.NAME(J)) GO TO 15
10    CONTINUE
      GO TO 20
15    CONTINUE
16    WRITE (IO,1000) NAME
1000  FORMAT(/* SERIES *,6A1,* NOT PRESENT*)
      INDEX=0
      RETURN
20    INDEX=I
      N1=ISTART(I)
      N2=ISTART(I+1)-1
      RETURN
30    INDEX=-1
      N1=1
      N2=LAST
      RETURN
      END
```

```
                                        IDENT      SEARCH
PROGRAM LENGTH                  00225
ENTRY POINTS     SEARCH         00015
BLOCK NAMES
                                00315
                 1              00004
EXTERNAL SYMBOLS
                 THEND,
                 QBQDICT,
                 STH,
                 SLC,
   00154 SYMBOLS
```

UNCLASSIFIED

```
      SUBROUTINE OTHER(XOLD,XNEW)
      REAL LATMIN,LATMAX,LONGMIN,LONGMAX
      DIMENSION IIYR(20),IIDAY(20),IIHR(20),DDMIN(20),RRLAT(20),RRLONG(
     120),AANOMAL(20)
      DIMENSION IFM(20)
      COMMON DELAT,DELON,XPOLAT,XPOLON
      COMMON LX
      COMMON W,DIST,ANGMCK,CHANGE,N1,LKK,KNUM,INUM,GINCH
      COMMON ANGM(     2),III,JJJ,KKK,XLAST
      COMMON X(2)
      COMMON INP,IC,U,V,XLAT,XLON
      COMMON POLAT,POLONG,ROT,UMIN,UMAX,VMIN,VMAX,HEIGHT,NLAT,NLON,
     +IPROJ,IBOX,SCALE,ISYMB,ILINE
      COMMON NALL,ISTART(11),NAMES(10,6),LENG,LAST
      COMMON YES,TITLE(10),ICOL,IA(76)
      COMMON/1/IDATE(2) ,HEDN(2)
      COMMON/3/LATMIN,LATMAX
      COMMON/5/JUDY1,JUDY2,ITM1,ITM2
      COMMON/7/LONGMIN,LONGMAX
      COMMON/8/IFM
      ATMIN=LATMIN
      ATMAX=LATMAX
      DIST2=80.
      HEDNL=279.
      IFLIGHT=1
      IMP=60
      IDAY=0
      ZNINE=99.0
      ZERO=0.0
      DEGRA=1.745329E-2
      CHANGE1=450.-(360.-CHANGE/2.)
      CHANGE2=(450.0-CHANGE/2.)-360.
      M=1
      L=1
      IF(LKK.EQ.1)GO TO 50
      I=1
 500  IF(I.EQ.1)200,201
 200  WRITE(20,701)ZNINE,ZERO,IDAY
      WRITE(30,700) M,ZNINE
C     WRITE(61,800)M,ZNINE,ZERO
C200  X(1)=99.0
C     X(2)=0.0
      M=M+2
C70   IF(IYR-66)501,99,501
C1000 FORMAT(11XI2,1X,I4    ,1X,I2,F4.1,F7.3,F8.3,13X,F5.3XF4.3XF4.8X)
C    1DIST2,HEDNL
C201  READ(IMP,1000        )IYR,IDAY    ,IHR,DMIN,RLAT,RLONG,ANOMAL,
      N=21
 201  IF(N-20) 801,801,977
 977  N=1
 800  READ(15,IFM)IIYR(N),IIDAY(N),IIHR(N),DDMIN(N),RRLAT(N),RRLONG(N),
     1AANOMAL(N)
      IF(IOCHECK.15) 800,779
 779  IF(EOF.15) 777,778
 777  IEND=IEND + 1
      IF(IEND . GE. 1) GO TO 401
```

```
778   N=N + 1
      IF(N.LT.21) GO TO 800
      N=1
801   IYR=IIYR(N)
      IDAY=IIDAY(N)
      IHR=IIHR(N)
      IMIN=DDMIN(N)
      IMIN=IHR*100 + IMIN
      RLAT=RRLAT(N)
      RLONG=RRLONG(N)
      ANOMAL=AANOMAL(N)
      IF(IBOX)350,355,350
350   ANOMAL=ANOMAL*1.8288
355   N=N+1
1000  FORMAT(13X,I2,I4,1XI2,F3.1,F8.4,F9.4,28XF5.5X)
      IF(IDAY.LT.JLDY1)GO TO 201
      IF(RLAT.GT.90.0)GO TO 201
      IF(IDAY.GT.JLDY2) GO TO 811
      IF(IYR.EQ.0) GO TO 811
      IF(RLAT.LT.LATMIN) GO TO 201
      IF(RLAT.GT.LATMAX) GO TO 201
      IF(IDAY.EQ.JLDY1.AND.IMIN.LT.ITM1)GO TO 201
      IF(IDAY.EQ.JLDY2.AND.IMIN.GT.ITM2) GO TO 811
      IF(RLONG.LT.LONGMIN) GO TO 201
      IF(RLONG.GT.LONGMAX) GO TO 201
C201  READ(15,1000)IYR,IDAY,IHR,DMIN,RLAT,RLONG,ANOMAL
401   IF(L-3)70,72,72
811   IF(L-3)99,810,810
810   WRITE(06,71)ANOM(1),HEDN(1)
      GO TO 99
72    WRITE(06,71) ANOM(1),HEDN(1)
71    FORMAT(2F10.4)
C     WRITE(61,75)ANOM(1),HEDN(1)
C75   FORMAT(1H0,3HA H,2F10.4)
70    IF(IEND.EQ.1) GO TO 99
C70   IF(EOF.15) 300,501
C300  IFLIGHT=IFLIGHT + 1
C     IF(IFLIGHT-13)201,201,99
501   IF(L.EQ.1)GO TO 5
      HEDN(  1)=HEDNL
    5 CONTINUE
      ANOM(1)=ANOM(2)
      ANOM(2)=ANOMAL
      IF(I.EQ.1)202,203
202   DIST1=DIST2
      GO TO 19
C     IF DISTANCE EXCEEDS VALUE GIVEN IN DIST LIFT PEN BEFORE PLOTTING
C     NEXT POINT
203   IF((DIST2-DIST1).GT.DIST)204,205
C204  X(M)=99.0
C     X(M+1)=0.0
C     IDATE(M)=0
C     IDATE(M+1)=0
  204 WRITE(20,701)ZNINE,ZERO,IDAY
      WRITE(30,700) M,ZNINE
C     WRITE(61,800)M,ZNINE,ZERO
```

UNCLASSIFIED

```
      DIST1=DIST2
      M=M+2
      GO TO 17
 205  DIST1=DIST2
      GO TO 19
 17   CONTINUE
   19 CONTINUE
C     X(M)=RLAT
C     X(M+1)=RLONG
      IAY1     =IDAY*100 + IYR
      IAY2       =IYR*100
      WRITE(20,701)RLAT,RLONG,ANOMAL
      WRITE(30,700) M,RLAT
C     WRITE(61,800)M,RLAT,RLONG
C800  FORMAT(1H0,I5,2F10.4)
 700  FORMAT(I10,F10.4)
 701  FORMAT(2F10.4,F10.1)
      IF(L.EQ.1)GO TO 60
      HEDN(  1)=450.-HEDN(  1)
      IF(HEDN(  1).GT.360.00)HEDN(  1)=HEDN(  1)-360.0
  910 HEDN(  1)=HEDN(  1)*DEGRA
   60 CONTINUE
      L=L+1
      M=M+2
      IF(I.GT.10)GO TO 20
 12   KK=I+1
 20   I=I+1
      GO TO 500
   99 LENG=M-1
      WRITE(06,71)ANOM(2),HEDN(1)
      REWIND 30
      REWIND 06
      III=M
      N1=I-1
C     WRITE(61,600) LENG,III,N1
C600  FORMAT(1H0,3I20)
C     DO 606 LLL=1,L
C     WRITE(61,605) HEDN(LLL),ANOM(LLL)
C605  FORMAT(1H0,2F10.4)
C606  CONTINUE
      IF(KKK.EQ.0)206,207
 206  LKK=0
      RETURN
 207  LKK=1
      RETURN
C     THE VALUE OF III IS SET IN SUB CONV WHEN SERIES TRACK IS BEING PLOTTED
C  FOR EACH VALUE OF TRACK PLOTTED A CORRESPONDING VALUE OF SERIES ANOM
C IS FOUND (THE SERIES WILL BE AN X,Y VALUE IN INCHES STOREDIN COMMON X(1000)
C  WHEN SERIES ANOM IS PLOTTED SUBROUTINE CONVERT IS BYPASSED
C  SINCE THE SERIES ANOM IS ALREADY IN INCHES
C III IS ONE MORE THAN THE TOTAL LENTH OF ARRAYX(10000)
C     LENG IS THE LENGTH OF EACH SERIES CALCULATED IN *OTHER* WHICH IS STORED
C     IN X(10000) **NOT** THE LENGTH OF THE USED PORTION OF ARRAY X(10000)
   50 LENG=(III-3)/2+INUM
      LKK=2
      RETURN
      END
```

```
                                        IDENT      OTHER
PROGRAM LENGTH                 01032
ENTRY POINTS     OTHER         00247
BLOCK NAMES
                               00315
                 1             00004
                 3             00002
                 5             00004
                 7             00002
                 8             00024
EXTERNAL SYMBOLS
                 THEND,
                 01C10100
                 08CDICT,
                 08CIFE0F
                 08CIFI0C
                 REW,
                 TSW,
                 STW,
                 ONSINGL,
   00312 SYMBOLS
```

UNCLASSIFIED

```
      SUBROUTINE CONV(LAT,IPEN,IPROJ)
C
CCC BASIC SUBROUTINE CONTAINING 12 STANDARD  PROJECTIONS.
C
      REAL LAT
      DIMENSION HEAD(2)
      DIMENSION S(2)
      DIMENSION LAT(2)
      DIMENSION A(4)
      COMMON DELAT,DELON,XPOLAT,XPOLON
      COMMON LX
      COMMON W,DIST,ANOMCK,CHANGE,NP,LKK,KNUM,INUM,GINCH
      COMMON ANOM(     2),IIII,JJJ,KKK,XLAST
      COMMON X(2)
      COMMON INP,IC,U(2),XLAT,XLON
      COMMON POLAT,PHIC,ROT,UV(4),HEIGHT,NLAT,NLON,III(2),SCALE,ISYMB
     1,ILINE
      COMMON NALL,ISTART(11),NAMES(10,6),LENG,LAST
      COMMON YES,TITLE(10),ICOL,IA(76)
      COMMON/1/IDATE(2) ,HEDN(2)
      DATA(RATIO=1,00092),(ECCSQ=0,0067227)
      DATA(F=,0174533),(ULAST=9999,)
      DATA((A(I),I=1,4)=1,37027,-,28771,,080412,-,14842)
      DATA(EE=1,7182818)
      RCF=57,2957795
      IDAY=0,0
      ZERO=0,0
      ZNINE=99,0
      PI=3,14159
C     UV(4)=VMAX(UV(3)=VMIN
      SCAL=HEIGHT/(UV(4)-UV(3))
C
C  GENERAL ENTRY POINT FOR ALL AZIMUTHAL PROJECTIONS.
C
1     SINPH=SIN(F*(LAT(2)-PHIO))
      COSPH=COS(F*(LAT(2)-PHIO))
      SINRT=-COS(F*ROT)
      COSRT= SIN(F*ROT)
      SINLO= SIN(F*POLAT)
      COSLO =COS(F*POLAT)
      SINLA=SIN(F*LAT(1))
      COSLA=SQRT(1,-SINLA*SINLA)
      COSA=SINLA*SINLO+COSLA*COSLO*COSPH
      SINA=SQRT(1,0001-COSA*COSA)
      SINB=COSLA*SINPH/SINA
      COSB=(SINLA*COSLO-COSLA*SINLO*COSPH)/SINA
      GO TO 70
101   U(1)= R*(COSB*COSRT-SINB*SINRT)
      U(2)=-R*(SINB*COSRT+COSB*SINRT)
C    THIS SECTION CALCULATES SERIES ANOM
C     I COULD HAVE BEEN REPLACED WITH III BUT IT WAS NOT WORTH THE EFFORT
C     JJJ IS A COUNTER, IT IS IN COMMON BECAUSE IT NEEDS TO BE INCREMENTED
C     EACH TIME IT SWITCHES FROM *OUTPUT* TO *CONV*
C     THE VALUE OF III IS SET IN *OTHER* IT IS THE NEXT UNUSED POSITION
C     IN ARRAY X(12000) WHICH WILL START SERIES ANOM.
      IF(LKK ,NE,1)GO TO 18
```

```
      JJJ=JJJ + 1

      K=JJJ
      J=K-1
      IF(J)30,30,31
 31   ANOM(1)=ANOM(2)
      HEDN(1)=HEDN(2)
      READ(06,33)ANOM(2),HEDN(2)
 33   FORMAT(2F10.4)
C     WRITE(61,34)ANOM(1),ANOM(2),HEDN(1),HEDN(2)
C34   FORMAT(1H0,7HANOMALY, 4F10.4)
C   KNUM==TIMES A 99. WAS FOUND IN SERIES TRACK
 30   M=2*J-1+2*KNUM
      IF(M)800,800,801
 801  READ(30,810)MVALUE,XVAL
      IF(EOF,30)200,201
 200  MVALUE=0
      XVAL=88.0
      GO TO 800
C     WRITE(61,605)MVALUE,XVAL
C605  FORMAT(1H0,7HMVALUE ,I5,F10.4)
 201  IF(MVALUE-M) 801,800,802
 802  REWIND 30
      GO TO 801
 810  FORMAT(I10,F10.4)
 800  I=IIII
C     WRITE(61,600) JJJ,KNUM,M,I
C600  FORMAT(1H0,4I10)
      IF(K.EQ.1) 501,502
 501  S(1)=U(1)
      S(2)=U(2)
      GO TO 20
 502  IF(K.EQ.2)503,504
 503  XDIFF=U(1)-S(1)
      YDIFF=U(2)-S(2)
      GO TO 17
 504  XDIFF=(XDIFF+W*(U(1)-S(1)))/(W+1)
      YDIFF=(YDIFF+W*(U(2)-S(2)))/(W+1)
   17 IF(XVAL.NE.99.0)GO TO 15
      KK=M+1
C     X(I)=99.0
C     X(I+1)=0.0
      WRITE(05,700) ZNINE,ZERO,IDAY,IDAY
C     WRITE(61,606) ZNINE,ZERO
C606  FORMAT(1H0,5HANOM ,2F10.4)
      XLAST=ZNINE
      KK=I+1
C     WRITE(61,12)I,X(I),KK,X(KK)
C12   FORMAT(* X*,I4,*=*,F7.3,5X,*X*,I4,*=*,F8.3)
      I=I+2
      KNUM=KNUM+1
C  FOLLOWING IS TRUE WHEN YOU ARE AT THE SAME POINT AS BEFORE
      IF((U(1)-S(1)).EQ.0.).AND.(U(2)-S(2)).EQ.0.)GO TO 19
   15 HEAD(2)=ATAN2(YDIFF,XDIFF)
      IF(HEAD(2).LT.0.)HEAD(2)=2*PI+HEAD(2)
C  HEAD(2) IS AN ANGLE BETWEEN 0 AND 2*PI
```

UNCLASSIFIED

```
      IF(K.EQ.2)GO TO 19
      IF(XVAL.EQ.99.0)GO TO 19
      DO 2 N=1,3,2
      IF(HEAD(2).GT.N*PI/2.AND.HEAD(2-1).LE.N*PI/2)GO TO 22
      IF(HEAD(2).LT.N*PI/2.AND.HEAD(2-1).GE.N*PI/2)GO TO 22
    2 CONTINUE
      IF(ABS(HEDN(2)-HEDN(  1)).GT.(CHANGE*F))402,310
C402  X(I)=99.
C     X(I+1)=0.0
 402  WRITE(05,700)ZNINE,ZERO,IDAY,IDAY
C     WRITE(61,606) ZNINE,ZERO
      XLAST=ZNINE
      INUM=INUM+1
      GO TO 305
 305  KK=I+1
C     WRITE(61,12)I,X(I),KK,X(KK)
      I=I+2
  310 IF(ABS(ANOM(2)-ANOM(  1)).GT.ANOMCK) GO TO 300
      GO TO 19
 300  IF(XLAST .NE.99.0)403,404
C403  X(I)=99.
C     X(I+1)=0.0
 403  WRITE(05,700) ZNINE,ZERO,IDAY,IDAY
C     WRITE(61,606) ZNINE,ZERO
      XLAST=ZNINE
      I=I+2
      INUM=INUM + 1
      GO TO 315
 404  CONTINUE
      GO TO 19
 315  KK=I+1
C     WRITE(61,12)I,X(I),KK,X(KK)
      GO TO 19
C NOW YOU WANT TO LIFT UP PEN BEFORE YOU PLOT NEXT PT.
C  22 X(I)=99.0
C     X(I+1)=0.0
 22   WRITE(05,700) ZNINE,ZERO,IDAY,IDAY
C     WRITE(61,606) ZNINE,ZERO
      XLAST=ZNINE
      KK=I+1
C     WRITE(61,12)I,X(I),KK,X(KK)
      I=I+2
C INUM IS A COUNTER TO KEEP TRACK OF HOW MANY TIMES A 99.0 WAS ADDED TO ARRAY
C ANOM DUE TO A CROSSING OF NORTH OR SOUTH
C     INUM COUNTS THE NUMBER OF EXTRA POINTS ADDED TO SERIES ANOM THAT
C     DO NOT OCCUR IN SERIES TRACK DUE TO SUCH THINGS AS THE VALUE OF
C     ANOMCK OR CHANGE WAS EXCEEDED
C     THE SIZE OF SERIES ANOM WILL ALWAYS BE .GE. TO THE SIZE OF SERIES TRACK
      INUM=INUM+1
      GO TO 310
  19  IF(0..LE.HEAD(2))411,408
 411  IF(HEAD(2).LE.PI/2)407,408
 407  ISIGN=1
      GO TO 23
 408  IF(3.*PI/2..LT.HEAD(2))412,410
 412  IF(HEAD(2).LT.2.*PI)409,410
```

```
 409  ISIGN=1
      GO TO 23
 410  ISIGN=-1
   23 XONE=S(1)+ISIGN*ANOM(2)*SIN(HEAD(2))/(GINCH*SCAL)
      XTWO  =S(2)+ISIGN*ANOM(2)*COS(HEAD(2))/(GINCH*SCAL)
C  23 XONE=S(1)+ISIGN*ANOM(2)                /(GINCH*SCAL)
C     XTWO  =S(2)+ISIGN*ANOM(2)                /(GINCH*SCAL)
      WRITE(05,700) XONE,XTWO,IDAY,IDAY
C     WRITE(61,606) XONE,XTWO
 700  FORMAT(2F10.4,2I10)
      XLAST=XONE
      IWRIT=I+1
C     WRITE(61,601) I,X(I),X(IWRIT),S(1),S(2),ISIGN,J,ANOM(J),GINCH,SCAL
C601  FORMAT(1H0,I5,4F10.4,2I5,3F10.4)
      S(1)=U(1)
      S(2)=U(2)
      HEAD(1)=HEAD(2)
   16 I=I+2
   20 IIII=I
   18 RETURN
C
C  STEREOGRAPHIC WITH ORIGIN AT POLAT,POLONG
C
70    R=2.0*SINA/(1.000001+COSA)
      GO TO 101
      END
```

```
                                        IDENT     CONV
PROGRAM LENGTH                  01002
ENTRY POINTS     CONV           00033
BLOCK NAMES
                                00315
                 1              00004
EXTERNAL SYMBOLS
                 01C10100
                 THEND,
                 01C04100
                 QBCDICT,
                 ATAN2
                 SQRTF
                 SINF
                 COSF
                 QBCIFECF
                 REW,
                 TSW,
                 STW,
                 QNSINGL,
   00275 SYMBOLS
```

UNCLASSIFIED

```
      SUBROUTINE ANGANO (XPT,XA,YPT,YA,ANG)                              G   1
C     COMPUTES ANGLE OF ROTATION FOR ANNOTATION                          G   2
C     PROGRAMMER - LEON LA LUMIERE - CODE 8174 - 10 DEC 1971             G   3
C                                                                        G   4
      PI=3.141592653                                                     G   5
      XRAD=0.01745329                                                    G   6
      TEMP=ANG                                                           G   7
      IF (XPT-XA) 1,10,1                                                 G   8
    1 ARG=(YPT-YA)/(XPT-XA)                                              G   9
      IF (ARG) 2,4,6                                                     G  10
    2 IF (YPT-YA) 3,4,9                                                  G  11
    3 ANG=(ATAN(ARG))/XRAD                                               G  12
      GO TO 14                                                           G  13
    4 IF (XPT-XA) 5,10,10                                                G  14
    5 ANG=PI/XRAD                                                        G  15
      GO TO 14                                                           G  16
    6 IF (YPT-YA) 7,4,8                                                  G  17
    7 ANG=(ATAN(ARG)+PI)/XRAD                                            G  18
      GO TO 14                                                           G  19
    8 ANG=(ATAN(ARG))/XRAD                                               G  20
      GO TO 14                                                           G  21
    9 ANG=(ATAN(ARG)+PI)/XRAD                                            G  22
      GO TO 14                                                           G  23
   10 IF (YPT-YA) 11,13,12                                               G  24
   11 ANG=270.0                                                          G  25
      GO TO 14                                                           G  26
   12  NG=90.0                                                           G  27
       O TO 14                                                           G  28
   13  NG=TEMP                                                           G  29
   14  ETURN                                                             G  30
       ND                                                                G  31
```

```
      SUBROUTINE FIXANG(XPT,YPT,ANG,CHAR,KOUNT,ILINE,IPEN)
C     ANNOTATES FIX NUMBERS                                              I   2
C     PROGRAMMER - LEON LA LUMIERE - CODE 8174 - 10 DEC 1971             I   3
C                                                                        I   4
      COMMON /11/ ITAL
      COMMON/12/IPT
      ITAL=ITAL + 1
      AHT=.07
      NFIG=5
      JDTEST=-1
      XRAD=0.01745329                                                    I   5
      CNG=ANG-90.                                                        I   6
      IF (MOD(KOUNT,2).NE.0) GO TO 1                                     I   7
C     ANNOTATES FIGURES TO THE RIGHT HAND SIDE OF THE DATA POINT         I   8
      XP=XPT                                                             I   9
      YP=YPT                                                             I  10
      GO TO 4                                                            I  11
C     ANNOTATES FIGURES TO THE LEFT HAND SIDE OF THE DATA POINT          I  12
    1 ENG=(CNG+180.)*XRAD                                                I  13
      IF (NFIG.GT.4) GO TO 2                                             I  14
      HYP=AHT*(6./7.)*(NFIG+1)                                           I  15
      GO TO 3                                                            I  16
    2 HYP=AHT*(6./7.)*NFIG                                               I  17
    3 YY=HYP*SINF(ENG)                                                   I  18
      XX=HYP*COSF(ENG)                                                   I  19
      YP=YPT+YY                                                          I  20
      XP=XPT+XX                                                          I  21
    4 IF (NFIG.GT.4) GO TO 5                                             I  22
      CALL NUMBER (XP,YP,AHT,IFN,CNG,2HI4)                               I  23
      CALL SYMBOL (XPT,YPT,0.02,3,0,0,-1)                                I  231
      GO TO 7                                                            I  24
    5 IF (NFIG.GT.5) GO TO 6                                             I  25
C ILINE=0 EVERY POINT    ILINE=1  EVERY N TH POINT ANNOTATED
      IF(ILINE) 11,10,11
 11   IF(MOD(ITAL,IPT).NE.1) GO TO 9
      CALL PLOT(XPT,YPT,2)
      CALL NUMBER(XP,YP,AHT,CHAR,CNG,2HF5)
      CALL PLOT(XPT,YPT,3)
      CALL SYMBOL(XPT,YPT,0.105,0,0,0,-1)
      GO TO 7
 9    CALL PLOT(XPT,YPT,2)
      GO TO 7
 10   CALL NUMBER (XP,YP,AHT,CHAR,CNG,2HF5)
      CALL PLOT(XPT,YPT,3)
 20   CALL SYMBOL(XPT,YPT,0.020,3,0,0,-1)
      GO TO 7                                                            I  28
    6 CALL NUMBER (XP,YP,AHT,IFN,CNG,2HI7)                               I  29
      CALL SYMBOL (XPT,YPT,0.02,3,0,0,-1)                                I  30
    7 CALL PLOT (XPT,YPT,3)                                              I  31
      IF (JDTEST.EQ.-1) GO TO 8                                          I  32
      IF (JUDY.EQ.JDTEST) GO TO 8                                        I  33
      JDTEST=JUDY                                                        I  34
      HYP=AHT*(6./7.)*(NFIG+4)                                           I  35
      ENG=CNG*XRAD                                                       I  36
      YY=HYP*SINF(ENG)                                                   I  37
      XX=HYP*COSF(ENG)                                                   I  38
```

UNCLASSIFIED

```
      YP=YPT+YY                                                          I  39
      XP=XPT+XX                                                          I  40
      CALL NUMBER (XP,YP,AHT,JUDY,CNG,2HI3)                              I  41
      CALL PLOT (XPT,YPT,3)                                              I  42
    8 RETURN                                                             I  43
      END                                                                I  44.
```

```
                                        IDENT      FIXANO
      PROGRAM LENGTH                 00450
      ENTRY POINTS     FIXANO        00010
      BLOCK NAMES
                       11            00001
                       12            00001
      EXTERNAL SYMBOLS
                       01C04100
                       08CDICT,
                       NUMBER
                       SYMBOL
                       PLOT
                       XMODF
                       SINF
                       COSF
         00143 SYMBOLS
```

```
      SUBROUTINE INPUT(NAME,IGO)
C
C  SUBROUTINE FOR INPUT OF DATA AND OTHER PARAMETERS, 8 MODES OF OPERATION
C
      REAL LAT
      REAL LATMIN,LATMAX,LONGMIN,LONGMAX
      INTEGER TITLE
      DIMENSION IFM(20),NAME(6)
      COMMON DELAT,DELON,XPOLAT,XPOLON
      COMMON LX
      COMMON W,DIST,ANGMCK,CHANGE,NP,LKK,KNUM,INUM,GINCH
      COMMON ANGM(     2),III,JJJ,KKK,XLAST
      COMMON LAT(2)
      COMMON INP,IC,U,V,XLAT,XLON
      COMMON POLAT,POLONG,ROT,UMIN,UMAX,VMIN,VMAX,HEIGHT,NLAT,NLON,
     *IPROJ,IBOX,SCALE,ISYMB,ILINE
      COMMON NALL,ISTART(11),NAMES(10,6),M,LAST
      COMMON YES,TITLE(10),ICOL,IA(76)
      COMMON/1/IDATE(2) ,HEDN(2)
      COMMON/3/LATMIN,LATMAX
      COMMON/5/JUDY1,JUDY2,ITM1,ITM2
      COMMON/7/LONGMIN,LONGMAX
      COMMON/8/IFM
      COMMON/12/IPT
      DATA(IBLANK=1H )
      GO TO (100 ,200,200,400,500,600,200,200                ),IGO
C
C  PRINTED MESSAGES BEFORE RETURNING ARE LOCATED IN THIS SECTION
C
20    WRITE(IO,1020)
1020  FORMAT(5(4H ***),* SERIES CONTAINS NO DATA*)
      GO TO 81
30    WRITE(IO,1030)
1030  FORMAT(5(4H ***),* IMPROPER PARAMETER - PLOT COMMANDS CANCELLED*/)
      IBOX=-3
      RETURN
40    WRITE(IO,1040)
1040  FORMAT(5(4H ***),* WARNING - NO NAMES ALLOWED IN LAST COMMAND*/ )
      ICOL=100
      RETURN
50    M=(ISTART(NALL+1)-ISTART(NALL))/2
      WRITE(IO,1050) NP
 1050 FORMAT(16HOPROGRAM READ IN,I9,2X6HPOINTS)
      IF (M.EQ.0) GO TO 20
      RETURN
60    WRITE (IO,1060) NAME
1060  FORMAT(5(4H ***),* TOO MANY NAMES FOR TABLES (MORE THAN 10)*/
     *       * SERIES *,6A1,* IGNORED, ASSOCIATED DATA FOLLOW*/)
      RETURN
70    WRITE (IO,1070)
 1070 FORMAT(5(4H ***),* NO NAME SUPPLIED FOR SERIES - DATA IGNORED*//)
      RETURN
80    WRITE(IO,1080)
1080  FORMAT(5(4H ***),* NO FORMAT ESTABLISHED FOR DATA*)
 81   RETURN
90    WRITE(IO,1090)
```

UNCLASSIFIED

```
1090  FORMAT(5(4H ***),* WARNING * DATA CARD OUT OF SEQUENCE*//)
      GO TO 300
C
C  MODE 1
C  READS OBJECT TIME FORMAT FOR DATA READ IN MODE 2.
C
100   READ (INP,1000) IFM
      WRITE(IO,1001) IFM
1000  FORMAT(20A4)
1001  FORMAT(/* DATA FORMAT ,,,*,20A4)
      GO TO 515
C
C  THIS SECTION LOADS NAME VECTORS INTO THE LOOK-UP TABLES IN /TABLES/, AFTER
C  FIRST CHECKING THAT THE NEW NAME IS NOT THE SAME AS THE LAST ONE . IF IT IS
C  THE NEW SERIES IS ASSUMED TO BE A CONTINUATION OF THE PREVIOUS ONE.
C  BLANK NAMES ARE A SPECIAL CASE AND ARE NOT NORMALLY PERMITTED.
C  MODES 2,3,4,7 USE THIS SECTION WHICH ALLOCATES NAMES ETC
200   IF (NAME(1).EQ.IBLANK) GO TO 70
      DO 205 J=1,6
      IF (NAME(J).NE.NAMES(NALL,J)) GO TO 206
205   CONTINUE
      GO TO 215
206   IF (NALL.EQ.10) GO TO 60
      NALL=NALL+1
      DO 210 J=1,6
210   NAMES(NALL,J)=NAME(J)
      FACTOR=1.0
215   GO TO (100,220,300,221,500,600,221,800                          ),IGO
C
C
C  MODE 2
C  NOTE THE DATA COUNT AT THE BEGINNING MUST INCLUDE THE DUMMY POINTS.
220   READ (INP,2000) M
      M=2*M
2000  FORMAT(5I4)
      IF (IFM(1)) 221,80,221
221   II=LAST+1
      LAST=LAST+M
      ISTART(NALL+1)=LAST+1
      IF (IGO.EQ.2) READ(INP,IFM) (LAT(J),J=II,LAST)
      GO TO 50
C
C  MODE 3
C
C  COMPRESSED FORMAT DATA READ IN WITH THIS SECTION. THE WORLD MAP IS IN THIS
C  FORM, ONLY IF THE DISTANCES BETWEEN CONSEQUTIVE POINTS IS LESS THAN 5 DEGREES
C  CAN THIS METHOD BE USED.
C
300   J1=LAST+1
      J2=J1+33
      READ (INP,3000) M,(LAT(J),J=J1,J2)
3000  FORMAT(I3,F6.1,F7.1,32F2.1)
      IF (M.LE.0) GO TO 50
      IF (LAT(J1).GT.90.) GO TO 90
      LAST=LAST+M
      II=J1+2
```

```
      J1=J2+1
      J2=J1+M-35
C
C  CHECK IF MORE THAN ONE CARD IN CURRENT COMPRESSED SECTION, IF NOT SKIP OUT
      IF (J2.GE.J1) READ(INP,3001)(LAT(J),J=J1,J2)
3001  FORMAT(40F2.1)
C
C  THIS SECTION EXPANDS THE COMPRESSED DATA WHICH IS IN DIFFERENCE FORM BIASSED
C  BY 5 TO ELIMINATE - SIGNS.
      DO 320 J=II,LAST
320   LAT(J)=LAT(J-2)+FACTOR*(LAT(J)-5.0)
      LAT(LAST+1)=99.0
      LAT(LAST+2)=0.
      LAST=LAST+2
      ISTART(NALL+1)=LAST+1
      GO TO 300
C
C  MODE 4
C
C  THE USER MAY SUPPLY A SPECIAL SUBROUTINE FOR READING DATA OR DOING ANY
C  OTHER CALCULATIONS, A BLANK CONTROL CARD IS ACCEPTABLE IN THIS MODE.
C  IF DATA ARE TO BE READ IN OR CALCULATED AND THEN PASSED BACK TO THE SUPERMAP
C  SYSTEM TO BE PLOTTED THE APPROPRIATE CALL AT THIS STAGE IS
400   CALL OTHER(LAT(1),LAT(LAST+1))
      IF (M.EQ.0) RETURN
      GO TO 200
C  WHERE THE NEW DATA VECTOR COMES INTO THE SECOND ARGUMENT WITH TOTAL LENGTH
C  *M* WORDS
C
C  MODE 5
C
C  READS PARAMETERS TO SPECIFY SIZE LOCATION ORIENTATION PROJECTION ETC. OF MAP
C   SIMPLIFIED VERSION WHERE SCALE IS INPUT HAS BEEN PROVIDED.
C
500   IBOX=1
      SCALE=0.
      WIDTH=HEIGHT
C  ISYMB FOR NAVIGATION DATA / LINE AND NOT ANNOT   1 SYMBOL AND ANNOT.
      READ(60,900)IBOX,ILINE,ISKIP,HEIGHT,LATMIN,LATMAX,LONGMIN,LONGMAX,
     1IPT,JUDY1,ITM1,JUDY2,ITM2
 900  FORMAT(3I2,6X5F8.2,I8,4I4)
C     READ(60,901)DIST,CHANGE,ANOMCK,GINCH
C901  FORMAT(2F3.0,2F5.0)
      DO 701 IX=1,ISKIP
 701  CALL SKIPFILE(15)
 700  IPROJ=7
      NP=55
      DIST=10.0
      CHANGE=20.0
      ANOMCK=1500.0
      F=.0174533
      NLAT=1
      NLON=1
      SKALE=COS(LATMIN*F)
      DELON=LONGMAX-LONGMIN
      DELAT=LATMAX-LATMIN
```

UNCLASSIFIED

```
      POLAT=LATMAX*(DELAT/2.0)
      POLONG=0.0
      VMAX=(3.141592653 6/360.)*DELAT
      VMIN=-VMAX
      UMAX=(3.141592653 6/360.)*DELON*SKALE
      UMIN=-UMAX
C
C  IF SCALE OPTION HAS NOT BEEN USED *SCALE* WILL REMAIN SET AT ZERO AND THE
C  FOLLOWING SECTION WHICH FINDS THE APPROPRIATE U-V WINDOW IS SKIPPED.
      IF (SCALE.EQ.0.) GO TO 510
      SCAL=SCALE
C  THE NUMERICAL CONSTANT IS .5/(MEAN EARTH RADIUS IN INCHES)
      UMAX=WIDTH *SCALE*.1993E-8
      VMAX=HEIGHT*SCALE*.1993E-8
      UMIN=-UMAX
      VMIN=-VMAX
      IF (UMAX.NE.0.) GO TO 510
      UMAX=VMAX
      UMIN=VMIN
C
 510  WRITE(IO,1003)
 1003 FORMAT(1H0,16HCHART PARAMETERS)
      WRITE(IO,1004)LATMIN,LATMAX
 1004 FORMAT(1H ,20HSOUTHMOST LATITUDE  ,F10.1,10X,20HNORTHMOST LATITUDE
     1  ,F10.1)
      WRITE(IO,1005) LONGMIN,LONGMAX
 1005 FORMAT(1H ,20HWESTMOST LONGITUDE  ,F10.1,10X,20HEASTMOST LONGITUDE
     1  ,F10.1)
      WRITE(61,6066)
 6066 FORMAT(1H0,37HDATES AND TIMES OF DATA TO BE PLOTTED)
      IK=JUDY1/100
      IB=JUDY1-IK*100
      IC=ITM1/100
      ID=ITM1-IC*100
      IE=JUDY2/100
      IF=JUDY2-IE*100
      IG=ITM2/100
      IH=ITM2-IG*100
      WRITE(61,6065)IK,IB,IC,ID
      WRITE(61,6065)IE,IF,IG,IH
 6065 FORMAT(1H ,I4,1X5HMONTH,2XI4,1X3HDAY,2XI4,1X5HHOURS,2XI4,1X
     17HMINUTES)
C
C  CHECKS FOR FATAL ERRORS IN MAP PARAMETERS
      IF (UMIN.GE.UMAX.OR.VMIN.GE.VMAX .OR. HEIGHT.LE.0.0) GO TO 30
  515 IF (NAME(1).NE.IBLANK) GO TO 40
      RETURN
C
C  MODE 6
C
C  READS IN TITLE VECTOR AND RESETS TITLE FLAG *YES*
600   READ(INP,8000) TITLE
 8000 FORMAT(10A8)
      WRITE(IO,6000) TITLE
6000  FORMAT(* MAP TITLE.",*,10A8)
      YES=100.0
```

```
      GO TO 515
C
C
C  MODE 7
C  THERE IS NO SPECIAL SET OF INSTRUCTIONS FOR MODE 7, IT SIMPLY CONSISTS OF A
C  SPECIAL ROUTE AROUND THE OTHER MODES IN *INPUT*.
C
C  MODE 8 IS ANOTHER FORM OF COMPRESSED DATA INPUT, HERE THE COMPRESSION FACTOR
C  IS 10 TIMES LARGER THAN IN THE STANDARD COMPRESSED MODE HENCE THE FACTOR 0.1
800   FACTOR=0.1
      GO TO 300
      END
```

```
                                        IDENT     INPUT
PROGRAM LENGTH                  01502
ENTRY POINTS     INPUT          00334
BLOCK NAMES
                                00315
                 1              00004
                 3              00002
                 5              00004
                 7              00002
                 8              00024
                 12             00001
EXTERNAL SYMBOLS
                 THEND,
                 OBCDICT,
                 OTHER
                 SKIPFILE
                 COSF
                 TSH,
                 STH,
                 SWE,
                 SLI,
                 ONSINGL,
   00353 SYMBOLS
```

UNCLASSIFIED

```
      SUBROUTINE OUTPUT (NAME,IGO)
C
      REAL LAT
      INTEGER TITLE
      DIMENSION RADIUS(2)
      REAL LATMIN,LATMAX,LONGMIN,LONGMAX
      REAL LATNOT(90),LONNOT(180)
      COMMON DELAT,DELON,XPOLAT,XPOLON
      COMMON LX
      COMMON W,DIST,ANGMCK,CHANGE,NP,LKK,KNUM,INUM,GINCH
      COMMON ANGM(     2),III,JJJ,KKK,XLAST
      COMMON LAT(2)
      COMMON INP,IC,U,V,XLAT,XLON
      COMMON POLAT,POLENG,ROT,UMIN,UMAX,VMIN,VMAX,HEIGHT,NLAT,NLON,
     +IPROJ,IBOX,SCALE,ISYMB,ILINE
      COMMON NALL,ISTART(11),NAMES(10,6),LENG,LAST
      COMMON YES,TITLE(10),ICOL,IA(76)
      COMMON/1/IDATE(2) ,HEDN(2)
      COMMON/3/LATMIN,LATMAX
      COMMON/7/LONGMIN,LONGMAX
      DATA(F=.0174533)
      DATA(ENDLAT=99.0),(IPEN=0)
      DATA(SIN1=.0174524),(COS1=.999848)
C
C     CALL PLOTS(IEUF,-1000)
      IPEN=3
      TESTMIN=LONGMIN*10.0
      TESTMAX=LONGMAX*10.0
      ITMIN=ABS(TESTMIN)
      ITMAX=ABS(TESTMAX)
      IF(ITMIN.EQ.ITMAX) GO TO 8502
      IF(ABS(LONGMAX).GT.ABS(LONGMIN))GO TO 8500
      XLAT =LATMIN
      XLON=LONGMAX
      CALL CONV(XLAT,IPEN,IPROJ)
      DIFF=V-VMIN
      XLAT=LATMIN
      XLON=LONGMIN
      CALL CONV(XLAT,IPEN,IPROJ)
      UDIFF=U-UMIN
      GO TO 8501
 8502  DIFF=0
      UDIFF=0
      GO TO 8501
 8500 XLAT=LATMIN
      XLON=LONGMIN
      CALL CONV(XLAT,IPEN,IPROJ)
      DIFF=V-VMIN
      XLAT=LATMAX
      XLON=LONGMIN
      CALL CONV(XLAT,IPEN,IPROJ)
      UDIFF=U-UMIN
 8501 VVIN=VMIN-.00001
      VVAX=VMAX+.00401
      UUIN=UMIN-.00001
      UUAX=UMAX+.00067
```

```
      ICOUNT=0
      ICHECK=0
      KOUNT=0
      LML=1
      FIRST=0
      GO TO (100,200,10,101) IGO
   10 ICHECK=1
      SINRT=-COS(F*ROT)
      COSRT= SIN(F*ROT)
      SINLO= SIN(F*POLAT)
      COSLO =COS(F*POLAT)
      SCALE=HEIGHT/(VMAX-VMIN)
      GO TO 200
C
C  *IBOX* IS USED TO CONTROL THE PRELIMINARY ACTION OF THE SUBROUTINE.
C
100   IF (IBOX+1) 180,105,101
C
C  IF *IPEN* IS 0 THIS IS THE FIRST MAP AND THE ORIGIN IS NOT SHIFTED.
101   IF (IPEN) 102,103,102
  102 CALL PLOT(WIDTH+1.0,0.0,-3)
C
103   WIDTH=HEIGHT*(UMAX-UMIN)/(VMAX-VMIN)
      CALL PLOT(0.0 ,-3)
      IF(YES.EQ.100.0)CALL SYMBOL(     -.5,-2.21,TITLE,90.0,80)
      YES=-1.0
      GO TO 105
C
C  DRAW RECTANGLE AROUND MAP
 104  CALL   PLOT(  0.0,HEIGHT,3)
      CALL   PLOT(WIDTH,HEIGHT,2)
      CALL   PLOT(WIDTH,   0.0,2)
      CALL   PLOT(  0.0,   0.0,2)
      CALL   PLOT(  0.0,HEIGHT,2)
 105  SINRT=-COS(F*ROT)
      COSRT= SIN(F*ROT)
      SINLO= SIN(F*POLAT)
      COSLO =COS(F*POLAT)
C
C  CERTAIN TRIG FUNCTION THAT ARE CONSTANT FOR A GIVEN MAP ARE CALCULATED AND
C  STORED IN *CONV* - *NEWMAP* IS AN ENTRY TO THAT ROUTINE
C
      SCALE=HEIGHT/(VMAX-VMIN)
C  IF NLAT OR NLON IS LESS THAN 1  A LAT LON GRID IS NOT DRAWN
      IF(NLAT.LT.1.OR.NLON.LT.1)GO TO 160
      XTOP=LONGMAX
      XBOT=LONGMIN
      IF(LONGMAX.LT.0)XTOP=360 + LONGMAX
      IF(LONGMIN.LT.0)XBOT=360 + LONGMIN
      DEG=FLOAT(NLON)
      IPEN=3
      XLAT=-90.0
      PLONG=POLONG
      IF(POLONG.LT.0.)PLONG=POLONG+360.
      XLON=PLONG-DEG
      DLAT=-.5
```

UNCLASSIFIED

```
      LTEMP=LKK
      LKK=0
C
C  MERIDIANS DRAWN IN ALTERNATING DIRECTIONS
      ZZMAX=DELAT/2.0
      ZTOP=POLAT + ZZMAX
      DO 110 I=1,360,NLON
      XLON=XLON+DEG
      IF(XLON.GT.360.)XLON=XLON-360.
      DLAT=-DLAT
      DO 110 J=1,360
      XLAT=XLAT+DLAT
      CALL CONV(XLAT,IPEN,IPROJ)
      Y=U-UDIFF
      W=V-DIFF
 555  FORMAT(1H0,8F10.5)
      IF(XLAT.LT.LATMIN.OR.XLAT.GT.LATMAX)GO TO 120
      IF(XLON.GT.XTOP.AND.XLON.LT.XBOT) GO TO 120
      IF(XLAT-ZTOP-.4)7000,120,7000
C9000 FORMAT(1H0,2F10.4)
 7000 CALL PLOT ((Y-UUIN)*SCALE,(W-VVIN)*SCALE,IPEN)
      IPEN=2
      GO TO 110
120   IPEN=3
110   CONTINUE
      DEG=.5
  150 XLAT=POLAT-DEG
      IG2=2*NLAT
C
C  LONGITUDE LINES DRAWN IN ALTERNATING DIRECTIONS
      DO 159 I=IG2,360,NLAT
      XLAT=XLAT+DEG
      IF(XLAT.GE.90.)900,901
 900  XLAT=POLAT-DEG
      DEG=-DEG
 901  DLON=FLOAT(NLAT)
      IPEN=3
      XLON=-DLON
155   XLON=XLON+DLON
      CALL CONV(XLAT,IPEN,IPROJ)
      W=V-DIFF
      Y=U-UDIFF
C     IF((XLAT.LT.82.0).OR.(XLAT.GT.83.0).OR.(XLON.LT.LONGMIN).OR.(XLON.G
C    1GT.LONGMAX)) GO TO 556
C     WRITE(61,555)XLAT,XLON,U,V,DIFF,W,Y,UDIFF
      IF(XLAT.LT.LATMIN.OR.XLAT.GT.LATMAX)GO TO 158
      IF(XLON.GT.XTOP.AND.XLON.LT.XBOT) GO TO 158
      CALL   PLOT((Y-UUIN)*SCALE,(W-VVIN)*SCALE,IPEN)
      IPEN=2
 156  IF(XLON-360.0)155,159,159
158   IPEN=3
      GO TO 156
159   CONTINUE
C
C     LATNOT=LATITUDE NOTATION
C     LONNOT=LONGITUDE NOTATION
```

```
C       DELAT,DELON ARE READ IN THROUGH THE MAP COMMAND
C       DELON==DEGREES LONGITUDE IN UV WINDOW
C       DELAT==DEGREES LATITUDE IN UV WINDOW
 756    IN=1
        IIMAX=DELAT/2.
        LATNOT(IN)=POLAT-ZZMAX
 6001   IN=IN + 1
        LATNOT(IN)=LATNOT(IN-1) + .5
        IF(LATNOT(IN)-ZTOP-.5)6001,6000,6000
 6000   LLMAX=DELON/2.0
        ITOP=DELON + 1.
        DO 169 IN=1,ITOP
        IF(IN.EQ.1)602,603
 602    LONNOT(IN)=POLONG - LLMAX
        GO TO 169
 603    LONNOT(IN)=LONNOT(IN-1)+NLON
  169   CONTINUE
C       LABEL LATITUDE LINES
        IN=0
        XPOLAT=LATMIN
  168   IPEN=3
        XPOLON=LONGMAX-(DELON/2.0)
        CALL CONV(XPOLAT,IPEN,IPROJ)
        W=V-DIFF
        Y=U-UDIFF
        CALL NUMBER((Y-ULIN)*SCALE+.05,(W-VVIN)*SCALE+.05,.07,XPOLAT,0,0
       1,4HF6.1)
 9000   XPOLAT=XPOLAT+.5
        IF(XPOLAT-ZTOP-.4)168,171,171
  171   CONTINUE
C       LABEL LONGITUDE LINES
 4442   IN=0
        XPOLON=LONGMIN
  175   IPEN=3
        XPOLAT=POLAT
        CALL CONV(XPOLAT,IPEN,IPROJ)
        W=V-DIFF
        Y=U-UDIFF
        IF((XPOLON.LT.-30.0).OR.(XPOLON.GT.240.0)) GO TO 450
        IF(Y.GT.ULAX.OR.Y.LT.UUIN.OR.W.GT.VVAX.OR.W.LT.VVIN)GO TO 450
        IN=IN+1
        XPOLON=XPOLON+NLON
C  A IS THE ANGLE AT WHICH THE LONGITUDE IS ANNOTATED
        IF(XPOLON-LONGMAX-1      )805,805,178
 805    IF(IPROJ.EQ.7)A=270.+XPOLON-POLONG
        XLONN=XPOLON-NLON
        CALL NUMBER((Y-ULIN)*SCALE+.05,(W-VVIN)*SCALE-.05,.07,XLONN      ,A,
       1,4HF6.1)
        GO TO 175
 450    XPOLON=XPOLON + NLON
        IF(XPOLON-LONGMAX-1      )175,175,178
  178   CONTINUE
        LKK=LTEMP
C  *LIMB* IS AN ENTRY POINT OF *CONV*, IF *NONE* IS NOT ZERO A LIMB IS DRAWN ON
C  THESE MAPS THAT OCCUPY ONLY A FINITE AREA OF THE U-V PLANE.
C  ONLY 2 KINDS OF LIMB ARE DEALT WITH - ELLIPSES OR CIRCLES. THIS MEETS PRESENT
```

UNCLASSIFIED

```
C  REQUIREMENTS
C60    CALL LIMB(RADIUS,NONE,IPROJ)
 160   NONE=0
       IF (NONE.EQ.0) GO TO 180
       AXIS=U
       U=RADIUS(1)
       V=0.0
       W=0.0
       IPEN=3
       DO 170 I=1,361
       V=V*AXIS
       IF (U.LE.UMAX.AND.U.GE.UMIN.AND.V.LE.VMAX.AND.V.GE.VMIN) GO TO 161
       IPEN=3
       GO TO 165
161    CALL   PLOT((U-UMIN)*SCALE,(V-VMIN)*SCALE,IPEN)
       IPEN=2
165    V=U*SIN1+W*COS1
       U=U*COS1-W*SIN1
170    W=V
C
C  THE STATE OF *IBOX* IS EXAMINED AND THE APPROPRIATE ACTION TAKEN (SEE EARLIER
C  COMMENTS).
180    IF (IBOX+2) 399,185,182
182    IBOX=-2
185    CONTINUE
C
C  IDENTIFICATION OF SERIES ON OUTPUT.A BLANK NAME MEANS OUTPUT EVERY
C  SERIES.
200    CALL SEARCH(NAME,INDEX,N2,N2)
       N1=INDEX
       N2=N1
       IF (N1) 202,399,205
202    N1=1
       N2=NALL
205    IF (IGO.EQ.2) GO TO 500
C
C  PLOTTED OUTPUT SECTION.
C
       REWIND 20
       DO 390 N=N1,N2
       KPDATE=0
       LABEL=0
       II=ISTART(N)
       IEND=ISTART(N+1)-1
       IPEN=3
       INMAP=0
C    THE VALUE OF LKK IS SET IN SUB OTHER DEPENDING ON THE VALUE OF KKK
       IF(LKK.EQ.2)GO TO 303
C
C  CHECK FOR BEGINNING OF NEW SERIES OR CHANGE TO POINT MODE
 301   READ(20,6000)LAT(1),LAT(2),ANOMAL
 6000  FORMAT(2F10.4,F10.1)
       IF(LAT(1)-ENDLAT)302,328,340
C
C  CONTINUOUS MODE DATA DRAWN
302    CALL CONV(LAT( 1),IPEN,IPROJ)
```

```
      GO TO 304
C 303 IF(LAT(II)-ENDLAT) 305,329,305
 303  READ(05,6000) LAT(1),LAT(2),IDATE(1),IDATE(2)
C     WRITE(61,9000) LAT(1),LAT(2)
      IF(LAT(1)-ENDLAT)305,329,305
  305 CONTINUE
      U=LAT(1)
      V=LAT(2)
  304 CONTINUE
      W=V-DIFF
      Y=U-UDIFF
C
C  CHECKS IF POINT LIES INSIDE MAP RECTANGLE, IF NOT SKIPS PLOT ROUTINE AND
C  COUNTING STATEMENT
      IF (Y.GT.UUAX.OR.Y.LT.UUIN.OR.W.GT.VVAX.OR.W.LT.VVIN) GO TO 330
      GO TO 9066
 700  CALL SYMBOL((U-UMIN)*SCALE,(V-VMIN)*SCALE,.07,5,0,0,-1)
      ICOUNT=ICOUNT + 1
      IF(ICOUNT-61)4000,4001,4000
 4001 WRITE(IO,405)
 4000 ICOUNT=1
      ITERM=2
      IHR=IDATE(ITERM)/100
      MIN=IDATE(ITERM)-(IHR*100)
      WRITE(IO,401)IDATE( 1),IHR,MIN,      LAT( 1),LAT(ITERM),U,V
 401  FORMAT(1H ,8X I6,9X I2,1X I2,10XF9.3,10XF9.3,2X,2F10.4)
      IF(IDATE( 1)-KPDATE) 702,702,701
 701  KPDATE=IDATE( 1)
      LABEL=0
 705  IFN=IDATE( 1)*10000 + IDATE(2)
      CALL NUMBER((U-UMIN)*SCALE,(V-VMIN)*SCALE,.07,IFN,0,3HI12)
      GO TO 210
 702  IF(LABEL-1)703,210,210
 703  IF(IDATE(2)    -1200)210,704,704
 704  LABEL=1
      GO TO 705
 906  IF(LKK .EQ. 2) GO TO 9066
      GO TO 9067
 9066 IF(IFIRST)8001,8000,8001
 8000 XLAST=(Y-UUIN)*SCALE
      YLAST=(W-VVIN)*SCALE
 8001 XPT=(Y-UUIN)*SCALE
      YPT=(W-VVIN)*SCALE
      CALL ANGANO(XPT,XLAST,YPT,YLAST,ANG)
      CHAR=ANOMAL
      IF(IFIRST) 810,809,810
 809  CALL PLOT(XPT,YPT,3)
 810  CALL FIXANO(XPT,YPT,ANG,CHAR,KOUNT,ILINE,IPEN)
      IFIRST=1
      KOUNT=KOUNT + 1
      LML=LML +1
      XLAST=(Y-UUIN)*SCALE
      YLAST=(W-VVIN)*SCALE
      GO TO 210
C9067 CALL SYMBOL((U-UMIN)*SCALE,(V-VMIN)*SCALE,.07,5,0,0,-2)
 9067 CALL PLOT((U-UMIN)*SCALE,(V-VMIN)*SCALE,IPEN)
```

UNCLASSIFIED

```
 9068 ICOUNT=ICOUNT + 1
      IF(ICOUNT-61) 4002,4003,4002
 4003 WRITE(IO,405)
 405  FORMAT(1H1)
 4002 ICOUNT=1
      ITERM=2
      IHR=IDATE(ITERM)/100
      MIN=IDATE(ITERM)-(IHR*100)
      GO TO 210
 444  WRITE(IO,401)IDATE( 1),IHR,MIN,      LAT( 1),LAT(ITERM),U,V
  210 CONTINUE
      INMAP=INMAP+1
      IPEN=2
320   II=II+2
      IF (II-IEND) 301,390,390
328   INMAP=INMAP+1
330   IPEN=3
      GO TO 320
 999  ICOUNT=ICOUNT + 1
      IF(ICOUNT-61)4004,4005,4004
 4005 WRITE(IO,405)
 4004 ICOUNT=1
      ITERM=2
      IHR=IDATE(ITERM)/100
      MIN=IDATE(ITERM)-(IHR*100)
      GO TO 320
 445  WRITE(IO,402)IDATE( 1),IHR,MIN,      LAT( 1),LAT(ITERM)
 402  FORMAT(1H ,8XI6,9XI2,1XI2,10XF9.3,10XF9.3,10X1H*)
      GO TO 320
  329 INMAP=INMAP+1
      IPEN=3
      II=II+2
  331 IF(II-IEND) 303,390,390
C
C  CHECK FOR CHANGE OF POINT SYMBOL OR REVERSION TO CONTINUOUS MODE
340   IF (LAT( 1)-ENDLAT) 341,328,360
C
C  POINT MODE SERIES DRAWN
341   CALL CONV(LAT( 1),IPEN,IPROJ)
C
C  CHECKS IF POINT LIES INSIDE MAP RECTANGLE, IF NOT SKIPS PLOT ROUTINE AND
      IF (U.GT.UMAX.OR.U.LT.UMIN.OR.V.GT.VMAX.OR.V.LT.VMIN) GO TO 350
      ITYPE=2
      CALL SYMBOL((U-UMIN)*SCALE,(V-VMIN)*SCALE,HT,ITYPE,0.,-1)
      WRITE(IO,907)U,UMIN,SCALE,V,VMIN,SCALE,HT,ITYPE
 907  FORMAT(1H ,7F10.3,I5)
348   INMAP=INMAP+1
350   II=II+2
      IF (II-IEND) 340,390,390
C
C  SYMBOL TYPE AND HEIGHT ESTABLISHED FOR SERIES IN POINT MODE
 360  ITYPE=2
      HT=0.14
      IF (LAT(2)    .NE.0.0) HT=LAT(2)
      GO TO 348
C  PRINTED OUTPUT LOG (LP-DATED
```

```
C
  390 CONTINUE
      IF(ICHECK.EQ.1)GO TO 395
 395  IMAP=INMAP + 1
      WRITE(IO,3000) IMAP
 3000 FORMAT(16H0PROGRAM PLOTTED,I9,2X17HPOINTS ON THE MAP)
C   IF LKK=0 SUB OTHER WII NOT CALCULATE ANOMALY SERIES ( IF=2 ANOMALY SERIES
C    HAD JUST BEEN PLOTTED THEREFOR REINITIALIZE EVERYTHING
      IF(LKK.EQ.0.OR.LKK.EQ.2)604,399
 604  KNUM=0
      INUM=0
      JJJ=0
      NALL=0
      LAST=0
  399 CONTINUE
C  PROGRAM PLOTTED TRACK  READY TO PLOT ANOMALY
      REWIND 05
      RETURN
C
C  PRINTED OUTPUT SECTION
C
500   DO 550 N=N1,N2
      II=ISTART(N)
      IEND=ISTART(N+1)-1
C5000 FORMAT(//* SERIES *,6A1//(8(2F7.2,1H,)))
 550  CONTINUE
      RETURN
      END

                                          IDENT      OUTPUT
      PROGRAM LENGTH                 02766
      ENTRY POINTS     OUTPUT        00532
      BLOCK NAMES
                                     00315
                       1             00004
                       3             00002
                       7             00002
      EXTERNAL SYMBOLS
                       Q1C10100
                       Q1C03100
                       Q1C02100
                       THEND,
                       Q8CDICT,
                       CONV
                       PLOT
                       SYMBOL
                       NUMBER
                       SEARCH
                       ANCANO
                       FIXANO
                       SINF
                       COSF
                       REW,
                       TSH,
                       STH,
                       QNSINGL,
         00542 SYMBOLS
LOAD
```