COMPUTER PROGRAMS FOR

DIRECT READ-OUT SATELLITE DATA USE

Supplement to ATS Paper No. 181

by

R. MADDEN, H. MILLER

M. POLIFKA

and

T. VONDER HAAR

MAY 1972

DEPARTMENT OF ATMOSPHERIC SCIENCE
COLORADO STATE UNIVERSITY
FORT COLLINS, COLORADO

Atmospheric Science Technical Report

COMPUTER PROGRAMS FOR DIRECT READ-OUT SATELLITE DATA USE

by

R. Madden, H. Miller, M. Polifka and T. Vonder Haar

The objective of this report is to provide direct read-out satellite data users with the necessary information and program documentation for computer processing and display of direct read-out (VHF) satellite data. This report is an extension of, and a supplement[1] to, Atmospheric Science Paper No. 181 (Miller and Vonder Haar, 1972). As noted in that paper, the versatility and utility of direct read-out satellite data can be greatly enhanced by digitizing and computer processing the data. Of course, to do this requires rather extensive computer programming. This programming has been accomplished, and is presented here in complete detail.

Caution must be exercised in attempting to use these programs on certain computer systems. The programs presented here were specifically designed for use on a Control Data Corporation Model 6400 computer, a 60-bit-per-word device. Also, the programs were written using Fortran Extended version language. To change these programs over for operation on another machine, only minor changes of some statements need be made and can be done easily by anyone reasonably proficient in computer programming, and familiar with the particular machine being used.

---

[1] References to standard terms, procedures and scientific papers concerning realtime satellite data are included in Atmospheric Science Paper No. 181.

The following flow diagram outlines the basic programs and their use in going from raw satellite signal to computerized data display. The sections of the flow chart labeled "Routine" are simple sub-routines of the program SRGRID.

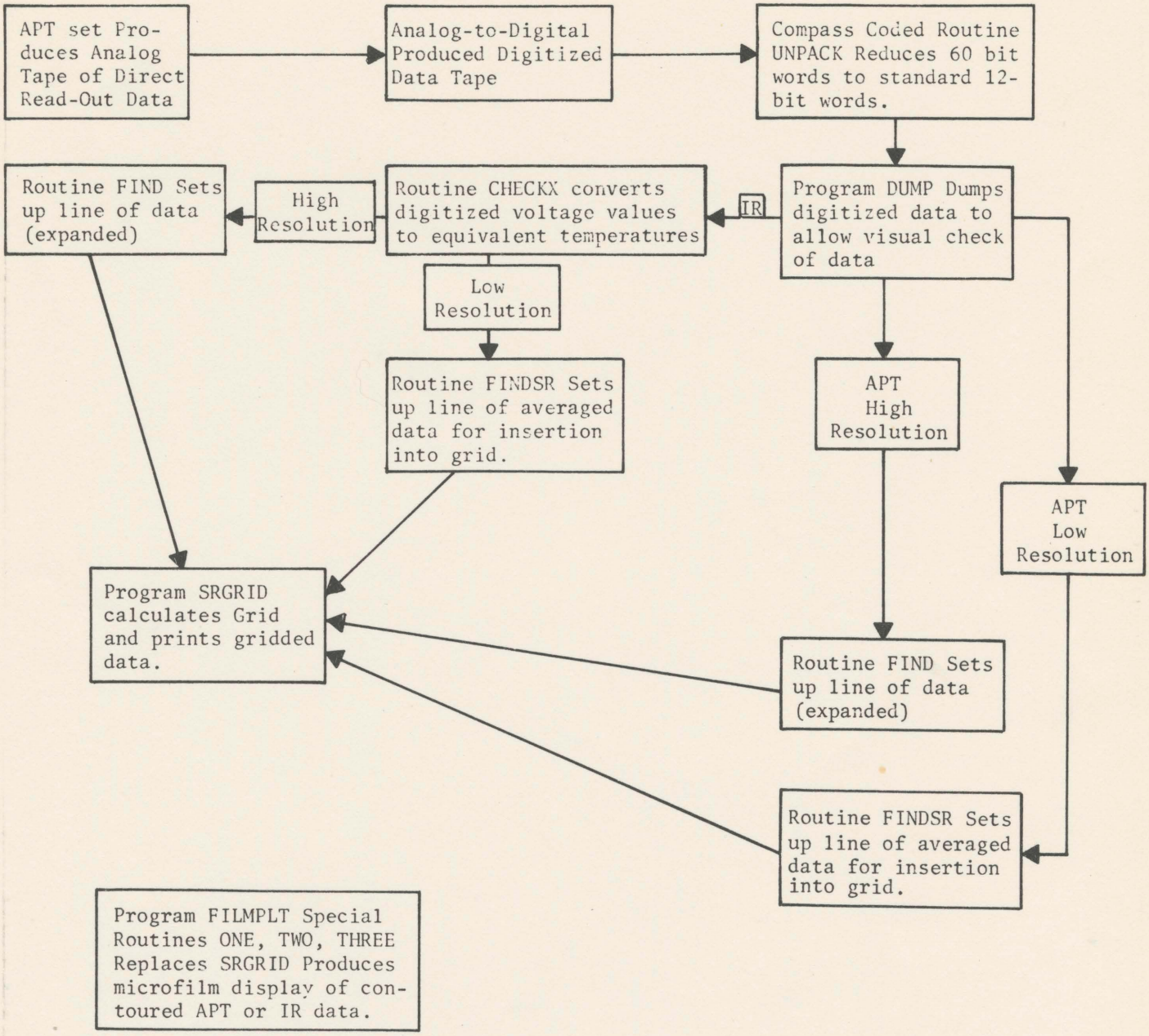
The remainder of this report consists of a short descriptive paragraph of:

(a) the satellite data tape

(b) each program

(c) each subroutine,

followed by the program listings.



FLOW CHART OF ROUTINES

Infrared Data Tape

An Infrared (IR) data tape is a digitized, binary off-line tape written at 800 bits per inch by the data conversion (analog-to-digital) service at Colorado State University. Each file on the tape consists of several satellite sensor scan lines. A line contains approximately 1980 data points. The data are packed in 60-bit words at 12-bits per point. There are 990 points per record or 198 (60-bit) words. Each scan line begins with 7 high amplitude pulses followed by several space values, approximately 582 data values, and a calibration wedge. Radiometer housing vaules follow the wedge until a new scan line begins.

Automatic Picture Transmission (video) Data Tape

An Automatic Picture Transmission (APT) data tape is a digitized binary offline tape written at 800 bits per inch by the data conversion (analog-to-digital) service at Colorado State University. Each file on the tape contains one APT picture. A picture has 800 scan lines with approximately 800 data points per record or 160 (60-bit) words. The first 20 records contain phase information. Each scan line begins with a 48 point sync pulse followed by 750 points of actual data. This is followed by another sync pulse and data line.

Program: FILMPLT

This program objectively contours IR or APT data and outputs the analysed fields on microfilm. It labels and plots latitude and longitude lines with the data. Each frame of microfilm is set up to contain 60 averaged lines of IR data or 50 averaged lines of APT data.

Input Cards:

CARD 1

Columns	Description
1-5	NT = Integer number of microfilm frames

CARD 2

Columns	Description
1-80	Alphanumeric label for microfilm

CARD 3

Columns	Description
1-80	more labels

CARD 4

Columns	Name	Description
1	D	1 = North to South Orbit 2 = South to North Orbit
2-3	NMO	Month of message
4-5	NDY	Day of message
6-7	SAT	Identifier
8-12	RO	Reference orbit no.

CARD 4 (cont.)

Columns	Name	Description
14-15	RDY	Reference day of ascending node
16-17	RHD	Reference hour of ascending node
19-20	RMIN	Reference min. of ascending node
21-22	RSEC	Reference sec. of ascending node
23	RQ	Quadrant
24-27	RL	Reference longitude in degrees
29-30	PMIN	Orbital period (min.)
31-32	PSEC	Orbital period (sec.)
34-37	PL	Orbital displacement to the west
38-41	H	Height

CARD 5

Columns	Name	Description
1-5	ORBIT	Orbit number
6-7	TSRH	First scan hour
8-9	TSRM	First scan minute
10-11	TSRS	First scan second
12-14	SRPM	First scan RPM
15-19	ORIN	Orbit inclination

Subroutines used

Plot routines on system

MAP	FRSTPT
GRID	GRID
PWRT	LINE
OPTION	NUMBR
FRAME	PSYM

Special routines

ONE:

There is a subroutine ONE for APT data and one with the same name for IR data. Each fills matrix AM with the data to be contoured.

The APT subroutine ONE reads and unpacks 400 lines of data for each frame. Each line contains 750 data values. This routine averages every 8 lines and every 8 1/3 data values. The averaged data is stored in matrix AM starting in column 16 of each row. The first and last 15 columns of each row are filled with the closest averaged data value.

Subroutines needed: UNPACK

Core needed: 2223 octal

The IR subroutine ONE reads and unpacks 120 lines of data for each frame. Each line contains 582 data values. This routine averages every 2 lines and every 5 1/3 data values. The averaged data is stored in matrix AM starting in column 6 of each row. The first 5 and last 6 columns of each row are filled with the closest averaged data value.

Subroutines needed: UNPACK - CHECKX

Core needed: 5000 octal

Special routines:

TWO:

This is a version of the SRGRID routine. It calculates the positions of latitude and longitude lines through matrix AM. It also plots a line of asterices on each contoured frame to represent latitude lines and longitude lines then labels each line.

Core: 2504 octal words

Special routines:

THREE

This routine objectively contours the data stored in matrix AM and outputs the analysed field on microfilm.

Subroutines needed: from NCAR

CALCNT
QUICK
HILO
PROJUV

Core needed: including all subroutines 1461 octal

Core and time needed for total program on CDC-6400

IR version

Core: 55000 octal

Time: 125 CP seconds
66 PP seconds
} for one frame

APT version

Core: 51000 octal

Time: 62 CP seconds
90 PP seconds
} for one frame

Program SRGRID

This routine calculates the positions of latitude and longitude lines through APT or IR data[1]. The routine then outputs on paper a grid of the data with slashes representing the latitude and longitude lines and alphanumeric characters representing a data range. The grid may be a low resolution of the data and cover from the right horizon to the left horizon, or it may be a high resolution of the data and cover only a small portion of the field available. If the grid is low resolution the data is averaged. For APT data, every 8 lines by 8 1/3 points are averaged and represented by one character. For IR data, every 2 lines by 5 1/3 points are averaged and represented by one character.

Core needed:	on CDC 6400	43000 octal
Time needed:	for 200 lines	175 CP seconds
	high resolution	80 PP seconds

Subroutines:

	FIND	for high resolution
(or)	FINDSR	for low resolution
	UNPACK	
	CHECKX	

---

[1] For conventional APT pictures (i.e., those not digitized) the gridding geometry does not hold. Reference Paper No. 181.

INPUT:

CARD 1

Columns	Name	Description (Integers)
1-5	KF	Start grid position to output
6-10	KL	End grid position to output

If a low resolution is wanted set KF = J1 and KL = J2 after J1 and J2 are calculated. Note comment cards in the program.

CARD 2

Columns	Name	Description
1	D	= 1 North to South Orbit = 2 South to North Orbit
2-3	NMO	Month of message
4-5	NDY	Day of message
6-7	SAT	Identifier
8-12	RO	Reference orbit number
14-15	RDY	Reference day of ascending node
16-17	RHR	Reference hour of ascending node
19-20	RMIN	Reference min. of ascending node
21-22	RSEC	Reference sec. of ascending node
23	RQ	Quadrant
24-27	RL	Reference longitude in degrees
29-30	PMIN	Orbital period (Min.)
31-32	PSEC	Orbital period (Sec.)
34-37	PL	Orbital displacement to the west
38-41	H	Height

CARD 3

Columns	Name	Description
1-5	ORBIT	Orbit number
6-7	TSRH	First scan hour
8-9	TSRM	First scan min.
10-11	TSRS	First scan sec.
12-14	SRPM	First scan RPM
15-19	ORIN	Orbit inclination

Subroutine FIND

There is a subroutine FIND for APT data and one with the same name for IR data. Each finds one line of data on tape and represents each data value in that line by a character between A and Z. The characters are stored in array SUM.

Subroutine FINDSR

There is a subroutine FINDSR for APT data and one with the same name for IR data. The APT subroutine FINDSR finds and averages 8 lines by 8 1/3 elements. Each averaged value is represented in array SUM by a character between A and Z.

The IR subroutine FINDSR finds and averages 2 lines by 5 1/3 elements. Each averaged value is represented in array SUM by a character between A and Z.

UNPACK

This routine unpacks 15 (22-bit) words from 3 packed (60-bit) words. The resulting 15 words are stored as integers. UNPACK was written in Control Data 6000 series assembly language called Compass. It was written specifically to unpack data digitized by the data conversion service.

Core: 15 octal

Subroutine CHECKX(C,T)

This routine determines temperature values (Kelvin) of IR data where:

$$\text{Temperature} = \text{function of } E$$

$$E = \frac{0.052593\ (\text{IR data-average space value per line})}{(\text{average housing value-average space value per line})}$$

Core: 260 octal

The values in the Temperature table are calcualted as in the ATS Paper 181.

DUMP

This program prints out 20 records from a scanner data tape digitized by the data conversion service via the sync pulse program. The data can be APT or IR data. The output is used to manually find the record number of the first data line, the position of syncs, the number of data values per line, etc. Once the record number of the first data line is known, the data tape can be positioned on the first data line to start the programs.

Core	:	16000 octal
CP time	:	15 seconds
PP time	:	4 seconds
Compile time	:	.9 seconds
Pages	:	20 pages

```
      PROGRAM FILMPLT
     1(INPUT=2028,OUTPUT=2028,TAPE1,TAPE6=OUTPUT,FILMPL,TAPE5=INPUT)
C ***************************************************************************
C ***** THIS IS THE MAIN ROUTINE THAT COMBINES  MADDENS ROUTINE
C ***** AND THE CONTOUR ROUTINE TO CONTOUR IR OR APT DATA ON MICROFILM
C ***************************************************************************
      COMMON C(10
      COMMON AM(120,60)
      COMMON/KEEP/TR,TIME,DT,DTSR,RRL,H,REFTIM,RO1,D,RL,DATA(2970),IF,
     * INDEX,NREC,NOST,LINE
      DIMENSION C(15)
      NTIME = 0
      TR = 0
      IF = 0
      READ(5,1000) NT
      READ(5,1001) (C(I),I=1,8)
      READ(5,1001) (C(I),I=9,15)
1001 FORMAT(8A10)
1000 FORMAT(I5)
  10 CONTINUE
      NTIME = NTIME +1
      CALL MAP(1.,120.,1.,60.,.05,.95,.01,.91)
      CALL GRID(1,1,1,1)
      CALL PWRT(1. ,.65,.C,80,1,0)
      CALL PWRT(1. ,.63.5,C(9),70,1,0)
      CALL PWRT(120.,61.5,4HTIME,4,1,0)
      CALL PWRT(1.,61.5,4HLON,.4,1,0)
      CALL PWRT(-5.,61.,3HLAT ,3,1,0)
      CALL OPTION(0,0,0,0,0)
      CALL TWO
      CALL ONE
      CALL THREE
      CALL FRAME
      IF(NTIME.LT.NT) GO TO 10
      STOP
      END
```

```
      SUBROUTINE ONE
C ***********************************************************************
C ***** THIS ROUTINE FILLS ARRAY AM WITH APT DATA USED IN CONTOUR ROUTINE THREE
C ***** IF          BEGINNING INDICATOR
C ***** ISTPS       STOP  POSITION TO SEARCH FOR THE SYNC
C ***** ISTRTS      START POSITION TO SEARCH FOR THE SYNC
C ***** ITIM        COUNTER  FOR MISSING PARTS OF A LINE
C ***** LINE        LINE COUNT
C ***** NOD2        POSITION OF NEW DATA LINE
C ***** NOS1        BEGINNING OF PREVIOUS SYNC
C ***** NOS2        BEGINNING OF NEW SYNC
C ***** NREC        RECORD COUNT
C ***** TOP         HIGHEST SYNC PULSE FOR A LINE
C ***********************************************************************
      COMMON AM(120,50)
      COMMON/KEEP/DUM(10)                                    ,DATA(2415),IF,
     * INDEX,NREC,NOS1,LINE
      DIMENSION DAT(25)
      DIMENSION IBUF(360)
      DIMENSION SUM(360)
      COMMON/UNP/UWD(3),JPT(15)
      SCALE = 1.414/2047.
      ICHE = 0
      LINE = 0
      ITIM =0
      IF(IF)1,1,5
    1 CONTINUE
      IF = 1
      INDEX = 0
      NREC = 0
      NOS1 = 0.
C ***** AT THE FIRST SEARCH ONE RECORD AT A TIME FOR THE SYNC PULSE
      ISTRTS = 0
      ISTPS = 800
      DO 6 I=1,360
      SUM(I) = 0.
    6 CONTINUE
      GO TO 10
    5 ISTRTS = NOS1 + 730
      IF(ISTRTS.LT.0) ISTRTS = 0
      IF(ISTRTS.LT.0) WRITE(6,4000)ISTRTS
      ISTPS = NOS1 + 850
      IF(ISTPS-2400)7,7,305
    7 CONTINUE
      IF(INDEX-2400)10,10,10
   10 CONTINUE
C ***** FIND  50 LINES AT A TIME
      IF(LINE.GE. 50) GO TO 400
C ***** THE APT DATA IS PACKED ON A TAPE WITH 800 (12-BIT WORDS)
C ***** PER RECORD
C ***** EACH SCAN LINE CONTAINS APPROXIMATELY 800  WORDS
C ***** STORE 3 RECORDS AT A TIME TO FIND ONE COMPLETE LINE
      NREC = NREC +1
      BUFFER IN(1,1) (IBUF(1),IBUF(250))
      IF(UNIT(1))13,11,12
   11 CONTINUE
      WRITE(6,2000) NREC
 2000 FORMAT(' EOF AT RECORD ',I10)
      GO TO 410
   12 CONTINUE
      WRITE(6,3000) NREC
 3000 FORMAT(' PARITY ERROR ON RECORD ',I10)
   13 CONTINUE
```

```
      NWDS = LENGTH(1)
      II = 0
      DO  17 I=1,NWDS,3
      DO 15 J=1,3
      JWD(J) = IBUF(I+J-1)
   15 CONTINUE
      CALL UNPACK
C ***** UNPACK IS A COMPASS-CODED SUBROUTINE USING CRP FOR COMMUNICATION.
      DO 16 J=1,15
      II = II+1
      IF(II.GT.800) GO TO 16
      INDEX = INDEX +1
      DATA(INDEX)= SCALE*FLOAT(JPT(J))
   16 CONTINUE
   17 CONTINUE
      IF(INDEX-2400)10,18,18
   18 CONTINUE
      I = ISTRTS+1
   20 CONTINUE
C ***** FIND THE SYNC PULSE WHICH SHOULD LAST AT LEAST 25 POINTS
      I = I+1
      IF(I-ISTPS)25,25,301
   25 IF(DATA(I)-.50)20,30,30
   30 KOUN = 0
      NOS2 = I
      TOP = DATA(I)
      DO 40 J=1,35
      IF(DATA(I+J)-.50)35,41,41
   35 KOUN = KOUN +1
      IF(KOUN-5)40,40,20
   41 CONTINUE
      IF(DATA(I).GT.TOP) TOP = DATA(I)
   40 CONTINUE
      ITIM = 0
   45 CONTINUE
      NOD2 = NOS2 +48
      II = 0
   99 DO 100 I=1,90,3
      DO 101 J=1,25
      II= II+1
      NN = NOD2+II-1
      DAT(J) = DATA(NN)
  101 CONTINUE
C ***** AVERAGE EVERY 8 AND 17 POINTS AND STORE IN SUM
      SUM(I) =(3.*(DAT(1)+DAT(2)+DAT(3)+DAT(4)+DAT(5)+DAT(6)+DAT(7)
     *  + DAT(8))+DAT(9))/25.  +SUM(I)
      SUM(I+1) = (3.*(DAT(10)+DAT(11)+DAT(12)+DAT(13)+DAT(14)+DAT(15)
     *+DAT(16)) +2.*(DAT(9)+DAT(17)))/25.   +SUM(I+1)
      SUM(I+2) = (3.*(DAT(18)+DAT(19)+DAT(20)+DAT(21)+DAT(22)+DAT(23)
     *+DAT(24)+DAT(25)) +DAT(17)) 25.  +SUM(I+2)
  100 CONTINUE
      N = NOS2+730
      N = N/800
      NBI = N*800
      IF(N)130,130,110
  110 CONTINUE
      IF(N.GT.3) STOP 3
C ***** SHIFT THE DATA IN THE STORED ARRAY TO MAKE ROOM FOR NEXT LINE
      NBB = 2400-NBI
      DO 120 I=1,NBB
      DATA(I) = DATA(I+NBI)
  120 CONTINUE
      INDEX = NBB
  130 CONTINUE
```

```
      NOS1 = NOS2-NN
      LINE = LINE +1
      IF(LINE.LT.2) GO TO 5
C ***** STORE EACH DATA LINE IN MATRIX AM
      DO 180 I=1,109
      XMEAN = SUM(I) 2
      X = 190.16
      DO 160 J=1,16
      X = X+10
      IF(XMEAN- X)170,170,160
  160 CONTINUE
      J = 16
  170 CONTINUE
      SUM(I+125) = CHAR(J)
  180 CONTINUE
      DO 161 I=1,125
  161 SUM(I) = 1H
      DO 162 I=255,360
  162 SUM(I) = 1H
  200 RETURN
  210 STOP 6
  301 ICHEX = 1
C ***** COULD  NOT FIND THE 7 PULSE-HIGH AMPLITUDE IN THE RANGE FROM ISTRTS
C ***** TO ISTPS
      IF(ISTRTS.EQ.0.AND.ISTPS.EQ.990) GO TO 306
      ITIM = ITIM +1
      IF(NOS2.GT.ISTRTS.AND.NOS2.LT.ISTPS) GO TO 310
      NOS2 = NOS1 +1943
      GO TO 310
  302 ICHEX = 2
C ***** COULD NOT FIND THE DATA END IN THE RANGE FROM ISTRTD TO ISTPD
      ITIM = ITIM+1
      IF(NOD2.GE.ISTRTD.AND.NOD2.LT.ISTPD) GO TO 310
      NOD2 = NOS2 + 675
      GO TO 310
  303 ITIM = ITIM +1
C ***** COULD NOT FIND THE WEDGE IN THE RANGE FROM ISTRTC TO ISTPC
      ICHEX = 3
      IF(NOC2.GE.ISTRTC.AND.NOC2.LT.ISTPC) GO TO 310
      NOC2 = NOD2 + 224
      GO TO 310
  306 CONTINUE
C ***** DATA LINE DOES NOT START IN FIRST RECORD TRY NEXT RECORD
      DO 307 I=1,1980
      DATA(I) = DATA(I+990 )
  307 CONTINUE
      INDEX = 1980
      GO TO 10
  305 CONTINUE
C ***** THE RANGE IS OUT OF BOUNDS
      ISTPS = 2970
      ICHEX = 5
  310 CONTINUE
      WRITE(6,4000)NOS2,NOS1,NOD2,NOD2,ICHEX,IREC
 4000 FORMAT(8I10)
      IF(ITIM.GT.2) GO TO 410
  350 GO TO (401, 99.99, 101, ICHEX
  410 CONTINUE
      END
```

```
      SUBROUTINE ONE
C ***************************************************************************
C ***** THIS ROUTINE FILLS ARRAY AM WITH IR DATA USED IN CONTOUR ROUTINE THREE
C ***** IF          BEGINNING INDICATOR
C ***** ISTPC       STOP  POSITION TO SEARCH FOR THE WEDGE
C ***** ISTPD       STOP  POSITION TO SEARCH FOR THE END OF THE DATA
C ***** ISTPS       STOP  POSITION TO SEARCH FOR THE 7 HIGH AMPS
C ***** ISTRTC      START POSITION TO SEARCH FOR THE WEDGE
C ***** ISTRTD      START POSITION TO SEARCH FOR THE END OF THE DATA
C ***** ISTRTS      START POSITION TO SEARCH FOR THE 7 HIGH AMPS
C ***** ITIM        COUNTER  FOR MISSING PARTS OF A LINE
C ***** ITIME       COUNTER  FOR MISSING PARTS OF A LINE
C ***** LINE        LINE COUNT
C ***** NOC2        POSITION OF WEDGE START
C ***** NOD2        POSITION OF NEW DATA LINE
C ***** NOS1        BEGINNING OF PREVIOUS 7 PULSE- HIGH AMPLITUDE
C ***** NOS2        BEGINNING OF NEW      7 PULSE- HIGH AMPLITUDE
C ***** NREC        RECORD COUNT
C ***** SUMHS       AVERAGE 200 VALUES TO DETERMINE THE HOUSING VALUE
C ***** SUMSP       AVERAGE 45 VALUES TO DETERMINE THE SPACE VALUE
C ***************************************************************************
      COMMON AM(120,60)
      COMMON/KEEP/DUM(10)                                    ,DATA(2970),IF,
     * INDEX,NREC,NOS1,LINE
      DIMENSION DAT(16)
      DIMENSION IBUF(360)
      DIMENSION SUM(360)
      COMMON/UNP/JWD(3),JPT(15)
      SCALE = 1.414/2047.
      ICHE = 0
      LINE = 0
      ITIM =0
      IF(IF)1,1,5
    1 CONTINUE
      IF = 1
      INDEX = 0
      NREC = 0
      NOS1 = 0.
C ***** AT THE FIRST SEARCH ONE RECORD AT A TIME FOR THE 7 HIGH AMPS
      ISTRTS = 0
      ISTPS = 990
      DO 6 I=1,360
      SUM(I) = 0.
    6 CONTINUE
      GO TO 10
    5 ISTRTS = NOS1 + 1910
      IF(ISTRTS.LE.0) STOP 7
      ISTPS = NOS1 + 2400
      IF(ISTPS-2970)10,10,305
   10 CONTINUE
C ***** FIND  60 LINES AT A TIME
      IF(LINE.GE. 60) GO TO 400
      NREC = NREC +1
C ***** THE IR DATA IS PACKED ON A TAPE WITH 990 (12-BIT WORDS)
C ***** PER RECORD
C ***** EACH SCAN LINE CONTAINS APPROXIMATELY 1980 WORDS
C ***** STORE 3 RECORDS AT A TIME TO FIND ONE COMPLETE LINE
      BUFFER IN(1,1) (IBUF(1),IBUF(250))
      IF(UNIT(1))13,11,12
   11 CONTINUE
      WRITE(6,2000) NREC
 2000 FORMAT(*   EOF AT RECORD  *,I10)
      STOP
```

```
   12 CONTINUE
      WRITE(6,3000) NREC
 3000 FORMAT(*   PARITY ERROR ON RECORD *,I10)
   13 CONTINUE
      NWDS = LENGTH(1)
      DO  17 I=1,NWDS,3
      DO 15 J=1,3
      JWD(J) = IBUF(I+J-1)
   15 CONTINUE
      CALL UNPACK
C ***** UNPACK IS A COMPASS-CODED SUBROUTINE USING  JWP  FOR COMMUNICATION
      DO 16 J=1,15
      INDEX = INDEX +1
      DATA(INDEX)= SCALE*FLOAT(JPT(J))
   16 CONTINUE
   17 CONTINUE
      IF(INDEX-2970)10,18,18
   18 CONTINUE
C ***** FIND THE 7 HIGH AMPLITUDE PULSES
C **** THE 7 HIGH AMPS SHOULD BE WITHIN 45 WORDS AND SHOULD SHOW AN
C **** INCREASE OF AT LEAST .15 FOR AT LEAST 6 PULSES
      I = ISTRTS+1
   20 CONTINUE
      I = I+1
      SAVS = DATA(I-1)+.15
      IF(I-ISTPS)25,25,301
   25 IF(SAVS-DATA(I))30,30,20
   30 KOUN = 1
      NOS2 = I
      DO 39 J=1,45
      K = I+J
      SAVS = DATA(K) + .15
      IF(SAVS-DATA(K+1))35,35,39
   35 KOUN = KOUN +1
   39 CONTINUE
      IF(KOUN-5)20,40,40
   40 CONTINUE
      ITIM = 0
      SUMSP = 0
C ***** AVERAGE 45 SPACE VALUES
      DO 42 J=1,45
      K = NOS2 + J +55
      SUMSP = SUMSP + DATA(K)
   42 CONTINUE
      SUMSP = SUMSP  45.
   41 CONTINUE
C ***** FIND THE END OF A DATA LINE
      ISTRTD = NOS2 +630
      ISTPD = NOS2+710
   50 I=ISTRTD -1
   55 I=I+1
      SAVD = DATA(I-1)
      SAVDS = SAVD+.15
      IF(I-ISTPD)60,60,302
   60 IF(SAVDS  )-DATA(I+1))65,65,55
   65 IF((SAVDS-.05)-DATA(I))75,75,70
   70 NOD2 = I+1
      GO TO 80
   75 NOD2 = I
   80 KOUN = 0
      DO 90 J=1,30
      IF(DATA(J+NOD2)-SAVDS  )85,90,90
   85 KOUN = KOUN +1
      IF(KOUN-5)90,90,55
```

```
   90 CONTINUE
      ITIM = 0
   9' CONTINUE
C ***** FIND THE START OF THE WEDGE
      ISTRTC = NOD2 + 195
      ISTPC = NOD2 + 255
      NOD2 = NOD2 -582
      I = ISTRTC
  500 I=I+1
      SAVC = DATA(I-1)+.2
      IF(I-ISTPC)501,501,303
  501 IF(SAVC-DATA(I))502,502,500
  502 NOC2 = I
      KOUN = 0
      DO 510 J=3,27
      IF(DATA(J+I)-SAVC)505,505,510
  505 KOUN = KOUN +1
  510 CONTINUE
      IF(KOUN-3)520,500,500
  520 CONTINUE
      ITIM = 0
      SUMHS = 0
C ***** AVERAGE 200 HOUSING VALUES
      DO 98 J=1,200
      K = NOC2 + J + 220
      SUMHS = SUMHS + DATA(K)
   98 CONTINUE
      SUMHS = SUMHS/200
   99 CONTINUE
      EHS = 0.052593
      CON  = EHS/(SUMHS-SUMSP)
      II = 0
C ***** THERE IS A SCAN ANGLE OF 109 DEGREES
      DO 100 I=1,109,3
      DO 101 J=1,16
      II = II+1
      NN = NOD2+II-1
      E = CON*(DATA(NN)-SUMSP)
C ***** SUBROUTINE CHECKX DETERMINES THE TEMPERATURE IN KELVIN
      CALL CHECKX(E,DAT(J))
  101 CONTINUE
C ***** AVERAGE 5 AND 1/3 FOR EACH DATA POINT TO GET 109 VALUES
      SUM(I) =(3.*(DAT(1)+DAT(2)+DAT(3)+DAT(4)+DAT(5)) + DAT(6))/16.
     * + SUM(I)
      IF(I.EQ.109) GO TO 100
      SUM(I+1) = (3.*(DAT(7)+DAT(8)+DAT( 9)+DAT(10)) +2.*(DAT(6)+DAT(11)
     *))/16.
     * + SUM(I+1)
      SUM(I+2) = (3.*(DAT(12)+DAT(13)+DAT(14)+DAT(15) +DAT(16)) +DAT(11)
     *)/16.
     * + SUM(I+2)
 '022 FORMAT(1X,19F7.2)
  '00 CONTINUE
  '05 CONTINUE
C ***** SHIFT THE DATA IN THE STORED ARRAY TO MAKE ROOM FOR NEXT LINE
      '. = NOS2 + '909
      '. = '.990
      '8. = '.'990
      IF('.'130,130,'10
  '10 CONTINUE
      IF('.GT.3) STOP 3
      '88. = 2970-'8.
      DO '20 I='.'88.
      DATA(I) = DATA(I+'8.)
```

```
  120 CONTINUE
      INDEX = NNN
  130 CONTINUE
      NOS1 = NOS2-NN
C ***** STORE EACH DATA LINE IN MATRIX AM
      ICHE = ICHE +1
      IF(ICHE .LT.2) GO TO 5
      ICHE = 0
      LINE = LINE +1
      DO 180 I=1,109
      AM(I+5,61-LINE) = SUM(I) 2.
  180 CONTINUE
      DO 181 I=1,6
      AM(I,61-LINE) = AM(6,61-LINE)
      J = I+114
      AM(J,61-LINE) = AM(114,61-LINE)
  181 CONTINUE
      DO 161 I=1,360
  161 SUM(I) = 0.
  200 CONTINUE
      GO TO 5
  210 STOP 6
  301 ICHEX = 1
C ***** COULD  NOT FIND THE 7 PULSE-HIGH AMPLITUDE IN THE RANGE FROM ISTRTS
C ***** TO ISTPS
      IF(ISTRTS.EQ.0.AND.ISTPS.EQ.990) GO TO 306
      ITIM = ITIM +1
      IF(NOS2.GT.ISTRTS.AND.NOS2.LT.ISTPS) GO TO 310
      NOS2 = NOS1 +1943                                                    S
      GO TO 310
  302 ICHEX = 2
C ***** COULD NOT FIND THE DATA END IN THE RANGE FROM ISTRTD TO ISTPD
      ITIM = ITIM+1
      IF(NOD2.GE.ISTRTD.AND.NOD2.LT.ISTPD) GO TO 310
      NOD2 = NOS2 + 680                                                    S
      GO TO 310
  303 ITIM = ITIM +1
C ***** COULD NOT FIND THE WEDGE IN THE RANGE FROM ISTRTC TO ISTPC
      ICHEX = 3
      IF(NOC2.GE.ISTRTC.AND.NOC2.LT.ISTPC) GO TO 310
      NOC2 = NOD2 + 220                                                    S
      GO TO 310
  306 CONTINUE
C ***** DATA LINE DOES NOT START IN FIRST RECORD TRY NEXT RECORD
      DO 307 I=1,1980
      DATA(I) = DATA(I+990 )
  307 CONTINUE
      INDEX = 1980
      GO TO 10
  305 CONTINUE
      ICHEX = 5
C ***** THE RANGE IS OUT OF BOUNDS
      ISTPS = 2970
      IF(ITIM.GT.2.AND.NREC.GT.450) GO TO 410
 4000 FORMAT(8I10)
      WRITE(6,4000)NOS2,NOS1,NOC2,NOC2,ICHE1,NREC
  310 CONTINUE
      IF(ITIM.GT.10) STOP 10
  350 GO TO (41,9, 99,99,10),ICHE1
  410 CONTINUE
      IF(LINE.GE.60) GO TO 400
      LINE = LINE +1
      DO 420 I=1,120
      AM(I,61-LINE) = AM(I,62-LINE)
```

```
420 CONTINUE
    GO TO 410
400 CONTINUE
    RETURN
    END
```

```
      SUBROUTINE TWO
C ***********************************************************************
C *****  THIS IS MADDENS ROUTINE TO BE USED WITH CONTOURED DATA
C *****   D          =1   NORTH TO SOUTH
C *****   D          =2 SOUTH TO NORTH
C *****   H          HEIGHT
C *****   NDY        DAY OF MESSAGE
C *****   NMO        MONTH OF MESSAGE
C *****   ORBIT      ORBIT NUMBER
C *****   PL         ORBITAL DISPLACEMENT TO WEST
C *****   PMIN       ORBITAL PERIOD
C *****   PSEC       ORBITAL PERIOD (SEC)
C *****   RO         REFERENCE ORBIT NO.
C *****   RDY        REFERENCE DAY OF ASCENDING NODE
C *****   RSEC       REFERENCE SEC  OF ASCENDING NODE
C *****   RMIN       REFERENCE MIN  OF ASCENDING NODE
C *****   RHR        REFERENCE HOUR OF ASCENDING NODE
C *****   RQ         QUADRANT
C *****   RQ         =0   NORTH HEM. 0 TO 90 DEGREES WEST
C *****   RQ         =SEE APT USERS MANUAL
C *****   RL         REF LONGITUDE IN DEGREES
C *****   SAT        IDENTIFIER 08 = ESSA 8
C *****   SRPM       FIRST SCAN RPM
C *****   TSRH       FIRST SCAN HOUR
C *****   TSRM       FIRST SCAN MIN
C *****   TSRS       FIRST SCAN SEC
C ***********************************************************************
      COMMON AM(120,60)
      INTEGER A
      DIMENSION A(360)
      COMMON BLK
      COMMON/KEEP/TR,TIME,DT,DTSR,RRL,H,REFTIM,RDY,D,RL,DUM(297),      S
     *IDUM(4),LINE
C     USEFUL CONSTANTS
      EP=.001
C      EP COUNTERACTS ROUND OFF ERROR IN GOING FROM FLOATING TO INTEGER
      PI=3.1416
C *****  RADIUS OF EARTH
      R=6368.
C *****  DEGREES TO RADIANS
      C1=0.0174533
C *****  RADIANS TO DEGREES
      C2=1.0/C1
      RI = 11.9*C1                                                     S
      KTIME=0
      DELP=2.0
      RDELP=DELP*C1
      IF(TR)1,1,2
C     READ IN ORBITAL INFORMATION
    1 CONTINUE
      READ(5,100)D,NMO,NDY,SAT,RO,RDY,RHR,RMIN,RSEC,RQ,RL,PMIN,PSEC,
     *PL,H
  100 FORMAT(F1.0,I2,I2,F2.0,F5.0,1X,2F2.0,1X,2F2.0,F1.0,F4.2,1X,2F2.0,
     *1X,F4.2,F4.0)
C      PUT TIME INTO DECIMAL FORM AND REF ORBIT EQ OR LON IN DEG EAST
C      READ IN ORBIT NUM,HR MIN SEC OF FIRST SR SCAN AND RPM OF SR
      DT=PMIN*100.+PSEC/60.
      IF(PMIN.GT.50.)DT=DT-100.
      DT=DT/60.
      REFTIM=RHR+RMIN/60.+RSEC/3600.
      IF(RQ.EQ.2..AND.RL.LT.90.)RL=RL+100.
      IF(RQ.EQ.1..AND.RL.LT.90.)RL=360.-RL-100.
      IF(RQ.EQ.0..OR.RQ.EQ.5.)RL=360.-RL
```

```
      IF(RQ.EQ.7..AND.RL.LT.90.)RL=RL+100.
      IF(RQ.EQ.6..AND.RL.LT.90.)RL=360.-RL-100.
C      READ IN ORBIT DY,HR,MN,SC,OF FIRST SCAN, AND RPM OF SR
      READ (5,101)ORBIT,TSRH,TSRM,TSRS,SRPM,DRIV
  101 FORMAT(F5.0,3F2.0,F3.0,F5.0)
      TSR=TSRH+TSRM/60.+TSRS/3600.
      TIME=TSR
      TSR = TSR -1.7/60.                                                    S
      DTSR=1.0/SRPM/60.
      DTSR = DTSR*2.                                                        S
    3 CONTINUE
      IF(RO.EQ.ORBIT)GO TO 10
      RO=RO+1.
      REFTIM=REFTIM+DT
      IF(REFTIM.LT.24.)GO TO 21
      RDY=RDY+1.
      REFTIM=REFTIM-24.
   21 CONTINUE
      RL=RL-PL
      IF(RL.LT.0.)RL=360.+RL
      GO TO 3
   10 CONTINUE
C      1. DETERMINE THE LATITUDE (PHI) AND LONGITUDE (THET) OF SUBPOINT
      IF(REFTIM.GT.TSR)TSR=TSR+24.
      TR=TSR-REFTIM
      TR=TR-DTSR
      TIME=TIME-DTSR
      RRL=RL*C1
    2 CONTINUE
      TR=TR+DTSR
      TIME=TIME+DTSR
C       READ IN DIGITIZED SR DATA
C      FOR TEST FILL DATA ARRAY WITH NUMBERS IN A DO LOOP
      DO 1000 I=1,360
      A(I) = 0.
 1000 CONTINUE
      DEGAM=TR/DT*360.
      RDEGAM=DEGAM*C1
      RPHI=ASIN(SIN(RDEGAM)*COS(RI))
      RDLONG=ASIN(SIN(RDEGAM)*SIN(RI)/COS(RPHI))
      IF(D.EQ.1.)RDLONG=PI-RDLONG
C      ADJUST LONGITUDE (THET) TO ROTATING EARTH
      RTHET=RRL-RDLONG-15.*TR*C1
      IF(RTHET.LT.0.)RTHET=2.*PI+RTHET
      PHI=RPHI*C2
      THET=RTHET*C2
C      2. DETERMINE SCAN ANGLES FOR FIRST (SIGH) AND SECOND (SIGH2)
C         HORIZONS
      RETAH=ASIN(R/(R+H))
      ETAH=RETAH*C2
      RGAMH=PI*.5-RETAH
      GAMH=90.-ETAH
      SIGH=180.-ETAH
      SIGH2=180.+ETAH
      C1=SIGH-5.
      C2=C1+119
C      3. DETERMINE AZIMUTH OF HEADING LINE (RIS) AT SUBPOINT
      RIS=ASIN(SIN(RI)*COS(RPHI))
C      4. DETERMINE LAT LON. OF FIRST (PHIH THETH) AND SECOND (PHIH2
C         THETH2) HORIZONS
      RIST=RIS
      RPHIH=ASIN(SIN(RPHI)*COS(RGAMH)+COS(RPHI)*SIN(RGAMH)*SIN(RIST))
      FJD=ASIN(SIN(RGAMH)*COS(RIS)/COS(RPHIH))
      IF(D.EQ.1.)FJD=-FJD
```

```
      RTHETH=RTHET+FUD
      PHIH=RPHIH*C2
      THETH=RTHETH*C2
C      DETERMINE LAT LON OF SECOND HORIZON
      RIST2=RIS
      RPHIH2=ASIN(SIN(RPHI)*COS(RGAMH)-COS(RPHI)*SIN(RGAMH)*SIN(RIST2))
      FUD=ASIN(SIN(RGAMH)*COS(RIS)/COS(RPHIH2))
      IF(D.NE.1.)FUD=-FUD
      RTHETH2=RTHET+FUD
C        COME IN FROM HORIZON 5 DEGREES AND DETERMINE LONGITUDE CROSSINGS
      DEL=5.
      IF(D.NE.1.)DEL=-DEL
      RDEL=DEL*C1
      TEMP=THETH+DEL
      KTEMP=TEMP*.2
      THETG=KTEMP*5
      RTHETG=THETG*C1
   30 CONTINUE
C       5. DETERMINE GREAT CIRCLE ARC LINGTHS (GAMG) TO GIVEN LONGITUDES
C          (THETG)
      RDTHET=ABS(RTHETG-RTHET)
      IF(D.EQ.1..AND.RTHETG.GT.RTHET)GO TO 31
      IF(D.NE.1..AND.RTHETG.LT.RTHET)GO TO 31
C        SCAN IS ON RIGHT SIDE OF SUBPOINT
      RALPHA=ACOS(SIN(RDTHET)*COS(RIS)*SIN(RPHI)-COS(RDTHET)*SIN(RIS))
      GO TO 34
   31 CONTINUE
C        SCAN IS ON LEFT SIDE OF SUBPOINT
      RALPHA=ACOS(SIN(RDTHET)*COS(RIS)*SIN(RPHI)+COS(RDTHET)*SIN(RIS))
   34 CONTINUE
      RGAMG=ASIN(SIN(RDTHET)*COS(RPHI)/SIN(RALPHA))
C       6. DETERMINE SCAN ANGLE (SIGG) FOR A GIVEN GREAT CIRCLE ARC
C          LENGTH
      SIDE=SQRT(R*R+(R+H)*(R+H)-2.*R*(R+H)*COS(RGAMG))
      RETAG=ASIN(R*SIN(RGAMG)/SIDE)
      ETAG=RETAG*C2
      IF(D.NE.1..AND.RTHETG.LT.RTHET)ETAG=-ETAG
      IF(D.EQ.1..AND.RTHETG.GT.RTHET)ETAG=-ETAG
      SIGG=180.-ETAG
      JSIG=SIGG
      IF(JSIG.LT.130.OR.JSIG.GT.230) GO TO 37
      TT = 60-KTIME
      SIGG = JSIG-J1+1
      IF(SIGG.LT.1.) GO TO 37
      IF(SIGG.GT.120) GO TO 37
      IF(KTIME.GT.0) GO TO 35
C       FIRST SCAN NAME LONGITUDES
      THETG=RTHETG*C2
      KTHETG = THETG +EP
      CALL OPTION(0.0,0.0,1)
      CALL FRSTPT(SIGG,61.5)
      CALL NUMBR(KTHETG,2H13)
      CALL OPTION(0.0,0.0,0)
      GO TO 37
   35 CONTINUE
C        ALTER DATA ARRAY AT SCAN ANGLE SIGG
      A(JSIG) = 8
      CALL PWRT(SIGG,TT,1H*,1,1,0)
   37 CONTINUE
      RTHETG=RTHETG+RDEL
      IF(D.NE.1..AND.RTHETG.GT.RTHETH2)GO TO 30
      IF(D.EQ.1..AND.RTHETG.LT.RTHETH2)GO TO 30
C        DETERMINE LATITUDE CROSSINGS
C        FIRST LAT TO BE PLOTTED NEAREST 5 DEGREE TO HORIZON
```

```
      KTEMP=PHIH*0.5
      PHIG=KTEMP*2
        RPHIG=PHIG*C1
      LTIME=1
   61 CONTINUE
      PHIT=PHIG+DELP
      RPHIT=PHIT*C1
      TESTER=COS(RIS)*COS(RPHI)/COS(RPHIT)
      IF(TESTER.GT.1.)GO TO 40
      RPHIG=RPHIT
      PHIG=PHIT
      GO TO 61
   40 CONTINUE
      ICHEX = 10
C       7. DETERMINE GREAT CIRCLE ARC LENGTHS (GAMG) TO GIVEN LATITUDES
C          (PHIG)
      RALPHA=ASIN(COS(RIS)*COS(RPHI)/COS(RPHIG))
   48 CONTINUE
      IF(LTIME.EQ.1)GO TO 46
      IF(RPHIG.GT.RPHI.OR.ICHEX.EQ.0) RALPHA = PI-RALPHA
   46 CONTINUE
      FUD=COS(RPHI)*COS(RPHIG)*SIN(RIS)*COS(RALPHA)
      IF(RPHIG.GT.RPHI.OR.ICHEX.EQ.0) FUD = -FUD
      RGAMG=ACOS((SIN(RPHI)*SIN(RPHIG)+FUD)/(1.0-COS(RPHI)*COS(RPHIG)
     1       *COS(RIS)*SIN(RALPHA)))
C       6. DETERMINE SCAN ANGLE (SIGG) FOR A GIVEN GREAT CIRCLE ARC
C          LENGTH
      SIDE=SQRT(R*R+(R+H)*(R+H)-2.*R*(R+H)*COS(RGAMG))
      RETAG=ASIN(R*SIN(RGAMG)/SIDE)
      ETAG=RETAG*C2
      IF(RPHIG.GT.RPHI.OR.ICHEX.EQ.0) ETAG = -ETAG
      SIGG=180.+ETAG
C       ALTER DATA ARRAY AT SCAN ANGLE SIGG
      IPHIG = PHIG +EP
      TT = 60-KTIME
      IF(D.NE.1.)GO TO 41
      TEST=SIGG-SIGH
      IF(TEST.GT.3.)GO TO 42
      CALL OPTION(0.0,0.0,0.0)
      CALL FRSTPT(-5.0,TT)
      CALL NUMBR(IPHIG,2H13)
      GO TO 42
   41 CONTINUE
      TEST=SIGH2-SIGG
      IF(TEST.GT.3.)GO TO 42
      CALL OPTION(0.0,0.0,0.0)
      CALL FRSTPT(121.,TT)
      CALL NUMBR(IPHIG,2H13)
   42 CONTINUE
      JSIG=SIGG
      IF(JSIG.LT.128.OR.JSIG.GT.252) GO TO 45
      SIGG = JSIG-J1+1
      A(JSIG) = 9
      CALL PWRT(SIGG,TT,1H*,1,1,0)
   45 CONTINUE
      IF(LTIME.NE.1)GO TO 47
      LTIME=2
      GO TO 40
   47 CONTINUE
      LTIME = LTIME +1
      PHIG = PHIG - DELP
        RPHIG=PHIG*C1
      IF(RPHIG.GT.RPHIH2)GO TO 40
      IF(KTIME.GT.0)GO TO 45
```

```
      WRITE(6,103)ORBIT,TSRH,TSRM,TSRS,DEL,DELP
  103 FORMAT(1H1,' APT DATA FROM ORBIT',F7.0,'.   TIME OF FIRST SCAN IS
     1',3(1X,F3.0),//,' LONGITUDE AND LATITUDE LINES DRAWN IN AT ',F3.0,
     2' AND ',F3.0,' DEGREES RESPECTIVELY                    XXXXXXXXXXX
     3X',20X,'TIME')
   45 CONTINUE
      CALL OPTION(0,0,0,0,0)
      CALL FRSTPT(121.,TT)
      CALL NUMBR(TIME,4HF6.3)
      KTIME = KTIME +1
      WRITE(6,102)(A(I),I=J1,J2),TIME
  102 FORMAT(1X,12011,F7.3)
      IF(KTIME.LT. 60) GO TO 2
      RETURN
      END
```

```
      SUBROUTINE THREE
C ***************************************************************************
C
C ***** THIS ROUTINE CONTOURS DATA FROM APT TAPES OR IR TAPES
C ***************************************************************************
      COMMON AM(120,60)
      COMMON/C/IC
      COMMON /EDGE/ ILEDGE,IHEDGE,JLEDGE,JHEDGE
      CALL FRSTPT(120.,60.)
      CALL MXMY(MX,MY)
      IHEDGE = MX-12
      JHEDGE = MY-8
      CALL FRSTPT(1.,1.)
      CALL MXMY(MX,MY)
      ILEDGE = MX+12
      JLEDGE = MY+16
      CALL CALCNT(AM,120,60,200.,300.,20.,1,20,0,1)
      RETURN
      END
```

```
      SUBROUTINE CALCNT(Z,M,N,FLO,HI,FINC,NSET,MHI,MDOT,NOQ)
C ..................................................................
C
C ***** THIS IS THE BASIC CONTOUR ROUTINE
C ..................................................................
      DIMENSION Z(M,N)
      COMMON/CONT/M1,M2,M3,M4,N1,N2,N3,N4,ASH,MNT,NQ,NL,CL(100),JPROJ
      COMMON /EDGE/ ILEDGE,IHEDGE,JLEDGE,JHEDGE
      JPROJ=NOQ
      M1=1       $M2=M1+1    $M3=M-1     $M4=M
      N1=1       $N2=N1+1    $N3=N-1     $N4=N
      X1=.05     $X2=.95     $X3=M1      $X4=M4
      Y1=.05     $Y2=.95     $Y3=N1      $Y4=N4
      GLO=FLO    $HA=HI      $FANC=FINC  $IDASH=MDOT
      CRIT=16.   $NNT=3
      NQ=AMAX0 (1,ABS (NSET))
   10 IF(HA-GLO)11,12,20
   11 GLO=HI   $HA=FLO   $GO TO 20
   12 GLO=HA=Z(M1,N1)   $DO 13 J=N1,N4 $DO13 I=M1,M4
      GLO=MIN1 (Z(I,J),GLO)
   13 HA=MAX1 (Z(I,J),HA)
   20 IF(FANC)21,22,24
   21 CRIT=-FANC
   22 FANC=(HA-GLO)/CRIT     $IF(FANC)90,90,23
   23 P=10.**(INT (ALOG10(FANC)))
      IF(FANC.LE.1.)P=P/10.
      FANC=INT (FANC/P)*P
   24 IF(HI-FLO)26,25,26
   25 GLO=INT (GLO/FANC)*FANC
      HA=INT (HA/FANC)*FANC
C  26 AH=MAX1 (ABS (GLO),ABS (HA))
C     ASH=10.**(2-INT (ALOG10(AH)))
C     IF(AH.LT.1)ASH=ASH*10.
   26 ASH=1.
   27 IFLAG=0
C     IF (HA.GT.999.9.OR.GLO.LT.-99.9) IFLAG=1
C     IF (IFLAG.EQ.0) GO TO 28
C     ASH=ASH*.1
C     HA=HA*ASH
C     GLO=GLO*ASH
C     GO TO 27
C  28 CONTINUE
   30 CALL HILO(Z,M,N)    $GO TO 40
C  31 CALL PTVALU(Z,M,N)
   40 DO 41 I=1,100
      CL(I)=GLO+(I-1)*FANC$NL=I-1$IF(CL(I).GT.HA)42,41
   41 CONTINUE    $NL=100
   42 CONTINUE
      CALL QUICK(Z,M,N)
   50 RETURN
   90 CONTINUE                                                    TEMP
      RETURN
      END
```

```
      SUBROUTINE PROJUV(U1,V1)
      COMMON/JUV/EP,PI180,XLAT,XLO1,XLATS,XLONS,IHM
C     CALCNT CALLS PROJUV TO PROVIDE TRANFORMS TO STEREOGRAPHIC PROJECTION.
C     CF. SUBROUTINES QUICK AND HILO.
      X=Y=0. $ V=(V1-1.)*XLAT+XLATS $ IF(V.GT.EP)GO TO 12
      R=(1.-SIN(V))/COS(V)
      U=(U1-1.)*XLO1+XLONS $ IF(U.GT.180.)U=U-360.
      X=ABS(R*SIN(U*PI180)) $ Y=SQRT(R*R-X*X)
      IF(U.LT.0.)X=-X $ IF(IHM.GT.0) GO TO 11
      IF(ABS(U).GT.90.)Y=-Y $ GO TO 12
   11 IF(ABS(U).LT.90.)Y=-Y
   12 U1=X $ V1=Y $ RETURN
      END
```

```
      SUBROUTINE QUICK(Z,M,N)
      DIMENSION Z(M,N)
      COMMON/CONT/M1,M2,M3,M4,N1,N2,N3,N4,ASH,NLT,NG,NL,CL(100),JPROJ
      C(P1,P2,B)=B+(P1-CV)*(P1-P2)
      IDUB=0
      DO 1 J=N1,N3      $Y1=J     $Y2=J+1
      DO 2 I=M1,M3      $X1=I     $X2=I+1
      H1=Z(I,J)  $H2=Z(I,J+1)   $H3=Z(I+1,J+1)  $H4=Z(I+1,J)
      DO 3 K=1,NL       $CV=CL(K)
   10 IF(H1-CV)11,20,20
   11 IF(H2-CV)12,14,14
   12 IF(H3-CV)13,15,15
   13 IF(H4-CV)2,2,30
   14 IF(H3-CV)16,17,17
   15 IF(H4-CV)31,32,32
   16 IF(H4-CV)33,34,34
   17 IF(H4-CV)35,36,36
   20 IF(H2-CV)21,23,23
   21 IF(H3-CV)22,24,24
   22 IF(H4-CV)36,35,35
   23 IF(H3-CV)25,26,26
   24 IF(H4-CV)37,33,33
   25 IF(H4-CV)32,31,31
   26 IF(H4-CV)30,3,3
   30 XA=C(H1,H4,X1) $YB=C(H4,H3,Y1)  $XB=X2   $YA=Y1            $GOTO40
   31 XA=C(H2,H3,X1) $YB=C(H4,H3,Y1)  $XB=X2   $YA=Y2            $GOTO40
   32 XA=C(H1,H4,X1) $XB=C(H2,H3,X1)  $YA=Y1   $YB=Y2            $GOTO40
   33 XB=C(H2,H3,X1) $YA=C(H1,H2,Y1)  $XA=X1   $YB=Y2  $IDUB=0  $GOTO40
   34 IDUB=1  $GOTO30
   35 YA=C(H1,H2,Y1) $YB=C(H4,H3,Y1)  $XA=X1   $XB=X2            $GOTO40
   36 XB=C(H1,H4,X1) $YA=C(H1,H2,Y1)  $XA=X1   $YB=Y1  $IDUB=3  $GOTO40
   37 IDUB=-1 $GOTO31
   40 GO TO (41,39) JPROJ
   39 CALL PROJUV(XA,YA) $ CALL PROJUV(XB,YB)
   41 CALL LINE(XA,YA,XB,YB)      $IF(IDUB)36,3,99
    3 CONTINUE
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE HILO(Z,M,N)
      DIMENSION Z(M,N)
      COMMON /CONT/ M1,M2,M3,M4,N1,N2,N3,N4,ASH,NNT,NQ,NL,CL(100),JPROJ
      COMMON /EDGE/ ILEDGE,IHEDGE,JLEDGE,JHEDGE
      DO 10 J=N1,N4
      JJ=AMAX0 (J-NNT,N1)
      JJJ=AMIN0 (J+NNT,N4)
      IJ=JJ
      IF(IJ.EQ.J)IJ=JJJ
      DO 10 I=M1,M4
      II=AMAX0 (I-NNT,M1)
      III=AMIN0 (I+NNT,M4)
  100 FORMAT(I6)
      ZZ=Z(I,J)     $IF(ZZ-Z(I,IJ))1,10,5
    1 DO 2 JK=JJ,JJJ
      DO 2 IK=II,III
      IF(JK.EQ.J.AND.IK.EQ.I)GO TO 2
      IF(ZZ-Z(IK,JK))2,10,10
    2 CONTINUE  $JS=1HL
      GO TO 7
    5 DO 6 JK=JJ,JJJ
      DO 6 IK=II,III
      IF(I.EQ.IK.AND.J.EQ.JK)GO TO 6
      IF(ZZ-Z(IK,JK))10,10,6
    6 CONTINUE     $JS=1HH
    7 ZZ=ZZ*ASH  $X=I  $Y=J
      GO TO (9,8) JPROJ
    8 CALL PROJUV(X,Y)
    9 CALL FRSTPT (X,Y)
      CALL MXMY (II,III)
      IF (II.LT.ILEDGE.OR.II.GT.IHEDGE) GO TO 10
      IF (III.LT.JLEDGE.OR.III.GT.JHEDGE) GO TO 10
      IZZ=ZZ+.499999
      ENCODE(6,100,IK)IZZ
      CALL PSYM(II,III,JS,1,0,1)
      CALL PWRT(II-28,III-16,IK,6,0,0)
   10 CONTINUE
      RETURN
      ENTRY PTVALU
C     II=(M4-M1+24)/24  $JJJ=(N4-N1+48)/48 $NIQ=M1-1+NQ $NJQ=N1-1+NQ      PERM
C     DO 20 J=NJQ,N4,JJJ  $Y=J $DO 20 I=NIQ,M4,II   $X=I                  PERM
C     ZZ=Z(I,J)*ASH
C     ENCODE(4,100,IK)ZZ
C     IK=IK.AND.4777777777777777777777B
C  20 CALL PWRT(X,Y,IK,4,0,0)
      RETURN
      END
```

```
      PROGRAM SRGRID
     *(INPUT=2028,TAPE5=INPUT,OUTPUT=2028,TAPE6=OUTPUT,TAPE1)
C ***** ************************************************************************
C ***** THIS IS MADDENS DECK. IT CAN BE USED FOR A HIGH OR LOW RESOLUTION
C ***** IT CAN BE USED WITH APT OR IR DATA
C *****  D          =1   NORTH TO SOUTH
C *****  D          =2 SOUTH TO NORTH
C *****  KF         START POSITION TO OUTPUT
C *****             FOR LOW RESOLUTION KF = J1
C *****  KL         END   POSITION TO OUTPUT
C *****             FOR LOW RESOLUTION KL = J2
C *****  KSTOP      NUMBER OF LINES TO PRINT
C *****  H          HEIGHT
C *****  NDY        DAY OF MESSAGE
C *****  NMO        MONTH OF MESSAGE
C *****  PL         ORBITAL DISPLACEMENT TO WEST
C *****  PMIN       ORBITAL PERIOD
C *****  PSEC       ORBITAL PERIOD (SEC)
C *****  ORBIT      ORBIT NUMBER
C *****  ORIN       ORBIT INCLINATION
C *****  RDY        REFERENCE DAY OF ASCENDING NODE
C *****  RHR        REFERENCE HOUR OF ASCENDING NODE
C *****  RL         REF LONGITUDE IN DEGREES
C *****  RMIN       REFERENCE MIN  OF ASCENDING NODE
C *****  RO         REFERENCE ORBIT NO.
C *****  RQ         QUADRANT
C *****  RQ         =0   NORTH HEM. 0 TO 90 DEGREES WEST
C *****  RQ         =SEE APT USERS MANUAL
C *****  RSEC       REFERENCE SEC  OF ASCENDING NODE
C *****  SAT        IDENTIFIER 08 = ESSA 8
C *****  SRPM       FIRST SCAN RPM
C *****  TMULP      A FACTOR TO MULTIPLY POSITIONS OF LAT AND LON BY
C *****  TMULP      =8.3333333333333333 FOR APT DATA
C *****  TMULP      =5.3333333333333333 FOR IR DATA
C *****             FOR LOW RESOLUTION TMULP = 1.
C *****  TSRH       FIRST SCAN HOUR
C *****  TSRM       FIRST SCAN MIN
C *****  TSRS       FIRST SCAN SEC
C ***** ************************************************************************
      COMMON/MAIN/A(3000)                                                      S
C     USEFUL CONSTANTS
      TMULP = 8.3333333333333333                                               S
      EP=.001
C      EP COUNTERACTS ROUND OFF ERROR IN GOING FROM FLOATING TO INTEGER
      READ(5,1002) KF,KL
      WRITE(6,1002) KF,KL
 1002 FORMAT(2I5)
      IF(KL.GT.(KF+119)) KL = KF+119
      KSTOP = 0
      PI=3.1416
C *****  RADIUS OF EARTH
      R=6368.
C *****  DEGREES TO RADIANS
      C1=0.0174533
C *****  RADIANS TO DEGREES
      C2=1.0/C1
      KTIME=0
      KORE=5
      DELP=2.0
      RDELP=DELP*C1
C     READ IN ORBITAL INFORMATION
     * CONTINUE
      READ(5,1001)D,NMO,NDY,SAT,RO,RDY,RHR,RMIN,RSEC,RQ,RL,PMIN,PSEC,
```

```
     1PL,H
      WRITE(6,1002) NMO,NDY
  100 FORMAT(F1.0,I2,I2,F2.0,F5.0,1X,2F2.0,1X,2F2.0,F1.0,F4.2,1X,2F2.0,
     11X,F4.2,F4.0)
C       PUT TIME INTO DECIMAL FORM AND REF ORBIT EQ CR LON IN DEG EAST
C       READ IN ORBIT NUM,HR MIN SEC OF FIRST SR SCAN AND RPM OF SR
       DT=PMIN*100.+PSEC/60.
       IF(PMIN.GT.50.)DT=DT-100.
       DT=DT/60.
       REFTIM=RHR+RMIN/60.+RSEC/3600.
       IF(RQ.EQ.2..AND.RL.LT.90.)RL=RL+100.
       IF(RQ.EQ.1..AND.RL.LT.90.)RL=360.-RL-100.
       IF(RQ.EQ.0..OR.RQ.EQ.5.)RL=360.-RL
       IF(RQ.EQ.7..AND.RL.LT.90.)RL=RL+100.
       IF(RQ.EQ.6..AND.RL.LT.90.)RL=360.-RL-100.
C       READ IN ORBIT DY,HR,MN,SC,OF FIRST SCAN AND RPM OF SR
       READ (5,101)ORBIT,TSRH,TSRM,TSRS,SRPM,ORIN
       WRITE(6,101) ORBIT
  101 FORMAT(F5.0,3F2.0,F3.0,F5.0)
       RI = C1*ORIN
       TSR=TSRH+TSRM/60.+TSRS/3600.
       TIME=TSR
       DTSR=1.0/SRPM/60.
     3 CONTINUE
       IF(RO.EQ.ORBIT)GO TO 10
       RO=RO+1.
       REFTIM=REFTIM+DT
       IF(REFTIM.LT.24.)GO TO 21
       RDY=RDY+1.
       REFTIM=REFTIM-24.
    21 CONTINUE
       RL=RL-PL
       IF(RL.LT.0.)RL=360.+RL
       GO TO 3
    10 CONTINUE
C       1. DETERMINE THE LATITUDE (PHI) AND LONGITUDE (THET) OF SUBPOINT
       IF(REFTIM.GT.TSR)TSR=TSR+24.
       TR=TSR-REFTIM
       TR=TR-DTSR
       TIME=TIME-DTSR
       RRL=RL*C1
     2 CONTINUE
       TR=TR+DTSR
       TIME=TIME+DTSR
C         READ IN DIGITIZED SR DATA
C       ******************************************************************
       DO 1003 I=1,3000                                                    S
       A(I) = 1H
 1003 CONTINUE
C **** CALL FIND FOR HIGH RESOLUTION . CALL FINDSR FOR LOW RESOLUTION
       CALL FIND                                                           S
       KSTOP=KSTOP+1
C ***** THIS IS THE NUMBER OF THE LINE TO START PRINTING
       IF(KSTOP.LT.1  )GO TO 2
       DEGAM=TR/DT*360.
       RDEGAM=DEGAM*C1
       RPHI=ASIN(SIN(RDEGAM)*COS(RI))
       RDLONG=ASIN(SIN(RDEGAM)*SIN(RI)/COS(RPHI))
       IF(D.EQ.1.)RDLONG=PI-RDLONG
C       ADJUST LONGITUDE (THET) TO ROTATING EARTH
       RTHET=RRL+RDLONG-15.*TR*C1
       IF(RTHET.LT.0.)RTHET=2.*PI+RTHET
       PHI=RPHI*C2
       THET=RTHET*C2
```

```
C       2. DETERMINE SCAN ANGLES FOR FIRST (SIGH) AND SECOND (SIGH2)
C          HORIZONS
        RETAH=ASIN(R/(R+H))
        ETAH=RETAH*C2
        RGAMH=PI*.5-RETAH
        GAMH=90.-ETAH
        SIGH=180.-ETAH
        SIGH2=180.+ETAH
        J1=SIGH-5.
        J2=J1+119
C       SIMULATE OUTER SPACE WITH ZEROS
        JJJ1 =(J1-1)*TMULP + 1.5
        JJ1 = J1*TMULP +5
        DO 2000 I=JJJ1,JJ1
        A(I) = 1H*
 2000 CONTINUE
        JJ2 = (SIGH2-1)*TMULP +1.5
        JJJ2 = JJ2+20
        DO 2001 I=JJ2,JJJ2
        A(I) = 1H*
 2001 CONTINUE
C       ************************************************************
C       3. DETERMINE AZIMUTH OF HEADING LINE (RIS) AT SUBPOINT
        RIS=ASIN(SIN(RI)/COS(RPHI))
C       4. DETERMINE LAT/LON OF FIRST (PHIH THETH) AND SECOND (PHIH2
C          THETH2) HORIZONS
        RIST=RIS
        RPHIH=ASIN(SIN(RPHI)*COS(RGAMH)+COS(RPHI)*SIN(RGAMH)*SIN(RIST))
        FUD=ASIN(SIN(RGAMH)*COS(RIS)/COS(RPHIH))
        IF(D.EQ.1.)FUD=-FUD
        RTHETH=RTHET+FUD
        PHIH=RPHIH*C2
        THETH=RTHETH*C2
C       DETERMINE LAT/LON OF SECOND HORIZON
        RIST2=RIS
        RPHIH2=ASIN(SIN(RPHI)*COS(RGAMH)-COS(RPHI)*SIN(RGAMH)*SIN(RIST2))
        FUD=ASIN(SIN(RGAMH)*COS(RIS)/COS(RPHIH2))
        IF(D.NE.1.)FUD=-FUD
        RTHETH2=RTHET+FUD
C         COME IN FROM HORIZON 5 DEGREES AND DETERMINE LONGITUDE CROSSINGS
        DEL=5.
        IF(D.NE.1.)DEL=-DEL
        RDEL=DEL*C1
        TEMP=THETH+DEL
        KTEMP=TEMP*.2
        THETG=KTEMP*5
        RTHETG=THETG*C1
   30 CONTINUE
C       5. DETERMINE GREAT CIRCLE ARC LENGTHS (GAMG) TO GIVEN LONGITUDES
C          (THETG)
        RDTHET=ABS(RTHETG-RTHET)
        IF(D.EQ.1..AND.RTHETG.GT.RTHET)GO TO 31
        IF(D.NE.1..AND.RTHETG.LT.RTHET)GO TO 31
C         SCAN IS ON RIGHT SIDE OF SUBPOINT
        RALPHA=ACOS(SIN(RDTHET)*COS(RIS)*SIN(RPHI)-COS(RDTHET)*SIN(RIS))
        GO TO 34
   31 CONTINUE
C         SCAN IS ON LEFT SIDE OF SUBPOINT
        RALPHA=ACOS(SIN(RDTHET)*COS(RIS)*SIN(RPHI)+COS(RDTHET)*SIN(RIS))
   34 CONTINUE
        RGAMG=ASIN(SIN(RDTHET)*COS(RPHI)/SIN(RALPHA))
C       6. DETERMINE SCAN ANGLE (SIGG) FOR A GIVEN GREAT CIRCLE ARC
C          LENGTH
        SIDE=SQRT(R*R+(R+H)*(R+H)-2.*R*(R+H)*COS(RGAMG))
```

```
      RETAG=ASIN(R*SIN(RGAMG)/SIDE)
      ETAG=RETAG*C2
      IF(D.NE.1..AND.RTHETG.LT.RTHET)ETAG=-ETAG
      IF(D.EQ.1..AND.RTHETG.GT.RTHET)ETAG=-ETAG
      SIGG=180.-ETAG
      JSIG=SIGG *TMULP +.5
      IF(KTIME.EQ.1)GO TO 35
C      FIRST SCAN NAME LONGITUDES
      THETG=RTHETG*C2
      KHUN=THETG*.01+EP
      XHUN=KHUN*100
      KTEN=(THETG-XHUN)*.1+EP
      XTEN=KTEN*10
      KONE=THETG-(XHUN+XTEN)+EP
      CALL SNUM(A(JSIG),1.,1.,KTEN,IERR)
      IF(IERR.NE.0) STOP 1
      IF(D.NE.1.)GO TO 36
      CALL SNUM(A(JSIG-1),1.,1.,KHUN,IERR)
      IF(IERR.NE.0) STOP 2
      CALL SNUM(A(JSIG+1),1.,1.,KONE,IERR)
      IF(IERR.NE.0) STOP 3
      GO TO 37
   36 CONTINUE
      CALL SNUM(A(JSIG+1),1.,1.,KHUN,IERR)
      IF(IERR.NE.0) STOP 4
      CALL SNUM(A(JSIG-1),1.,1.,KONE,IERR)
      IF(IERR.NE.0) STOP 5
      GO TO 37
   35 CONTINUE
C       ALTER DATA ARRAY AT SCAN ANGLE SIGG
      A(JSIG) = 1H/
   37 CONTINUE
      RTHETG=RTHETG+RDEL
      IF(D.NE.1..AND.RTHETG.GT.RTHETH2)GO TO 50
      IF(D.EQ.1..AND.RTHETG.LT.RTHETH2)GO TO 50
C       DETERMINE LATITUDE CROSSINGS
C       FIRST LAT TO BE PLOTTED NEAREST 5 DEGREE TO HORIZON
      KTEMP=PHIH*0.5
      PHIG=KTEMP*2
        RPHIG=PHIG*C1
      LTIME=1
   61 CONTINUE
      PHIT=PHIG+DELP
      RPHIT=PHIT*C1
      TESTER=COS(RIS)*COS(RPHI)/COS(RPHIT)
      IF(TESTER.GT.1.)GO TO 40
      RPHIG=RPHIT
      PHIG=PHIT
      GO TO 61
   40 CONTINUE
      ICHEX = 10
C       7. DETERMINE GREAT CIRCLE ARC LENGTHS (GAMG) TO GIVEN LATITUDES
C          (PHIG)
      RALPHA=ASIN(COS(RIS)*COS(RPHI)/COS(RPHIG))
   48 CONTINUE
      IF(LTIME.EQ.1)GO TO 46
      IF(RPHIG.GT.RPHI.OR.ICHEX.EQ.0) RALPHA = PI-RALPHA
   46 CONTINUE
      FUD=COS(RPHI)*COS(RPHIG)*SIN(RIS)*COS(RALPHA)
      IF(RPHIG.GT.RPHI.OR.ICHEX.EQ.0) FUD = -FUD
      RGAMG=ACOS((SIN(RPHI)*SIN(RPHIG)+FUD)/(1.0-COS(RPHI)*COS(RPHIG)
     *       *COS(RIS)*SIN(RALPHA)))
C       6. DETERMINE SCAN ANGLE (SIGG) FOR A GIVEN GREAT CIRCLE ARC
C          LENGTH
```

```
      SIDE=SQRT(R*R+(R+H)*(R+H)-2.*R*(R+H)*COS(RGAMG))
      RETAG=ASIN(R*SIN(RGAMG)/SIDE)
      ETAG=RETAG*C2
      IF(RPHIG.GT.RPHI.OR.ICHK.EQ.0) ETAG = -ETAG
      SIGG=180.+ETAG
      IF(D.NE.1.)GO TO 41
      TEST=SIGG-SIGH
      IF(TEST.GT.3.)GO TO 42
      JTEN=PHIG*0.1+EP
      XTEN=JTEN*10
      JONE=(PHIG-XTEN)+EP
      JSIGH=SIGH*TMULP+.5
      CALL SNUM(A(JSIGH),1,1,JTEN,IERR)
      IF(IERR.NE.0) STOP 6
      CALL SNUM(A(JSIGH+1),1,1,JONE,IERR)
      IF(IERR.NE.0) STOP 7
      GO TO 43
   41 CONTINUE
      TEST=SIGH2-SIGG
      IF(TEST.GT.3.)GO TO 42
      JTEN=PHIG*0.1
      XTEN=JTEN*10
      JONE=(PHIG-XTEN)
      JSIGH2=SIGH2*TMULP+.5
      CALL SNUM(A(JSIGH2),1,1,JTEN,IERR)
      IF(IERR.NE.0) STOP 10
      CALL SNUM(A(JSIGH2-1),1,1,JONE,IERR)
      IF(IERR.NE.0) STOP 11
      GO TO 43
   42 CONTINUE
      JSIG=SIGG*TMULP+.5
      A(JSIG) = 1H
   43 CONTINUE
      IF(LTIME.NE.1)GO TO 47
      LTIME=2
      GO TO 40
   47 CONTINUE
      LTIME = LTIME +1
      PHIG = PHIG - DELP
        RPHIG=PHIG*C1
      IF(RPHIG.GT.RPHIH2)GO TO 40
      IF(KTIME.EQ.1)GO TO 45
      WRITE(6,105)ORBIT,TSRH,TSRM,TSRS,DEL,DELP
C *****  THIS CARD IS SET FOR APT DATA ONLY
  105 FORMAT(1H1,' APT DATA FROM ORBIT',F7.0,'.   TIME OF FIRST SCAN IS
     1',3(1X,F5.0),'  DATE 6/25 71'
     2'                .' LONGITUDE AND LATITUDE LINES DRAWN IN AT ',F5.0,
     2' AND ',F5.0,' DEGREES RESPECTIVELY'
     3X,'.20X,'TIME')
      KTIME=1
   45 CONTINUE
      WRITE(6,102)(A(I),I=KF,KL),TIME
  102 FORMAT(1X,120A1,F7.3)
      IF(KSTOP.LT.120)GO TO 2
      END
```

```
      SUBROUTINE FIND
C ***** ***************************************************************
C
C ***** THIS ROUTINE FINDS ONE LINE OF APT DATA FOR MADDENS HIGH RESOLUTION
C ***** IF          BEGINNING INDICATOR
C ***** ISTPS       STOP  POSITION TO SEARCH FOR THE SYNC
C ***** ISTRTS      START POSITION TO SEARCH FOR THE SYNC
C ***** ITIM        COUNTER  FOR MISSING PARTS OF A LINE
C ***** LINE        LINE COUNT
C ***** NOD2        POSITION OF NEW DATA LINE
C ***** NOS1        BEGINNING OF PREVIOUS SYNC
C ***** NOS2        BEGINNING OF NEW SYNC
C ***** NREC        RECORD COUNT
C ***** TOP         HIGHEST SYNC PULSE FOR A LINE
C ***** ***************************************************************
      COMMON DATA(2415)
      DIMENSION CHAR(16),IBUF(360)
      COMMON/MAIN/SUM(3000)                                              S
      COMMON/UNP/JWD(3),JPT(15)
      DATA CHAR/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,
     *1HO,1HP/
      DATA IF/0/
      LINE = 0
      IF(IF)1,1,5
    1 LINE = 0
      IF = 1
      INDEX = 0
      NREC = 0
      SCALE = 1.414/2047.
      NOS1 = 0.
      ITIM =0
C ***** AT THE FIRST SEARCH ONE RECORD AT A TIME FOR THE SYNC PULSE
      ISTRTS = 0
      ISTPS = 800
      GO TO 10
    5 ISTRTS = NOS1 + 750
      IF(ISTRTS.LT.0) ISTRTS = 0
      IF(ISTRTS.LT.0) WRITE(6,4000)ISTRTS
      ISTPS = NOS1 + 850
      IF(ISTPS-2400)7,7,305
    7 CONTINUE
      IF(INDEX-2400)10,10,10
   10 CONTINUE
      NREC = NREC +1
C ***** THE APT DATA IS PACKED ON A TAPE WITH 800 (12-BIT WORDS)
C ***** PER RECORD
C ***** EACH SCAN LINE CONTAINS APPROXIMATELY 800  WORDS
C ***** STORE 3 RECORDS AT A TIME TO FIND ONE COMPLETE LINE
      BUFFER IN(1,1) (IBUF(1),IBUF(250))
      IF(UNIT(1))13,11,12
   11 CONTINUE
      WRITE(6,2000) NREC
 2000 FORMAT(*   EOF AT RECORD  *,I10)
      STOP
   12 CONTINUE
      WRITE(6,3000) NREC
 3000 FORMAT(*   PARITY ERROR ON RECORD *,I10)
   13 CONTINUE
      NWDS = LENGTH(1)
      II = 0
      DO  17 I=1,NWDS,3
      DO 15 J=1,3
      JWD(J) = IBUF(I+J-1)
   15 CONTINUE
```

```
      CALL UNPACK
C ***** UNPACK IS A COMPASS-CODED SUBROUTINE USING 'UNP' FOR COMMUNICATION
      DO 16 J=1,15
      II = II+1
      IF(II.GT.800) GO TO 16
      INDEX = INDEX +1
      DATA(INDEX)= SCALE*FLOAT(JPT(J))
   16 CONTINUE
   17 CONTINUE
      IF(INDEX-2400)10,18,18
   18 CONTINUE
      I = ISTRTS+1
   20 CONTINUE
C ***** FIND THE SYNC PULSE WHICH SHOULD LAST AT LEAST 25 POINTS
      I = I+1
      IF(I-ISTPS)25,25,301
   25 IF(DATA(I)-.55)20,30,30
   30 KOUN = 0
      NOS2 = I
      TOP = DATA(I)
      DO 40 J=1,35
      IF(DATA(I+J)-.55)35,41,41
   35 KOUN = KOUN +1
      IF(KOUN-5)40,40,20
   41 CONTINUE
      IF(DATA(I).GT.TOP) TOP = DATA(I)
   40 CONTINUE
      ITIM = 0
   43 CONTINUE
      NOD2 = NOS2 +48
C ***** STORE THE DATA LINE IN SUM
      DO 100 I=1,750
      SUM(I) = DATA(NOD2+I-1)
  100 CONTINUE
      N = NOS2+750
      N = N/800
      NN = N*800
      IF(N)130,130,110
  110 CONTINUE
      IF(N.GT.3) STOP 3
C ***** SHIFT THE DATA IN THE STORED ARRAY TO MAKE ROOM FOR NEXT LINE
      NNN = 2400-NN
      DO 120 I=1,NNN
      DATA(I) = DATA(I+NN)
  120 CONTINUE
      INDEX = NNN
  130 CONTINUE
      NOS1 = NOS2-NN
      XINT = TOP/15
      DO 180 I=1,750
      XMEAN = SUM(I)
      X =-XINT
      DO 160 J=1,16
      X = X+XINT
      IF(XMEAN- X)170,170,160
  160 CONTINUE
      J = 16
  170 CONTINUE
      SUM(I+1125) = CHAR(J)
  180 CONTINUE
      DO 161 I=1,1125
  161 SUM(I) = 1H
      DO 162 I=1876,3000
  162 SUM(I) = 1H
```

```
  200 RETURN
  210 CONTINUE
      WRITE(6,5000) (DATA(I),I=1,2400)
 5000 FORMAT(1X,15F6.3)
      STOP 16
  301 ICHEX = 1
C ***** COULD NOT FIND SYNC PULSE
      IF(ISTRTS.EQ.0.AND.ISTPS.EQ.800) GO TO 306
      ITIM = ITIM +1
      IF(NOS2.GT.ISTRTS.AND.NOS2.LT.ISTPS) GO TO 310
      NOS2 = NOS1 +800
      GO TO 310
  304 ICHEX = 4
      GO TO 310
  305 CONTINUE
C ***** OUT OF RANGE OF STORED DATA
      ISTPS =2400
      ICHEX = 5
  306 CONTINUE
C ***** DATA LINE DOES NOT START IN; FIRST RECORD TRY NEXT RECORD
      DO 307 I=1,1600
      DATA(I) = DATA(I+800 )
  307 CONTINUE
      INDEX = 1600
      GO TO 10
  310 CONTINUE
      WRITE(6,4000)NOS2,NOS1,NOD2,      ICHEX,NREC
 4000 FORMAT(8I10)
      IF(ITIM- 3)350,350,210
  350 GO TO (45,10,400,10,10),ICHEX
  400 CONTINUE
      END
```

```
      SUBROUTINE FIND
C ***** ********************************************************************
C ***** THIS ROUTINE FINDS ONE LINE OF IR DATA FOR MADDENS HIGH RESOLUTION
C ***** IF          BEGINNING INDICATOR
C ***** ISTPC       STOP  POSITION TO SEARCH FOR THE WEDGE
C ***** ISTPD       STOP  POSITION TO SEARCH FOR THE END OF THE DATA
C ***** ISTPS       STOP  POSITION TO SEARCH FOR THE 7 HIGH AMPS
C ***** ISTRTC      START POSITION TO SEARCH FOR THE WEDGE
C ***** ISTRTD      START POSITION TO SEARCH FOR THE END OF THE DATA
C ***** ISTRTS      START POSITION TO SEARCH FOR THE 7 HIGH AMPS
C ***** ITIM        COUNTER  FOR MISSING PARTS OF A LINE
C ***** LINE        LINE COUNT
C ***** NOC2        POSITION OF WEDGE START
C ***** NOD2        POSITION OF NEW DATA LINE
C ***** NOS1        BEGINNING OF PREVIOUS 7 PULSE- HIGH AMPLITUDE
C ***** NOS2        BEGINNING OF NEW      7 PULSE- HIGH AMPLITUDE
C ***** NREC        RECORD COUNT
C ***** SUMHS       AVERAGE 200 VALUES TO DETERMINE THE HOUSING VALUE
C ***** SUMSP       AVERAGE 45 VALUES TO DETERMINE THE SPACE VALUE
C ***** ********************************************************************
      COMMON DATA(2970)
      DIMENSION CHAR(16),IBUF(360)
      COMMON/MAIN/SUM(1920)
      COMMON/UNP/JWD(5),JPT(15)
      DATA CHAR/1H ,1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,
     *1HN,1HO/
      DATA IF/0/
      LINE = 0
      IF(IF)1,1,5
    1 LINE = 0
      IF = 1
      INDEX = 0
      NREC = 0
      SCALE = 1.414/2047.
      NOS1 = 0.
      ITIM =0
C ***** AT THE FIRST SEARCH ONE RECORD AT A TIME FOR THE 7 HIGH AMPS
      ISTRTS = 0
      ISTPS = 990
      GO TO 10
    5 ISTRTS = NOS1 + 1910
      IF(ISTRTS.LE.0) STOP 7
      ISTPS = NOS1 + 2400
      IF(ISTPS-2970)10,10,505
   10 CONTINUE
      NREC = NREC +1
C ***** THE IR DATA IS PACKED ON A TAPE WITH 990 (12-BIT WORDS)
C ***** PER RECORD
C ***** EACH SCAN LINE CONTAINS APPROXIMATELY 1980 WORDS
C ***** STORE 3 RECORDS AT A TIME TO FIND ONE COMPLETE LINE
      BUFFER IN(1,1) (IBUF(1),IBUF(250))
      IF(UNIT(1))15,11,12
   ** CONTINUE
      WRITE(6,2000) NREC
 2000 FORMAT(' EOF AT RECORD ',I10)
      STOP
   12 CONTINUE
      WRITE(6,3000) NREC
 3000 FORMAT(' PARITY ERROR ON RECORD ',I10)
   15 CONTINUE
      NWDS = LENGTH(1)
      DO 17 I=1,NWDS,5
      DO 15 J=1,5
```

```
      JWD(J) = IBUF(I+J-1)
   15 CONTINUE
      CALL UNPACK
C ***** UNPACK IS A COMPASS-CODED SUBROUTINE USING  UNP  FOR COMMUNICATION
      DO 16 J=1,15
      INDEX = INDEX +1
      DATA(INDEX)= SCALE*FLOAT(JPT(J))
   16 CONTINUE
   17 CONTINUE
      IF(INDEX-2970)10,18,18
   18 CONTINUE
C ***** FIND THE 7 HIGH AMPLITUDE PULSES
C **** THE 7 HIGH AMPS SHOULD BE WITHIN 45 WORDS AND SHOULD SHOW AN
C ****  INCREASE OF AT LEAST .15 FOR AT LEAST 6 PULSES
      I = ISTRTS+1
   20 CONTINUE
      I = I+1
      SAVS = DATA(I-1)+.15
      IF(I-ISTPS)25,25,301
   25 IF(SAVS-DATA(I))30,30,20
   30 KOUN = 1
      NOS2 = I
      DO 39 J=1,45
      K = I+J
      SAVS = DATA(K) + .15
      IF(SAVS-DATA(K+1))35,35,39
   35 KOUN = KOUN +1
   39 CONTINUE
      IF(KOUN-5)20,40,40
   40 CONTINUE
      ITIM = 0
      SUMSP = 0
C ***** FIND THE AVERAGE OF 45 SPACE VALUES
      DO 42 J=1,45
      K = NOS2 + J +35
      SUMSP = SUMSP + DATA(K)
   42 CONTINUE
      SUMSP = SUMSP  45.
   41 CONTINUE
C ***** FIND THE END OF THE DATA LINE
      ISTRTD = NOS2 +630
      ISTPD = NOS2+700
   50 I=ISTRTD -1
   55 I=I+1
      SAVD = DATA(I-1)
      SAVDS = SAVD+.15
      IF(I-ISTPD)60,60,302
   60 IF((SAVDS  )-DATA(I+1))65,65,55
   65 IF((SAVDS-.05)-DATA(I))75,75,70
   70 NOD2 = I+1
      GO TO 80
   75 NOD2 = I
   80 KOUN = 0
      DO 90 J=1,30
      IF(DATA(J+NOD2)-SAVDS  )85,90,90
   85 KOUN = KOUN +1
      IF(KOUN-5)90,90,55
   90 CONTINUE
      ITIM = 0
   91 CONTINUE
C ***** FIND THE START OF THE WEDGE
      ISTRTC = NOD2 + 195
      ISTPC = NOD2 + 255
      NOD2 = NOD2 -502
```

```
      I = ISTRTC
  500 I=I+1
      SAVC = DATA(I-1)+.2
      IF(I-ISTPC)501,501,303
  501 IF(SAVC-DATA(I))502,502,500
  502 NOC2 = I
      KOUN = 0
      DO 510 J=3,27
      IF(DATA(J+I)-SAVC)505,505,510
  505 KOUN = KOUN +1
  510 CONTINUE
      IF(KOUN-3)520,500,500
  520 CONTINUE
      ITIM = 0
      SUMHS = 0
C ***** FIND THE AVERAGE OF 200 HOUSING VALUES
      DO 98 J=1,200
      K = NOC2 + J + 220
      SUMHS = SUMHS + DATA(K)
   98 CONTINUE
      SUMHS = SUMHS/200
   99 CONTINUE
      EHS = 0.052593
      CON  = EHS/(SUMHS-SUMSP)
      II = 0
      DO 100 I=1,582
      NN = NOD2+ I-1
      E = CON*(DATA(NN)-SUMSP)
C ***** SUBROUTINE CHECKX DETERMINES THE TEMPERATURE IN KELVIN
      CALL CHECKX(E,SUM(I))
  100 CONTINUE
  105 CONTINUE
      N = NOS2 + 1909
      N = N/990
      NN = N*990
      IF(N)130,130,110
  110 CONTINUE
      IF(N.GT.3) STOP 5
      NNN = 2970-NN
C ***** SHIFT THE DATA IN THE STORED ARRAY TO MAKE ROOM FOR NEXT LINE
      DO 120 I=1,NNN
      DATA(I) = DATA(I+NN)
  120 CONTINUE
  130 CONTINUE
      NOS1 = NOS2-NN
      INDEX = NNN
      DO 180 I=1,582
      XMEAN = SUM(I)
      X = 271
      DO 160 J=1,16
      X = X+2
      IF(XMEAN- X)170,170,160
  160 CONTINUE
      J = 16
  170 CONTINUE
      SUM(I+669) = CHAR(J)
  180 CONTINUE
      DO 161 I=1,669
  161 SUM(I) = 1H
      DO 162 I=1252,1920
  162 SUM(I) = 1H
  200 RETURN
  210 STOP 6
  301 ICHEX = 1
```

```
C ***** COULD  NOT FIND THE T PULSE-HIGH AMPLITUDE IN THE RANGE FROM ISTRTS
C ***** TO ISTPS
      IF(ISTRTS.EQ.0.AND.ISTPS.EQ.990) GO TO 306
      ITIM = ITIM +1
      IF(NOS2.GT.ISTRTS.AND.NOS2.LT.ISTPS) GO TO 310
      NOS2 = NOS1 +1943
      GO TO 310
  302 ICHEX = 2
      ITIM = ITIM+1
C ***** COULD NOT FIND THE DATA END IN THE RANGE FROM ISTRTD TO ISTPD
      IF(NOD2.GE.ISTRTD.AND.NOD2.LT.ISTPD) GO TO 310
      NOD2 = NOS2 + 673
      GO TO 310
  303 ITIM = ITIM +1
C ***** COULD NOT FIND THE WEDGE IN THE RANGE FROM ISTRTC TO ISTPC
      ICHEX = 3
      IF(NOC2.GE.ISTRTC.AND.NOC2.LT.ISTPC) GO TO 310
      NOC2 = NOD2 + 224
      GO TO 310
  306 CONTINUE
C ***** DATA LINE DOES NOT START IN FIRST RECORD TRY NEXT RECORD
      DO 307 I=1,1980
      DATA(I) = DATA(I+990 )
  307 CONTINUE
      INDEX = 1980
      GO TO 10
  305 CONTINUE
C ***** THE RANGE IS OUT OF BOUNDS
      ISTPS = 2970
      ICHEX = 5
  310 CONTINUE
      WRITE(6,4000)NOS2,NOS1,NOD2,NOC2,ICHEX,IREC
 4000 FORMAT(8I10)
      IF(ITIM.GT.2) GO TO 410
  350 GO TO (41,91, 99,99,10),ICHEX
  410 CONTINUE
      END
```

```
      SUBROUTINE FINDSR
C ***** ************************************************************************
C ***** THIS ROUTINE FINDS ONE LINE OF IR DATA FOR MADDENS LOW RESOLUTION
C ***** IF          BEGINNING INDICATOR
C ***** ISTPC       STOP  POSITION TO SEARCH FOR THE WEDGE
C ***** ISTPD       STOP  POSITION TO SEARCH FOR THE END OF THE DATA
C ***** ISTPS       STOP  POSITION TO SEARCH FOR THE 7 HIGH AMPS
C ***** ISTRTC      START POSITION TO SEARCH FOR THE WEDGE
C ***** ISTRTD      START POSITION TO SEARCH FOR THE END OF THE DATA
C ***** ISTRTS      START POSITION TO SEARCH FOR THE 7 HIGH AMPS
C ***** ITIM        COUNTER  FOR MISSING PARTS OF A LINE
C ***** ITIME       COUNTER  FOR MISSING PARTS OF A LINE
C ***** LINE        LINE COUNT
C ***** NOC2        POSITION OF WEDGE START
C ***** NOD2        POSITION OF NEW DATA LINE
C ***** NOS1        BEGINNING OF PREVIOUS 7 PULSE- HIGH AMPLITUDE
C ***** NOS2        BEGINNING OF NEW      7 PULSE- HIGH AMPLITUDE
C ***** NREC        RECORD COUNT
C ***** SUMHS       AVERAGE 200 VALUES TO DETERMINE THE HOUSING VALUE
C ***** SUMSP       AVERAGE 45 VALUES TO DETERMINE THE SPACE VALUE
C ***** ************************************************************************
      DIMENSION DAT(16)
      DIMENSION CHAR(16),IBUF(360)
      COMMON DATA(2970)
      COMMON/MAIN/SUM(360)
      COMMON/UNP/JWD(3),JPT(15)
      DATA CHAR/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,
     *1HO,1HP/
      DATA IF/0/
      LINE = 0
      DO 6 I=1,360
      SUM(I) = 0.
    6 CONTINUE
      IF(IF)1,1,5
    1 LINE = 0
      IF = 1
      INDEX = 0
      NREC = 0
      SCALE = 1.414/2047.
      NOS1 = 0.
      ITIM =0
C ***** AT THE FIRST SEARCH ONE RECORD AT A TIME FOR THE 7 HIGH AMPS
      ISTRTS = 0
      ISTPS = 990
      GO TO 10
    5 ISTRTS = NOS1 + 1910
      IF(ISTRTS.LE.0) STOP 7
      ISTPS = NOS1 + 2400
      IF(ISTPS-2970)10,10,305
   10 CONTINUE
      NREC = NREC +1
C ***** THE IR DATA IS PACKED ON A TAPE WITH 990 (12-BIT WORDS)
C ***** PER RECORD
C ***** EACH SCAN LINE CONTAINS APPROXIMATELY 1980 WORDS
C ***** STORE 3 RECORDS AT A TIME TO FIND ONE COMPLETE LINE
      BUFFER IN(11,1) (IBUF(1),IBUF(250))
      IF(UNIT(11))3,11,12
   11 CONTINUE
      WRITE(6,2000) NREC
 2000 FORMAT(*   EOF AT RECORD  *,I10)
      STOP
   12 CONTINUE
      WRITE(6,3000) NREC
```

```
 3000 FORMAT('   PARITY ERROR ON RECORD ',I10)
   13 CONTINUE
      NWDS = LENGTH(1)
      DO  17 I=1,NWDS,3
      DO 15 J=1,3
      JWD(J) = IBUF(I+J-1)
   15 CONTINUE
      CALL UNPACK
C ***** UNPACK IS A COMPASS-CODED SUBROUTINE USING  UNP  FOR COMMUNICATION
      DO 16 J=1,15
      INDEX = INDEX +1
      DATA(INDEX)= SCALE*FLOAT(JPT(J))
   16 CONTINUE
   17 CONTINUE
      IF(INDEX-2970)10,18,18
   18 CONTINUE
      I = ISTRTS+1
   20 CONTINUE
C ***** FIND THE 7 HIGH AMPLITUDE PULSES
C **** THE 7 HIGH AMPS SHOULD BE WITHIN 45 WORDS AND SHOULD SHOW AN
C **** INCREASE OF AT LEAST .15 FOR AT LEAST 6 PULSES
      I = I+1
      SAVS = DATA(I-1)+.15
      IF(I-ISTPS)25,25,301
   25 IF(SAVS-DATA(I))30,30,20
   30 KOUN = 1
      NOS2 = I
      DO 39 J=1,45
      K = I+J
      SAVS = DATA(K) + .15
      IF(SAVS-DATA(K+1))35,35,39
   35 KOUN = KOUN +1
   39 CONTINUE
      IF(KOUN-5)20,40,40
   40 CONTINUE
      ITIM = 0
      SUMSP = 0
C ***** FIND THE AVERAGE OF 45 SPACE VALUES
      DO 42 J=1,45
      K = NOS2 + J +35
      SUMSP = SUMSP + DATA(K)
   42 CONTINUE
      SUMSP = SUMSP / 45.
   41 CONTINUE
C ***** FIND THE END OF THE DATA LINE
      ISTRTD = NOS2 +630
      ISTPD = NOS2+700
   50 I=ISTRTD -1
   55 I=I+1
      SAVD = DATA(I-1)
      SAVDS = SAVD+.15
      IF(I-ISTPD)60,60,302
   60 IF((SAVDS  )-DATA(I+1))65,65,55
   65 IF((SAVDS-.05)-DATA(I))75,75,70
   70 NOD2 = I+1
      GO TO 80
   75 NOD2 = I
   80 KOUI = 0
      DO 90 J=1,30
      IF(DATA(J+NOD2)-(SAVDS  ))85,90,90
   85 KOUI = KOUI +1
      IF(KOUI-5)90,90,55
   90 CONTINUE
      ITIM = 0
```

```
   91 CONTINUE
C ***** FIND THE START OF THE WEDGE
      ISTRTC = NOD2 + 195
      ISTPC = NOD2 + 255
      NOD2 = NOD2 -582
      I = ISTRTC
  500 I=I+1
      SAVC = DATA(I-1)+.2
      IF(I-ISTPC)501,501,503
  501 IF(SAVC-DATA(I))502,502,500
  502 NOC2 = I
      KOUN = 0
      DO 510 J=3,27
      IF(DATA(J+1)-SAVC)505,505,510
  505 KOUN = KOUN +1
  510 CONTINUE
      IF(KOUN-3)520,500,500
  520 CONTINUE
      ITIM = 0
      SUMHS = 0
C ***** FIND THE AVERAGE OF 200 HOUSING VALUES
      DO 98 J=1,200
      K = NOC2 + J + 220
      SUMHS = SUMHS + DATA(K)
   98 CONTINUE
      SUMHS = SUMHS/200
   99 CONTINUE
      EHS = 0.052503
      CON  = EHS/(SUMHS-SUMSP)
      II = 0
C ***** THERE IS A SCAN ANGLE OF 109 DEGREES
      DO 100 I=1,109,3
      DO 101 J=1,16
      II = II+1
      NN = NOD2+II-1
      E = CON*(DATA(NN)-SUMSP)
C ***** SUBROUTINE CHECKX DETERMINES THE TEMPERATURE IN KELVIN
      CALL CHECKX(E,DAT(J))
  101 CONTINUE
C ***** AVERAGE 5 AND 1/3 FOR EACH DATA POINT TO GET 109 VALUES
      SUM(I) =(3.*(DAT(1)+DAT(2)+DAT(3)+DAT(4)+DAT(5)) + DAT(6)) /6.
     *  + SUM(I)
      IF(I.EQ.109) GO TO 100
      SUM(I+1) = (3.*(DAT(7)+DAT(8)+DAT( 9)+DAT(10)) +2.*(DAT(6)+DAT(11)
     *))/6.
     *  + SUM(I+1)
      SUM(I+2) = (3.*(DAT(12)+DAT(13)+DAT(14)+DAT(15) +DAT(16)) +DAT(11)
     *)/6.
     *  + SUM(I+2)
  100 CONTINUE
  105 CONTINUE
      N = NOS2 + 1909
      N = N/990
      N = N*990
      IF(N)130,130,110
  110 CONTINUE
      IF(N.GT.3) STOP 3
      NNN = 2970+N.
C ***** SHIFT THE DATA IN THE STORED ARRAY TO MAKE ROOM FOR NEXT LINE
      DO 120 I=1,NNN
      DATA(I) = DATA(I+N.)
  120 CONTINUE
      INDEX = NNN.
  130 CONTINUE
```

```
      NOS1 = NOS2-NN
      ICHE = ICHE +1
      IF(ICHE .LT.8) GO TO 5
      ICHE = 0
      LINE = LINE +1
      DO 180 I=1,90
C ***** STORE EACH DATA LINE IN MATRIX AM
      AM(I+15,51-LINE) =(SUM(I) *8.)*100.
  180 CONTINUE
      DO 181 I=1,15
      AM(I,51-LINE) = AM(16,51-LINE)
      J = I+105
      AM(J,51-LINE) = AM(105,51-LINE)
  181 CONTINUE
      DO 161 I=1,360
  161 SUM(I) = 0.
  200 CONTINUE
      GO TO 5
  210 STOP 6
  301 ICHEX = 1
C ***** COULD NOT FIND SYNC PULSE
      IF(ISTRTS.EQ.0.AND.ISTPS.EQ.800) GO TO 306
      ITIM = ITIM +1
      IF(NOS2.GT.ISTRTS.AND.NOS2.LT.ISTPS) GO TO 310
      NOS2 = NOS1 +800
      GO TO 310
  304 ICHEX = 4
      GO TO 310
  305 CONTINUE
C ***** OUT OF RANGE OF STORED DATA
      ISTPS =2400
      ICHEX = 5
  306 CONTINUE
C ***** DATA LINE DOES NOT START IN FIRST RECORD TRY NEXT RECORD
      DO 307 I=1,1600
      DATA(I) = DATA(I+800 )
  307 CONTINUE
      INDEX = 1600
      GO TO 10
  310 CONTINUE
      WRITE(6,4000)NOS2,NOS1,NOO2,      ICHE,IREC
 4000 FORMAT(8I10)
      IF(ITIM- 3)350,350,210
  350 GO TO (45,10,400,99,10),ICHE
  410 CONTINUE
      IF(LINE.GE.50) GO TO 400
      LINE = LINE +1
      DO 420 I=1,120
      AM(I,51-LINE) = AM(I,52-LINE)
  420 CONTINUE
      GO TO 410
  400 CONTINUE
      END
```

```
      SUBROUTINE FINDSR
C ***** ***************************************************************
C ***** THIS ROUTINE FINDS ONE LINE OF APT DATA FOR MADDEN'S  LOW RESOLUTION
C ***** IF          BEGINNING INDICATOR
C ***** ISTPS       STOP  POSITION TO SEARCH FOR THE SYNC
C ***** ISTRTS      START POSITION TO SEARCH FOR THE SYNC
C ***** ITIM        COUNTER  FOR MISSING PARTS OF A LINE
C ***** LINE        LINE COUNT
C ***** NOD2        POSITION OF NEW DATA LINE
C ***** NOS1        BEGINNING OF PREVIOUS SYNC
C ***** NOS2        BEGINNING OF NEW SYNC
C ***** NREC        RECORD COUNT
C ***** TOP         HIGHEST SYNC PULSE FOR A LINE
C ***** ***************************************************************
      COMMON DATA(2415)
      DIMENSION DAT(25)
      DIMENSION CHAR(16),IBUF(360)
      COMMON/MAIN/SUM(360)
      COMMON/UNP/JWD(3),JPT(15)
      DATA CHAR/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,
     *1HO,1HP/
      DATA IF/0/
      LINE = 0
      DO 6 I=1,360
      SUM(I) = 0.
    6 CONTINUE
      IF(IF)1,1,5
    1 LINE = 0
      IF = 1
      INDEX = 0
      NREC = 0
      SCALE = 1.414/2047.
      NOS1 = 0.
      ITIM =0
C ***** AT THE FIRST SEARCH ONE RECORD AT A TIME FOR THE SYNC PULSE
      ISTRTS = 0
      ISTPS = 800
      GO TO 10
    5 ISTRTS = NOS1 + 750
      IF(ISTRTS.LE.0) STOP 7
      ISTPS = NOS1 + 850
      IF(ISTPS-2400)7,7,305
    7 CONTINUE
      IF(INDEX-2400)10,18,18
   10 CONTINUE
      NREC = NREC +1
C ***** THE APT DATA IS PACKED ON A TAPE WITH 800 (12-BIT WORDS)
C ***** PER RECORD
C ***** EACH SCAN LINE CONTAINS APPROXIMATELY 800  WORDS
C ***** STORE 3 RECORDS AT A TIME TO FIND ONE COMPLETE LINE
      BUFFER IN(1,1) (IBUF(1),IBUF(250))
      IF(UNIT(1))13,11,12
   11 CONTINUE
      WRITE(6,2000) NREC
 2000 FORMAT(' EOF AT RECORD ',I10)
      STOP
   12 CONTINUE
      WRITE(6,3000) NREC
 3000 FORMAT(' PARITY ERROR ON RECORD ',I10)
   13 CONTINUE
      NWDS = LENGTH(1)
      DO 17 I=1,NWDS,3
      DO 15 J=1,3
```

```
      JWD(J) = IBUF(I+J-1)
   15 CONTINUE
      CALL UNPACK
C ***** UNPACK IS A COMPASS-CODED SUBROUTINE USING 'UNP' FOR COMMUNICATION
      DO 16 J=1,15
      INDEX = INDEX +1
      DATA(INDEX)= SCALE*FLOAT(JPT(J))
   16 CONTINUE
   17 CONTINUE
      IF(INDEX-2400)10,18,18
   18 CONTINUE
      I = ISTRTS+1
   20 CONTINUE
C ***** FIND THE SYNC PULSE WHICH SHOULD LAST AT LEAST 25 POINTS
      I = I+1
      IF(I-ISTPS)25,25,301
   25 IF(DATA(I)-.50)20,30,30
   30 KOUN = 0
      NOS2 = I
      TOP = DATA(I)
      DO 40 J=1,40
      IF(DATA(I+J)-.50)35,41,41
   35 KOUN = KOUN +1
      IF(KOUN-5)40,40,20
   41 CONTINUE
      IF(DATA(I).GT.TOP) TOP = DATA(I)
   40 CONTINUE
      ITIM = 0
   43 CONTINUE
      NOD2 = NOS2 +48
      II = 0
   99 DO 100 I=1,90,3
C ***** AVERAGE EVERY 8 AND 1/3 POINTS AND STORE IN SUM
      DO 101 J=1,25
      II= II+1
      NN = NOD2+II-1
      DAT(J) = DATA(NN)
  101 CONTINUE
      SUM(I) =(3.*(DAT(1)+DAT(2)+DAT(3)+DAT(4)+DAT(5)+DAT(6)+DAT(7)
     * + DAT(8))+DAT(9))/25.  +SUM(I)
      SUM(I+1) = (3.*(DAT(10)+DAT(11)+DAT(12)+DAT(13)+DAT(14)+DAT(15)
     *+DAT(16)) +2.*(DAT(9)+DAT(17)))/25.   +SUM(I+1)
      SUM(I+2) = (3.*(DAT(18)+DAT(19)+DAT(20)+DAT(21)+DAT(22)+DAT(23)
     *+DAT(24)+DAT(25)) +DAT(17))/25.  +SUM(I+2)
  100 CONTINUE
      N = NOS2+750
      N = N/800
      NN = N*800
      IF(N)130,130,110
  110 CONTINUE
      IF(N.GT.3) STOP 3
C ***** SHIFT THE DATA IN THE STORED ARRAY TO MAKE ROOM FOR NEXT LINE
      NNN = 2400-NN
      DO 120 I=1,NNN
      DATA(I) = DATA(I+NN)
  120 CONTINUE
      INDEX = NNN
  130 CONTINUE
      NOS1 = NOS2-NN
      LINE = LINE +1
      IF(LINE.LT.8) GO TO 5
      XINT = TOP/15
      DO 180 I=1,90
      XMEAN = SUM(I)/8
```

```
      X =-XINT
      DO 160 J=1,16
      X = X+XINT
      IF(XMEAN- X)170,170,160
  160 CONTINUE
      J = 16
  170 CONTINUE
      SUM(I+135) = CHAR(J)
  180 CONTINUE
      DO 161 I=1,135
  161 SUM(I) = 1H
      DO 162 I=226,360
  162 SUM(I) = 1H
  200 RETURN
  210 CONTINUE
      WRITE(6,5000) (DATA(I),I=1,2400)
 5000 FORMAT(1X,15F6.3)
      STOP 16
  301 ICHEX = 1
C ***** COULD NOT FIND SYNC PULSE
      IF(ISTRTS.EQ.0.AND.ISTPS.EQ.800) GO TO 306
      ITIM = ITIM +1
      IF(NOS2.GT.ISTRTS.AND.NOS2.LT.ISTPS) GO TO 310
      NOS2 = NOS1 +800
      GO TO 310
  304 ICHEX = 4
      GO TO 310
 4000 FORMAT(8I10)
  305 CONTINUE
C ***** OUT OF RANGE OF STORED DATA
      ISTPS =2400
      ICHEX = 5
  306 CONTINUE
C ***** DATA LINE DOES NOT START IN FIRST RECORD TRY NEXT RECORD
      DO 307 I=1,1600
      DATA(I) = DATA(I+800 )
  307 CONTINUE
      INDEX = 1600
      GO TO 10
  310 CONTINUE
      IF(ITIM- 5)350,350,210
  350 GO TO (45,10,400,99,10),ICHEX
  400 CONTINUE
      END
```

```
          IDENT    UNPACK
*  ROUTINE TO TAKE 3 CM WORDS (1ST 3 IN /UNP/) AND MAKE 15 CDC INTEGERS
*    (NEXT 15 IN /UNP/) FROM THEM WITH CONVERSION FROM 2-S COMPLEMENT
*    REPRESENTATION OF NEGATIVE NUMBERS TO 1-S COMPLEMENT.
*    ROUTINE INSPIRED BY NEED TO UNPACK DATA WRITTEN ON A/D CONVERSION
          ENTRY    UNPACK
UNPACK    BSSZ     1              RESERVED FOR RETURN LINK STRRAGE
          SB1      1              CONSTANT
          SX3      B1             CONSTANT
          SB2      3              INTIALIZE TO UNPACK A GROUP OF 3 WORDS
          SB3      15             INITIALIZE TO STORE 15 WORDS
          SB6      B2             INITIALIZE FOR COUNTING REFERENCE
          SB7      B3             INITIALIZE FOR COUNTING REFERENCE
          MX2      12             HIGH ORDER 12 BIT MASK
OUTER     SB4      5              INTIIALIZE TO UNPACK 5 BYTES PER WORD
          SB5      B6-B2          GET JWD ARRAY ONE BY ONE
          SA1      JWD+B5         IN ORDER AND PUT IN X1
INNER     BX6      X1*X2          ISOLATE HIGH 12 BITS
          LX1      12             GET NEXT BYTE IN READY POSITION
          NG       X6,NEG         GO TO NEG IF A/D RESULT WAS NEGATIVE
          LX6      12             PUT POSITIVE A/D RESULT IN LOW BITS
          JP       STORE            AND GO STORE IT
NEG       AX6      48             MAKE 2S COMPLEMENT INTEGER
          IX6      X3-X6          ONES COMPLEMENT = -(1-N) WHERE N IS
          BX6      -X6              IN TWOE COMPLEMENT
STORE     SB5      B7-B5          STORE INTO JPT IN ORDER
          SA6      JPT+B5           FROM X6
          SB3      B5-B1          INDEX JPT ARRAY
          SB4      B4-B1          COUNT BYTES PROCESSED
          NE       B4,INNER       LOOP IF NOT 5
          SB2      B2-B1
          NE       B2,OUTER
          JP       UNPACK         COUNT WORDS UNPACKED
JWD       USE      /UNP           LOOP IF NOT 3
JWD       BSS      3              DONE, EXIT
JPT       BSS      15             DEFINE
          USE      *                COMMON
          END                         BLOCK UNP
```

```
      SUBROUTINE CHECKX(D,T)
C ***************************************************************************
C *****  THIS ROUTINE DETERMINES TEMPERATURE VALUES IN KELVIN OF IR
C *****  DATA WHERE,
C *****   TEMPERATURE = F(E)
C *****   AND E = 0.052593*(IR DATA - AVERAGE SPACE VALUE PER LINE)
C *****           ------------------------------------------
C *****           (AVERAGE HOUSING VALUE - AVERAGE SPACE VALUE PER LINE)
C ***************************************************************************
      DIMENSION TAB(120)
      DATA(TAB(I),I=1,70)/
     10.0078703,0.0081201,0.0083753,0.0086359,0.0089020,0.0091755,
     20.0094507,0.0097334,0.010022,0.010316,0.010616,0.010921,0.011233,
     30.011550,0.011874,0.012203,0.012538,0.012879,0.013227,0.013580,
     40.01394,0.014306,0.014678,0.015056,0.015441,0.015832,0.016229,
     50.016633,0.017043,0.017460,0.017883,0.018313,0.018749,0.019192,
     60.019641,0.020097,0.020560,0.021029,0.021505,0.021988,0.022478,
     70.022974,0.023478,0.023988,0.024505,0.025028,0.025559,0.026097,
     80.026641,0.027193,0.027751,0.028316,0.028889,0.029468,0.030055,
     90.030648,0.031249,0.031856,0.032471,0.033093,0.033722,0.034358,
     A0.035001,0.035651,0.036309,0.036973,0.037645,0.038324,0.039010,
     *0.039703/
      DATA(TAB(I),I=71,120)/
     B0.040404,0.041112,0.041827,0.042549,0.043278,0.044014,0.044758,
     C0.045509,0.046267,0.047033,0.047805,0.048585,0.049372,0.050167,
     D0.050968,0.051777,0.052593,0.053416,0.054247,0.055084,0.055929,
     *0.056781,
     E0.057641,0.058507,0.059381,0.060262,0.061150,0.062045,0.062948,
     F0.063857,0.064774,0.065698,0.066629,0.067568,0.068513,0.069466,
     G0.070425,0.071392,0.072366,0.073347,0.074336,0.075331,0.076333,
     H0.077343,0.078359,0.079382,0.080413,0.081451,0.082495,0.083547
      DO 10 I=2,120
      IF(D.LT.TAB(I)) GO TO 20
   10 CONTINUE
   20 CONTINUE
      DIF1 = ABS(D-TAB(I))
      DIF2 = ABS(D-TAB(I-1))
      T = I-1+200.16
      IF(DIF1.GT.DIF2) T = I-2+200.16
      RETURN
      END
```

```
      PROGRAM DUMP
     *(INPUT=202B,TAPE5=INPUT,OUTPUT=202B,TAPE6=OUTPUT,TAPE1)
C ***** ************************************************************************
C ***** THIS PROGRAM PRINTS OUT 20 RECORDS FROM THE SCANNER DATA
C ***** DIGITIZED BY THE DATA CONVERSION SERVICE VIA THE SYNC PULSE
C ***** PROGRAM.  THE DATA CAN BE APT OR IR DATA.  THE ROUTINE IS USED
C ***** TO FIND THE RECORD NUMBER OF THE FIRST DATA LINE, THE POSITION
C ***** OF THE SYNCS, THE NUMBER OF DATA VALUES PER LINE, ETC.  ONCE
C ***** THE RECORD NUMBER OF THE FIRST DATA LINE IS KNOWN, THEN THE
C ***** DATA TAPE CAN BE POSITIONED AT THIS POINT TO START THE PROGRAM.
C ***** ************************************************************************
      DIMENSION IBUF(250)
      COMMON DATA(2970),INDEX,NREC
      COMMON/UNP/JWD(3),JPT(15)
      NREC = 0
      NS = 0
      SCALE = 1.414/2047.
   20 CONTINUE
      NREC = NREC +1
C ***** DUMP ONLY 20 RECORDS
      IF(NREC.GT.20) STOP
      BUFFER IN(1,1) (IBUF(1),IBUF(250))
      IF(UNIT(1))50,30,40
   30 WRITE(6,2000) NREC
 2000 FORMAT('   EOF AT RECORD  ',I10)
      STOP
   40 WRITE(6,4000) NREC
 4000 FORMAT('   PARITY ERROR ON RECORD ',I10)
   50 CONTINUE
      NWDS = LENGTH(1)
      WRITE(6,5000) NREC                                                    S
 5000 FORMAT(1H0,' RECORD NUMBER',I10)
      INDEX = 0
      DO 100 I=1,NWDS,3
      DO 70 J=1,3
      JWD(J) = IBUF(I+J-1)
   70 CONTINUE
      CALL UNPACK
C ***** UNPACK IS A COMPASS-CODED SUBROUTINE USING  UNP  FOR COMMUNICATION
      DO 65 J=1,15
      INDEX = INDEX +1
      DATA(INDEX)= SCALE*FLOAT(JPT(J))
   65 CONTINUE
      K = 5*(NWDS-I+1)
      L = INDEX-14
      K = MIN0(K,15)
      K = L+K-1
      WRITE(6,7000) (DATA (J),J=L,K)                                        S
 7000 FORMAT(1X,15(F6.3,1X))
  100 CONTINUE
      GO TO 20
      END
```