#712

Dynamic Explorer 2

ind and Temperature Spectrometer (WATS)

Neutral Velocity and Temperature

81-070B-04C

Dynamic Explorer 2

Wind and Temperature Spectrometer (WATS)

Neutral Velocity and Temperature

81-070B-04C

This data set consists of three magnetic tapes. The D tapes are 9 track, 6250 BPI and the C tapes are 3480 cartriges. These data are written in Vax backup format with multiple save_sets on each tape. The directories are listed for each tape on the following pages. The D and C numbers along with the time spans follow:

D#	C#	Time Span
D-86234	C-29389	08/08/81 - 10/21/81
D-86235	C-29390	02/03/82 - 09/14/82
D-86236	C-29391	09/14/82 - 02/18/83

81-070B-04C
DSC#712

CCSDYDNM000200NSSDjjjjSMRK0001

TYPE_OF_FILE_NAME: WATS TEMPERATURE AND WIND

FILE_ATTRIBUTES: Sequential, unformatted, two record types, fixed lengt records, blocked with blocksize 2024 bytes. Placed in 3 tapes labeled WATSn, in backup format, in backup files named WATSn.BCK where n is 1,2, or 3 depending upon the tape number.

TYPE_OF_FILE_DESCRIPTION:

Each file corresponds to an original Sigma 9 MAF file. Each logical record of the file has ambient temperature and winds obtained in a two second slot. File names are of the form D[orbit number]N, where: the letter D is at the beginning of all files, the orbit number contains leading zeros if needed to force it to be 4 digits long, and N is a sequence number within a given orbit from 0 to 9 to ensure unique file names.

FILE_STRUCTURE:

The files are written as VAX unformatted sequential with logical records of equal length.

RECORD_TYPE_NAMES: Horizontal, Vertical -- Only one of the two baffles may be operating at one time, and the one operating in a given 2 second measurement slot determines the record type.

RECORD_TYPES:

RECORD_TYPE NAME: Horizontal

FIXED_RECORD_TYPE_LENGTH: 44 bytes 46

FORMAT_OF_THE_LOGICAL_RECORD:

Not scanned: (More pages from tape)

WORD NUMBER	FIELD NAME	TYPE I/R	DESCRIPTION
1	DATE	VI4	Date in YYDDD, YY = last 2 digits of the year, DDD=day number (Jan 1 = 1)
2	TIME	VI4	Universal time of day in milliseconds
3	MODE	VI4	Values 3 or 4 indicates horizontal
4	OUTIN	VI4	1 = baffle going out and 0 = going in (This field is used by the correction program to correct systematic wind errorS that are a function of the baffle direction.)
5	MASS	VI4	Mass in AMU, usually 28 or 32, (Mass 32 measured indicates ambient

WATS 3 [illegible]

81-070B-04C

DSC #712

(not scanned - (more pages from tape 5))

CCSDXZLM0001SMARK001CCSDXVNM0002SMRK0001

LOG_VOL_IDENT: USANASANSSDDEB4_0001

LOG_VOL_CLOSING_DATE: YYYY-MM-DD

LOG_VOL_FILE_STRUCTURE: FILES-11

TAPE_DENSITY=6250 BPI

TAPE_TRACKS=9

TAPE_LENGTH=2400 INCHES

COMPUTER_MFGR: DIGITAL EQUIPMENT CORPORATION

OPERATING_SYSTEM: VAX/VMS 5.4

COMPUTER_SYSTEM: MICRO VAX II

TECHNICAL_CONTACT: DR. L. E. WHARTON

CODE 914

NASA/GSFC

GREENBELT, MD 20771

PHONE: 301-286-3486

SPAN: DEIO::WHARTON

PREV_VOL_IDENT: NONE

CCSDXVNM0002EMRK0001CCSDYDNM0002SMRK0003

DATA_SET_NAME: NEUTRAL VELOCITY AND TEMPERATURE

DATA_SOURCES: DYNAMICS EXPLORER B, WIND AND TEMPERATURE SPECTROMETER (WATS)

INVESTIGATOR_CONTACT: MR. N. W. SPENCER

CODE 910

NASA/GSFC

GREENBELT, MD 20771

PHONE: 301-286-5001

SPAN: PACF::SPENCER

SOURCE_CHARACTERISTICS:

A. DESCRIPTION_OF_SPACECRAFT:

The Dynamics Explorer 2 spacecraft was one of two satellites launched for the

WATS1
D-86234

(more pages from tapes)

```
Listing of save set(s)

Save set:            FORMAT.BCK
Written by:          MANAGER
UIC:                 [000001,000004]
Date:                14-OCT-1992 17:26:36.93
Command:             BACKUP/REW/INTER/LABEL=WATS1/VERIFY DE$DATA2:
                     [WHARTON.WATS]FORMAT.SFD; MKA500:FORMAT.BCK;
Operating system:    VAX/VMS version V5.5
BACKUP version:      V5.5
CPU ID register:     12000003
Node name:           _MITE::
Written on:          _MITE$MKA500:
Block size:          8192
Group size:          10
Buffer count:        600

[WHARTON.WATS]FORMAT.SFD;3                    12  26-JUN-1992 11:02

Total of 1 file, 12 blocks
End of save set

Save set:            VOLDESC.BCK
Written by:          MANAGER
UIC:                 [000001,000004]
Date:                14-OCT-1992 17:27:03.65
Command:             BACKUP/INTER/LABEL=WATS1/VERIFY DE$DATA2:
                     [WHARTON.WATS]VOLDESC.SFD; MKA500:VOLDESC.BCK;
Operating system:    VAX/VMS version V5.5
BACKUP version:      V5.5
CPU ID register:     12000003
Node name:           _MITE::
Written on:          _MITE$MKA500:
Block size:          8192
Group size:          10
Buffer count:        600

[WHARTON.WATS]VOLDESC.SFD;3                   35  26-JUN-1992 11:02

Total of 1 file, 35 blocks
End of save set

Save set:            WATSCOR.BCK
Written by:          MANAGER
UIC:                 [000001,000004]
Date:                14-OCT-1992 17:27:34.42
Command:             BACKUP/INTER/LABEL=WATS1/VERIFY DE$DATA2:
                     [WHARTON.WATS]WATSCOR.FOR; MKA500:WATSCOR.BCK;
Operating system:    VAX/VMS version V5.5
BACKUP version:      V5.5
CPU ID register:     12000003
Node name:           _MITE::
Written on:          _MITE$MKA500:
Block size:          8192
Group size:          10
Buffer count:        600

[WHARTON.WATS]WATSCOR.FOR;9                   19  12-JUN-1992 10:03

Total of 1 file, 19 blocks
```

Sample output of WHTSKEHD.FOR

81220	57240718	3	1	28	0.19E+08	1054.2	188.6	4BA0363	92BD0203	-121.2
81220	57248718	3	1	28	0.18E+08	1084.1	179.9	4790337	92BD0203	-126.0
81220	57256719	3	1	28	0.17E+08	1085.4	174.4	4420309	92BD0203	-127.4
81220	57264719	3	1	28	0.16E+08	1083.1	177.6	40702DA	92BD0203	-120.0
81220	57272719	3	1	28	0.15E+08	1127.3	170.3	3D502BA	92BD0203	-123.2
81220	57280719	3	1	28	0.15E+08	1154.2	183.0	3A50295	92BD0203	-106.3
81220	57288719	3	1	28	0.13E+08	1049.5	170.6	372025B	92BD0203	-114.6
81220	57304719	3	1	28	0.12E+08	1098.9	157.0	32E0226	92BD0203	-119.6
81220	57306719	5	1	28	0.12E+08	1038.1	201.9	3320231	65900305	54.7
81220	57312719	3	1	28	0.12E+08	1131.4	158.3	31B021D	92BD0203	-114.0
81220	57314719	5	1	28	0.12E+08	1058.0	206.8	31F021D	65900305	52.9
81220	57320719	3	1	28	0.12E+08	1064.0	163.9	30A020C	92BD0203	-104.1
81220	57322719	5	1	28	0.12E+08	1046.4	181.8	30D020F	658F0305	80.6
81220	57328719	3	1	28	0.11E+08	1061.4	155.2	2F001F0	92BD0203	-108.5
81220	57330719	5	1	28	0.11E+08	1034.8	185.4	2F90200	658F0305	80.2
81220	57336719	3	1	28	0.11E+08	1150.6	144.4	2DB01EE	92BD0203	-114.9
81220	57338719	5	1	28	0.11E+08	1102.7	173.6	2DD01F2	658F0305	95.8
81220	57344719	3	1	28	0.10E+08	1083.4	136.9	2BB01CF	92BD0203	-118.0
81220	57346720	5	1	28	0.10E+08	1084.1	181.3	2BC01D2	658F0305	90.7
81220	57352720	3	1	28	0.97E+07	1031.7	113.4	29D01B2	92BD0203	-137.0
81220	57354720	5	1	28	0.10E+08	1121.4	198.8	2A301C4	658F0305	76.3
81220	57360720	3	1	28	0.90E+07	989.2	113.6	2800193	92BD0203	-132.4
81220	57362720	5	1	28	0.94E+07	1057.7	194.3	28601A8	658F0305	83.0
81220	57368720	3	1	28	0.96E+07	1035.8	117.3	2680184	92BD0203	-124.1
81220	57370720	5	1	28	0.91E+07	1100.9	203.8	26E019A	658F0305	75.8
81220	57376720	3	1	28	0.84E+07	1040.7	112.9	253017B	92BD0203	-124.0
81220	57378720	5	1	28	0.86E+07	1114.7	205.9	2540181	658F0305	76.4
81220	57384720	3	1	28	0.81E+07	1059.6	126.2	23D016B	93BD0203	-106.2
81220	57386720	5	1	28	0.84E+07	1121.7	210.9	2400179	658F0305	74.2
81220	57392720	3	1	28	0.76E+07	1016.6	127.2	2280155	93BD0203	-100.7
81220	57394720	5	1	28	0.79E+07	1066.3	207.9	22D0164	658F0305	79.8
81220	57400720	3	1	28	0.75E+07	1107.4	118.9	2170151	93BD0203	-104.4
81220	57402720	5	1	28	0.76E+07	1097.8	223.8	21C0156	658F0305	66.5
81220	57408720	3	1	28	0.69E+07	1025.8	125.1	2020137	93BD0203	-93.5
81220	57410720	5	1	28	0.73E+07	1054.9	218.2	2080148	658F0305	75.4
81220	57416720	3	1	28	0.67E+07	1042.5	114.1	1F2012C	93BD0203	-99.8
81220	57418720	5	1	28	0.70E+07	1050.1	226.3	1F80139	658F0305	69.7
81220	57424720	3	1	28	0.68E+07	1173.5	104.8	1E50130	93BD0203	-104.4
81220	57426720	5	1	28	0.69E+07	1069.1	221.2	1E90135	658F0305	76.4
81220	57432720	3	1	28	0.65E+07	1140.8	101.7	1D70121	93BD0203	-102.8

```
C*********************************************************************
C*                                                                   *
C*   PROGRAM RDIR.FOR                                                *
C*   FOR ANY QUESTIONS CONTACT:                                      *
C*   LAWRENCE E. WHARTON (301) 286-3486  SPAN PACF::WHARTON          *
C*   NASA/GODDARD SPACE FLIGHT CENTER CODE 914                       *
C*   GREENBELT MD 20771                                              *
C*                                                                   *
C*   INPUT FILE: DIR.DAT WHICH CONTAINS A LIST OF ORBIT NUMBERS,     *
C*               BEGINNING AND END TIME AND DATE, LATITUDE,          *
C*               LONGITUDE,AND LOCAL TIME.                           *
C*   OUTPUT FILE:OUTPUT FILE NAME INPUT BY USER AND CONTAINS:        *
C*   ORBIT NUMBER, START DATE, START TIME(S), END DATE, END TIME(S)  *
C*   ONE RECORD FOR EACH SEGMENT OF DATA                             *
C*                                                                   *
C*********************************************************************
      INTEGER*4 SD,ST,ED,ET,SD2,ST2,ED2,ET2,DT,ENO,SNO,SEC1,SEC2
      CHARACTER*30 OFN
90    FORMAT(I6,2(I6,I9))
98    FORMAT(I6,I6,I9,21X,I6,I9)
      OPEN(1,FILE='DIR.DAT',FORM='FORMATTED',STATUS='OLD')
      TYPE*,'ENTER OUTPUT FILE NAME:(IF NONE THEN <RET>)'
      READ(5,80)OFN
80    FORMAT(A20)
      IF(OFN.NE.' ')OPEN(2,FILE=OFN,FORM='FORMATTED',STATUS='NEW')
7     TYPE*,'ENTER :0=STOP,1=ORBIT NO,2=START/END
     1 ORBIT,3=TIME/DATE RANGE'
      READ(5,*)DT
      IF (DT.EQ.0) GOTO 10
      REWIND(1)
      IF (DT.EQ.1) THEN
         TYPE*,'ENTER ORBIT NUMBER'
         READ(5,*)NO
          IS=1
      ENDIF
      IF (DT.EQ.2) THEN
         TYPE*,'ENTER START AND END ORBIT'
         READ(5,*)SNO,ENO
         IS=2
      ENDIF
      IF (DT.EQ.3) THEN
         TYPE*,'ENTER START DATE AND TIME(SEC) AND
     1   END DATE AND TIME(SEC)'
         READ(5,*)SD,ST,ED,ET
         SD=(SD-81200)*86400+ST
         ED=(ED-81200)*86400+ET
         IS=3
      ENDIF
      GOTO (3,4,5),IS
3     READ (1,98,ERR=7) NO2,SD2,ST2,ED2,ET2
      L = NO2/10
      IF(L.LT.NO) GOTO 3
      IF(L.GT.NO) GOTO 7
      WRITE(*,90) NO2,SD2,ST2,ED2,ET2
      IF(OFN.NE.' ')WRITE(2,90) NO2,SD2,ST2,ED2,ET2
      GOTO 3
4     READ(1,98,ERR=7) NO2,SD2,ST2,ED2,ET2
      L = NO2/10
      IF(L.LT.SNO) GOTO 4
      IF(L.GT.ENO) GOTO 7
```

```
      WRITE(*,90) NO2,SD2,ST2,ED2,ET2
      IF(OFN.NE.' ')WRITE(2,90)NO2,SD2,ST2,ED2,ET2
      GOTO 4
5     READ(1,98,ERR=7) NO2,SD2,ST2,ED2,ET2
      SEC1=(SD2-81200)*86400 + ST2/1000
      SEC2=(ED2-81200)*86400 + ET2/1000
      IF (SEC2.GT.SD.AND.SEC1.LT.ED)THEN
         WRITE(*,90) NO2,SD2,ST2,ED2,ET2
         IF(OFN.NE.' ')WRITE(2,90)NO2,SD2,ST2,ED2,ET2
      ENDIF
      IF (SEC1.GT.ED) GOTO 7
      GOTO 5
10    CLOSE(1)
      CLOSE(2)
      STOP
      END
```

```
      PROGRAM WATSREAD
      CHARACTER DFN*40,DFN2*22
      DIMENSION BUFF(11),IBUFF(11)
      DATA BUFF(11)/99999./
      EQUIVALENCE (BUFF,IBUFF)
      PRINT*,'ENTER ORBIT OF DATA TO BE READ'
      READ (5,10) DFN
10    FORMAT (A40)
      PRINT*,'ENTER NAME OF OUTPUT FILE'
      READ (5,11) DFN2
11    FORMAT (A22)
      OPEN (UNIT=20,FILE=DFN,STATUS='OLD',FORM='UNFORMATTED')
      OPEN (UNIT=21,FILE=DFN2,STATUS='NEW')
30    READ (20,END=99) (BUFF(J),J=1,11)
      WRITE (21,31) (IBUFF(J),J=1,5),(BUFF(J),J=6,8),IBUFF(9),
     $                IBUFF(10),BUFF(11)
31    FORMAT (1X,I5,I9,1X,I1,1X,I1,1X,I2,1X,E10.2,1X,F8.1,1X,F8.1,1X,
     $         Z8,1X,Z8,F10.1)
      GOTO 30
99    STOP
      END
```

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
c *                                                                    *
c *                                                                    *
c * Program WATSCOR.FOR                                                *
c *                                                                    *
c * Purpose:                                                           *
c *         This program is to make the following correction on        *
c *         Wats data:                                                 *
c *         a. Let temperature = 9999 if it is from the slot           *
c *            following RPA mode.                                     *
c *         b. Increase horizontal baffle temperature by 1%            *
c *            and vertical baffle temperature by 5%                   *
c *         c. Decrease oxygen temperature by 3%                       *
c *         d. Average out the sweep assymmetry for 0 winds            *
c *                                                                    *
c *                                                                    *
c *          It will rewrite the correct data into the data file       *
c *         called 'Dxxxxx.dat'. A formatted file containing the       *
c *         the date time temperature horizontal and vertical wind     *
c *         is written into file 'Txxxxx.dat'.  The format is          *
c *         1xi6,i10,3f10.2 and no velocities are indicated by 0.       *
c *                                                                    *
c *                                                                    *
c *                                                                    *
c *   Using procedure:                                                 *
c *                     FORT WATSCOR                                   *
c *                     LINK WATSCOR
c *                     RUN WATSCOR                                    *
c *                    >ENTER ALLOCATED DEV(with []):                  *
c *                     DEVICE:[ID] FOR ANOTHER DEVICE                 *
c *                     [ID] FOR ANOTHER ACCOUNT                       *
c *                     RETURN FOR THE SAME ACCOUNT                    *
c *                    >ENTER PREFIX FILE NAME(WITH *)                 *
c *                     C*.DAT                                         *
c *                                                                    *
c * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION ARY(1000),B(10),E(10)
      INTEGER B,E
      CHARACTER*40 LOGDEVA,ARY
      CHARACTER*10 FNA
      TYPE*,'ENTER ALLOCATED DEV:(WITH ''[]'' RETURN FOR SAME ACCOUNT)'
      READ(5,88) LOGDEVA
      TYPE*,'ENTER FILE NAME(WITH *, EXAMPLE C*.DAT)'
      READ(5,89)FNA
      CALL MAKEARRAY(LOGDEVA,FNA,ARY,I)
      CALL CORRECT(ARY,I)
   88 FORMAT(A17)
   89 FORMAT(A10)
      STOP
      END
C*************************************************************
      SUBROUTINE MAKEARRAY (LOGDEVA,FA,ARY,I)
      DIMENSION ARY(1000)
      INTEGER*4 CONTEXTA,CONTEXT,ON
      CHARACTER LOGDEVA*20,ARY*40,FINA*40,FOUTA*40,FNA*40,AON*5
     &          ,C1,C2,C3,FA*10,CD
      FINA=LOGDEVA//FA
      CALL LIB$FIND_FILE (FINA,FOUTA,CONTEXT,,)
      C1=';'
      CD='.DAT'
```

```
      C2=']'
      C3=' '
      K1=INDEX(FOUTA,C2)
      K3=INDEX(FA,C3)
      I=0
      FNA=FOUTA(1:K1)//FA(1:K3-1)//';'
1     CALL LIB$FIND_FILE (FINA,FOUTA,CONTEXTA,,)
      K=INDEX(FOUTA,CD)-5
      AON=FOUTA(K:K+4)
      READ(AON,100,ERR=2) ON
100   FORMAT(I5)
         IF (FOUTA.NE.FNA) THEN
           I=I+1
           ARY(I)=FOUTA
           GOTO 1
         ENDIF
2     RETURN
      END


C*****************************************************************
      SUBROUTINE CORRECT(ARY,I)
      DIMENSION ARY(1000)
      DIMENSION IDATA(11,1800)
      CHARACTER ARY*40,C*4,FN*40
      REAL DATA(11,1800)
      EQUIVALENCE(IDATA,DATA)
      C='.DAT'
      IM='000000FF'X
      DO 1,J=1,I
      K=INDEX(ARY(J),C)
        OPEN(1,FILE=ARY(J),STATUS='OLD',ERR=1,FORM='UNFORMATTED')
        FN='D'//ARY(J)(K-5:K+3)
        OPEN(2,FILE=FN,STATUS='NEW',FORM='UNFORMATTED')
        FN='T'//ARY(J)(K-5:K+3)
        OPEN(3,FILE=FN,STATUS='NEW',FORM='FORMATTED')
        II = 1
  3     READ(1,END=2)(DATA(K,II),K=1,11)
        II = II + 1
        LL=L
        L=IAND(IM,J)
        IF (L.EQ.4) THEN
        IF (LL.NE.3) DATA(7,II)=9999.
        ENDIF
        IF(DATA(7,II).NE.9999.)THEN
        IF(IDATA(3,II).EQ.3.OR.IDATA(3,II).EQ.4)
     *     DATA(7,II)= DATA(7,II) * 1.01
        IF(IDATA(3,II).EQ.5.OR.IDATA(3,II).EQ.6)
     *     DATA(7,II)= DATA(7,II) * 1.05
        IF(IDATA(5,II).EQ.32) DATA(7,II) = DATA(7,II) * 0.97
        ENDIF
        GOTO 3
  2     CLOSE(1)
        II=II-1
        CALL VELCOR(DATA,IDATA,II)
        DO 4 K = 1,II
         WRITE(2) (DATA(KK,K),KK=1,11)
          IF (IDATA(3,K).EQ.3.OR.IDATA(3,K).EQ.4) THEN
            WRITE(3,10) IDATA(1,K),IDATA(2,K),DATA(7,K),0.,DATA(11,K)
          ELSE
```

```
            WRITE(3,10) IDATA(1,K),IDATA(2,K),DATA(7,K),DATA(11,K),0.
          ENDIF
 4      CONTINUE
      CLOSE(2)
        CLOSE(3)
 1    CONTINUE
10    FORMAT(1X,I6,I10,3F10.2)
      RETURN
      END

C*********************************************************************
      SUBROUTINE VELCOR(DATA,IDATA,II)
      INTEGER IDATA(11,1800)
      REAL DATA(11,1800),SV(1800)
      REAL AVER(3,1800),MV(1800)
      DATA IS/8/
      J = IS/2
      JT=IS*1000+1
      DO 10 I = 1,II
        IF (IDATA(3,I).EQ.3.OR.IDATA(3,I).EQ.4) THEN
          INDEX1 = 1
          MV(I)=1
        ELSEIF (IDATA(3,I).EQ.5.OR.IDATA(3,I).EQ.6) THEN
          INDEX1 = 2
          MV(I)=2
        ELSE
          MV(I)=0
          GOTO 10
        ENDIF
        AVER(1,I)=99999.
        AVER(2,I)=99999.
        AVER(3,I)=99999.
        SV(I)=99999.
        IT=IDATA(2,I)
        S1 = 0
        S2 = 0
        S3 = 0
        N1 = 0
        N2 = 0
        N3 = 0
        INDEX2 = 0
C AVERAGING THE VELOCITY
        ILOW=MAX(1,I-J)
        IHIGH=MIN(II,I+J)
        DO 20 K = ILOW,IHIGH
        IF(IABS(IT-IDATA(2,K)).GT.JT)GO TO 20
          IF (IDATA(3,K).EQ.3.OR.IDATA(3,K).EQ.4) THEN
            INDEX2 = 1
          ELSEIF (IDATA(3,K).EQ.5.OR.IDATA(3,K).EQ.6) THEN
            INDEX2 = 2
          ELSE
            GOTO 20
          ENDIF
          IF (INDEX1.NE.INDEX2) GOTO 20
          IF (IDATA(5,K).EQ.28) THEN
            S1 = S1 + DATA(11,K)
            N1 = N1 + 1
          ELSEIF (IDATA(5,K).EQ.32) THEN
            IF (IDATA(4,K).EQ.0) THEN
              S2 = S2 + DATA(11,K)
```

```
              N2 = N2 + 1
            ELSEIF (IDATA(4,K).EQ.1) THEN
              S3 = S3 + DATA(11,K)
              N3 = N3 + 1
            ENDIF
          ENDIF
 20     CONTINUE
      IF (N1.EQ.0) THEN
        AVER(1,I) =99999.
      ELSE
        AVER(1,I) = S1 / N1
      ENDIF
      IF (N2.EQ.0) THEN
        AVER(2,I) =99999.
      ELSE
        AVER(2,I) = S2 / N2
      ENDIF
      IF (N3.EQ.0) THEN
        AVER(3,I) =99999.
      ELSE
        AVER(3,I) = S3 / N3
      ENDIF
 10   CONTINUE
C .5 * ( O2OUT - O2IN )
      DO 2 I=1,II
      IF (IDATA(5,I).EQ.32) THEN
        S = (AVER(3,I)-AVER(2,I))/2
        IF(ABS(S).LT.400.)GO TO 3
        IF(AVER(1,I).GT.11111.)GO TO 2
        S=AVER(3,I)-AVER(1,I)
        IF(ABS(S).GT.400.)THEN
        S=AVER(1,I)-AVER(2,I)
        ENDIF
        IF(ABS(S).GT.400.)GO TO 2
3       CONTINUE
        IF(MV(I).EQ.1) SV(I)=S
        IF (IDATA(4,I).EQ.0) THEN
          DATA(11,I) = AVER(2,I) + S
        ELSEIF (IDATA(4,I).EQ.1) THEN
          DATA(11,I) = AVER(3,I) - S
        ENDIF
      ENDIF
2     CONTINUE
      SL=99999
      SH=-99999
      IB=0
      DO 4 I=1,II
      IF(MV(I).NE.1)GO TO 4
      S=SV(I)
      IF(S.GT.500)GO TO 4
      IF(IB.EQ.0)IB=I
      IF(S.GT.SH)THEN
      SH=S
      IH=I
      ENDIF
      IF(S.LT.SL)THEN
      SL=S
      IL=I
      ENDIF
 4    CONTINUE
```

```
      IT=0
      IF(IH.GT.IB+J.AND.IH.LT.IE-J)IT=IH
      IF(IL.GT.IB+J.AND.IL.LT.IE-J)IT=IL
      IF(IT.EQ.0)RETURN
      KL=0
      KH=0
      DO 6 K=IT-J,IT-J-8,-1
      IF(MV(I).NE.1)GO TO 6
      IF(IDATA(5,I).NE.32)GO TO 6
      KL=K
      GO TO 7
6     CONTINUE
7     CONTINUE
      DO 8 K=IT+J,IT+J+8
      IF(MV(I).NE.1)GO TO 8
      IF(IDATA(5,I).NE.32)GO TO 8
      KH=K
      GO TO 9
8     CONTINUE
9     CONTINUE
      DO 5 I=IT-J,IT+J
      IF(MV(I).NE.1)GO TO 5
      IF(I.LT.1.OR.I.GT.II)GO TO 5
      IF(IDATA(5,I).NE.32)GO TO 5
      IF(I.LE.IT) THEN
      IF(KL.LT.1)GO TO 5
      S=SV(KL)
      ENDIF
      IF(I.GT.IT) THEN
      IF(KH.GT.II)GO TO 5
      S=SV(KH)
      ENDIF
      IF(ABS(S).GT.500)GO TO 5
        IF (IDATA(4,I).EQ.0) THEN
          DATA(11,I) = AVER(2,I) + S
        ELSEIF (IDATA(4,I).EQ.1) THEN
          DATA(11,I) = AVER(3,I) - S
        ENDIF
5     CONTINUE
      RETURN
      END
```