

QB4
D66 IR

This document was produced
by scanning the original publication.

Ce document est le produit d'une
numérisation par balayage
de la publication originale.

PROCEDURES AND PROGRAMS FOR THE REDUCTION
OF CANADIAN MAGNETIC REPEAT STATION DATA

by

GEOPHYSICS / GÉOPHYSIQUE
LIBRARY / BIBLIOTHÈQUE

AUG 1 1991

L.R. Newitt

GEOLOGICAL SURVEY
COMMISSION GÉOLOGIQUE

Geophysics Division

Geological Survey of Canada

Internal Report 91-3

INTRODUCTION

An earlier report (Newitt, 1986) has described the procedures used during the mid-1980s to process Canadian repeat station data and to estimate the undisturbed level of the magnetic field. The report also described several factors affecting the accuracy of the data, such as temperature effects, the use of F/F* corrections and the effect of the small angle approximation on D and H baselines. It also summarized the evolution of repeat station data reduction procedures, pointing out that changing technology both necessitates and permits changes in data reduction procedures (as well as observational procedures).

The recent development of a ring-core solid-state memory magnetometer (a.k.a. lunch-box magnetometer) has resulted in such changes. First, it is now standard procedure to align the magnetometer sensor in the XYZ co-ordinate system instead of DHZ. (See Newitt, 1990 for a description of the magnetometer and the set-up procedure.) This allows data processing procedures to conform more closely to magnetic observatory data processing procedures. Secondly, it is necessary to upload data from the magnetometer into a personal computer at the end of each recording session. This makes it both possible and desirable to carry out data reduction on the PC, often while still in the field.

This manual describes the use of the standard observatory programs, suitably modified for PC use, required to

process repeat station data, as well as those programs which are unique to repeat station data processing. A flow diagram summarizing the data reduction procedure is given in Figure 1.

SOLID-STATE MEMORY MAGNETOMETER DATA PROCESSING

The procedure for uploading data from the lunch-box magnetometer is described fully in the Magnetic Repeat Station Field Manual (Newitt, 1990). The data are stored in a binary file called

OBMCYR1.BIN

(All examples in this manual, unless otherwise stated, are for station Clyde River D, which has the three letter code CYR, IAGA code 020291, and two-digit code 31.)

This file is converted to an ASCII file by entering the lunchbox magnetometer set-up program (as described on page 36 of the Field Manual) and using option "D" (reduce raw data from obm). This will produce an ASCII file named

OBMCYR.001

This file must be reformatted so that it contains records identical to those used in observatory data processing. However, before this can be done, OBMCYR.001 must be manually edited to remove *'s which appear in column 1 in various places throughout the file. Do this by entering XYWRITE and

using the universal command **CH /*/ /.** Also, delete the last line of the file which states "END OF DATA".

Now the file is converted to the standard observatory WKF-type file using program **REFORML**, which is listed in Appendix 1. The program first asks for the IAGA code. These are listed in Table 1 for all Canadian repeat stations. The output file is assigned the name **CYR.WKF**. Make certain that the output file is created in the **DATA** subdirectory; i.e., the full path name will be **C:\MT\DATA\CYR.WKF**. This comment applies to almost all the files which will be discussed subsequently. Thus, unless otherwise stated, it will be assumed that all data files are in the **DATA** subdirectory.

Programs file are, of course, in the **MT** directory.

The data file contains 10-second data, which must be averaged to form 1-minute data. However, before this is done, a program must be run to pad the file so that each day contains a full day's worth of values (8640 points). The time period before the start of the data, and after the end of the data, are filled with 999999s. The program which carries out this task is called **REBLOCK** and is listed in Appendix 2. The output file is assigned a name of **CYR1.WKF**. The averaging is done using program **MINAVE**, listed in Appendix 3. The output file name is **CYR2.WKF**.

PROTON MAGNETOMETER DATA PROCESSING

Both the GSM-18 and the GSM-10 proton magnetometers store data in solid-state memory. These data must be uploaded into the computer before they can be processed.

The procedure to do so is listed step-by-step below:

- 1) Remove the internal modem from the Data General computer.
- 2) Turn on the magnetometer and connect it to port 1 of the computer through the ACI-18 interface.
- 3) Key CNTL/ALT/CMD to enter the computer set-up mode.
Verify that COM2 is set to 9600 baud. Reboot.
- 4) Enter the XTALK directory. Start XTALK.
- 5) Load the VAX command file (#7). Verify that SP=9600 and PO=2. (When the internal modem is removed, port 1 is taken as port 2 by XTALK).
- 6) Turn capture on using a file name GEM.DAT.
- 7) Go Local.
- 8) Press the SEND button (blue mode) on the magnetometer.
The data should appear on the screen and the hard disc light should flash periodically.
- 9) When all data have been transferred, turn Capture Off.

This file contains data from all stations occupied during the survey, and must be broken into individual files for each station. To do this, edit the file using XYWRITE. Use the SAVEDEF command to save a block of data from one sta-

tion. Give the appropriate block a name of CYR.F. It may not be possible to fit an entire station file in the define block. If this is the case, save several smaller define blocks and concatenate them using the DOS COPY command.

Records in CYR.F will most likely not be FORTRAN records; they will therefore be unreadable by any of the editing programs. To correct this problem, use program FTNREC, which is listed in Appendix 4. This is a compiled BASIC program. Note that it also requires the program BASRUN.EXE. To run FTNREC, the input file must reside in the same directory as the program. Therefore, copy the data to the program directory or vice versa. Be sure to copy the output file, named CYR1.F, back to the data subdirectory.

The file of ppm data is next converted to a WKF-type file using REFORMG (Appendix 5). The output file is name CYR2.F. Finally, REBLCKF (Appendix 6) is run to fill out the days at the beginning and end of the file. The output file is CYR3.F. Since the data are minute values, no averaging is needed.

EDITING AND PLOTTING THE COMBINED FILE

The file of F data is now merged with the file of 3-component data using program FMERGE (Appendix 7). The output file is called CYRMER.WKF.

The data can now be plotted on the screen using **WKFPLT** (Appendix 8). The program asks for the input file name (CYRMER.WKF) and then the start day, end day to be plotted, and the sensitivity. The start day (and end day) must be the sequential day of the year; the sensitivity is normally 10 (nT/mm). However, if there are large disturbances present, it may be desirable to plot at a sensitivity of 20 or even 40. A hard copy of each day may be obtained by pressing the **SHIFT/PRTSC** keys before going on to the next day.

If a proper pen and ink plot is desired, it will be necessary to transfer the file to the microVax. To do this, the WKF file must be converted to a short-record file using **PRETRANS**. (Appendix 9). The file is then copied to a floppy and sent from the Compaq 386 to the microVax using DECNET. The copy command is:

```
COPY F:CYRMER.WKF MVAX4"NEWITT LEFROY":::DUB1:[NEWITT.REPEAT]*
```

The file must be converted to a WKF file using **POSTRANS**, residing on the microVax (Appendix 10). It may now be plotted on the Zeta 8 plotter using the Vax version of **WKFPLT**.

The plot should be examined for spikes. There are no editing programs to remove such spikes. If any are present, the file (CYRMER.WKF) is placed in the editor, and the bad values are manually replaced by 999999s.

PROCESSING OF ABSOLUTE OBSERVATIONS AND BASELINES

Absolute observations are entered into computer file using XYWRITE. The file name is CYRD.ABS; the "D" refers to the fact that the station is at the Clyde River "D" location. The data are entered in free format in the following order:

- 1) Two digit station identifier
- 2) Year (2 digits)
- 3) Sequential day of the year
- 4) Up and north D reading (degrees)
- 5) Up and north D reading (minutes)
- 6) Up and south D reading (degrees)
- 7) Up and south D reading (minutes)
- 8) Mean time of D observation (hhmm)
- 9) Down and north D reading (degrees)
- 10) Down and north D reading (minutes)
- 11) Down and south D reading (degrees)
- 12) Down and south D reading (minutes)
- 13) True north direction, in minutes of arc
- 14) D instrument correction (minutes)
- 15) Mean corrected D observation (degrees)
- 16) Mean corrected D observation (minutes)
- 17) Up and east I reading (degrees)
- 18) Up and east I reading (minutes)

- 19) Up and west I reading (degrees)
- 20) Up and west I reading (minutes)
- 21) Mean time of I observation (hhmm)
- 22) Down and east I reading (degrees)
- 23) Down and east I reading (minutes)
- 24) Down and west I reading (degrees)
- 25) Down and west I reading (minutes)
- 26) F observation (may be uncorrected for station difference)
- 27) I instrument correction (minutes)
- 28) Mean corrected I observation (degrees)
- 29) Mean corrected I observation (minutes)
- 30) / - record terminator; follow by ENTER

Each set of observations is separated by a record containing only 99/. A sample file is shown in Figure 2.

The file of observations is checked for errors using program **ABSCHK**. It creates a reformatted output file called **CYRD1.ABS**. A file listing errors, called **ABSDIAG.ERR** is also created which may be viewed or printed out. If any errors are indicated, **CYRD.ABS** should be corrected accordingly, and the program rerun. Be sure to delete the output file, and the error file, before rerunning. The program also asks for the F station difference, that is, the value which must be added to the F observations given in the file to correct them to values at the repeat station marker.

This correction must be calculated before hand. The program asks for a value after each set of absolutes in case the position of the ppm was changed from one set of observations to the next. The F values listed in **CYRD1.ABS** have the F correction added.

Next, **ABSCONV** (Appendix 12) is run. This program converts the absolute file to a form which can be used subsequently by **DIFTRE**. The output file is named **CYRD2.ABS**.

DIFTRE (Appendix 13) creates a file of spot baseline values. Both **CYRDMER.WKF** and **CYRD2.ABS** are input files. The output file is name **DIF90.CYR**. The requested 2-digit identifier code can be found in Table 1. The printout file should be checked for any obvious errors and should be kept for the records.

Sometimes a baseline value is not calculated for the last set of absolutes. If this is the case, proceed as follows. Use the editor and the **SAVEDEF** command to copy the last set of absolutes from **CYRD2.ABS** to a temporary file. Then run **DIFTRE** on that file. Finally, use the DOS copy command to add the output file to **DIF90.CYR**.

The next program to be run is **BASELINE** (Appendix 14). **BASELINE** requires two input files, the file of spot baseline files, **DIF90.CYR**, and a file listing rejected values, discontinuities and baseline segments, called **CYRBASE.INP**. A sample of this file, as it is set up before the initial run,

is given in Figure 3. The first line gives the IAGA code, start year, start day, end year and end day. 'DLET' refers to deleted spot baseline values, and 'DISC' refers to baseline segments, for the Z, Y and X components in turn. Since there are no deletions or baseline segments established for the initial run, zero is entered after 'DISC' and 'DLET'. When asked for the plot type and scale, enter 0 for the initial run. This option displays the spot baseline values only. (On subsequent runs, when baseline segments have been specified in CYRBASE.INP, enter 2.) Scale is normally entered as 20, but can be 50 or even more depending on the change in baseline values over the interval. This must be determined by trial.

A hard copy of the output may be obtained by printing the file called "OUTPUT" found in the MT directory. A hard copy of the baseline plot is obtained by keying SHIFT/PRTSC. Note that **BASELINE** can only plot 5 days on the screen.

Baseline segments are established by examining the plot of spot baseline values and all other available information. The segments are entered into CYRBASE.INP; an example is given in Figure 4. Note that the 'DISC' lines must be changed to give the number of entries to follow. The first value on each line is the date and time of the start of the baseline segment. Next is the start value; if the value is 999, the end value of the previous segment is used. Next is

the end value of the segment; if it is 999, **BASELINE** determines a value though least squares.

Once the baseline segments have been established rerun the program, this time asking for an output baseline file. The output file is named **CYRBASE.OUT**.

This output baseline file is added to **CYRMER.WKF** to form a file of absolute one minute values called **CYRABS.WKF**. This is accomplished using program **ADDBASE** (Appendix 15). Note that **ADDBASE** creates a printout file in the MT directory that is normally too long to be printed. Be sure to examine the end of this file on the screen to see that the job terminated normally, then delete it. **CYRABS.WKF** is the final output file, and data from it are used to calculate an undisturbed level of the magnetic field. A summary of the data, in hourly mean value form, is provided by program **MHVCHECK** (Appendix 16). A sample listing is given in Figure 5.

Ideally, the undisturbed level of the magnetic field is obtained by averaging over a few hours near local midnight on one or more days. There are occasions when this is not possible. If the nighttime level is somewhat disturbed or unsettle, it may be preferable to take an average of 24 hours (or 48 or 72 hours if the days are all of comparable activity). It might also be better to take a daily mean if there is a large scatter in the baselines, especially if

there is a pronounced difference between morning and evening baselines indicating a temperature related effect. In practice, each situation is different, and requires a certain amount of judgment on the part of the person doing the data reduction.

One final program remains, called **FFPLT** (Appendix 17). This program plots F-F* differences between the baseline corrected fluxgate values and the ppm values. Ideally, this plot should be a straight line. Deviations from it indicate uncorrected variations in the fluxgate magnetometer. Naturally F-F* is most sensitive to changes in Z, especially at high latitudes, and least sensitive to changes in Y. The plotting sensitivity used with FFPLT is normally 1. Again, a hard copy plot can be obtained by keying CNTL/PRTSC.

REFERENCES

Newitt, L.R., Reduction of magnetic repeat station data in Canada. G.S.C. Geophysics Division Internal Report 86-16, 1986.

Newitt, L.R., Magnetic repeat station field manual. Geological Survey of Canada, Geophysics Division. Second Revision, April 1990.

TABLE 1
REPEAT STATION CODES

Number	Name	Letter Code	IAGA Code
20	Frobisher Bay	FRB	026291
21	Goose Bay	GOO	037300
22	Halifax	HAL	045296
23	Inuvik	INK	022227
26	Big Trout Lake	BTL	036270
27	Bonne Bay	BNB	040302
28	Brandon	BRD	040260
29	Chibougamau	CBG	040286
30	Claremont	CRM	046281
31	Clyde River	CYR	020291
32	Coral Harbour	CBR	026277
33	Dawson	DWY	026221
34	Ennadai	ENN	029259
35	Essex	ESS	048277
36	Estevan	ESV	041257
37	Eureka	EUA	010274
38	Fort Chimo	FCH	032292
39	Fort Macleod	FMD	040247
40	Fort Nelson	FNS	031247
41	Iqloolik	IGL	021278
42	Lynn Lake	LYN	033259
43	Moosonee	MSE	039279
44	Nain	NAL	033298
45	New Liskeard	NLD	043280
46	Norman Wells	NOR	025233
47	Penticton	PNT	041240
48	Pond Inlet	PDT	017282
49	Port Hardy	PHC	039233
50	Port Harrison	PHS	032282
51	Prince George	PGG	036237
52	Sachs Harbour	SAH	018235
53	Saskatoon	SAS	038253
54	Sault Ste Marie	SRE	044276
55	Schefferville	SCH	035293
56	Sept Iles	SIC	040294
57	Smithers	SMH	035233
58	The Pas	TPS	036259
59	Twin City	TCY	042271
60	Uranium City	URC	030251
61	Whitehorse	WHC	029225
62	Woodstock	WDK	044292
71	Ayer's Cliff	AYC	045288

72	Battle Harbour	BHR	038304
73	Cape Dorset	CDT	026283
74	Charlottetown	CTT	044297
75	Coppermine	CPM	022245
76	Fort Resolution	FRS	029246
77	Ile-a-la-Crosse	ILC	035252
78	Louisburg	LBG	044300
79	Matapedia	MPD	042293
80	Ocean Falls	OFB	038232
81	Pangnirtung	PGT	024294
82	Peace River	PCR	034243
83	Port Burwell	PBL	030295
84	Quebec	QCQ	043289
85	Rainy River	RNR	041265
86	Repulse Bay	RPB	023274
87	Tyrrell Lake	TRL	027255
88	York Factory	YOF	033268

CANADIAN MAGNETIC REPEAT STATION DATA FLOW 1991

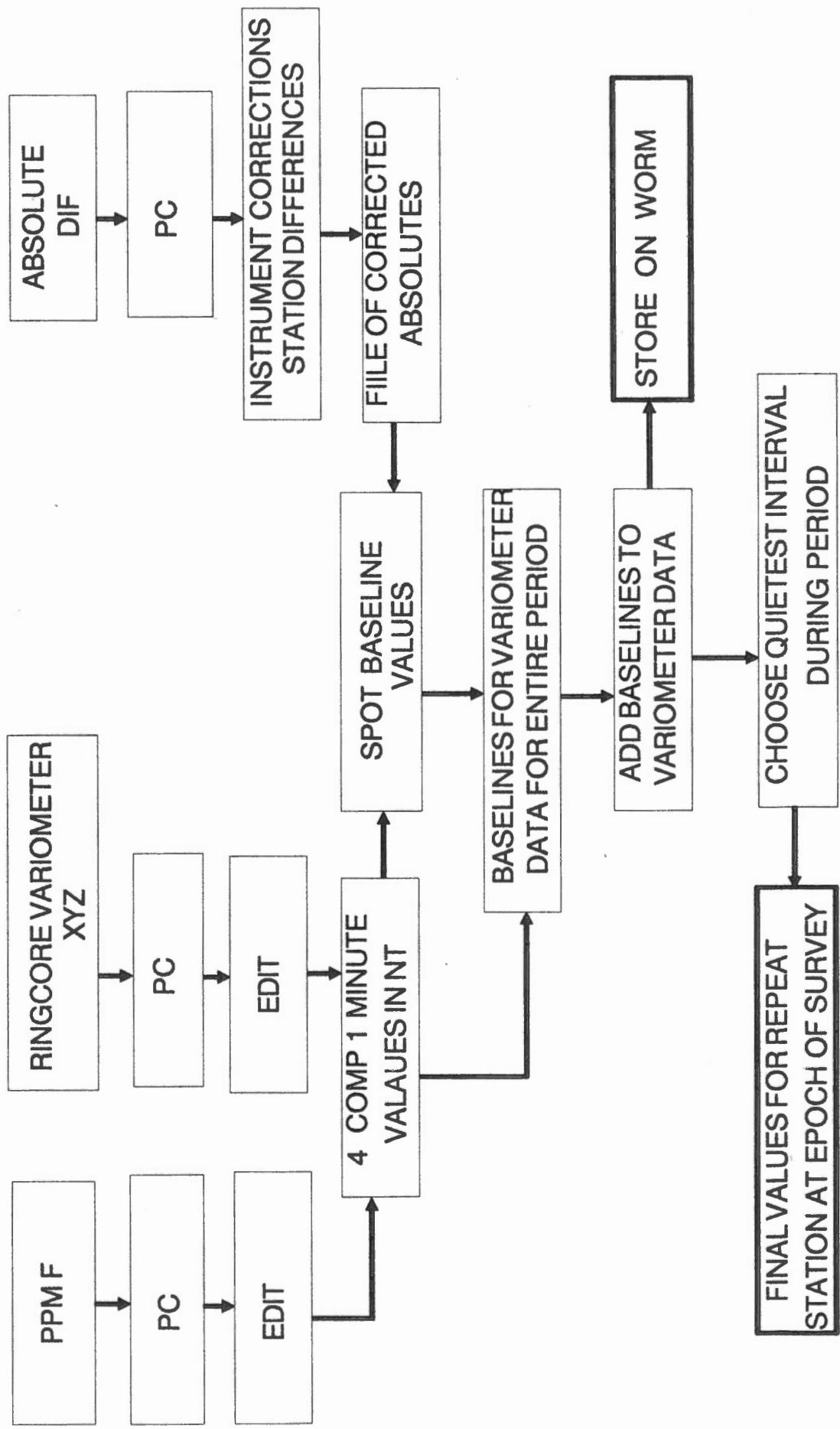


FIGURE 2

58 90 257 101 20.6 281 28.4 2212 101 41.6 281 38.8 0.0 -0.5 11 31.9 101 33.1 78
28.2 2217 281 31.8 258 27.2 60186.4 0.0 78 27.7/
58 90 257 101 18.2 281 27.0 2224 101 40.0 281 34.6 0.0 0.0 11 30.0 101 33.0 78
28.6 2228 281 31.6 258 27.2 60181.0 0.0 78 27.8 /
58 90 257 101 19.0 281 26.6 2233 101 39.2 281 36.4 0.0 0.0 11 30.3 101 33.4 78
28.6 2238 281 31.8 258 27.2 60179.4 0.0 78 27.7/
58 90 257 101 18.0 281 29.0 2243 101 41.2 281 36.4 0.0 0.0 11 31.2 101 33.6 78
27.8 2245 281 31.8 258 26.0 60181.0 0.0 78 27.1/
58 90 257 101 17.2 281 24.2 2249 101 37.6 281 35.6 0.0 0.0 11 28.7 101 33.6 78
27.8 2252 281 33.2 258 27.2 60180.2 0.0 78 27.1/
58 90 257 101 22.0 281 25.6 2256 101 40.0 281 38.0 0.0 0.0 11 31.4 101 33.0 78
27.2 2259 281 32.6 258 27.0 60178.2 0.0 78 27.2/
58 90 257 101 22.0 281 25.4 2302 101 40.0 281 37.0 0.0 -0.2 11 30.9 101 33.0 78
27.6 2311 281 32.0 258 27.0 60175.5 0.0 78 27.4/
58 90 257 101 21.6 281 26.4 2315 101 39.6 281 36.0 0.0 0.0 11 30.9 101 33.0 78
28.0 2318 281 32.0 258 27.2 60173.2 0.0 78 27.6/
58 90 257 101 20.0 281 26.8 2327 101 40.2 281 37.0 0.0 -0.2 11 30.8 101 33.2 78
27.6 2330 281 32.2 258 27.0 60172.8 0.0 78 27.3/
58 90 257 101 20.2 281 29.8 2333 101 40.2 281 35.6 0.0 0.1 11 31.6 101 33.0 78
27.8 2336 281 32.0 258 27.0 60171.6 0.0 78 27.5/
99/
58 90 258 101 41.2 281 36.0 1522 101 57.0 281 47.2 0.0 0.2 11 45.6 101 29.8 78
30.0 1526 281 28.2 258 29.0 60152.8 0.0 78 30.3/
58 90 258 101 41.0 281 38.4 1529 101 58.6 281 47.4 0.0 0.2 11 46.6 101 28.8 78
30.7 1532 281 27.8 258 29.5 60155.8 0.0 78 30.9/
58 90 258 101 40.0 281 36.5 1539 101 61.2 281 44.4 0.0 0.0 11 45.5 101 28.2 78
30.9 1542 281 27.3 258 30.2 60151.4 0.0 78 31.4/
58 90 258 101 32.6 281 33.6 1547 101 54.2 281 37.9 0.0 0.0 11 39.6 101 28.6 78
30.3 1552 281 27.0 258 29.6 60150.8 0.0 78 31.1/
58 90 258 101 34.0 281 33.8 1556 101 57.4 281 41.0 0.0 -0.2 11 41.4 101 28.5 78
31.0 1559 281 27.1 258 30.1 60153.0 0.0 78 31.4/
58 90 258 101 36.0 281 33.8 1606 101 60.4 281 43.2 0.0 0.2 11 43.6 101 28.1 78
31.4 1609 281 26.8 258 30.2 60160.6 0.0 78 31.7/
58 90 258 101 36.2 281 34.0 1612 101 57.4 281 42.0 0.0 0.0 11 42.4 101 28.1 78
32.0 1615 281 27.0 258 30.8 60160.6 0.0 78 31.9/
58 90 258 101 31.6 281 33.2 1619 101 53.9 281 40.8 0.0 0.0 11 39.9 101 28.0 78
31.9 1621 281 26.6 258 30.8 60163.8 0.0 78 32.0/
99/

FIGURE 3

36259
90
257
90
260
'DLET
' 0
'DISC
' 0
'DLET
' 0
'DISC
' 0
'DLET
' 0
'DISC
' 0

FIGURE 4

36259
90
257
90
260
'DLET
' 0
'DISC
' 3
25700
01,-
520,9
99,'R
'
26000
01,99
9,999
, 'R'
26100
01,99
9,999
, 'R'
'DLET
' 0
'DISC
' 2
25700
01,-
2230,
999,'
R'
26100
01,99
9,999
, 'R'
'DLET
' 0
'DISC
' 7
25700
01,69
5,999
, 'R'
25905
28,12
50,99
9,'R'
25915
01,99
9,105
5,'R'

25918

56,19

57,19

57,'R

'

26000

01,19

57,21

14,'R

'

26015

01,21

14,99

9,'R'

26100

01,99

9,999

,'R'

FIGURE 5

END OF REQUESTED DATA IN S/R RETRIEVED FROM UNIT 1

1

TPS 90 258 9 15 X 11000 808 819 870 886 681 699 827 780 755 763 775 555
574 776 778 738 737 763 779 780 804 812 803 800 765

TPS 90 259 9 16 X 11000 829 871 901 884 8879999 722 473 724 687 395 609
732 703 747 776 826 835 806 786 794 805 816 8409999

1

TPS 90 258 9 15 Y 2000 414 419 453 448 319 318 401 401 405 421 417 313
325 461 453 431 397 385 390 379 395 407 409 405 399

TPS 90 259 9 16 Y 2000 417 476 510 464 456 441 314 179 384 389 245 400
436 393 401 385 412 416 393 394 399 402 407 420 397

1

TPS 90 258 9 15 Z 58000 955 958 966 889 847 875 943 937 954 936 935 960
947 929 952 966 981 9831006100010141003 986 967 954

TPS 90 259 9 16 Z 58000 9741035 9991019 986 949 9651003 922 922 999 821
906 912 951 934 983 995 996 990 986 982 980 989 967

1

TPS 90 258 9 15 X 0 3 5 6 9 54 20 13 7 8 9 4 46
39 6 4 5 6 2 4 6 7 3 3 1 64

TPS 90 259 9 16 X 0 6 12 12 6 89999 34 24 19 40 17 27
23 13 17 15 9 59006 2 2 3 4 59999

OEND OF JOB

APPENDIX 1

REFORML.FOR

PROGRAM REFORML

C L.R. NEWITT 1989.053

C PROGRAM REFORMATS DATA FROM THE LUNCHBOX MAGNETOMETER

C INTO STANDARD *.WKF ASCII FILES

C FOR USE ON THE DATA-GENERAL ONE

C

C*****NOTE*****

C WHEN TRANSFERRING DATA FILES TO THE PC USING X-TALK

C MAKE CERTAIN X-TALK IS SET UP TO ALLOW THE TRANSMISSION OF CR AND LF

C

```
DIMENSION MON(12), IVAR(3,128),ITEMP(256),JVAR(3,128),ICOMP(8)
DIMENSION IHEAD(128), JTEMP(750), TEMP(16)
CHARACTER*80 INFILE,OUTFILE, TEMFILE, AA
CHARACTER*30 IDATA(128)
CHARACTER*1 IXY
DATA MON/31,28,31,30,31,30,31,31,30,31,30,31/
DATA ICOMP/1,1,1,0,0,0,0,0,0/
INFILE = '
OUTFILE = '
TEMFILE = '
```

C READ IN THE IAGA CODE WHICH WILL BE PRINTED ON THE HEADER

C

```
PRINT 22
FORMAT(1H1,'ENTER IAGA CODE... ')
READ(*,*)IOBS
```

C THIS IS THE TEMPERATURE SAMPLING RATE IS SECONDS

C

```
ITMSCL=999
LREC=1
```

C

```
PRINT 1
FORMAT(1H1,'THIS PROGRAM CONVERTS DATA FROM THE LUNCHBOX',//,
*1H , 'MAGNETOMETER TO A *.WKF ASCII FILE')
```

C THE NAME OF THE INPUT FILE IS REQUESTED

C

```
PRINT 2
FORMAT(1H0,'ENTER NAME OF INPUT FILE.... ')
READ(*,3)INFILE
FORMAT(A)
OPEN(UNIT=1, FILE=INFILE, STATUS='OLD',access='sequential')
```

C THE NAME OF THE OUTPUT FILE IS REQUESTED

C

```
PRINT 4
FORMAT(1H0,'ENTER NAME OF OUTPUT FILE.... ')
```

```
READ(*,3)OUTFILE
OPEN(UNIT=2, FILE=OUTFILE, STATUS='NEW', ACCESS='SEQUENTIAL',
*FORM='FORMATTED', RECL=1024)
C
C THIS IS A SCRATCH FILE USED FOR TEMPORARY STORAGE OF DATA
C
C     OPEN(UNIT=3, STATUS='SCRATCH', FORM='UNFORMATTED', ACCESS='DIRECT'
*C,RECL=512)
C
C THE NAME OF THE TEMPERATURE FILE IS REQUESTED AT THIS POINT
C
C
C READ IN THE YEAR AND INSTRUMENT SERIAL NUMBER
C ALL OTHER PARAMETERS ARE OBTAINABLE FROM THE INPUT FILE.
C ASSUME THAT THE SAMPLING RATE IS 10 SECONDS
C AND THAT THE SCALE FACTOR IS 10
    INT=10
    ISCALE =10
    PRINT 5
5   FORMAT(1H0,'ENTER YEAR....')
    READ(*,*) JYR
    IF(MOD(JYR,4).EQ.0) MON(2)=29
    SCALE=ISCALE
    PRINT 31
31  FORMAT(1H0,'ENTER INSTRUMENT SERIAL NUMBER... ')
    READ(*,*) INSTID
C
C READ THE START TIME OF THE FILE
C
    IBLOCK=1
    KKNT=1
100 CONTINUE
    READ(1,27)AA
    READ(1,27)AA
27   FORMAT(A)
    LSEC=0
    READ(1,*)JSTDAY,LHR,LMIN
    PRINT 23, JSTDAY,LHR,LMIN,LSEC
23   FORMAT(1H0,'START TIME: DAY',I5,' HOUR',I3,' MIN',I3,' SEC',I3)
    LMINOLD=LMIN
    KDIF=(JYR-1970)*365
    FLP=(JYR-1970.)/4.
    KLP=(JYR-1970)/4
    IF(FLP.GT.(KLP+.5))KLP=KLP+1
    KDIF=KDIF+KLP
    JSEQ=JSTDAY-KDIF+1
    MONTOT=0
    JJ=1
    DO 53 K=1,12
    MONTOT=MONTOT+MON(K)
    IF(MONTOT.GT.JSEQ)GO TO 54
    JJ=JJ+1
```

```
53    CONTINUE
54    MONTOT=MONTOT-MON(JJ)
      JMON=JJ
      JDAY=JSEQ-MONTOT
      PRINT 24,JYR,JMON,JDAY,JSEQ
24    FORMAT(' CONVERTED: YEAR',I5,' MONTH',I3,' DAY',I3,' SEQ DAY',I4)
      DO 55 J=1,13
55    READ(1,27)AA
      READ(1,*)XBL,YBL,ZBL
      XBL=XBL/1000.
      YBL=YBL/1000.
      ZBL=ZBL/1000.
      PRINT 25, XBL,YBL,ZBL
25    FORMAT(1H0,'BASELINE VALUES: X',F8.1,' Y',F8.1,' Z',F8.1)
      DO 56 J=1,7
56    READ(1,27)AA
C
C   READ DATA FROM INPUT FILE
C
101   ISEC=0
      JCNT=0
      NREC=3
      IEND=0
      NOLD=N
      N=1
      ISECTOT=LHR*3600+LMIN*60+LSEC
90    READ(1,*,IOSTAT=IERR,ERR=997,END=999)X,Y,Z
      if(kknt.gt.1020)print *, kknt,isectot, x,y,z,iblock
      IF(KKNT.EQ.1024)THEN
      READ(1,333)AA
      READ(1,333)AA
333   FORMAT(A)
      PRINT 334,AA,KKNT
334   FORMAT(1H ,A,I6)
      KKNT=0
      ENDIF
      if(iblock.gt.8550)print *,iblock,lmin,lminold,x,y,z
      IF(IBLOCK.EQ.(8639-LMINOLD*6))THEN
      READ(1,333)AA
      READ(1,333)AA
      READ(1,*)DUM1,DUM2,DUM3
      PRINT *,DUM1,DUM2,DUM3
      DO 755, I=1,13
755   READ(1,333)AA
      READ(1,*)DUM1,DUM2,DUM3
      PRINT*, DUM1,DUM2,DUM3
      DO 756, I=1,7
756   READ(1,333)AA
      IBLOCK=0
      KKNT=0
      ENDIF
      IBLOCK=IBLOCK+1
```

```
KKNT=KKNT+1
ISECTOT=ISECTOT+10
XXX=X/10000.+XBL
YYY=Y/10000.+YBL
ZZZ=Z/10000.+ZBL
IVAR(1,N)=XXX*10.+.5
IVAR(2,N)=YYY*10.+.5
IVAR(3,N)=ZZZ*10.+.5
IF(ISECTOT.EQ.86390)GO TO 210
IF(N.EQ.128)GO TO 200
N=N+1
GO TO 90
C
C IF 128 VALUES HAVE BEEN READ IN, WRITE THEM ON A SCRATCH FILE
C REINITIALIZE COUNTER AND CONTINUE READING
C THIS WILL BE ONE RECORD OF DATA ON OUTPUT FILE
C
200 JREC=NREC-2
    DO 201 II=JREC,NREC
        JJ=II-JREC+1
201 WRITE(3,REC=II)(IVAR(JJ,I),I=1,128)
    N=1
    NREC=NREC+3
    GO TO 90
C
C WHEN THE END OF A DAY HAS BEEN REACHED, WRITE DATA ONTO SCRATCH FILE
C THEN WRITE HEADER RECORD ONTO OUTPUT FILE, AND COPY ALL DATA RECORDS
C FROM SCRATCH FILE TO OUTPUT FILE
C
C IOBS - OBSERVATORY CODE
C INSTID - INSTRUMENT SERIAL NUMBER OR IDENTIFIER
C JDAY - DAY
C JMON - MONTH
C JYR - YEAR
C JSEQ - SEQUENTIAL DAY OF THE YEAR
C JSTHR - HOUR AT WHICH DATA STARTS
C JSTMIN - MINUTE AT WHICH DATA STARTS
C JSTSEC - SECOND AT WHICH DATA STARTS
C LENDHR - HOUR AT WHICH DATA STOPS
C LENDMIN - MINUTE AT WHICH DATA STOPS
C LENDSEC - SECOND AT WHICH DATA STOPS
C INT - SAMPLING INTERVAL IN SECONDS
C ICOMP - INDICATES WHICH COMPONENTS ARE RECORDED
C             PRESENCE OF COMPONENT INDICATED BY 1, ABSENCE BY 0
C ICOMP(1) - X
C ICOMP(2) - Y
C ICOMP(3) - Z
C ICOMP(4) - D
C ICOMP(5) - H
C ICOMP(6) - F
C ICOMP(7) - I
C ICOMP(8) - OTHER
```

C ISCALE - THE SCALE FACTOR OF THE MAGNETIC DATA
C ITMSCL - THE TIME INCREMENT OF THE TEMPERATURE DATA
210 CONTINUE
NP=N+1
DO 230 J=1,3
DO 230 K=NP,128
230 IVAR(J,K)=9999999
JREC=NREC-2
DO 231 II=JREC,NREC
JJ=II-JREC+1
231 WRITE(3,REC=II)(IVAR(JJ,I),I=1,128)
NREC=NREC+3
LENDHR=ISECTOT/3600
LENDMIN=(ISECTOT-LENDHR*3600)/60
LENDSEC=ISECTOT-LENDHR*3600-LENDMIN*60
NREC=NREC-3
DO 280 KK=1,128
280 IHEAD(KK)=9999999
286 continue
icomp(1)=1
icomp(2)=1
icomp(3)=1
JSTHR=LHR
JSTMIN=LMIN
LSTSEC=LSEC
WRITE(2,20)IOBS,INSTID,JDAY,JMON,JYR,JSEQ,JSTHR,JSTMIN,
1JSTSEC,LENDHR,LENDMIN,LENDSEC,INT,(ICOMP(I),I=1,8),ISCALE,ITMSCL,
2(IHEAD(JJ),JJ=24,128)
20 FORMAT(128I8)
PRINT 704, JDAY,JMON,JYR,JSTHR,JSTMIN,JSTSEC,LENDHR,LENDMIN,
*LENDSEC
704 FORMAT(1H0,'INFORMATION WRITTEN IS FOR DAY',3I5,/,1H ,
1'FIRST TIME WRITTEN IS',3I3,/,1H , 'LAST TIME WRITTEN IS',3I3)
LREC=LREC+1
C
C WRITE THE TEMPERATURES FOR THE DAY ON THE FILE IMMEDIATELY FOLLOWING
C THE HEADER RECORD
C
DO 110 I=1,256
ITEMP(I)=9999999
110 CONTINUE
GO TO 122
ITMCNT=1
112 READ(4,14,END=122)KDAY,KHR,KMIN,(TEMP(I),I=1,NSENSOR)
14 FORMAT(3I5,10X,F5.1)
IF(KDAY.EQ.JSEQ)GOTO 114
IF(KDAY.EQ.JSEQ+1.AND.KHR.EQ.0.AND.KMIN.EQ.0) THEN
ITDUM=1440
GOTO 120
ENDIF
IF(KDAY.LT.JSEQ)THEN
IF(1TMcnt.EQ.1) PRINT 17

```
ITMCNT=2
GOTO 112
ENDIF
IF(KDAY.GT.JSEQ) THEN
    REWIND 4
    IF(1TMCNT.EQ.1)PRINT 18
    ITMCNT=2
    GOTO 122
ENDIF
16 FORMAT(1H0,'***WARNING***',//,1H , 'TEMPERATURE AND DATA STARTING
*TIMES DO NOT MATCH')
17 FORMAT(1H0,'***WARNING***',//, ' TEMPERATURE DATA STARTS DAY(S) BEFO
*RE VECTOR PPM DATA.')
18 FORMAT(1H0,'***WARNING***',//, ' TEMPERATURE DATA STARTS DAY(S) AFTE
*R VECTOR PPM DATA.')
114 ITDUM=KHR*60+KMIN
IF(ITDUM.EQ.0) GOTO 112
IDDUM=JSTHR*60+JSTMIN
IF(ABS(ITDUM-IDDUM).GT.10) THEN
    IF(1TMCNT.EQ.1) PRINT 16
    ITMCNT=2
ENDIF
120 ITEMP(ITDUM/10)=(TEMP(NSENSOR)+.05)*10.0
IF(ITDUM.LT.1440) GOTO 112
122 WRITE(2,20)(ITEMP(I),I=1,128)
      WRITE(2,20)(ITEMP(I),I=129,256)
      LREC=LREC+2
C
C   WRITE THE MAGNETIC DATA FOR THE DAY ONTO THE OUTPUT FILE,
C   ONE COMPONENT AT A TIME
C
DO 220 L=1,NREC,3
READ(3,REC=L)(JVAR(1,I),I=1,128)
WRITE(2,20)(JVAR(1,I),I=1,128)
220 LREC=LREC+1
DO 221 L=2,NREC,3
READ(3,REC=L)(JVAR(2,I),I=1,128)
WRITE(2,20)(JVAR(2,I),I=1,128)
221 LREC=LREC+1
DO 222 L=3,NREC,3
READ(3,REC=L)(JVAR(3,I),I=1,128)
WRITE(2,20)(JVAR(3,I),I=1,128)
222 LREC=LREC+1
CLOSE(UNIT=3, STATUS='DELETE')
OPEN(UNIT=3, STATUS='SCRATCH', ACCESS='DIRECT', FORM='UNFORMATTED'
*,RECL=512)
IF(IEND.EQ.99)STOP
JDAY=JDAY+1
JSEQ=JSEQ+1
LHR=0
LMINOLD=LMIN
LMIN=0
```

```
LSEC=0
GO TO 101
998 PRINT 19, IRAT, INT, IHR, IMIN, LHR, LMIN
19  FORMAT(1H0,'*****WARNING***',//,1H , 'INCREMENT PROBLEM',//,
*1H ,6I8)
997 PRINT 996,IERR,IBLOCK,KKNT,ISECTOT,X,Y,Z
996 FORMAT(1H0,' ERROR #', I5,3I10,3F10.0)
999 IEND=99
N=N-1
GO TO 210
END
```

APPENDIX 2

REBLOCK.FOR

PROGRAM REBLOCK

C
C VERSION 1987.357 G. JANSEN VAN BEEK
C MODIFIED 1989.053 BY L. NEWITT
C
C PROGRAM TAKES A *.WKF FILE CONTAINING 10-SECOND DATA
C AND ENSURES THAT THE FIRST AND LAST
C DAYS CONTAIN 8640 VALUES FOR EACH COMPONENT.
C PREFIX AND SUFFIX VALUES HAVE BEEN SET TO 999999.
C START AND END TIMES FOR EACH DAY HAVE BEEN SET TO 0001 AND 2400.
C
C NOTE: TIMES ARE SHIFTED BY +10 SECONDS
C I.E., 24 00 00 IS REALL 23 59 50
C
CHARACTER*20 TMP1,TMP2,FILE1,FILE2
COMMON ID(128),ITMP1(128),ITMP2(128)
DIMENSION IADAT(8768)
DIMENSION ID1(128)

C
C OPEN THE PROGRAM/FILES
C
WRITE(*,900)
900 FORMAT(1H1,'This program takes a *.WKF file and ensures that the',
 *' first and last day',//,' contain 2880 values for each component.',
 //,' Prefix and suffix values have been set to 999999. Start and ',
 *'times for each day have',//,' been set to 00 01 and 24 00.')
WRITE(*,902)
902 FORMAT(1HS,'Input data file name -----> ')
READ(*,800) FILE1
800 FORMAT(A)
 OPEN(1,FILE=FILE1,STATUS='OLD',IOSTAT=IOS)
 IF(IOS.EQ.29)STOP '*** INPUT FILE NOT FOUND ***'
 WRITE(*,904)
904 FORMAT(1H\$,'Output data file name -----> ')
READ(*,800) FILE2
OPEN(2,FILE=FILE2,STATUS='NEW',ACCESS='SEQUENTIAL',RECL=1024)

C
C SET UP THE WORKING LOOP
C
10 READ(1,810,END=700)(ID(I),I=1,128)
810 FORMAT(128I8)
 READ(1,810,IOSTAT=IOS)(ITMP1(I),I=1,128)
 IF(IOS.NE.0)STOP 'ABNORMAL STOP WHEN READING ITMP1'
 READ(1,810,IOSTAT=IOS)(ITMP2(I),I=1,128)
 IF(IOS.NE.0)STOP 'ABNORMAL STOP WHEN READING ITMP2'
 CALL DEFINE(NC,NREC)
 PRINT *, NC,NREC
 WRITE(*,906)(ID(I),I=1,13)
 DO 220 JJ=1,128
220 ID1(JJ)=ID(JJ)
 ID1(7)=00
 ID1(8)=00

```
ID1(9)=10
ID1(10)=24
ID1(11)=00
ID1(12)=00
WRITE(2,810)(ID1(I),I=1,128)
WRITE(2,810)(ITMP1(I),I=1,128)
WRITE(2,810)(ITMP2(I),I=1,128)
906 FORMAT(1H , 'Process at day block ',13I8)
C
DO 40 L=1,NC
DO 20 I=1,8768
IADAT(I)=999999
20 CONTINUE
CALL FILARA(NREC,IOS,IADAT)
IF(IO.NE.0)STOP 'ABNORMAL STOP WHILE READING COMP. DATA 1'
CALL OUTPT(IADAT)
40 CONTINUE
GOTO 10
C
C      END THE PROGRAM
C
700 WRITE(*,908)
908 FORMAT(1H0,'*** NORMAL END OF PROGRAM ***')
END
SUBROUTINE OUTPT(IT)
C
C      SUBROUTINE TO OUTPUT THE COMPONENT DATA
C
COMMON ID(128),ITMP1(128),ITMP2(128)
DIMENSION IT(8768)
C
DO 20 J=1,8640,128
IEND=J+127
IF(IEND.GT.8640) THEN
    DO 10 IXX=8641,IEND
    IT(IXX)=999999
10 CONTINUE
ENDIF
WRITE(2,800)(IT(I),I=J,IEND)
800 FORMAT(128I8)
20 CONTINUE
C
RETURN
END
SUBROUTINE DEFINE(NC,NREC)
C
C      G. JANSEN VAN BEEK 1987.335
C
COMMON ID(128),ITMP1(128),ITMP2(128)
C
C      SUBROUTINE DEFINES THE NUMBER OF COMPONENTS AND THE NUMBER
C      OF RECORDS PER COMPONENT THAT EXIST FOR THE DATA TIME INTERVAL
```

```
C   FOUND IN THE DATA HEADER ARRAY "ID"
C
C   CALCULATE THE NUMBER OF COMPONENTS PRESENT
C
NC=0
DO 10 I=14,20
IF(ID(I).NE.0)NC=NC + 1
10 CONTINUE
C
C   CALCULATE THE NUMBER OF RECORDS PER COMPONENT
C
MMDAY=ID(7)*360 + ID(8)*6 + ID(9)/10
NMDAY=ID(10)*360 + ID(11)*6 + ID(12)/10
MNDIF=NMDAY-MMDAY + 1
NREC=MNDIF/128.
IF(MOD(MNDIF,128).NE.0) NREC=NREC+1
RETURN
END
SUBROUTINE FILARA(NREC,IOS,IN)
C
C   G. JANSEN VAN BEEK 1987.335
C
COMMON ID(128),ITMP1(128),ITMP2(128)
DIMENSION NN(128),IN(8768)
C
C   SUBROUTINE READS THE DATA FOR ONE COMPONENT DAY INTO THE 8640 DATA
C   POINT ARRAY "IN"
C
LSTRT=ID(7)*360 + ID(8)*6 + ID(9)/10
NSTRT=LSTRT
LSTP=ID(10)*360 + ID(11)*6 + ID(12)/10
DO 20 K=1,NREC
READ(1,900,10STAT=IOS) (NN(I),I=1,128)
IF (IOS.NE.0) RETURN
900 FORMAT(128I8)
M=1
DO 10 L=NSTRRT,NSTRRT+127
IN(L)=NN(M)
M=M + 1
10 CONTINUE
NSTRT=NSTRRT + 128
20 CONTINUE
C
RETURN
END
```

APPENDIX 3

MINAVE.FOR

PROGRAM MINAVE

C

C PROGRAM TAKES 10-SEC DATA FROM A WKF FILE AND AVERAGES OVER 7 VALUES,
C CENTRED ON A MINUTE AND WRITES ONTO A NEW WKF FILE
C NOTE THAT THE LAST MINUTE OF THE DAY DOES NOT CONTAIN A FULL AVERAGE
C NOTE ALSO THAT THE INPUT FILE MUST CONTAIN A FULL DAYS WORTH OF DATA.
C IF NOT, USE PROGRAM REBLOCK FIRST

C

C L.R. NEWITT, FEB 23, 1989

C

DIMENSION IPAR(128), ITEMP(128), IDAT(128), VAL(8704), MVAL(1536)
CHARACTER*80 INFILE, OUTFILE
PRINT 1
1 FORMAT(1H0,'ENTER NAME OF INPUT FILE--->')
READ(*,2)INFILE
2 FORMAT(A)
PRINT 3
3 FORMAT(1H0,'ENTER NAME OF OUTPUT FILE--->')
READ(*,2)OUTFILE
OPEN(UNIT=1,FILE=INFILE,STATUS='OLD',RECL=1024)
OPEN(UNIT=2,FILE=OUTFILE,STATUS='NEW',RECL=1024)

C

C READ AND WRITE HEADER AND TEMPERATURE RECORDS AT BEGINNING OF EACH
C DAYS DATA
C CHANGE START AND STOP MINUTE AND SECOND VALUES, AND DATA INTERVAL
C TO APPROPRIATE MINUTE VALUES

C

100 READ(1,4,END=999)(IPAR(I),I=1,128)
4 FORMAT(128I8)
IPAR(8)=1
IPAR(9)=0
IPAR(11)=0
IPAR(12)=0
IPAR(13)=60
WRITE(2,4)(IPAR(I),I=1,128)
PRINT *, (IPAR(I),I=1,24)
READ(1,4)(ITEMP(I),I=1,128)
WRITE(2,4)(ITEMP(I),I=1,128)
READ(1,4)(ITEMP(I),I=1,128)
WRITE(2,4)(ITEMP(I),I=1,128)

C

C READ EACH COMPONENT IN TURN AND AVERAGE OVER 7 VALUES CENTRED ON
C THE MINUTE

C

DO 200 JJJ=1,3
K=1
DO 101 J=1,68
READ(1,4)(IDAT(I),I=1,128)
DO 111 III=1,128
VAL(K)=IDAT(III)
IF(K.LT.300)PRINT *, III, IDAT(I), K, VAL(K)
111 K=K+1

```
101    CONTINUE
C
C   CALCULATE MEAN ONE-MINUTE VALUE
C   CENTRE ON 7TH 10-SEC VALUE TO CORRECT 10-SECOND TIME SHIFT
C   INTRODUCED BY PROGRAM REBLOCK
C
DO 102 I=7,8641,6
XSUM=0.
XN=0.
DO 103 JJ=I-3,I+3
IF(VAL(JJ).GE.900000.)GO TO 103
XSUM=XSUM+VAL(JJ)
XN=XN+1
103  CONTINUE
L=(I-1)/6
IF(XN.NE.0.)THEN
MVAL(L)=XSUM/XN
ELSE
MVAL(L)=999999
ENDIF
102  CONTINUE
DO 104 M=1441,1536
104  MVAL(M)=999999
ISTRRT=1
IEND=128
DO 105 L=1,12
WRITE(2,4)(MVAL(I),I=ISTRRT,IEND)
ISTRRT=ISTRRT+128
IEND=IEND+128
105  CONTINUE
200  CONTINUE
GO TO 100
999  STOP
END
```

APPENDIX 4

FTNREC.BAS

```
10 REM Program FTNREC.BAS
20 REM Program returns a fortran record using the first occurrence of a
30 REM carriage return as a record delimiter, subsequent carriage returns
40 REM and linefeeds are ignored.
50 CLS: KEY OFF: SCREEN 0,1,0,0: COLOR 14,2,9
60 LOCATE 25,1: PRINT"Program FTNREC.BAS; G. Jansen van Beek; 30 July, 1987";
70 LOCATE 1,1: CR$=CHR$(13): LF$=CHR$(10): IFLG=0
80 INPUT"Enter name of input file ----> ";INNAME$
90 OPEN"I",#1,INNAME$
100 INPUT"Enter name of output file ----> ";OUTNAME$
110 OPEN"O",#2,OUTNAME$
120 RECS=""
130 IF EOF(1) GOTO 220
140 A$ = INPUT$(1,#1)
150 IF A$=CR$ AND IFLG=0 THEN GOTO 190 ELSE IF A$=CR$ THEN GOTO 130
160 IF A$=LF$ GOTO 130
170 RECS=RECS+A$: IFLG=0
180 GOTO 130
190 PRINT RECS: PRINT #2,RECS: IFLG=1
200 GOTO 120
210 CLOSE #1,#2: KEY ON
220 PRINT "End of job."
230 PRINT "Input file name was ----> ";INNAME$
240 PRINT "Output file name is ----> ";OUTNAME$
250 KEY ON
260 END
```

APPENDIX 5
REFORMG.FOR

PROGRAM REFORMG

C

C PROGRAM REFORMATS DATA FROM THE GSM-18 PPM INTO A STANDARD WKF FILE

C REVISED 1989.194 BY L NEWITT

C

DIMENSION MON(12), IVAR(128), ITEMP(144), JVAR(128),ICOMP(8)
DIMENSION IHEAD(128), JTEMP(750), TEMP(16)
CHARACTER*80 INFILE,OUTFILE
DATA MON/31,28,31,30,31,30,31,31,30,31,30,31/
DATA ICOMP/0,0,0,0,0,1,0,0/
INFILE = '
OUTFILE = '

C

C THIS IS THE TEMPERATURE SAMPLING RATE IS SECONDS

C

ITMSCL=600
LREC=1

C

PRINT 1
1 FORMAT(1H1,'THIS PROGRAM CONVERTS DATA FROM THE GSM-18',//,
*'INTO A WKF FILE')

C

C THE NAME OF THE INPUT FILE IS REQUESTED

C

PRINT 2
2 FORMAT(1H0,'ENTER NAME OF INPUT FILE....')
READ(*,3)INFILE
3 FORMAT(A)
OPEN(UNIT=1, FILE=INFILE, STATUS='OLD')

C

C THE NAME OF THE OUTPUT FILE IS REQUESTED

C

PRINT 4
4 FORMAT(1H0,'ENTER NAME OF OUTPUT FILE....')
READ(*,3)OUTFILE
OPEN(UNIT=2, FILE=OUTFILE, STATUS='NEW', ACCESS='SEQUENTIAL',
*FORM='FORMATTED', RECL=1024)

C

C ENTER THE IAGA CODE FOR THE REPEAT STATION

C

PRINT 15
15 FORMAT(1H0,'ENTER IAGA CODE FOR STATION--->')
READ(*,*)IOBS

C

C THIS IS A SCRATCH FILE USED FOR TEMPORARY STORAGE OF DATA

C

OPEN(UNIT=3, STATUS='SCRATCH', FORM='UNFORMATTED', ACCESS='DIRECT'
*,RECL=512)

C

C READ IN THE YEAR

```

C
PRINT 5
5 FORMAT(1H0,'ENTER YEAR....')
READ(*,*) JYR
C
C THIS IS THE SCALE FACTOR AND INSTRUMENT ID
C
ISCALE=10
SCALE=ISCALE
INSTID = 41227
PRINT 11, INSTID
11 FORMAT(1H0,'INSTRUMENT SERIAL NUMBER IS',I8)
C
C READ DATA FROM INPUT FILE
C
100 N=1
ISW=0
NREC=1
IEND=0
101 READ (1,7,END=999,ERR=999)IMON,1DAY,IHR,IMIN,ISEC,F
7 FORMAT(17X,2I2,I3,2I2,F8.1)
print 7, imon,iday,ihr,imin,iseq,f
C
C DETERMINE THE START DAY AND TIME FOR INSERTION IN THE HEADER RECORD
C WHICH IS WRITTEN AT THE START OF EACH DAY'S DATA
C
IF(N.NE.1)GO TO 102
IF(ISW.NE.0)GO TO 102
JDAY=1DAY
JMON=IMON
JSEQ=0
KMON=IMON-1
DO 301 K=1,KMON
301 JSEQ=JSEQ+MON(K)
JSEQ=JSEQ+JDAY
JSTHR=IHR
JSTMIN=IMIN
JSTSEC=ISEC
ISW=1
C
C DETERMINE THE TIME INCREMENT BY SUBTRACTING THE TIME BETWEEN THE FIRST
C TWO RECORDS
C
102 IF(N.NE.2)GO TO 103
IF(ISW.NE.1)GO TO 103
IT1=JSTHR*60*60+JSTMIN*60+JSTSEC
IT2=IHR*60*60+IMIN*60+ISEC
INT=IT2-IT1
ISW=2
103 CONTINUE
C

```

```

C STORE F VALUES READ IN IN AN ARRAY
C VALUES ARE IN TENTHS OF NANOTESLAS
C
IF(N.EQ.1)GO TO 104
ITM2=IHR*60*60+IMIN*60+ISEC
IF(IHR.LT.LHR.AND.IMIN.EQ.0)ITM2=ITM2+24*60*60
ITM1=LHR*60*60+LMIN*60+LSEC
IDIF=ITM2-ITM1
IRAT=IDIF/INT
IF(IRAT.EQ.1)GO TO 104
IF(IRAT.EQ.0)GO TO 998
IRAT=IRAT-1
N=N+IRAT
DO 307 J=1,IRAT
C FILL IN MISSING VALUES WITH 888888'S
307 IVAR(N-J)=888888
104 IVAR(N)=(F+.5/SCALE)*SCALE
print *, n,ivar(n)
IF(N.GE.128)GO TO 200
IF(IDAY.GT.JDAY.AND.IMIN.EQ.0.AND.ISEC.EQ.0)GO TO 210
N=N+1
LHR=IHR
LMIN=IMIN
LSEC=ISEC
GO TO 101
C
C IF 128 VALUES HAVE BEEN READ IN, WRITE THEM ON A SCRATCH FILE
C REINITIALIZE COUNTER AND CONTINUE READING
C THIS WILL BE ONE RECORD OF DATA ON OUTPUT FILE
C
200 WRITE(3,REC=NREC)(IVAR(I),I=1,128)
print *, nrec
N=1
NREC=NREC+1
GO TO 101
C
C WHEN THE END OF A DAY HAS BEEN REACHED, WRITE DATA ONTO SCRATCH FILE
C THEN WRITE HEADER RECORD ONTO OUTPUT FILE, AND COPY ALL DATA RECORDS
C FROM SCRATCH FILE TO OUTPUT FILE
C
C IOBS - OBSERVATORY CODE
C INSTID - INSTRUMENT SERIAL NUMBER OR IDENTIFIER
C JDAY - DAY
C JMON - MONTH
C JYR - YEAR
C JSEQ - SEQUENTIAL DAY OF THE YEAR
C JSTHR - HOUR AT WHICH DATA STARTS
C JSTMIN - MINUTE AT WHICH DATA STARTS
C JSTSEC - SECOND AT WHICH DATA STARTS
C LENDHR - HOUR AT WHICH DATA STOPS
C LENDMIN - MINUTE AT WHICH DATA STOPS
C LENDSEC - SECOND AT WHICH DATA STOPS

```

```

C INT - SAMPLING INTERVAL IN SECONDS
C ICOMP - INDICATES WHICH COMPONENTS ARE RECORDED
C PRESENCE OF COMPONENT INDICATED BY 1, ABSENCE BY 0
C ICOMP(1) - X
C ICOMP(2) - Y
C ICOMP(3) - Z
C ICOMP(4) - D
C ICOMP(5) - H
C ICOMP(6) - F
C ICOMP(7) - I
C ICOMP(8) - OTHER
C ISCALE - THE SCALE FACTOR OF THE MAGNETIC DATA
C ITMSCL THE TIME INCREMENT OF THE TEMPERATURE DATA
210 CONTINUE
NP=N+1
DO 230 K=NP,128
230 IVAR(K)=999999
WRITE(3,REC=NREC)(IVAR(I),I=1,128)
NREC=NREC+1
LENDHR=IHR
LENDMIN=IMIN
LENDSEC=ISEC
IF(IHR.EQ.0.AND.IMIN.EQ.0.AND.ISEC.EQ.0)LENDHR=24
NREC=NREC-1
DO 280 KK=1,128
280 IHEAD(KK)=999999
WRITE(2,701)IOBS,INSTID,JDAY,JMON,JYR,JSEQ,JSTHR,JSTMIN,
1JSTSEC,LENDHR,LENDMIN,LENDSEC,INT,(ICOMP(I),I=1,8),ISCALE,ITMSCL,
2(IHEAD(JJ),JJ=24,128)
701 FORMAT(128I8)
PRINT 704, JDAY,JMON,JYR,JSTHR,JSTMIN,JSTSEC,LENDHR,LENDMIN,
*LENDSEC
704 FORMAT(1H0,'INFORMATION WRITTEN IS FOR DAY',3I5,/,1H ,
1'FIRST TIME WRITTEN IS',3I3,/,1H , 'LAST TIME WRITTEN IS',3I3)
LREC=LREC+1
C
C WRITE DUMMY TEMPERATURE RECORDS ON THE FILE IMMEDIATELY FOLLOWING
C THE HEADER RECORD
C
110 CONTINUE
DO 281 KK=1,256
281 JTEMP(KK)=999999
WRITE(2,701)(JTEMP(I),I=1,128)
LREC=LREC+1
WRITE(2,701)(JTEMP(I),I=129,256)
282 LREC=LREC+1
C
C WRITE THE MAGNETIC DATA FOR THE DAY ONTO THE OUTPUT FILE
C
DO 220 L=1,NREC
READ(3,REC=L)(JVAR(I),I=1,128)
WRITE(2,701)(JVAR(I),I=1,128)

```

```
LREC=LREC+1
220  CONTINUE
      CLOSE(UNIT=3, STATUS='DELETE')
      OPEN(UNIT=3, STATUS='SCRATCH', ACCESS='DIRECT', FORM='UNFORMATTED'
*,RECL=512)
      IF(IEND.EQ.99)STOP
      GO TO 100
998  PRINT 19, IRAT,ICNT,IHR,IMIN,LHR,LMIN
19   FORMAT(1H0,'*****WARNING***',//,1H ,'INCREMENT PROBLEM',//,
*1H ,6I8)
999  IEND=99
      GO TO 210
      END
```

APPENDIX 6

REBLCKF.FOR

C PROGRAM REBLOCK
C
C VERSION 1987.357 G. JANSEN VAN BEEK
C MODIFIED 1989.194 BY L. NEWITT
C
C PROGRAM TAKES A *.WKF FILE CONTAINING 1 MINUTE DATA
C AND ENSURES THAT THE FIRST AND LAST
C DAYS CONTAIN 1440 VALUES FOR EACH COMPONENT.
C PREFIX AND SUFFIX VALUES HAVE BEEN SET TO 999999.
C START AND END TIMES FOR EACH DAY HAVE BEEN SET TO 0001 AND 2400.
C
CHARACTER*20 TMP1,TMP2,FILE1,FILE2
COMMON ID(128),ITMP1(128),ITMP2(128)
DIMENSION IADAT(1536)
DIMENSION ID1(128)
C
C OPEN THE PROGRAM/FILES
C
WRITE(*,900)
900 FORMAT(1H1,'This program takes a *.WKF file and ensures that the',
*' first and last day',//,' contain 1440 values for each component.',
//,' Prefix and suffix values have been set to 999999. Start and ',
*'times for each day have',//,' been set to 00 01 and 24 00.')
WRITE(*,902)
902 FORMAT(1H\$,'Input data file name -----> ')
READ(*,800) FILE1
800 FORMAT(A)
OPEN(1,FILE=FILE1,STATUS='OLD',IOSTAT=IOS)
IF(IOS.EQ.29)STOP '*** INPUT FILE NOT FOUND ***'
WRITE(*,904)
904 FORMAT(1H\$,'Output data file name -----> ')
READ(*,800) FILE2
OPEN(2,FILE=FILE2,STATUS='NEW',ACCESS='SEQUENTIAL',RECL=1024)
C
C SET UP THE WORKING LOOP
C
10 READ(1,810,END=700)(ID(I),I=1,128)
810 FORMAT(128I8)
READ(1,810,IOSTAT=IOS)(ITMP1(I),I=1,128)
IF(IOS.NE.0)STOP 'ABNORMAL STOP WHEN READING ITMP1'
READ(1,810,IOSTAT=IOS)(ITMP2(I),I=1,128)
IF(IOS.NE.0)STOP 'ABNORMAL STOP WHEN READING ITMP2'
CALL DEFINE(NC,NREC)
PRINT *, NC,NREC
WRITE(*,906)(ID(I),I=1,13)
DO 220 JJ=1,128
220 ID1(JJ)=ID(JJ)
ID1(7)=00
ID1(8)=01
ID1(9)=00
ID1(10)=24
ID1(11)=00

```
ID1(12)=00
WRITE(2,810)(ID1(I),I=1,128)
WRITE(2,810)(ITMP1(I),I=1,128)
WRITE(2,810)(ITMP2(I),I=1,128)
906 FORMAT(1H , 'Process at day block ',13I8)
C
DO 40 L=1,NC
DO 20 I=1,1536
IADAT(I)=999999
20 CONTINUE
CALL FILARA(NREC,IOS,IADAT)
IF(IO.NE.0)STOP 'ABNORMAL STOP WHILE READING COMP. DATA 1'
CALL OUTPT(IADAT)
40 CONTINUE
GOTO 10
C
C      END THE PROGRAM
C
700 WRITE(*,908)
908 FORMAT(1HO,'*** NORMAL END OF PROGRAM ***')
END
SUBROUTINE OUTPT(IT)
C
C      SUBROUTINE TO OUTPUT THE COMPONENT DATA
C
COMMON ID(128),ITMP1(128),ITMP2(128)
DIMENSION IT(1536)
C
DO 20 J=1,1440,128
IEND=J+127
IF(IEND.GT.8640) THEN
    DO 10 IXX=1441,IEND
    IT(IXX)=999999
10 CONTINUE
ENDIF
WRITE(2,800)(IT(I),I=J,IEND)
800 FORMAT(128I8)
20 CONTINUE
C
RETURN
END
SUBROUTINE DEFINE(NC,NREC)
C
C      G. JANSEN VAN BEEK 1987.335
C
COMMON ID(128),ITMP1(128),ITMP2(128)
C
SUBROUTINE DEFINES THE NUMBER OF COMPONENTS AND THE NUMBER
C      OF RECORDS PER COMPONENT THAT EXIST FOR THE DATA TIME INTERVAL
C      FOUND IN THE DATA HEADER ARRAY "ID"
C
C      CALCULATE THE NUMBER OF COMPONENTS PRESENT
```

C
C
NC=0
DO 10 I=14,20
IF(ID(I).NE.0)NC=NC + 1
10 CONTINUE
C
C CALCULATE THE NUMBER OF RECORDS PER COMPONENT
C
MMDAY=ID(7)*60 + ID(8)*1 + ID(9)/60
NMDAY=ID(10)*60 + ID(11)*1 + ID(12)/60
MNDIF=NMDAY-MMDAY + 1
NREC=MNDIF/128.
IF(MOD(MNDIF,128).NE.0) NREC=NREC+1
RETURN
END
SUBROUTINE FILARA(NREC,IOS,IN)
C
C G. JANSEN VAN BEEK 1987.335
C
COMMON ID(128),ITMP1(128),ITMP2(128)
DIMENSION NN(128),IN(1536)
C
C SUBROUTINE READS THE DATA FOR ONE COMPONENT DAY INTO THE 8640 DATA
C POINT ARRAY "IN"
C
LSTRT=ID(7)*60 + ID(8)*1 + ID(9)/60
NSTRT=LSTRT
LSTP=ID(10)*60 + ID(11)*1 + ID(12)/60
DO 20 K=1,NREC
READ(1,900,IOSTAT=IOS) (NN(I),I=1,128)
IF (IOS.NE.0) RETURN
900 FORMAT(128I8)
M=1
DO 10 L=NSTRT,NSTRT+127
IN(L)=NN(M)
M=M + 1
10 CONTINUE
NSTRT=NSTRT + 128
20 CONTINUE
C
RETURN
END

APPENDIX 7

FMERGE.FOR

```
PROGRAM FMERGE
C
C PROGRAM COMBINES 3-COMPONENT XYZ DATA FROM THE RING-CORE MAG AND F DATA
C FROM THE GSM-18 INTO A SINGLE WKF TYPE FILE
C INPUT DATA ARE WKF FILES
C FILES MUST CONTAIN FULL DAYS DATA; IF NOT, RUN REBLOCK AND REBLCKF FIRST
C
C PROGRAMMED BY L NEWITT, 1989.194
C
DIMENSION IVAR(128),JVAR(128)
CHARACTER*80 INPTXYZ, INPUTF, OUTPUT
PRINT 101
101 FORMAT(1H0,'ENTER NAME OF 3-COMP INPUT FILE--->')
READ(*,112)INPTXYZ
112 FORMAT(A)
OPEN(UNIT=1,FILE=INPTXYZ,STATUS='OLD')
PRINT 102
102 FORMAT(1H0,'ENTER NAME OF F INPUT FILE--->')
READ(*,112)INPUTF
OPEN(UNIT=2,FILE=INPUTF,STATUS='OLD')
PRINT 103
103 FORMAT(1H0,'ENTER NAME OF OUTPUT FILE--->')
READ(*,112)OUTPUT
OPEN(UNIT=3,FILE=OUTPUT,STATUS='NEW',RECL=1024)
C
IEND=0
1000 CONTINUE
C
C READ 3-COMPONENT HEADER RECORD
C
READ(1,104,END=999)ID1,ID2,IDAD,IMON,IYR,ISEQ,IHRS,IMINS,ISECS,
*IHRF,IMINF,ISECF,ISMP,(IVAR(I),I=14,128)
104 FORMAT(128I8)
C
C READ F HEADER RECORD AND COMPARE WITH 3-COMPONENT RECORD
C
READ(2,104,END=998)JD1,JD2,JDAY,JMON,JYR,JSEQ,JHRS,JMINS,JSECS,
*JHRF,JMINF,JSECF,JSMP,(JVAR(I),I=14,128)
IF(ID1.NE.JD1)THEN
PRINT 105
105 FORMAT(1H0,'STATION IDENTIFIERS DO NOT MATCH')
STOP
ENDIF
IF(IDAY.NE.JDAY.OR.IMON.NE.JMON.OR.ISEQ.NE.JSEQ)THEN
PRINT 106
106 FORMAT(1H0,'DAYS DO NOT MATCH')
PRINT 107, IDAY,IMON,ISEQ
PRINT 107, JDAY,JMON,JSEQ
107 FORMAT(1H ,3I8)
STOP
ENDIF
IF(IHRS.NE.JHRS.OR.IMINS.NE.JMINS.OR.ISECS.NE.JSECS)THEN
```

```
PRINT 108
108 FORMAT(1H0,'START TIMES DO NOT MATCH')
PRINT 107, IHRS,IMINS,ISECS
PRINT 107, JHRS,JMINS,JSECS
STOP
ENDIF
IF(IHRF.NE.JHRF.OR.JMINF.NE.IMINF.OR.ISECF.NE.JSECF)THEN
PRINT 109
109 FORMAT(1H0,'STOP TIMES DO NOT MATCH')
PRINT 107, IIMINF,ISECF
PRINT 107, JMINF,JSECF
STOP
ENDIF

C
C WRITE OUT NEW HEADER
C
1100 CONTINUE
IX=1
IY=1
IZ=1
ID=0
IH=0
IF=1
WRITE(3,104)ID1,ID2,IDAY,IMON,IYR,ISEQ,IHRS,IMINS,ISECS,
*IHRF,IMINF,ISECF,ISMP,IX,IY,IZ,ID,IH,IF,(IVAR(I),I=20,128)
PRINT 104, ID1, ID2, IDAY, IMON, ISEQ, IHRS, IMINS, ISECS,
*IHRF, IMINF, ISECF, ISMP, IX, IY, IZ, ID, IH, IF

C
C READ AND WRITE TWO TEMPERATURE RECORDS
C
READ(1,104)(IVAR(I),I=1,128)
IF(IEND.EQ.0) READ(2,104)(JVAR(I),I=1,128)
WRITE(3,104)(IVAR(I),I=1,128)
READ(1,104)(IVAR(I),I=1,128)
IF(IEND.EQ.0) READ(2,104)(JVAR(I),I=1,128)
WRITE(3,104)(IVAR(I),I=1,128)

C
C READ AND WRITE X DATA
C
DO 201 K=1,12
READ(1,104)(IVAR(I),I=1,128)
201 WRITE(3,104)(IVAR(I),I=1,128)

C
C READ AND WRITE Y DATA
C
DO 202 K=1,12
READ(1,104)(IVAR(I),I=1,128)
202 WRITE(3,104)(IVAR(I),I=1,128)

C
C READ AND WRITE Z DATA
DO 203 K=1,12
READ(1,104)(IVAR(I),I=1,128)
```

```
203  WRITE(3,104)(IVAR(I),I=1,128)
C
C  READ WRITE F DATA
C
IF(IEND.EQ.0)THEN
DO 204 K=1,12
READ(2,104)(JVAR(I),I=1,128)
204  WRITE(3,104)(JVAR(I),I=1,128)
ELSE
DO 205 L=1,128
205  JVAR(L)=999999
DO 206 K=1,12
206  WRITE(3,104)(JVAR(I),I=1,128)
ENDIF
C
C  LOOP BACK TO PROCESS NEXT DAY
C
      GO TO 1000
C
998  PRINT 110
110  FORMAT(1H0,'F DATA ENDS BEFORE 3-COMP DATA',//,
*'A DAY OF NULL F VALUES WILL BE ADDED')
      IEND=99
      GO TO 1100
999  CONTINUE
      CLOSE(UNIT=3)
      STOP
      END
```

APPENDIX 8

WKFPLT.FOR

```

CC      PROGRAM WKFPLT
C
C version for plotting to pc screen
C modified by L. Newitt, Sept, 1990
C compiled using MS-Fortran with graphics
C
INCLUDE 'FGRAFH.FI'
INCLUDE 'FGRAFH.FD'
CHARACTER*20 INFILE,NAME
CHARACTER*40 CHARS
DIMENSION BUF(1),IDAT(128),DAT(128),ICOM(8)
CHARACTER*1 ICOMP(8),COMP,BX
CHARACTER*2 STR1
RECORD /XYCOORD/ XY
RECORD /WXYCOORD/WXY
DATA ICOMP/'X','Y','Z','D','H','F','I','*'/
C
C PLOT 'GRAMS FROM *.WKF TYPE FILES AT VARIABLE MM/HR ON ZETA
C PLOTTER FOR DISPLAY PURPOSES
C
C VERSION 358.1987 G. JANSEN VAN BEEK
C
      WRITE(*,'(1HS,A)') 'Input file -----> '
      READ(*,880) INFILE
880 FORMAT(A)
      OPEN(15,FILE=INFILE,STATUS='OLD',ACCESS='SEQUENTIAL')
      WRITE(*,'(1HS,A)') 'Start day, end day and scale value -----> '
      READ(*,*) KDAY,KFDAY,XSCALE
      ITMSCL = 10
      IBX=0
      IDA = KDAY
      IF(XSCALE.EQ.0.0)XSCALE=20
      FSC = XSCALE
      FIN=0.0
      KOUNT=0
      XB=0.0
      NFILE=1
      YB=0.0
      IBOX=0
      SCALE=25.4*XSCALE
C
C KDAY=INITIAL STARTING DAY AFTERWARDS THE CURRENT DAY BEING
C PROCESSED
C KFDAY=FINAL DAY THAT IS REQUESTED TO BE PLOTTED
C IDA=START DAY THAT IS REQUESTED TO BE PLOTTED
C SMPRT=NUMBER OF DATA POINT PER HOUR
C ISMPRT=SAMPLE RATE IN SECONDS
C
10 read(15,800,IOSTAT=IOS) (IDAT(I),I=1,128)
   IF(IOS.LT.0) GOTO 999
   IF(IOS.GT.0) GOTO 16
   YB=0.0

```

```

C.....WRITE(*,'(1H0,13I8)')(IDAT(I),I=1,13)
   II=1
   DO 11 JJ=14,21
   ICOM(II)=IDAT(JJ)
   II=II+1
11 CONTINUE
   KOUNT = KOUNT + 1
   FYR = IDAT(5)
   IDAY = IDAT(6)
   DTSCl=IDAT(22)
   DIDENT = IDAT(2)
   SMPRT = 3600/IDAT(13)
   ISMPRT = IDAT(13)
1000 IBG = IDAT(7)*3600 + IDAT(8)*60 + IDAT(9)
   IEN = IDAT(10)*3600 + IDAT(11)*60 + IDAT(12)
   NUM = (IEN-IBG+IDAT(13))/IDAT(13)
C
C   IBG=time of first data point in seconds of the day
C   IEN=time of the last data point in seconds of the day
C   NUM=number of data points per component until the end of the day
C   IRR=number of records per component until the end of the day
C   NC=number of components in the data file
C   ISTRT=sequential number of the first data point of the day
C   DTSCl=data multiplication factor
C   NMAX=sequential number of the last data point of the day
C
C
800 format(128I8)
   NC = 0
   DO 1010 I=14,21
   IF(IDAT(I).NE.0) NC=NC+1
1010 CONTINUE
   READ(15,810,ERR=16,END=15) IBUF
810 FORMAT(I8)
   KOUNT = KOUNT + 1
   READ(15,810,ERR=16,END=15) IBUF
   KOUNT=KOUNT+1
   IRR = NUM/128
   N = MOD(NUM,128)
   IF(N.NE.0) IRR=IRR+1
   ISTRT = IBG/ISMPRT
   NMAX = IEN/ISMPRT
   GO TO 20
15 FIN=5.0
   GO TO 25
16 KOUNT = KOUNT + 1
   WRITE(*,5992)KOUNT
5992 FORMAT(1H0,'ERROR IN RECORD NO. ',I5)
   WRITE(*,'(1H0,13I8)')(IDAT(I),I=1,13)
   WRITE(*,'(1H ,13F7.0)') (DAT(IN),IN=1,13)
   WRITE(*,'(1H ,A,I10)')'READ ERROR NO. ',IOS
   GOTO 9999
20 CONTINUE

```

```

21 IF(IDAY.LT.KDAY) GO TO 250
   IF(IDAY.GT.KFDAY) GOTO 999
25 continue
   GOTO 257
250 DO 255 I=1,NC
   DO 255 J=1,IRR
   READ(15,810,ERR=16,END=15) IBUF
   KOUNT = KOUNT + 1
255 CONTINUE
   GOTO 10

C
C      YD=MEAN POSITION OF COMPONENT TRACE
C      XB,YB=PLOT ORIGIN OF EACH 'GRAM
C      X,Y=CURRENT PEN COORDINATES
C      ENTR=TOTAL LENGTH OF THE COMPONENT TRACE FOR A FULL DAY
C
C.....DATA OBTAINED FOR ONE FULL DAY
C*****ESTABLISH THE DISTANCE IN INCHES BETWEEN EACH DATA POINT
C*****FINC=THE PLOTTING DISTANCE/HOUR DIVIDED BY THE NO. OF SAMPLES/HOUR
257 FINC = ITMSCL/(SMPRT*25.4)
   ENTR = FINC*86400/ISMPRT
   CALL GRAPHICSMODE()
   CALL SETVIEWORG(10,10,XY)
   DUMMY = SETWINDOW(.TRUE.,-1.,-1.,11.5,10.)
C      CALL POS(ibox,XB,YB,IBX)
   X=XB
   Y=YB
   INCT=1.5
   YD=YB+INCT
   FDY=FLOAT(IDAY)
   DO 400 J=1,NC
   OPEN(UNIT=12,STATUS='SCRATCH')
C.....WORKING SPACE FOR EACH COMPONENT TRACE IS +/- 2 INCHES
   YMx=YD+2.5
   YMn=YD-2.5
   IF(YMn.LT.-1.0) YMn=-1.0
   IF(YMx.GT.10.) YMx=10.
   NAV=0
   TOT=0.0
C.....WRITE(*,'(1H0,10I8)') IBG,IEN,NUM,NMAX
C.....WRITE(*,'(1H0,10I8)') NC,IRR
   DO 50 I=1,IRR
C.....ASSUME THE DATA SCALE (IDAT(22)) IS SET TO 10
   READ(15,800,ERR=16,END=15) (IDAT(IN),IN=1,128)
820 FORMAT(128F8.1)
C.....WRITE(*,'(1H ,13F7.0)') (DAT(IN),IN=1,13)
   KOUNT = KOUNT + 1
   DO 50 II=1,128
   DAT(II)=IDAT(II)/DTSCl
   WRITE(12,840) DAT(II)
   IF(IDAT(II).GT.800000) GO TO 50
   NAV=NAV+1

```

```

TOT=TOT+DAT(II)
50 CONTINUE
REWIND 12
IF(NAV)65,65,53
53 FMN=TOT/NAV
X = XB
XS=X-0.2
LW=0
CALL SYMK(XS,YD,FMN,LW)
IT=0
DO 531 JJ=1,7
IF(ICOM(JJ).NE.0)IT=IT+1
IF(IT.EQ.J) GOTO 532
531 CONTINUE
JJ=8
532 COMP=ICOMP(JJ)
YDD=YD+0.2
CALL MOVETO_W(XS,YD,WXY)
CALL GETCURRENTPOSITION(XY)
CALL SETTEXTPOSITION(XY.YCOORD/8+2,XY.XCOORD/8,XY)
CALL OUTTEXT(COMP)
IF(NUM.EQ.NMAX) GOTO 530
X = XB + (ISTRRT)*FINC
GOTO 535
530 CONTINUE
535 READ(12,840,END=400) FV
840 FORMAT(F10.3)
55 IF(FV.GT.80000.) FV=FMN
Y=(FV-FMN)/SCALE + YD
IF(Y.GT.YMX) Y=YMX
IF(Y.LT.YMN) Y=YMN
Y1=Y
CALL MOVETO_W(X,Y,WXY)
READ(12,840,END=400)FV
DO 60 II=ISTRRT+1,NMAX
X=X+FINC
IF(FV.GT.80000.)THEN
CALL MOVETO_W(X,Y,WXY)
ELSE
Y=(FV-FMN)/SCALE+YD
IF(Y.LT.YMN) GO TO 58
IF(Y.GT.YMX) GO TO 59
GO TO 581
58 Y=YMN
GO TO 581
59 Y=YMX
581 CONTINUE
DUMMY=LINETOW(X,Y)
ENDIF
Y1=Y
IF(II.LT.NMAX+1)READ(12,840,END=400)FPLUS
IF(FPLUS.GT.80000.)CALL MOVETO_W(X,Y,WXY)

```

```

IF(FV.GT.80000..AND.FPLUS.LT.88000.) THEN
    XPLUS=X+FINC
    FPLUS1=(FPLUS-FMN)/SCALE + YD
    CALL MOVETO_W(XPLUS,FPLUS1,WXY)
ENDIF
FV=FPLUS
60 CONTINUE
XS=X+0.2
IF(NMAX.NE.86400/ISMPRT) XS=XB+ENTR+0.2
LW=1
CALL SYMK(XS,YD,FMN,LW)
YDD=YD+0.2
CALL MOVETO_W(XS,YDD,WXY)
CALL GETCURRENTPOSITION(XY)
CALL SETTEXTPOSITION(XY.YCOORD/8+2,XY.XCOORD/8+5,XY)
CALL OUTTEXT(COMP)
GO TO 69
65 X=XB + NMAX*FINC/2.0
chars = 'NO DATA'
C
X=XB
69 XNX=2.25
IF(IBX.EQ.1) XNX=1.0
YD=YD+XNX
CLOSE (12,STATUS='DELETE')
400 CONTINUE
C.....INITIALIZE ARRAY
Y=YB+0.4
X = XB
CALL MOVETO_W(X,Y,WXY)
C.....WRITE(*,'(1H ,A)')'Start to plot the hour tiks.'
DO 450 J=1,24
CALL TIKS(X,Y,0.1,0.0)
IF(MOD(J-1,6).NE.0) GO TO 452
FJ = FLOAT(J-1)
IF(FJ.EQ.24.OR.FJ.EQ.0) GO TO 452
CALL MOVETO_W(X,Y-0.2,WXY)
CALL GETCURRENTPOSITION(XY)
CALL SETTEXTPOSITION(XY.YCOORD/8+3,XY.XCOORD/8+2,XY)
IFJ=FJ
WRITE(STR1,'(I2)')IFJ
CALL OUTTEXT(STR1)
CALL MOVETO_W(X,Y,WXY)
452 CONTINUE
X=X+FINC*SMPRT
DUMMY = LINETO_W(X,Y)
450 CONTINUE
C.... WRITE(*,'(1H A)')'End of hour tik plotting.'
KDAY=IDAY
IF(FIN.NE.0.0) GO TO 999
IBOX=IBOX+1
CALL SETTEXTPOSITION(26,5,XY)

```

```

PRINT 151, IDAY
151 FORMAT(1H0,' DAY',I4,' PLOTTED, PRESS ANY KEY TO CONTINUE--->')
READ(*,'(A)')ICONT
CALL ENDGRAPH()
GO TO 10
999 CONTINUE
X = 1
9999 CONTINUE
CLOSE(12,STATUS='DELETE')
WRITE(*,'(1H ,A)') 'Plot file name is "WKFPLT.PLT".'
CALL ENDGRAPH()
END
SUBROUTINE POS(IBX,X,Y,IB)
IF(IBX.LT.1) RETURN
XINC=1.0
IF(IB.EQ.0) GOTO 10
IF(MOD(IBX,2).NE.0) GOTO 50
10 X=XINC
Y=0.0
RETURN
50 Y=5.0
IF(IBX.EQ.1)XINC=1.5
X=XINC
RETURN
END
SUBROUTINE TIKS(XM,YM,HI,ANG)
INCLUDE 'FGRAPH.FD'
RECORD /WXYCOORD/WXY
RECORD /XYCOORD/XY
X=XM
Y=YM
IF(X.NE.999..OR.Y.NE.999.) GO TO 40
CALL GETCURRENTPOSITION(XY)
40 CONTINUE
X1=X
Y1=Y+HI
Y2=Y-HI
IF(ABS(ANG).NE.90.) GO TO 50
S=SIGN(1.,ANG)
X1=X+HI*S
Y1=Y
50 CALL MOVETO_W(X1,Y1,WXY)
DUMMY = LINETO_W(X1,Y2)
CALL MOVETO_W(X,Y,WXY)
RETURN
END
SUBROUTINE SYMK(XT,YT,FMN,KEY)
INCLUDE 'FGRAPH.FD'
RECORD /WXYCOORD/ WXY
RECORD /XYCOORD/POSITION
CHARACTER*5, STR
CALL MOVETO_W(XT,YT,WXY)

```

```
DUMMY = LINETO_W(XT+0.1,YT)
DUMMY = LINETO_W(XT-0.1,YT)
DUMMY = LINETO_W(XT,YT)
DUMMY = LINETO_W(XT,YT+0.1)
DUMMY = LINETO_W(XT,YT-0.1)
DUMMY = LINETO_W(XT,YT)
YS=YT+0.1
IF(KEY.LT.1) RETURN
CALL MOVETO_W(XT+0.2,YS,WXY)
CALL GETCURRENTPOSITION(POSITION)
CALL SETTEXTPOSITION(POSITION.YCOORD/8+3,POSITION.XCOORD/8+5,
*POSITION)
IFMN =FMN
WRITE(STR,'(I5)')IFMN
CALL OUTTEXT(STR)
RETURN
END
SUBROUTINE GRAPHICSMODE()
INCLUDE 'FGRAPH.FD'
RECORD /VIDEOCONFIG/ VDINFO
CALL GETVIDEOCONFIG(VDINFO)
DUMMY = SETVIDEOMODE(6)
RETURN
END
SUBROUTINE ENDGRAPH()
INCLUDE 'FGRAPH.FD'
DUMMY = SETVIDEOMODE( $DEFAULTMODE)
RETURN
END
```

APPENDIX 9
PRETRANS.FOR

PROGRAM PRETRANS

C

C CONVERTS WKF RECORDS TO SHORT 8 BYTE RECORDS BEFORE TRANSFERRING TO

C MICROVAX USING DECNET

C

DIMENSION INDAT(128)

CHARACTER*80 INFILE, OUTFILE

PRINT 1

1 FORMAT(1H0,'ENTER NAME OF INUT FILE--->')

READ(*,2)INFILE

2 FORMAT(A)

PRINT 3

3 FORMAT(1H0,'ENTER NAME OF OUTPUT FILE--->')

READ(*,2)OUTFILE

OPEN(UNIT=1,FILE=INFILE,STATUS='OLD',RECL=1024)

OPEN(UNIT=2,FILE=OUTFILE,STATUS='NEW')

100 READ(1,4,END=999)(INDAT(I),I=1,128)

4 FORMAT(128I8)

DO 10 J=1,128

WRITE(2,5)INDAT(J)

5 FORMAT(I8)

10 CONTINUE

GO TO 100

999 STOP

END

APPENDIX 10
POSTTRANS.FOR

```
PROGRAM PSTRNS
C
C TRANSFORMS 8 BYTE RECORDS TRANSFERRED FROM THE 386 BY DECNET
C TO 1024 BYTE WKS FILES
C
C L NEWITT 1989.058
C
DIMENSION IDAT(128)
CHARACTER*80 INFILE, OUTFILE
PRINT 1
1 FORMAT(1H0,'ENTER NAME OF INPUT FILE--->')
READ(*,2)INFILE
2 FORMAT(A)
OPEN(UNIT=1,FILE=INFILE,STATUS='OLD')
PRINT 3
3 FORMAT(1H0,'ENTER NAME OF OUTPUT FILE--->')
READ(*,2)OUTFILE
OPEN(UNIT=2,FILE=OUTFILE,STATUS='NEW',RECL=1024)
I=1
100 READ(1,4,END=999)IVAL
4 FORMAT(I8)
IDAT(I)=IVAL
IF(I.EQ.128)GO TO 110
I=I+1
GO TO 100
110 CONTINUE
WRITE(2,5)(IDAT(I),I=1,128)
5 FORMAT(128I8)
I=1
GO TO 100
999 STOP
END
```

APPENDIX 11

ABSCHK.FOR

```
PROGRAM ABSCHK
DIMENSION IFAI(900),ITMD(900),DD(4),DM(4),XID(900),XIM(4),XIN(4)
DIMENSION ITMI(900),T1(4),DECL(900),D(900),T2(4),XINCL(900)
DIMENSION DUMM(13)
REAL*8 MIN
CHARACTER*1 ANS
CHARACTER*40 CHKIN,CHKOUT
INTEGER ERRCT

C
C** CHECKS ABSOLUTE READINGS AS SUBMITTED BY OPERATORS AT MAGNETIC
C OBSERVATORIES AND PRODUCES AN OUTPUT FILE WITH RECORDS
C SUFFICIENTLY SHORT SO THEY MAY BE HANDLED BY TXTEDIT ROUTINES
C
C REPEAT STATION VERSION WRITTEN FROM DATA GENERAL ONE
C BY L. NEWITT, 1989.195
C
C OPEN INPUT AND OUTPUT FILES
C
PRINT 9599
9599 FORMAT(1H0,'ENTER NAME OF INPUT FILE ----->')
READ(*,9600) CHKIN
9600 FORMAT(A)
OPEN(UNIT=12,FILE=CHKIN,STATUS='OLD')
PRINT 9602
9602 FORMAT(1H0,'ENTER NAME OF OUTPUT FILE ----->')
READ(*,9600) CHKOUT
9601 FORMAT(1H0,A)
OPEN(UNIT=13,FILE=CHKOUT,STATUS='NEW')
OPEN(UNIT=7,FILE='ABSDIAG.ERR',STATUS='NEW')
PRINT 18
18 FORMAT(1H0,'Error file is named ABSDIAG.ERR')
C
WRITE(7,9650)
9650 FORMAT(1H , 'ABSCHK')
WRITE(7,9651) CHKIN
9651 FORMAT(1H , 'INPUT FILE -----> ',A)
WRITE(7,9652) CHKOUT
9652 FORMAT(1H , 'OUTPUT FILE -----> ',A)
DERR=0.3/60.
XINCER=0.3/60.
PRINT 9700
9700 FORMAT(1H0,'DO YOU WANT THE STATISTICS? Y OR N')
READ (*,9701) ANS
9701 FORMAT(A)
DUM=0.
IDUM=0
ITEST = 0
ISTAR = 0
ERRCT = 0
PRINT 2
2 FORMAT(1H1)
C MAIN LOOP
```

```
K=0
NUMF=1
5    CONTINUE
K=K+1
C *** SPECIAL FIX ***
C INPUT ONE RECORD FOR ONE SET OF ABSOLUTES
READ(12,*,END=997,ERR=3500) CHK, YR,YRDA,DD(1),DM(1),DD(2),
* DM(2), TDEC,
* DD(4),DM(4),DD(3),DM(3),DCORR,XIDCOR,DECLD,DECLM,
*XIN(1),XIM(1),XIN(4),XIM(4), TINC,XIN(2),XIM(2),XIN(3),XIM(3),
* F,XINCOR,XINCLD,XINCLM
3002 CONTINUE
IF(K.GT.900) STOP 2
IPRERR=0
ISTN=CHK
ITINC=TINC
ITDEC=TDEC
IYR=YR
IF(ISTN.GT.90) GO TO 999
C
C SUBTRACT TRUE NORTH BEARING FROM READING AND THEN SET TRUE NORTH BEARING
C TO ZERO
C THIS IS NNECESSARY BECAUSE THE OUTPUT FORMAT CANNOT HANDLE LARGE VALUES
C
DO 111 J=1,4
TEMP=DD(J)+DM(J)/60.-DCORR/60.
IF(TEMP.LT.0.)TEMP=TEMP+360.
ID=TEMP
DD(J)=ID
DM(J)=(TEMP-ID)*60.
111 CONTINUE
DCORR=0.
C
C INPUT AND ADD F STATION DIFFERENCE CORRECTION TO EACH SET OF ABSOLUTES
C
IF(NUMF.EQ.1)THEN
PRINT 17,YRDA,TDEC
17  FORMAT(1H0,'Enter F station difference which is applicable',//,
*' for the next set of observations beginning on Day',F5.0,' time',
*F7.0,/,---->')
READ(*,*)FCORR
NUMF=-1
ENDIF
F=F+FCORR
6   FORMAT(1H ,2I2,F4.0,2(F4.0,F4.1),I4,2(F4.0,F4.1),2F4.1,F5.0,
* F4.1,2(F4.0,F4.1),I4,2(F4.0,F4.1),F8.1,F4.1,F4.0,F4.1)
WRITE(13,7) ISTN,IYR,YRDA,DD(1),DM(1),DD(2),DM(2),ITDEC,DD(4),
* DM(4),DD(3),DM(3),DCORR,XIDCOR,DECLD,DECLM,
* XIN(1),XIM(1),XIN(4),XIM(4),ITINC,XIN(2),XIM(2),XIN(3),XIM(3),
* F,XINCOR,XINCLD,XINCLM
7   FORMAT(2I2,F4.0,2(F4.0,F4.1),I4,2(F4.0,F4.1),2F4.1,F5.0,F4.1,
* 2(F4.0,F4.1),I4,2(F4.0,F4.1),F8.1,F4.1,F4.0,F4.1)
```

```
IF(ISTN.NE.ISTAR) DATST = 0
IF(YRDA.GE.DATST) GO TO 5000
WRITE(7,121)
IPRERR=1
ERRCT = ERRCT+1
5000 IF(ITDEC.GT.ITEST) GO TO 5001
WRITE(7,121)
IPRERR=1
ERRCT = ERRCT+1
5001 IF(ITINC.GT.ITDEC) GO TO 5002
WRITE(7,121)
IPRERR=1
ERRCT = ERRCT+1
5002 CONTINUE
ITEST = ITINC
DATST = YRDA
ISTAR = ISTN
S=0
DO 100 J=1,4
C CONVERT D READINGS TO DEGREES
T1(J)=DD(J)+DM(J)/60.
S=S+T1(J)
100 CONTINUE
DECL(K)=DECLD + DECLM/60.
D(K)=S/4. +180. + XIDCOR/60. - DCORR/60.
IF(D(K).GE.360.) D(K)=D(K)-360.
DCHK = D(K) - DECL(K)
ITMP=IFIX(D(K))
DEGD=ITMP
DMIN=(D(K)-ITMP)*60.
IF(ABS(DCHK).LE.DERR) GO TO 5003
WRITE(7,121)
IPRERR=1
ERRCT = ERRCT+1
5003 CONTINUE
121 FORMAT(1H , 'ERROR',80(1H*))
S=0
DO 200 J=1,4
IF(T1(J).LT.180.) T1(J)=T1(J)+180.
S=S+T1(J)
200 CONTINUE
DMEAN=S/4.
S=0
DO 220 J=1,4
T1(J)=T1(J)-DMEAN
S=S+T1(J)*T1(J)
220 CONTINUE
RMSD=SQRT(S/4.)
C CONVERT I READINGS TO DEGREES
DO 300 J=1,4
T3=XIN(J)+XIM(J)/60.
IF(T3.LT.90.) T3=90.-T3
```

```
IF(T3.GT.270.) T3=T3-270.
IF(T3.GT.180.0.AND.T3.LT.270.) T3=270.-T3
IF(T3.GT.90.0.AND.T3.LT.180.) T3=T3-90.
300 T2(J)=T3
S=0
DO 500 J=1,4
500 S=S+T2(J)
XIMN=S/4.
XINCL(K)=XINCLD+XINCLM/60.
XID(K)=90.-XIMN+XINCOR/60.
ITMP=IFIX(XID(K))
XIDEG=ITMP
XIMIN=(XID(K)-ITMP)*60.
XICHK=XID(K)-XINCL(K)
IF(ABS(XICHK).LE.XINCER) GO TO 5004
WRITE(7,121)
IPRERR=1
ERRCT = ERRCT+1
5004 CONTINUE
S=0
DO 550 J=1,4
T2(J)=T2(J)-XIMN
S=S+T2(J)*T2(J)
550 CONTINUE
RMSI =SQRT(S/4.)
ITMD(K)=ITDEC
ITMI(K)=ITINC
IFAI(K)=F
IF(IPRERR.GT.0) THEN
    WRITE(7,6) ISTN,IYR,YRDA,DD(1),DM(1),DD(2),DM(2),ITDEC,
* DD(4),DM(4),DD(3),DM(3),DCORR,XIDCOR,DECLD,DECLM,
* XIN(1),XIM(1),XIN(4),XIM(4),ITINC,XIN(2),XIM(2),XIN(3),
* XIM(3),F,XINCOR,XINCLD,XINCLM
    WRITE(7,120) K,D(K),DEGD,DMIN,DECL(K),DECLD,DECLM
120  FORMAT(' DECLINATION FOR SET ',I3,' IS ',F10.3,F6.0,F4.1/
* 1H , 'DECLINATION CALCULATED IS      ',F10.3,F6.0,F4.1)
    IF(ANS.EQ.'Y'.OR.ANS.EQ.'Y') THEN
        WRITE(7,230) (T1(J),J=1,4)
230  FORMAT(1H ,49X,'SCATTERS IN DEG. FROM THE MEAN ARE ',4F10.3)
        WRITE(7,240) RMSD
240  FORMAT(1H ,49X,'RMS DEVIATION FROM MEAN IS ',F10.3)
        ENDIF
        WRITE(7,510) K,XID(K),XIDEG,XIMIN,XINCL(K),XINCLD,XINCLM
510  FORMAT(1H , 'INCLINATION FOR SET ',I3,' IS ',F10.3,F6.0,F4.1/
*1H , 'INCLINATION CALCULATED IS      ',F10.3,F6.0,F4.1)
        IF(ANS.EQ.'Y'.OR.ANS.EQ.'Y') THEN
            WRITE(7,560) (T2(J),J=1,4)
560  FORMAT(1H ,49X,'SCATTERS FROM THE MEAN ARE ',4F10.3)
            WRITE(7,570) RMSI
570  FORMAT(1H ,49X,'RMS DEVIATION FROM THE MEAN IS ',F10.3)
            ENDIF
            ENDIF
```

```
C      WRITE(7,6000)
6000 FORMAT(1H , '** END OF SET **'//)
      GO TO 5
999  CONTINUE
C      WRITE(7,7000)
7000 FORMAT(1H , '**END OF GROUP **'/1H1)
      ITEST = 0
      WRITE(13,7) ISTN, IDUM, DUM, DUM, DUM, DUM, DUM, IDUM, DUM, DUM,
* DUM, IDUM, DUM, DUM,
* DUM, DUM, DUM, DUM, DUM
900  IST=K-1
      NUMF=1
      GO TO 5
997  CONTINUE
      WRITE(7,8000) ERRCT
8000 FORMAT(1H0,'END OF RUN      ERROR COUNT IS',I10)
      CLOSE(UNIT=13)
      CLOSE(UNIT=6)
      STOP
3500 READ(12,3501,END=997) DUMM
3501 FORMAT(13A10)
      WRITE(7,3502) K
3502 FORMAT(1H0,'** RECORD READ ERROR, REC.',I5)
3600 WRITE(7,3601) DUMM
3601 FORMAT(1H ,13A10)
      WRITE(7,121)
      ERRCT = ERRCT+1
      WRITE(13,3602)K
3602 FORMAT(I10)
      GO TO 5
C*** SPECIAL READ
3700 READ(12,*) CHK, YR,YRDA,DD(1),DM(1),DD(2),DM(2), TDEC,DD(4),DM(4),
* DD(3),DM(3),EXTRA,DCORR,XIDCOR ,DECLD,DECLM,
*XIN(1),XIM(1),XIN(4),XIM(4), TINC,XIN(2),XIM(2),XIN(3),XIM(3),
*F,XINCOR,XINCLD,XINCLM
      GO TO 3002
      END
```

APPENDIX 12

ABSCONV.FOR

```
PROGRAM ABSCONV
DIMENSION DD(4),DM(4),XIM(4),XIN(4)
CHARACTER*40 CHKIN,CHKOUT
INTEGER ERRCT
LOGICAL EX

C
C      CONVERTS FORMAT OF ABSOLUTE OBSERVATIONS TO FORMAT COMPATIBLE
C      WITH DIFTREXB
C
C      CONVERTED TO DATA GENERAL ONE BY L. NEWITT, 1989.194
C
C      OPEN INPUT AND OUTPUT FILES
C
C      OPEN(UNIT=7,FILE='ABSCONV.ERR',STATUS='UNKNOWN')
C
PRINT 9599
9599 FORMAT(1H0,'ENTER NAME OF INPUT FILE ----->')
READ(*,9701) CHKIN
WRITE(7,9401) CHKIN
9401 FORMAT(1H0,'ABSCONV INPUT FILE -----> ',A)
OPEN(UNIT=12,FILE=CHKIN,STATUS='OLD')
PRINT 9602
9602 FORMAT(1H0,'ENTER NAME OF OUTPUT FILE ----->')
READ(*,9701) CHKOUT
OPEN(UNIT=13,FILE=CHKOUT,STATUS='NEW')
9701 FORMAT(A)
C
DUM=0.
IDUM=0
ITEST = 0
ISTAR = 0
ERRCT = 0
C      PRINT 2
2      FORMAT(1H1)
C      MAIN LOOP
      K=0
5      CONTINUE
      K=K+1
C
C      INPUT ONE RECORD FOR ONE SET OF ABSOLUTES
      READ(12,7,ERR=3500,END=997) ISTN,IYR,YRDA,DD(1),DM(1),DD(2),
      * DM(2),ITDEC,DD(4),DM(4),DD(3),DM(3),DCORR,XIDCOR,DECLD,DECLM,
      * XIN(1),XIM(1),XIN(4),XIM(4),ITINC,XIN(2),XIM(2),XIN(3),XIM(3),
      * F,XINCOR,XINCLD,XINCLM
      IPRERR=0
      IF(ISTN.GT.90) GO TO 999
      LAST=0
      IYRL=IYR
      YRDAL=YRDA
      ITDECL=ITDEC
      ISTNL=ISTN
C
```

```
6   FORMAT(1H ,2I2,F4.0,2(F4.0,F4.1),I4,2(F4.0,F4.1),2F4.1,F5.0,
* F4.1,2(F4.0,F4.1),I4,2(F4.0,F4.1),F8.1,F4.1,F4.0,F4.1)
C
      WRITE(13,7) ISTN,IYR,YRDA,DD(1),DM(1),DD(2),DM(2),ITDEC,DD(4),
* DM(4),DD(3),DM(3),DCORR,XIDCOR,DECLD,DECLM,
* XIN(1),XIM(1),XIN(4),XIM(4),ITINC,XIN(2),XIM(2),XIN(3),XIM(3),
* F,XINCOR,XINCLD,XINCLM, LAST
C
7   FORMAT(2I2,F4.0,2(F4.0,F4.1),I4,2(F4.0,F4.1),2F4.1,F5.0,F4.1,
* 2(F4.0,F4.1),I4,2(F4.0,F4.1),F8.1,F4.1,F4.0,F4.1,I3)
C
      IF(ISTN.NE.ISTAR) DATST = 0
      IF(YRDA.GE.DATST) GO TO 5000
      WRITE(7,121)
      IPRERR=1
      ERRCT = ERRCT+1
5000 IF(ITDEC.GT.ITEST) GO TO 5001
      WRITE(7,121)
      IPRERR=1
      ERRCT = ERRCT+1
5001 IF(ITINC.GT.ITDEC) GO TO 5002
      WRITE(7,121)
      IPRERR=1
      ERRCT = ERRCT+1
5002 CONTINUE
      ITEST = ITINC
      DATST = YRDA
      ISTAR = ISTN
121  FORMAT(1H ,'ERROR',80(1H*))
      IF(IPRERR.GT.0) THEN
          WRITE(7,6) ISTN,IYR,YRDA,DD(1),DM(1),DD(2),DM(2),ITDEC,
* DD(4),DM(4),DD(3),DM(3),DCORR,XIDCOR,DECLD,DECLM,
* XIN(1),XIM(1),XIN(4),XIM(4),ITINC,XIN(2),XIM(2),XIN(3),
* XIM(3),F,XINCOR,XINCLD,XINCLM
          ENDIF
      GO TO 5
999  CONTINUE
      ITEST = 0
      LAST=999
      ITDECPL=ITDECL+1
      WRITE(13,7) ISTNL,IYRL,YRDAL,DUM,DUM,DUM,DUM,ITDECPL,
* DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,
* DUM,DUM,DUM,DUM,1DUM,DUM,DUM,DUM,
* DUM,DUM,DUM,DUM,DUM,LAST
900  IST=K-1
      GO TO 5
997  CONTINUE
      WRITE(7,8000) ERRCT
8000 FORMAT(1H0,'END OF RUN      ERROR COUNT IS',I10)
      CLOSE(UNIT=7)
      CLOSE(UNIT=13)
      STOP
```

-A12- 4-

```
3500 READ(12,3501,END=997) DUMM
3501 FORMAT(13A10)
    WRITE(7,3502) K
3502 FORMAT(1H0,'** RECORD READ ERROR, REC.',I5)
3600 WRITE(7,3601) DUMM
3601 FORMAT(1H ,13A10)
    WRITE(7,121)
    ERRCT = ERRCT+1
3602 FORMAT(I10)
    GO TO 5
    END
```

APPENDIX 13

DIFTRE.FOR

PROGRAM DIFTRE

C
C Program originally written for CYBER by G.Jansen van Beek
C VERSION 1988.053 adapted for VAX
C & new input format by F.Plet
C

C MODIFIED 1987.342 G. JANSEN VAN BEEK
C DEBUGGED 1988.304 G. JANSEN VAN BEEK
C

C DATA GENERAL VERSION 1989.198 BY L. NEWITT
C

```
REAL*8 IFAI,DD,DM,XID,XIM,HD,HI,AX,AY,AZ
REAL*8 IXAD,IYAD,IXAI,IYAI,IZAD,IZAI
COMMON IN(128),IYAD(20),IXAD(20),IZAD(20)
COMMON IYAI(20),IXAI(20),IZAI(20),IFAI(20),IFT(20)
COMMON ITMD(20),DD(20),DM(20),XID(20),XIM(20),IDATM(20)
COMMON ITMI(20),IDAY,IHR,IMIN,IDAY1,IHR1,IMIN1,IDBTM(20)
COMMON HD(20),HI(20),AX(20),AY(20),AZ(20)
COMMON IND(1440),INH(1440),INZ(1440),INF(1440)
COMMON PREV, IEOF, ITIMER, NCMP
CHARACTER*80 PROUT,INTRE,INABS,OUTFILE,STN,TMP1,TMP2
REAL*8 SUM,SUMSQ,RAD,AVER
DIMENSION KDAY(20)
```

C
C THIS PROGRAM TAKES THE ABSOLUTE INFORMATION FROM THE ABSOLUTE
C OBSERVATION FORMS, APPLIES IT TO THE *.WKF FILES AND CALCULATES
C NEW BASELINES.

C
C THE PURPOSE OF THE PROGRAM IS TO CREATE AN OUTPUT FILE FILLED WITH
C ABSOLUTE INFORMATION AS APPLIED TO *.WKF FILES.
C

C THIS VERSION OF THE PROGRAM CALCULATES X,Y,Z BASELINES FOR ONE-MINUTE
C AMOS DATA ONLY.

C
C PROGRAM DOES PROCESS SPLIT DAYS

C
C IYAD AND IYAI REFER TO Y
C IXAD AND IXAI REFER TO X

C
C
C ---IYAD, IXAD, IZAD - COMPONENT VALUES AT TIME OF D
C ---IYAI, IXAI, IZAI - COMPONENT VALUES AT TIME OF I
C ---ITMD - TIME OF D
C ---ITMI - TIME OF I
C ---DD, DM - D READINGS IN DEGREES AND MINUTES
C ---XID,XID - I READINGS IN DEGREES AND MINUTES

C
C
C FILE7 - SAVED FILE CONTAINING THE PRINTED OUTPUT
C FILE1 - *.WKF FILE (ONE-MINUTE AMOS DATA INPUT FILE)
C FILE2 - ABSOLUTE OBSERVATION FILE

C FILE4 - BASELINE OUTPUT FILE
C
700 FORMAT(A)
RAD=180./3.1415926536
INDIC= 1
C !FLAG TO INDICATE A SPLIT DAY
IFO= 0
C !FIRST PASS FLAG
NUMB= 0
C !TIME OF D OR I OBS.
IEOF= 0
C !END FLAG FOR AMOS DATA FILE
PRINT 601
601 FORMAT(1H0,'ENTER NAME OF PRINTOUT FILE--->')
READ(*,700)PROUT
OPEN(UNIT=7,FILE=PROUT,STATUS='NEW')
PRINT 602
602 FORMAT(1H0,'ENTER NAME OF .WKF FILE--->')
READ(*,700)INTRE
301 FORMAT(1H1,'*.WKF FILE ----> ',A)
WRITE(7,301) INTRE
OPEN(UNIT=1,FILE=INTRE,STATUS='OLD')
PRINT 603
603 FORMAT(1H0,'ENTER NAME OF ABSOLUTE FILE--->')
READ(*,700)INABS
302 FORMAT(1H0,'ABSOLUTE FILE----> ',A)
WRITE(7,302) INABS
OPEN(UNIT=2,FILE=INABS,STATUS='OLD')
PRINT 604
604 FORMAT(1H0,'ENTER NAME OF OUTPUT BASELINE FILE--->')
READ(*,700)OUTFILE
303 FORMAT(1H0,'OUTPUT FILE ----> ',A)
WRITE(7,303) OUTFILE
OPEN(UNIT=4,FILE=OUTFILE,STATUS='NEW')
304 FORMAT(1H0,'LISTING FILE ----> ',A)
WRITE(7,304) PROUT
PRINT 605
605 FORMAT(1H0,'ENTER 2-DIGIT STATION CODE--->')
READ(*,*)NSTN
CALL GETTIM(JHR,JMIN,JSEC,JHSEC)
CALL GETDAT(JYR,JMON,JDAY)
WRITE(7,305)JHR,JMIN,JSEC,JYR,JMON,JDAY
305 FORMAT(1H0,10X,'Time of run - ',I4,2I3,'. Date of run - ',I4,2I3)
C
1 READ(2,750,END=900)ISTN
C !POSITION THE ABS. OBS. FILE AT
750 FORMAT(I2)
IF (ISTN.LT.NSTN) GOTO 1
IF (ISTN.NE.NSTN) THEN
 WRITE(7,846)
846 FORMAT(1H0,'***OBSERVATORY NOT FOUND IN ABSOLUTE',
 '* 'OBSERVATION FILE***')

```
GOTO 900
ENDIF
BACKSPACE 2
C
C
READ(1,800,END=910) (IN(I),I=1,128)
C !READ HEADER RECORD FROM
800 FORMAT (128I8)
NCOMP = IN(14)+IN(15)+IN(16)+IN(17)+IN(18)+IN(19)+IN(20)
C !THE AMOS DATA FILE
C IF(IN(2).NE.ISTN) THEN
C   WRITE(7,848)
C 848      FORMAT(1H0,'***WRONG AMOS DATA FILE***')
C GOTO 900
C ENDIF
DO 6 I=1,1440
IND(I)=999999
INH(I)=999999
INZ(I)=999999
INF(I)=999999
6 CONTINUE
8 IN3= IN(6)
C           !SEQ. DAY FROM AMOS DATA
GO TO 10
9 WRITE(7,12)
WRITE(7,11)
WRITE(7,12)
12 FORMAT (///)
11 FORMAT (1H , 'CHECK TIMES OF ABSOLUTE OBSERVATIONS')
10 ISTP = 1
C           !NO. OF ABSOLUTE OBS. IN THE SET
ITIMER= 1
DO 15 I=1,20
IFAI(I)= 999999
15 CONTINUE
DO 30 I=1,20
READ(2,850,END=900)ISTN,KDAY(I),ITMD(I),DD(I),DM(I),ITMI(I),
*IFAI(I),XID(I),XIM(I),LAST
850 FORMAT(I2,2X,I3,17X,I4,24X,F5.0,F4.1,16X,I4,16X,F8.1,4X,F4.0,
*F4.1,I3)
C           WRITE(7,850)ISTN,KDAY,ITMD(I),DD(I),DM(I),ITMI(I),
C *IFAI(I),XID(I),XIM(I),LAST
IF(LAST.NE.0) GOTO 40
IF (ISTN.GT.NSTN) THEN
  IF(I.GT.1) THEN
    WRITE(6,852)
852      FORMAT(1H0,'***MISSING END FLAG IN ABSOLUTE',
*          ' OBSERVATION SET***')
    GOTO 40
  ELSE
    GOTO 900
  ENDIF
```

```
ENDIF
IDATM(I) = KDAY(I)*10000+ITMD(I)
IDBTM(I) = KDAY(I)*10000+ITMI(I)
ISTP = ISTP + 1
30 CONTINUE
40 CALL TIMER(KDAY,ITMD(1),ISEQ)
ISTP=ISTP-1
IF(KDAY(1).LT.IN3.AND.IFO.EQ.0)GOTO 10
C !POSITION THE ABS. OBS. FILE
C
C
C A PRINT OF THE ABSOLUTE OBSERVATIONS HAS BEEN COMMENTED OUT
C TO SHORTEN THE PRINT FILE
C
C CALL OBSPR
C
IFO= 1
C
C ISEQ = 1 FOR ONE MINUTE BEFORE OBSERVATION
C ISEQ = 2 FOR ONE MINUTE AFTER OBSERVATION
C
C
C POSITION THE FILE
C
C
DO 110 J=1,ISTP
IF(J.GT.1) THEN
    IPREVDAY=KDAY(J-1)
ELSE
    IPREVDAY=0
ENDIF
IF(ITMI(J).LE.ITMD(J)) GOTO 9
60 IF(IN(6).LE.KDAY(J).AND.IEOF.EQ.0)THEN
C !POSITION THE AMOS DATA FILE
    CALL INPUT(KDAY(J))
C           !KDAY=SEQ. DAY FROM ABS.OBS.FILE
    GOTO 60
ENDIF
ISEQ = 1
IF (IPREVDAY.LT.KDAY(J)) THEN
    IPREV=0
ELSE
    IPREV=NUMB
ENDIF
NUMB = ITMD(J)
IF(IPREV.GE.NUMB) GOTO 9
70 CALL TIMER(KDAY(J),NUMB,ISEQ)
C !FIND FIRST MINUTE OF THE OBS.
IF (ITIMER.EQ.0) GO TO 9
C !FLAG FOR OBS. TIME ERROR
IDAY = IDAY1
C           !SEQ. DAY OF FIRST MINUTE OF OBS.
```

```
IHR = IHR1
C           !     HOUR   "   "   "   "
C           !     MIN.   "   "   "   "
100 ISEQ = 2
      CALL TIMER(KDAY(J),NUMB,ISEQ)
C     !FIND LAST MINUTE OF OBS.
      IF(IDAY1.GT.IDAY) THEN
          INDIC = 2
C     !IF START AND END MINUTE IN SEQ. DAYS
      ELSE
          INDIC = 1
C     !IF START AND END MINUTE IN SAME DAYS
      ENDIF
      CALL IDAT(INDIC,J,NSTN)
C     !FORM 3-MINUTE AMOS DATA AVERAGE
      IF(NUMB.EQ.ITMI(J)) GO TO 110
C IYAD IS Y VARIATION
C IXAD IS X VARIATION IN NT
      IYAD(J) = IYAI(J)
      IXAD(J) = IXAI(J)
      IZAD(J) = IZAI(J)
      NUMB = ITMI(J)
      ISEQ = 1
      GO TO 70
110 CONTINUE
C
      DO 120 I=1,ISTP
      DD(I) = DD(I)+DM(I)/60.
120 XID(I) = XID(I)+XIM(I)/60.
C
C     CALCULATE THE Z ABSOLUTES AND H ABSOLUTES AT TIME OF I
C
      DO 130 I=1,ISTP
      AZ(I) = IFAI(I)*DSIN(XID(I)/RAD)
      HI(I) = IFAI(I)*DCOS(XID(I)/RAD)
130 CONTINUE
C
C     CALCULATE THE CORRECTED H AT TIME OF D AND THE CORRESPONDING X AND
C     Y ABSOLUTE VALUES
C
      DO 140 I=1,ISTP
      CSD = DCOS(DD(I)/RAD)
      SND = DSIN(DD(I)/RAD)
      HD(I) = HI(I) + (IXAD(I)-IXAI(I))*CSD + (IYAD(I)-IYAI(I))*SND
      AX(I) = HD(I)*CSD
      AY(I) = HD(I)*SND
140 CONTINUE
C
C     PRINT THE ABSOLUTE VALUES
C
      WRITE(7,1040)ISTN,IN(5)
```

```
1040 FORMAT(1H0,'ABSOLUTE VALUES FOR STATION #',I3,',',I5)
      WRITE(7,1050)
1050 FORMAT(1H , 'TIME OF D',4X,'H(D)',4X,'X(D)',4X,'Y(D)',4X,'TIME OF I
      *',4X,'H(I)',4X,'Z(I)',4X,'F(I)',4X,'F*(I)')
      DO 150 I=1,ISTP
      WRITE(7,1060)IDATM(I),HD(I),AX(I),AY(I),IDBTM(I),HI(I),AZ(I)
      *,IFAI(I),IFT(I)
1060 FORMAT(1H ,I8,F9.1,2F8.1,4X,I7,F10.1,F8.1,F8.1,I9)
150 CONTINUE
      WRITE(7,1065)
1065 FORMAT(1H , 'F*(I) IS THE MEAN OF THREE F VALUES FOUND IN THE DATA
      * AT TIME OF I')
C
C      CALCULATE THE BASELINES
C
      DO 160 I=1,ISTP
      HVR=IYAD(I)
      DVR=IXAD(I)
      IF (HVR.GT.90000.OR.DVR.GT.90000) GOTO 1400
C THESE ARE THE BASELINES
      AX(I) = AX(I) - IXAD(I)
      AY(I) = AY(I) - IYAD(I)
      AZ(I) = AZ(I) - IZAI(I)
      GO TO 155
1400 AX(I)=999999.
      AY(I)=999999.
      AZ(I)=999999.
155 CONTINUE
160 CONTINUE
C
C      CALCULATE THE MEAN AND STANDARD DEVIATION
C
      IFLAG=0
      AVX=0
      AVY=0
      AVZ=0
      SXS=0
      SYS=0
      SZS=0
175 DO 280 K=1,3
      GO TO (180,200,220) K
180 DO 190 J=1,ISTP
      HI(J) = AY(J)
190 CONTINUE
      AV = AVY
      DE = SYS
      GO TO 240
200 DO 210 J=1,ISTP
      HI(J) = AX(J)
210 CONTINUE
      AV = AVX
      DE = SXS
```

```
GO TO 240
220 DO 230 J=1,ISTP
    HI(J) = AZ(J)
230 CONTINUE
    AV = AVZ
    DE = Szs
240 CONTINUE
    CALL AVDEV1(SUM,SUMSQ,N,DUM,DUM1,IDUM)
    UV = AV + DE
    BV = AV - DE
    DO 250 J=1,ISTP
        IF(IFLAG.NE.1) GO TO 245
        IF(HI(J).LT.UV.AND.HI(J).GT.BV)
        *CALL AVDEV2(SUM,SUMSQ,N,HI(J),DUM,1DUM)
        GO TO 250
245 CALL AVDEV2(SUM,SUMSQ,N,HI(J),DUM,1DUM)
250 CONTINUE
    CALL AVDEV3(SUM,SUMSQ,N,AVER,DEV,0)
    IF(K.NE.1)GO TO 260
    AVY = AVER
    SYS = DEV
    IF(IFLAG.EQ.1) GO TO 280
    AVY1 = AVER
    SYS1 = DEV
    GO TO 280
260 IF(K.NE.2) GO TO 270
    AVX = AVER
    SXS = DEV
    IF(IFLAG.EQ.1) GO TO 280
    AVX1 = AVER
    SXS1 = DEV
    GO TO 280
270 AVZ = AVER
    Szs = DEV
    IF(IFLAG.EQ.1) GO TO 280
    AVZ1 = AVER
    Szs1 = DEV
280 CONTINUE
    IF(IFLAG.EQ.1) GO TO 290
C
C      REMOVE THOSE VALUES WHICH DEVIATE FROM THE MEAN VALUE MORE THAN
C      ONE STANDARD DEVIATION
C
    IFLAG = 1
    GO TO 175
C
C      PRINT THE RESULTS
C
290 WRITE(7,1070)ISTN,IN(5)
1070 FORMAT(1H0,'BASELINES FOR #',I3,', ',I4,' WITH A MEAN FROM THOSE',
     */,10X,' VALUES WHICH VARIED LESS THAN ONE STANDARD DEVIATION FROM',
     */,10X,' THE ORIGINAL MEAN')
```

```
      WRITE(7,1080)
1080 FORMAT(1H , 'TIME    X BASELINE Y BASELINE Z BASELINE')
      DO 300 J=1,ISTP
      WRITE(7,1090)IDATM(J),AX(J),AY(J),AZ(J)
1090 FORMAT(1H , I7,3(F10.1,1X))
300 CONTINUE
      WRITE(7,2060)
2060 FORMAT(1H , 7X,3(10H-----,1X))
      WRITE(7,2040)AVX1,AVY1,AVZ1
2040 FORMAT(1H , 'MEANS ',3(F10.1,1X))
      WRITE(7,2050)SXS1,SYS1,SZS1
2050 FORMAT(1H , 'STD DV.',3(F10.1,1X))
      WRITE(7,2070)
2070 FORMAT(1H , 'MODIFIED')
      WRITE(7,2040)AVX,AVY,AVZ
      WRITE(7,2050)SXS,SYS,SZS
      WRITE(7,2030)
2030 FORMAT(1H0,130(1H*),//)
      WRITE(4,3000)IN(1),IDATM(1),IDBTM(ISTP),AVX1,SXS1,AVY1,SYS1,
      *AVZ1,SZS1,AVX,SXS,AVY,SYS,AVZ,SZS
3000 FORMAT(I6,2I10,12F7.0)
      IF(IEOF.EQ.1) GOTO 910
      NUMB = 0
      GO TO 10
C
C      END THE PROGRAM
C
900 WRITE(7,2010)NSTN
2010 FORMAT(1H0,'END OF ABSOLUTE INPUT DATA FOR STATION #',12)
      GO TO 920
910 WRITE(7,2020)
2020 FORMAT(1H0,'E-O-F FOUND IN AMOS INPUT FILE')
920 CONTINUE
      CLOSE(UNIT=6)
      END
      SUBROUTINE OBSPR
      REAL*8 IFAI,DD,DM,XID,XIM,HD,HI,AX,AY,AZ
      REAL*8 IXAD,IYAD,IXAI,IYAI,IZAD,IZAI
      COMMON IN(128),IYAD(20),IXAD(20),IZAD(20)
      COMMON IYAI(20),IXAI(20),IZAI(20),IFAI(20),IFT(20)
      COMMON ITMD(20),DD(20),DM(20),XID(20),XIM(20),IDATM(20)
      COMMON ITMI(20),IDAY,IHR,IMIN,IDADY1,IHR1,IMIN1,IDBTM(20)
      COMMON HD(20),HI(20),AX(20),AY(20),AZ(20)
      COMMON IND(1440),INH(1440),INZ(1440),INF(1440)
      COMMON PREV, IEOF, ITIMER, NCMP
      WRITE(7,1000)
1000 FORMAT(1H1,'THE ABSOLUTE OBSERVATIONS ARE AS FOLLOWS')
      WRITE(7,1010)
1010 FORMAT(1H0,'DAY/HR/MIN',7X,'D OBS.',6X,'DAY/HR/MIN',7X,'1 OBS.',6X
      *, 'TOTAL FORCE')
      DO 50 I=1,ISTP
      WRITE(7,1020)IDATM(I),DD(I),DM(I),IDBTM(I),XID(I),XIM(I),IFAI(I)
```

```
1020 FORMAT(1H ,2(I8.5X,F7.3,1X,F5.1,3X),F9.1)
50 CONTINUE
RETURN
END
SUBROUTINE TIMER(KDAY,NUMB,ISEQ)
REAL*8 IFAI,DD,DM,XID,XIM,HD,HI,AX,AY,AZ
REAL*8 IXAD,IYAD,IXAI,IYAI,IZAD,IZAI
COMMON IN(128),IYAD(20),IXAD(20),IZAD(20)
COMMON IYAI(20),IXAI(20),IZAI(20),IFAI(20),IFT(20)
COMMON ITMD(20),DD(20),DM(20),XID(20),XIM(20),IDATM(20)
COMMON ITMI(20),IDAY,IHR,IMIN,IDADY1,IHR1,IMIN1,IDBTM(20)
COMMON HD(20),HI(20),AX(20),AY(20),AZ(20)
COMMON IND(1440),INH(1440),INZ(1440),INF(1440)
COMMON PREV, IEOF, ITIMER, NCMP
C
C THIS SUBROUTINE CALCULATES THE START AND STOP TIME OF EACH
C ANGLE OBSERVATION
C
C
ITIMER= 1
IDAY1 = KDAY
IHR1 = NUMB/100
IMIN1 = NUMB-IHR1*100
IF ((IHR1.GT.23).OR.(IMIN1.GT.59)) GO TO 50
IF (IDAY1.LT.377.AND.IHR1.LT.24.AND.IMIN1.LT.60) GO TO 10
STOP 'PROGRAM ABORTED DUE TO TIME FIELD ERROR'
C
C CALCULATE TIME OF FIRST MINUTE
C
10 CONTINUE
IF(ISEQ.EQ.2) GO TO 20
IF(IMIN1.GT.1)IMIN1 = IMIN1 - 1
IF(IMIN1.GT.0) RETURN
IMIN1 = 60 + IMIN1
IHR1 = IHR1 - 1
IF(IHR1.GE.0) RETURN
IHR1= 24 + IHR1
IDAY1 = IDAY1 - 1
RETURN
C
C CALCULATE TIME OF LAST MINUTE
C
20 IMIN1 = IMIN1 + 1
IF(IMIN1.LT.61) RETURN
IMIN1 = IMIN1 - 60
IHR1 = IHR1 + 1
IF(IHR1.LT.24) RETURN
IHR1 = IHR1 - 24
IDAY1 = IDAY1 + 1
RETURN
50 ITIMER= 0
WRITE(7,60)
```

```
60 FORMAT (1H , 'INPUT HAS BAD FORMAT')
END
SUBROUTINE IDAT(INDIC,JJ,NSTN)
REAL*8 IFAI,DD,DM,XID,XIM,HD,HI,AX,AY,AZ
REAL*8 IXAD,IYAD,IXAI,IYAI,IZAD,IZAI
COMMON IN(128),IYAD(20),IXAD(20),IZAD(20)
COMMON IYAI(20),IXAI(20),IZAI(20),IFAI(20),IFT(20)
COMMON ITMD(20),DD(20),DM(20),XID(20),XIM(20),IDATM(20)
COMMON ITMI(20),IDAY,IHR,IMIN,IDAD1,IHR1,IMIN1,IDBTM(20)
COMMON HD(20),HI(20),AX(20),AY(20),AZ(20)
COMMON IND(1440),INH(1440),INZ(1440),INF(1440)
COMMON PREV, IEOF, ITIMER, NCOMP
C
C THIS SUBROUTINE GATHERS THE DATA AT TIME OF ANGLE MEASUREMENTS
C AND PLACES IT INTO THE COMPONENT ARRAYS
C
      Y=0
      X=0
      Z=0
      F=0
      NY=0
      NX=0
      NZ=0
      NF=0
      IFLAG = 1
      KDAY = IN(6)
      IF (INDIC.EQ.2) GO TO 150
      IEND = IMIN1 + IHR1*60
      IST = IMIN + IHR*60
      5 DO 30 I = IST,IEND
      IF(IND(I).GT.79999) GO TO 10
      Y = Y + IND(I)
      NY = NY + 1
      10 IF(INH(I).GT.79999) GO TO 20
      X = X + INH(I)
      NX = NX + 1
      20 IF(INZ(I).GT.79999) GO TO 25
      Z = Z + INZ(I)
      NZ = NZ + 1
      25 IF(INF(I).GT.79999) GO TO 30
      F = F + INF(I)
      NF = NF + 1
      30 CONTINUE
      IF(INDIC.EQ.2) GO TO 150
      35 IF(NZ.EQ.0) GO TO 40
      IZAI(JJ) = Z/NZ + 0.5001
      GO TO 44
      40 IZAI(JJ) = 999999.0
      44 IF(NF.EQ.0) GO TO 48
      IFT(JJ) = F/NF + 0.5001
      GO TO 50
      48 IFT(JJ) = 999999
```

```
50 CONTINUE
  IF(NY.EQ.0) GO TO 60
  IYAI(JJ) = Y/NY + SIGN(0.5001,Y)
  GO TO 70
60 IYAI(JJ) = 999999.0
70 IF(NX.EQ.0) GO TO 80
  IXAI(JJ) = X/NX + SIGN(0.5001,X)
  GO TO 90
80 IXAI(JJ) = 999999.0
90 CONTINUE
      RETURN

C
C      PROCESS THE SPLIT RECORD
C
150 GO TO (160,170,35) IFLAG
160 IST = IMIN
  IEND = 1440
  IFLAG = 2
  GO TO 5
170 CALL INPUT(KDAY)
  IST = 1
  IEND = IMIN1 + IHR1*60
  IDAY = IN(6)
  IDAY1= IN(6)
  IFLAG = 3
  GO TO 5
192 RETURN
      END
      SUBROUTINE AVDEV(SUM,SUMSQ,N,AVER,DEV,K)
      REAL*8 IFAI,DD,DM,XID,XIM,HD,HI,AX,AY,AZ
      REAL*8 IXAD,IYAD,IXAI,IYAI,IZAD,IZAI
      COMMON IN(128),IYAD(20),IXAD(20),IZAD(20)
      COMMON IYAI(20),IXAI(20),IZAI(20),IFAI(20),IFT(20)
      COMMON ITMD(20),DD(20),DM(20),XID(20),XIM(20),IDATM(20)
      COMMON ITMI(20),IDAY,IHR,IMIN,IDADY1,1HR1,IMIN1,IBDTM(20)
      COMMON HD(20),HI(20),AX(20),AY(20),AZ(20)
      COMMON IND(1440),INH(1440),INZ(1440),INF(1440)
      COMMON PREV, IEOF, ITIMER, NCOMP
      REAL*8 SUM, SUMSQ, AVER
      CALCULATE MEAN AND STANDARD DEVIATION.
C      INITIALIZE :
C          CALL AVDEV1 (SUM,SUMSQ,N,DUM,DUM1,1DUM)
C      COMPUTE SUMS, ETC :
C          CALL AVDEV2 (SUM,SUMSQ,N,DUM,DUM1,1DUM)
C      CALCULATE MEAN AVER AND STANDARD DEVIATION DEV (WITH N-K DEGREES F
C          CALL AVDEV3 (SUM,SUMSQ,N,AVER,DEV,K)
C      TO CALCULATE OVERALL MEAN, STANDARD DEVIATION :
C          CALL AVDEV1 (SUMT,SUMSQT,NT,DUM,DUM1,1DUM)
C          CALL AVDEV4 (SUMT,SUMSQT,NT,AVER,DEV,N)
C          CALL AVDEV3 (SUMT,SUMSQT,NT,AVERT,DEVT,KT)
ENTRY AVDEV1(SUM,SUMSQ,N,DUM,DUM1,1DUM)
SUM=0.
```

```
SUMSQ=0.
N=0
RETURN
ENTRY AVDEV2(SUM,SUMSQ,N,AVER,DUM,1DUM)
SUM=SUM+AVER
SUMSQ=SUMSQ+AVER*AVER
N=N+1
RETURN
ENTRY AVDEV3(SUM,SUMSQ,N,AVER,DEV,K)
IF (N.LE.K .OR. K.LT.0) GO TO 10
GO TO 20
10 AVER=SUM
DEV=SUMSQ
C   WRITE(7,15)      N,K,SUM,SUMSQ
C   15 FORMAT(3H N= I5, 5X 2HK= I5, 5X 4HSUM= E16.8, 5X 6HSUMSQ= E16.8,
C   *           10X 39HPROGRAM HAS SET AVER=SUM AND DEV=SUMSQ.)
      RETURN
20 FN=FLOAT(N)
AVER=SUM/FN
DEV=SQRT ((SUMSQ-FN*AVER*AVER)/FLOAT(N-K))
RETURN
ENTRY AVDEV4(SUM,SUMSQ,N,AVER,DEV,K)
SUM=SUM+AVER
SUMSQ=SUMSQ+DEV
N=N+K
RETURN
END
SUBROUTINE INPUT(KDAY)
REAL*8 IFAI,DD,DM,XID,XIM,HD,HI,AX,AY,AZ
REAL*8 IXAD,IYAD,IXAI,IYAI,IZAD,IZAI
COMMON IN(128),IYAD(20),IXAD(20),IZAD(20)
COMMON IYAI(20),IXAI(20),IZAI(20),IFAI(20),IFT(20)
COMMON ITMD(20),DD(20),DM(20),XID(20),XIM(20),IDATM(20)
COMMON ITMI(20),IDAY,IHR,IMIN,1DAY1,IHR1,IMIN1,1DBTM(20)
COMMON HD(20),HI(20),AX(20),AY(20),AZ(20)
COMMON IND(1440),INH(1440),INZ(1440),INF(1440)
COMMON PREV, IEOF, ITIMER, NCOMP
DIMENSION IDAT(1536)
C
C THIS SUBROUTINE SEARCHES OUT THE APPROPRIATE DAY IN THE TREORG-
C TYPE FILE.
C IT FIRST RESTRUCTURES THE NEW "*.WKF" FILES INTO THE OLD
C FAMILIAR FORMAT (1440 BASE)
C
800 FORMAT (128I8)
READ(1,800,END=900) IDUM
READ(1,800,END=900) IDUM
IF(NCOMP.GT.4)WRITE(7,'(A)') ' TOO MANY COMPONENTS'
9 DO 91 J=1,NCOMP
K2 = 0
DO 92 IREC=1,12
```

```
K1 = K2+1
K2 = K1+127
92 READ (1,800,END=900) (IDAT(L),L=K1,K2)
  IF (J.GT.1) GOTO 10
  DO 5 I=1,1440
    IF (IDAT(I).GT.799999) IDAT(I)=9999990
    5 INH(I) = IDAT(I)/10
    GOTO 91
10 IF (J.GT.2) GOTO 20
  DO 15 I=1,1440
    IF (IDAT(I).GT.799999) IDAT(I)=9999990
15 IND(I) = IDAT(I)/10
  GOTO 91
20 IF (J.GT.3) GOTO 30
  DO 25 I=1,1440
    IF (IDAT(I).GT.799999) IDAT(I)=9999990
25 INZ(I) = IDAT(I)/10
  GOTO 91
30 DO 35 I=1,1440
  IF (IDAT(I).GT.799999) IDAT(I)=9999990
35 INF(I) = (IDAT(I)+5)/10
91 CONTINUE
  IN3=IN(6)
  IF(IN3.EQ.KDAY) CALL BANNER
  READ(1,800,END=910) (IN(I),I=1,128)
  RETURN

C
C      DIAGNOSTIC MESSAGES
C
900 WRITE(7,1000) (IN(I),I=1,10)
1000 FORMAT(1H0,'ABNORMAL E-O-F FOUND AT ',10I10)
  STOP
910 WRITE(7,1010) IN3
1010 FORMAT(1H0,'NORMAL E-O-F ON *.WKF-TYPE FILE AFTER/ON DAY ',14)
  IEOF = 1
  RETURN
  END
  SUBROUTINE BANNER
C
C      G. JANSEN VAN BEEK 1987.335
C
REAL*8 IFAI,DD,DM,XID,XIM,HD,HI,AX,AY,AZ
REAL*8 IXAD,IYAD,IXAI,IYAI,IZAD,IZAI
COMMON IN(128),IYAD(20),IXAD(20),IZAD(20)
COMMON IYAI(20),IXAI(20),IZAI(20),IFAI(20),IFT(20)
COMMON ITMD(20),DD(20),DM(20),XID(20),XIM(20),IDATM(20)
COMMON ITMI(20),IDAY,IHR,IMIN,IDAD1,IHR1,IMIN1,IBDTM(20)
COMMON HD(20),HI(20),AX(20),AY(20),AZ(20)
COMMON IND(1440),INH(1440),INZ(1440),INF(1440)
COMMON PREV, IEOF, ITIMER, NCMP
CHARACTER*20 STATION(14),MON(12)
DATA STATION/'OTTAWA','ST. JOHNS','MEANOOK','VICTORIA',
```

```
*      'FORT CHURCHILL', 'BAKER LAKE', 'CAMBRIDGE BAY',
*      'POSTE-DE-LA-BALEINE', 'RESOLUTE BAY',
*      'YELLOWKNIFE', 'GLENLEA', 'ALERT',
*      'MOULD BAY', 'UNLISTED'
DATA MON/'JANUARY', 'FEBRUARY', 'MARCH', 'APRIL', 'MAY', 'JUNE',
*      'JULY', 'AUGUST', 'SEPTEMBER', 'OCTOBER', 'NOVEMBER',
*      'DECEMBER'/
C
C      SUBROUTINE WRITES A BANNER OF THE OUTPUT FILE CONTAINING THE
C      OBSERVATORY NAME, THE YEAR-DAY AND THE DAY/MONTH
C
C      IST = IN(2)
C      IF(IST.GT.13)IST=14
C
C      WRITE(7,800)STATION(IST),IN(5),IN(6),IN(3),MON(IN(4))
800 FORMAT(1H0,1X,4(1H*),' INFORMATION FOR ',A,' OBSERVATORY. ',',
*4(1H*),' Year - ',I4,'. Sequential day - ',I3,' or - ',I2,', ',
*,A,'.')
      RETURN
      END
```

APENDIX 14

BASELINE.FOR

CC PROGRAM BASLINE.FOR
C THIS PROGRAM USES THE SPOT BASELINES VALUES PRODUCED BY DIFTRE.
C THE SPOT BASELINES ARE PLOTTED FOR EACH COMPONENT, FOR IBTYP = 0.
C SPOT BASELINES CAN BE REJECTED FROM INCORPORATION INTO FINAL BASELINE
C BY INPUTTING CORRESPONDING COUNTER NUMBER VIA DLET CARD,
C DELETES ARE STILL PLOTTED, BUT CIRCLED, THEY DO NOT ENTER
C SUBSEQUENT CALCULATIONS.
C
C VERSION DEVELOPED TO ENCORPORATE REPEAT STATION OBSERVATIONS
C BY L. NEWITT
C
C PC VERSION COMPILED UNDER MSFORTRAN 5.0 INCLUDED GRAPHICS
C BY L. NEWITT
C
C FOR IBTYP = 2, THE PROGRAM PLOTS AS FOR IBTYP = 0, BUT CONTINUES TO
C COMPUTE BESTFIT BASELINE FROM DATA.
C IF IBTYP = 1 THE PROGRAM PLOTS AS FOR IBTYP=0 EXCEPT THAT REJECTED PO
C WILL NOT BE PLOTEED.
C IF IBTYPE=3 THE PROGRAM PLOTS AS FOR IBTYP=2 EXCEPT THAT THE REJECTED
C WILL NOT BE PLOTTED
C DISCONTINUITIES MUST BE DEFINED BY INPUTTING TIMES (DISC) AND
C STARTING VALUES (ST).
C IF(ST.EQ.999) THERE WILL BE NO DISCONTINUITY IN VALUE
C BUT ONLY A CHANGE IN SLOPE.
C
INCLUDE 'FGRAPH.FI'
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
INCLUDE 'FGRAPH.FD'
REAL*4 INDATA(12)
CHARACTER*3 TMOLA(24)
COMMON/BASE/ TMON(24),TMODA(5),TMONS(24) ,TDAY(24),
1TDA(366), THR(24), THRLA(24)
COMMON/INST/ X(20,2),Y(20,2),Z(20,2),XS(20,2),YS(20,2),
* ZS(20,2),IT1(20),IT2(20)
COMMON/BITS/ TSTRT,TEND,STYR,ENDYR,SCAL,YSCAL
DIMENSION IXBSLN(20,5), IYBSLN(20,5), IZBSLN(20,5)
DIMENSION XBSLN(20,5),YBSLN(20,5),ZBSLN(20,5)
DIMENSION XDISC(20),YDISC(20),ZDISC(20)
DIMENSION IDLET(20), ST(20)
DIMENSION STP(20), TI(20),T(20)
CHARACTER*1 ITYP(20), YESNO, ICPT, AMVEC, ICNT
CHARACTER*80 INFILE, OUTFILE, STA, INFLBS, STA2
CHARACTER*4 KCARD
RECORD / XYCOORD / XY
RECORD/ RCCOORD/ CP
DATA TMOLA/'JAN','FEB','MAR','APR','MAY','JUN',
* 'JUL','AUG','SEP','OCT','NOV','DEC',
* 'JAN','FEB','MAR','APR','MAY','JUN',
* 'JUL','AUG','SEP','OCT','NOV','DEC'/
ID2=0
ID=0
STA=' '

```
STA2=' '
PRINT 2002
2002 FORMAT(1H0,'ARE DATA FOR OBSERVATORIES OR REPEAT STATIONS? ',/
*,1H , 'ENTER 0 FOR OBS OR 9 FOR REP---->')
READ(*,*)IREP
PRINT 2000
2000 FORMAT(1H0,'ENTER NAME OF SPOT BASELINE (DIFTRE) INPUT FILE---->
*')
READ(*,2001)INFILE
2001 FORMAT(A)
IF(IREP.EQ.0)THEN
OPEN(UNIT=1,FILE=INFILE,STATUS='OLD',FORM='UNFORMATTED')
ELSE
OPEN(UNIT=1,FILE=INFILE,STATUS='OLD')
ENDIF
PRINT 2003
2003 FORMAT(1H , 'DO YOU WANT AN OUTPUT FILE? (Y/N)---->')
READ(*,2001)YESNO
IF(YESNO.EQ.'Y'.OR.YESNO.EQ.'y')THEN
PRINT 2004
2004 FORMAT(1H , 'ENTER NAME OF OUTPUT FILE---->')
READ(*,2001)OUTFILE
OPEN(UNIT=2,FILE=OUTFILE,STATUS='NEW')
ELSE
OPEN(UNIT=2,STATUS='SCRATCH')
ENDIF
PRINT 2009
2009 FORMAT(1H , 'ENTER NAME OF FILE CONTAINING DISCONTINUITIES---->')
READ(*,2001)INFLBS
OPEN(UNIT=4,FILE=INFLBS,STATUS='OLD')
OPEN(UNIT=7,FILE='OUTPUT',STATUS='NEW')
OPEN(UNIT=20,FILE='STACODE.DAT',STATUS='OLD')
READ(4,*)ID,ISTYR,ISTMO,IENDYR,IENDMO
STYR=ISTYR
STMO=ISTMO
ENDYR=IENDYR
ENDMO=IENDMO
CALL STATCOD(ID2,ID,STA,20)
IF(ID2.EQ.1)THEN
PRINT 2473
2473 FORMAT(1H0,'ARE THESE AMOS OR VECTOR PPM DATA? (A/V)--->')
READ(*,2001)AMVEC
ENDIF
PRINT 2008
2008 FORMAT(1H , 'ENTER PLOT TYPE, SCALE---->')
READ(*,*)IBTYP,YSCAL
IF(IREP.GT.0)THEN
SCAL=0.5
J=0
IFIN=ENDMO+1
DO 9001 I=STMO,IFIN
TDA(I)=J/SCAL
```

```
J=J+1
9001 CONTINUE
    J=0
    DO 9002 I=1,24
    J=J+1
    THR(I)=J/24./SCAL
    THRLA(I)=I
9002 CONTINUE
    ELSE
        SCAL=25.086
        DO 10 I=1,24
10     TMONS(I)=TMON(I)/SCAL
        DO 11 I=1,5
11     TMODA(I)=TMON(I)/SCAL
        END IF
100    FORMAT(A4,16,4F10.0,I5,F10.0,15)
        IF(IREP.GT.0)THEN
            PRINT 9995, STA, ID, STYR, STMO, ENDYR, ENDMO, IBTYP, YSCAL, IREP
            WRITE(7,9995)STA, ID, STYR, STMO, ENDYR, ENDMO, IBTYP, YSCAL, IREP
        ELSE
            WRITE(7,95)STA, ID, STYR, STMO, ENDYR, ENDMO, IBTYP, YSCAL, IREP
            PRINT 95, STA, ID, STYR, STMO, ENDYR, ENDMO, IBTYP, YSCAL, IREP
        ENDIF
        PRINT 96
        WRITE(7,96)
95    FORMAT(1H1,'BASELINE DATA PLOTTED FOR ',A4,16/1H0,
*'STARTING WITH YEAR',F4.0,' MONTH',F4.0,' ENDING WITH YEAR',
*F4.0,' MONTH',F4.0,5X,'IBTYP IS',I3,5X,'YSCAL IS',F10.0,
*5X,'IREP IS',I5)
9995 FORMAT(1H1,'BASELINE DATA PLOTTED FOR ',A4,16/1H0,
*'STARTING WITH YEAR',F4.0,' DAY ',F4.0,' ENDING WITH YEAR',
*F4.0,' DAY ',F4.0,5X,'IBTYP IS',I3,5X,'YSCAL IS',F10.0,
*5X,'IREP IS',I5)
96    FORMAT(1H0,T4,'ID',T13,'IT1',T23,'IT2',T32,'X',T37,'SD',T44,'Y',
*T49,'SD',T56,'Z',T61,'SD',T67,'XS',T73,'SD',T79,'YS',T85,'SD',
*T91,'ZS',T98,'SD',T105,'COUNTER')
        YRS=STYR+1900.
        IF(IREP.GT.0)THEN
            STYR=0.
            ENDIF
            I=1
105    CONTINUE
C ***** REPEAT STATION DATA ARE ON A FORMATTED FILE
        READ(1,5,END=110)ID,IT1(I),IT2(I),(INDATA(J),J=1,12)
        X(I,1)=DBLE(INDATA(1))
        X(I,2)=DBLE(INDATA(2))
        Y(I,1)=DBLE(INDATA(3))
        Y(I,2)=DBLE(INDATA(4))
        XS(I,1)=DBLE(INDATA(7))
        XS(I,2)=DBLE(INDATA(8))
        YS(I,1)=DBLE(INDATA(9))
```

```
YS(I,2)=DBLE(INDATA(10))
Z(I,1)=DBLE(INDATA(5))
Z(I,2)=DBLE(INDATA(6))
ZS(I,1)=DBLE(INDATA(11))
ZS(I,2)=DBLE(INDATA(12))
5   FORMAT(I6,2I10,12F7.0)
    CALL STATCOD(ID2,ID,STA2,20)
    IF(STA2.NE.STA)THEN
      PRINT 6541
      WRITE(7,6541)
    ENDIF
6541  FORMAT(1H0,'***** WARNING - STATION NAMES DO NOT MATCH *****',/)
    IF(IREP.GT.0)THEN
      IF(ABS(X(I,1)).GT.99999.) GO TO 105
      IF(ABS(Y(I,1)).GT.999999.) GO TO 105
      IF(ABS(Z(I,1)).GT.99999.) GO TO 105
      IF(ABS(XS(I,1)).GT.99999.) GO TO 105
      IF(ABS(YS(I,1)).GT.999999.) GO TO 105
      IF(ABS(ZS(I,1)).GT.99999.) GO TO 105
      ELSE
        IF(ABS(X(I,1)).GT.999.) GO TO 105
        IF(ABS(Y(I,1)).GT.999.) GO TO 105
        IF(ABS(Z(I,1)).GT.999.) GO TO 105
        IF(ABS(XS(I,1)).GT.999.) GO TO 105
        IF(ABS(YS(I,1)).GT.999.) GO TO 105
        IF(ABS(ZS(I,1)).GT.999.) GO TO 105
      END IF
      PRINT 98, ID, IT1(I), IT2(I), X(I,1), X(I,2), Y(I,1), Y(I,2), Z(I,1),
* Z(I,2), XS(I,1), XS(I,2), YS(I,1), YS(I,2), ZS(I,1), ZS(I,2) ,I
      WRITE(7,98) ID, IT1(I), IT2(I), X(I,1), X(I,2), Y(I,1), Y(I,2), Z(I,1),
* Z(I,2), XS(I,1), XS(I,2), YS(I,1), YS(I,2), ZS(I,1), ZS(I,2), I
98   FORMAT(1H ,I6,2I10,12F8.1,5X,I5)
    IF(I.EQ.1) GO TO 106
    IF(IT1(I).LT.IT1(I-1)) IT1(I) = IT1(I) + 3650000.
106  CONTINUE
    T(I)=IT1(I) + STYR*10000000.
    KTD=T(I)/10000
    THM=T(I)-KTD*10000.
    KTH=THM/100
    KTM=(THM-KTH*100.)
    TDEC=(KTH*60.+KTM)/1440.
    TT(I)=(KTD+TDEC)*10000.
C   USE MODIFIED MEAN ABSOLUTES
C   DO 97 L=1,2
C     XS(I,L)=X(I,L)
C     YS(I,L)=Y(I,L)
C     ZS(I,L)=Z(I,L)
C97  CONTINUE
    I=I+1
    GO TO 105
110  CONTINUE
    NPTS=I-1
```

```
IF(IREP.GT.0)THEN
DST=STMO
DSTP=ENDMO
ELSE
MST=STMO
TYR=(ENDYR-STYR)*12.
MOMAX=ENDMO-STMO+TYR+1
END IF
C CONVERT YRMO TO YRDAY
IF(IREP.GT.0)THEN
TSTART=STYR*1000.+STMO
TEND=ENDYR*1000.+ENDMO+1
ELSE
ISTMO=STMO
TSTART=STYR*1000.+TMON(ISTMO)
IENDMO=ENDMO
TEND=ENDYR*1000.+TMON(IENDMO +1)
END IF
C PLOT Z BASELINE POINTS
ICPT ='Z'
IF(AMVEC.EQ.'V')ICPT='F'
C REJECT POINTS
C THESE DELETES CAN BE IN ANY ORDER
PRINT 80,ICPT
WRITE(7,80)ICPT
80 FORMAT(1H1,A1,' REJECTS',
*//,1H , 'COUNTER NUMBERS ARE')
READ(4,*)KCARD,MD
IF(MD.NE.0)THEN
READ(4,*)(IDLET(K),K=1,MD)
ENDIF
IF(KCARD.NE.'DLET') GO TO 800
PRINT 82,(IDLET(K),K=1,MD)
WRITE(7,82)(1DLET(K),K=1,MD)
82 FORMAT(1H0,12I5)
C COMPUTE BASELINE FOR Z
C IBTYP=0 POINTS ONLY
C IBTYP=1 POINTS ONLY, REJECTS NOT PLOTTED
C IBTYP = 2 BEST FIT LINES
C ITYPE=3 BEST FIT LINES, REJECTS NOT PLOTTED
C READ IN DISCONTINUITIES FOR EACH COMPONENT
C FIRST MUST BE AT OR BEFORE FIRST ABSOLUTE
C LAST MUST BE AT OR AFTER LAST ABSOLUTE
C ZDISC CONTAINS TIME OF DISCONTINUITY, EG 830321245,
C ST CONTAINS THE STARTING VALUE AT DISCONTINUITY
C IF(ST.EQ.999) THIS MEANS NO DISCONTINUITY OF VALUE AT SLOPE CHANGE
C STP CONTAINS THE END VALUE APPLICABLE AT THE TIME OF THE NEXT DISCONT
C STP SHOULD EQUAL 999. UNLESS A FORCE FIT LINE IS REQUIRED
READ(4,*)KCARD,N
PRINT 149, N
WRITE(7,149)N
149 FORMAT(1H0,'THERE ARE',I5,' DISCONTINUITIES',
```

```
*//,T9,'START',T22,'LEVEL',T36,'END',T47,'LEVEL',T53,'TYPE')
IF(KCARD.NE.'DISC') GO TO 810
DO 1151 K=1,N
READ(4,*)ZDISC(K),ST(K),STP(K),ITYP(K)
1151 CONTINUE
PRINT 151, (ZDISC(K),ST(K),ZDISC(K+1),STP(K),ITYP(K),K=1,N)
WRITE(7,151)(ZDISC(K),ST(K),ZDISC(K+1),STP(K),ITYP(K),K=1,N)
151 FORMAT(1H0,F15.0,F10.2,F15.0,F10.2,5X,A5)
IF(IBTYP.EQ.0.OR.IBTYP.EQ.1)GO TO 298
IF(IBTYP.EQ.2.OR.IBTYP.EQ.3)CALL BSTFIT(ZS,NPTS,ZDISC,ZBSLN,N,
*NBSLN,IDLST,MD,ST,STP,T)
PRINT 170,ICPT
WRITE(7,170)ICPT
170 FORMAT(1H0,A1,' BASELINE',
*//,T5,'START',T17,'LEVEL',T30,'END',T41,'LEVEL',T55,'SLOPE',//)
KCPT=3
DO 155 I=1,NBSLN
PRINT 160, (ZBSLN(I,J),J=1,5),ITYP(I)
WRITE(7,160)(ZBSLN(I,J),J=1,5),ITYP(I)
160 FORMAT(1H ,2(F10.0,F10.2,4X),E16.10,5X,A5)
WRITE(2,161) ID,KCPT,(ZBSLN(I,J),J=1,4),ZBSLN(I,5),ITYP(I)
161 FORMAT(16,I4,2(F10.0,F12.3),E16.10,5X,A5)
155 CONTINUE
298 CONTINUE
PRINT 171
READ(*,2001)ICONT
CALL GRAPHICSMODE()
CALL SETVIEWORG(20, 20, XY)
C THIS CONVERTS THE PLOTTING COORDINATES TO INCHES TO CONFORTM WITH HARD
C COPY PLOTS; MAXIMUM FIVE DAYS IN HORIZONTAL DIRECTION
DUMMY = SETWINDOW(.TRUE., -1., -1., 10.5, 10.)
CALL SETTEXTPOSITION(24,75,CP)
CALL OUTTEXT('Z')
JCPT=3
XPAGE=0.
YPAGE=0.
CALL MINMAX(ZS,NPTS,YMAX,YMIN)
YZERO=YPAGE-YMIN/YSCAL
IF(IREP.GT.0)THEN
CALL BASPLTR(DST,DSTP,YMIN,YMAX,XPAGE,YPAGE)
ELSE
C CALL BASPLT(MST,MOMAX,YMIN,YMAX,XPAGE,YPAGE,ICPT,TMOLA)
END IF
CALL CPTPLT(NPTS,ZS,XPAGE,YZERO,IDLST,MD,IBTYP,TT)
IF(IBTYP.EQ.2.OR.IBTYP.EQ.3)CALL CPLT(XPAGE,YZERO,ZBSLN,NBSLN)
299 CONTINUE
PRINT 171
171 FORMAT(1H0,//,' PRESS ANY KEY TO CONTINUE--->')
READ(*,2001)ICONT
CALL ENDGRAPH()
C PLOT Y BASELINE POINTS
300 ICPT='Y'
```

```
IF(AMVEC.EQ.'V')ICPT='I'
C REJECT POINTS
PRINT 80,ICPT
WRITE(7,80)ICPT
READ(4,*) KCARD,MD
IF(MD.NE.0)THEN
READ(4,*)(IDLET(K),K=1,MD)
ENDIF
IF(KCARD.NE.'DLET') GO TO 800
PRINT 82,(IDLET(K),K=1,MD)
WRITE(7,82)(IDLET(K),K=1,MD)
C COMPUT BASELINE FOR Y
READ(4,*) KCARD,N
PRINT 149, N
WRITE(7,149)N
IF(KCARD.NE.'DISC') GO TO 810
DO 1152 K=1,N
READ(4,*)YDISC(K),ST(K),STP(K),ITYP(K)
1152 CONTINUE
PRINT 151, (YDISC(K),ST(K),YDISC(K+1),STP(K),ITYP(K),K=1,N)
WRITE(7,151)(YDISC(K),ST(K),YDISC(K+1),STP(K),ITYP(K),K=1,N)
IF(IBTYP.EQ.0.OR.IBTYP.EQ.1)GO TO 499
IF(IBTYP.EQ.2.OR.IBTYP.EQ.3)CALL BSTFIT(YS,NPTS,YDISC,YBSLN,N,
*NBSLN,IDLET,MD,ST,STP,T)
PRINT 170,ICPT
WRITE(7,170)ICPT
270 FORMAT(11H0Y BASELINE,
*//,,T5,'START',T17,'LEVEL',T30,'END',T41,'LEVEL',T55,'SLOPE',//)
KCPT=1
DO 255 I=1,NBSLN
PRINT 160, (YBSLN(I,J),J=1,5),ITYP(I)
WRITE(7,160)(YBSLN(I,J),J=1,5),ITYP(I)
WRITE(2,161) ID,KCPT,(YBSLN(I,J),J=1,4),YBSLN(I,5),ITYP(I)
255 CONTINUE
499 CONTINUE
PRINT 171
READ(*,2001)ICONT
JCPT=2
CALL GRAPHICSMODE()
CALL SETVIEWORG(20, 20, XY)
C THIS CONVERTS THE PLOTTING COORDINATES TO INCHES TO CONFORTM WITH HARD
C COPY PLOTS; MAXIMUM FIVE DAYS IN HORIZONTAL DIRECTION
DUMMY = SETWINDOW(.TRUE., -1., -1., 10.5, 10.)
CALL SETTEXTPOSITION(24,75,CP)
CALL OUTTEXT('Y')
XPAGE=0.
YPAGE=0.
CALL MINMAX(YS,NPTS,YMAX,YMIN)
YZERO=YPAGE-YMIN/YSCAL
IF(IREP.GT.0)THEN
CALL BASPLTR(DST,DSTP,YMIN,YMAX,XPAGE,YPAGE)
ELSE
```

```
C CALL BASPLT(MST,MOMAX,YMIN,YMAX,XPAGE,YPAGE,ICPT,TMOLA)
END IF
CALL CPTPLT(NPTS,YS,XPAGE,YZERO,IDLLET,MD,IBTYP,TT)
IF(IBTYP.EQ.2.OR.IBTYP.EQ.3)CALL CPLT(XPAGE,YZERO,YESLN,NBSLN)
PRINT 171
READ(*,2001)ICONT
CALL ENDGRAPH()
C PLOT X BASELINE POINTS
500 ICPT='X'
IF(AMVEC.EQ.'V')ICPT='D'
C REJECT POINTS
PRINT 80,ICPT
WRITE(7,80)ICPT
READ(4,*) KCARD,MD
IF(MD.NE.0)THEN
READ(4,*)(IDLLET(K),K=1,MD)
ENDIF
IF(KCARD.NE.'DLET') GO TO 800
PRINT 82,(IDLLET(K),K=1,MD)
WRITE(7,82)(IDLLET(K),K=1,MD)
C COMPUTE BASELINE FOR X
READ(4,*)KCARD,N
PRINT 149, N
WRITE(7,149)N
IF(KCARD.NE.'DISC') GO TO 810
DO 1153 K=1,N
READ(4,*)XDISC(K),ST(K),STP(K),ITYP(K)
1153 CONTINUE
WRITE(7,151)(XDISC(K),ST(K),XDISC(K+1),STP(K),ITYP(K),K=1,N)
PRINT 151, (XDISC(K),ST(K),XDISC(K+1),STP(K),ITYP(K),K=1,N)
IF(IBTYP.EQ.0.OR.IBTYP.EQ.1)GO TO 599
IF(IBTYP.EQ.2.OR.IBTYP.EQ.3)CALL BSTFIT(XS,NPTS,XDISC,XBSLN,N,
*NBSLN,IDLLET,MD,ST,STP,T)
PRINT 170,ICPT
WRITE(7,170)ICPT
KCPT=2
DO 555 I=1,NBSLN
PRINT 160, (XBSLN(I,J),J=1,5),ITYP(I)
WRITE(7,160)(XBSLN(I,J),J=1,5),ITYP(1)
WRITE(2,161) ID,KCPT,(XBSLN(I,J),J=1,4),XBSLN(I,5),ITYP(I)
555 CONTINUE
599 CONTINUE
PRINT 171
READ(*,2001)ICONT
JCPT=1
XPAGE=0.
YPAGE=0.
CALL GRAPHICSMODE()
CALL SETVIEWORG(20, 20, XY)
C THIS CONVERTS THE PLOTTING COORDINATES TO INCHES TO CONFORTM WITH HARD
C COPY PLOTS; MAXIMUM FIVE DAYS IN HORIZONTAL DIRECTION
DUMMY = SETWINDOW(.TRUE., -1., -1., 10.5, 10.)
```

```
CALL SETTEXTPOSITION(24,75,CP)
CALL OUTTEXT('X')
CALL MINMAX(XS,NPTS,YMAX,YMIN)
YZERO=YPAGE-YMIN/YSCAL
IF(IREP.GT.0)THEN
CALL BASPLTR(DST,DSTP,YMIN,YMAX,XPAGE,YPAGE)
ELSE
CALL BASPLT(MST,MOMAX,YMIN,YMAX,XPAGE,YPAGE,ICPT,TMOLA)
END IF
CALL CPTPLT(NPTS,XS,XPAGE,YZERO,IDLTD,MD,IBTYP,TT)
IF(IBTYP.EQ.2.OR.IBTYP.EQ.3)CALL CPLT(XPAGE,YZERO,XBSLN,NBSLN)
PRINT 171
READ(*,2001)ICONT
CALL ENDGRAPH()
GO TO 925
800 PRINT 801, KCARD
WRITE(7,801)KCARD
801 FORMAT(1H0,20(1H*),'ERROR- CARD READ WAS TYPE',A5,' DLET NEEDED')
GO TO 925
810 PRINT 811, KCARD
WRITE(7,811)KCARD
811 FORMAT(1H0,20(1H*),'ERROR- CARD READ WAS TYPE',A5,' DISC NEEDED')
925 CLOSE(UNIT=7,STATUS='KEEP')
STOP
END
BLOCKDATA
COMMON/BASE/ TMON(24),TMODA(5),TMONS(24) ,TDAY(24),
1TDA(366), THR(24), THRLA(24)
DATA TMODA/5.,10.,15.,20.,25./
C CALENDAR IS FOR 1985 AND 1986
DATA TMON/0.,31.,59.,90.,120.,151.,181.,212.,243.,273.,304.,334.,
*365.,396.,424.,455.,485.,516.,546.,575.,608.,638.,669.,699./
END
SUBROUTINE BASPLT(MST,MOMAX,YMIN,YMAX,XPAGE,YPAGE,ICPT,TMOLA)
C PLOTS AXES FOR BASELINE PLOTS
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
COMMON/BASE/ TMON(24),TMODA(5) ,TMONS(24) ,TDAY(24),
1TDA(366), THR(24), THRLA(24)
COMMON/BITS/ TSTRT,TEND,STYR,ENDYR,SCAL,YSCAL
CHARACTER*3 TMOLA(24)
IPDN=2
IPUP=3
Y1=-0.25
T1=-0.1
T2=-0.05
C PLOT YAXIS
ANNOT=YSCAL/2.
YSCLINV = 1./YSCAL
C CALL AXES(XPAGE,YPAGE,YMIN,YMAX,YSCLINV,ANNOT,ANNOT,'F7.1',90.,
C *ICPT,1)
C PLOT XAXIS
C ARRAY CONTAINS MONTH TICKS - TMONS
```

```
C  ARRAY CONTAINS MONTH DAY TICKS - TMODA
C  ARRAY CONTAINS MONTH LABELS - TMOLA
C  MST IS STARTING MONTH, MOMAX IS NUMBER OF MONTHS TO PLOT
    TMONST=TMONS(MST)
C  PLOT MONTH TICKS
C      CALL PLOT(XPAGE,YPAGE,IPUP)
        XM=0.
        M=MST
50   CONTINUE
C      CALL NUMBER(XM,YPAGE+0.05,0.07,TMON(M),0.,'F3.0')
C      CALL PLOT(XM,YPAGE,IPUP)
C      CALL PLOT(XM,YPAGE+T1,IPDN)
C      CALL PLOT(XM,YPAGE-T1,IPDN)
C      CALL PLOT(XM,YPAGE,IPDN)
C  PLOT MONTH DAY TICKS
        X=TMODA(1) + XM
        I=0
100  CONTINUE
C      CALL PLOT(X,YPAGE,IPDN)
C      CALL PLOT(X,YPAGE+T2,IPDN)
C      CALL PLOT(X,YPAGE,IPDN)
        I=I+1
        IF(I.GT.5) GO TO 101
        X=TMODA(I)+XM
        GO TO 100
101   CONTINUE
        M=M+1
        XD=TMONS(M)-TMONST
C      CALL PLOT(XD,YPAGE,IPDN)
C  WRITE MONTH LABEL
        X=TMODA(2) +XM
        CALL SYMBOL(X,YPAGE+Y1,0.15,TMOLA(M-1),0.,3)
C  START NEXT MONTH
        XM=XD
        IF(M.LE.MOMAX) GO TO 50
        YSCLINV=-YSCLINV
        ANNOT=-ANNOT
C      CALL AXES(XM,YPAGE+8.,YMAX,YMIN,YSCLINV,ANNOT,ANNOT,'F7.1',270.,
C      *ICPT,1)
C  RETURN TO MAIN
        RETURN
        END
        SUBROUTINE CPTPLT(NPTS,R,XP,YP,IDLLET,MD,IBTYP,TT)
C  S/R TO PLOT THE SPOT BASELINE VALUES
C  IF IBTYP=1 OR IBTYP=3 DOES NOT PLOT REJECTED POINTS
        IMPLICIT DOUBLE PRECISION(A-H,O-Z)
        INCLUDE 'FGRAFH.FD'
        COMMON/BITS/ TSTRRT,TEND,STYR,ENDYR,SCAL,YSCAL
        COMMON/INST/ X(20,2),Y(20,2),Z(20,2),XS(20,2),YS(20,2),
* ZS(20,2),IT1(20),IT2(20)
        DIMENSION R(20,2),IDLLET(20), TT(20)
        RECORD /WXYCOORD/ WXY
```

C SELECT TIME RANGE
DO 200 I=1,NPTS
DO 100 K=1,MD
IF(I.EQ.IDLET(K).AND.IBTYP.EQ.1)GO TO 200
100 IF(I.EQ.IDLET(K).AND.IBTYP.EQ.3)GO TO 200
IF(TT(1).LT.TSTRT*10000.) GO TO 200
IF(TT(1).GT.TEND*10000.) GO TO 200
XPAGE=(TT(1)/10000.-TSTRT)/SCAL + XP
YPAGE=R(I,1)/YSCAL + YP+1.
SD=R(I,2)/YSCAL
10 IF(YPAGE.LT.0..OR.YPAGE.GT.10.)GO TO 11
GO TO 12
11 IF(YPAGE.GT.10.)YPAGE=YPAGE-2.
IF(YPAGE.LT.0.)YPAGE=YPAGE+2.
GO TO 10
12 CONTINUE
YG=YPAGE
CALL MOVETO_W(XPAGE,YG,WXY)
DUMMY=LINETO_W(XPAGE+.1,YG+.1)
DUMMY=LINETO_W(XPAGE-.1,YG-.1)
DUMMY=LINETO_W(XPAGE,YG)
DUMMY=LINETO_W(XPAGE-.1,YG+.1)
DUMMY=LINETO_W(XPAGE+.1,YG-.1)
DUMMY=LINETO_W(XPAGE,YG)
YG=YPAGE+SD
IF((YPAGE+SD).GT.10.) YG=10.
DUMMY=LINETO_W(XPAGE,YG)
YG=YPAGE-SD
IF(YG.LT.0.) YG=0.
DUMMY=LINETO_W(XPAGE,YG)
C IDENTIFY REJECTED POINTS
DO 150 K=1,MD
YG=YPAGE
IF(I.NE.IDLET(K))GO TO 150
CALL MOVETO_W(XPAGE,YG,WXY)
DUMMY=RECTANGLE_W(\$GBORDER,XPAGE-.1,YG+.1,XPAGE+.1,YG-.1)
150 CONTINUE
200 CONTINUE
RETURN
END
SUBROUTINE CPLT(XP,YP,BSLN,NBSLN)
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
INCLUDE 'FGRAPH.FD'
DIMENSION BSLN(20,5)
COMMON/BITS/ TSTRT,TEND,STYR,ENDYR,SCAL,YSCAL
RECORD /WXYCOORD/ WXY
DO 100 I=1,NBSLN
AA=BSLN(I,1)/10000.
IF((IDINT(AA)/1000).GT.STYR)AA=AA-1000.+365.
IAA=AA
THM=AA-IAA
IH=THM*100.

```
IM=THM*10000.-IH*100.  
AA=IAA+(IH*60.+IM)/1440.  
X1=(AA-TSTRT)/SCAL+XP  
Y1=BSLN(I,2)/YSCAL + YP+1.  
A=BSLN(I,3)/10000.  
IF((IDINT(A)/1000).GT.STYR) A=A-1000.+365.  
IA=A  
THM=A-IA  
IH=THM*100.  
IM=THM*10000.-IH*100.  
A=IA+(IH*60.+IM)/1440.  
X2=(A-TSTRT)/SCAL + XP  
Y2=BSLN(I,4)/YSCAL + YP+1.  
10 IF(Y1.GT.10..OR.Y2.GT.10..OR.Y1.LT.0..OR.Y2.LT.0.)GO TO 11  
    GO TO 13  
11 IF(Y1.LT.0..OR.Y2.LT.0.)GO TO 12  
    Y1=Y1-2.  
    Y2=Y2-2.  
    GO TO 10  
12 Y1=Y1+2.  
    Y2=Y2+2.  
    GO TO 10  
13 CONTINUE  
    CALL MOVETO_W(X1,Y1,WXY)  
    DUMMY=LINETO_W(X2,Y2)  
100 CONTINUE  
    RETURN  
    END  
    SUBROUTINE MINMAX(ARRAY,NPTS,YMAX,YMIN)  
C TO FIND YMAX YMIN FOR PLOTS  
    IMPLICIT DOUBLE PRECISION(A-H,O-Z)  
    COMMON/BITS/ TSTRT,TEND,STYR,ENDYR,SCAL,YSCAL  
    DIMENSION ARRAY(20,2)  
    YMEAN=0.  
    ISCALE=YSCAL  
    DO 10 I=1,NPTS  
    YMEAN=YMEAN+ARRAY(I,1)  
10 CONTINUE  
    YMEAN=YMEAN/NPTS  
    IMEAN=IDINT(YMEAN)/ISCALE*ISCALE  
    YMIN=IMEAN-4*YSCAL  
    YMAX=IMEAN+4*YSCAL  
    RETURN  
    END  
    SUBROUTINE BSTFIT(R,NPTS,DISC,BSLN,NDISC,NBSLN,IDLLET,MD,ST,STP,T)  
C FITS WEIGHTED LEAST SQUARES LINE TO BASELINE SEGMENTS  
C WEIGHTS INVERSELY PROPORTIONAL TO STANDARD DEVIATIONS  
C READ IN STARTING VALUE FROM PREVIOUS BASELINE  
    IMPLICIT DOUBLE PRECISION(A-H,O-Z)  
    COMMON/INST/ X(20,2),Y(20,2),Z(20,2),XS(20,2),YS(20,2),  
    * ZS(20,2),IT1(20),IT2(20)  
    COMMON/BITS/ TSTRT,TEND,STYR,ENDYR,SCAL,YSCAL
```

```
DIMENSION BSLN(20,5),DISC(20) ,R(20,2) ,ADISC(20)
DIMENSION IDLET(20),ST(20),P(20),STP(20),T(20)
REAL*8 SUMPRD, SUMSQ
RST=ST(1)
RSTP = STP(1)
DO 17 J=1,NDISC
IYD=DISC(J)/10000000
IDISC=DISC(J)-IYD*10000000
IDY=IDISC/10000
IHR=(IDISC-IDY*10000)/100
IMIN=IDISC-IHR*100-IDY*10000
C   CONVERT TIMES TO YEARMINUTES
      ADISC(J)=IMIN+IHR*60+IDY*1440-1440
      IF(IYD.GT.STYR)ADISC(J)=ADISC(J)+365*1440
17   CONTINUE
      I=1
      K=1
      KK=1
15   SUMPRD=0
      SUMSQ=0
20   CONTINUE
C   SKIP REJECTS
      DO 50 L=1,MD
      IF(I.EQ.IDLET(L))GO TO 90
50   CONTINUE
      IF(I.GT.NPTS) GO TO 100
      JYD=T(I)/10000000
      JT=T(I)-JYD*10000000
      JDY=JT/10000
      JHR=(JT-JDY*10000)/100
      JMIN=JT-JHR*100-JDY*10000
C   CONVERT TO YEARMINUTE
      P(I)=JMIN+JHR*60+JDY*1440-1440
      IF(JYD.GT.STYR)P(I)=P(I)+365*1440
      IF(P(I).GT.ADISC(K+1)) GO TO 100
      U=P(I)-ADISC(K)
      V=R(I,1) -RST
      RSD=R(I,2)
      IF(RSD.LT.1.) RSD=1.
      SUMPRD=SUMPRD+V*U/RSD
      SUMSQ=SUMSQ+U*U/RSD
90   CONTINUE
      I=I+1
      GO TO 20
100  IF(RSTP.NE.999.)GO TO 105
      A=SUMPRD/SUMSQ
105  CONTINUE
      BSLN(KK,1)=DISC(K)
      BSLN(KK,2)=RST
      BSLN(KK,3)=DISC(K+1)
      IF(RSTP.NE.999.)GO TO 107
      RST=RST + A*(ADISC(K+1)-ADISC(K))
```

```
BSLN(KK,4)=RST
BSLN(KK,5)=A
GO TO 108
107 BSLN(KK,4)=RSTP
A=(RSTP-RST)/(ADISC(K+1)-ADISC(K))
BSLN(KK,5)=A
RST=RSTP
108 CONTINUE
K=K+1
RSTP=STP(K)
IF(ST(K).EQ.999.) GO TO 110
RST=ST(K)
110 CONTINUE
KK=KK+1
IF(K.LT.NDISC) GO TO 15
NBSLN=KK -1
RETURN
END
SUBROUTINE BASPLTR(DST,DSTP,YMIN,YMAX,XPAGE,YPAGE)
C PLOTS AXES FOR BASELINE PLOTS
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
INCLUDE 'FGRAPH.FD'
COMMON/BASE/ TMON(24),TMODA(5) ,TMONS(24) ,TDAY(24),
*TDA(366),THR(24),THRLA(24)
COMMON/BITS/ TSTRT,TEND,STYR,ENDYR,SCAL,YSCAL
RECORD /WXYCOORD/ WXY
RECORD /XYCOORD/ POSITION
CHARACTER*3 STR
CHARACTER*5 STR1
IPDN=2
IPUP=3
Y1=-0.25
T1=-0.1
T2=-0.05
C PLOT YAXIS
YM = YPAGE+1.
XX = XPAGE
CALL MOVETO_W(XX,YM,WXY)
NUMY=YMIN
DO 10 YYM = YM,YM+8.
DUMMY = LINETO W(XX,YYM)
DUMMY = LINETO_W(XX+T1,YYM)
DUMMY = LINETO_W(XX-T1,YYM)
DUMMY = LINETO W(XX,YYM)
CALL MOVETO W(XX-.5,YYM-1.5,WXY)
CALL GETCURRENTPOSITION(POSITION)
CALL SETTEXTPOSITION(POSITION.YCOORD/8,POSITION.XCOORD/8,
+ POSITION)
WRITE(STR1, '(15)') NUMY
CALL OUTTEXT(STR1)
CALL MOVETO W(XX,YYM,WXY)
IF(YYM.EQ.YM+8.)GO TO 10
```

```
DUMMY = LINETO_W(XX,YYM+.5)
DUMMY = LINETO_W(XX+T2,YYM+.5)
DUMMY = LINETO_W(XX-T2,YYM+.5)
DUMMY = LINETO W(XX,YYM+.5)
NUMY=NUMY+YSCAL
10    CONTINUE
C PLOT X AXIS
C ARRAY CONTAINS DAY TICKS - TDA
C ARRAY CONTAINS HOUR TICKS -THR
C ARRAY CONTAINS HOUR LABELS THRLA
C DST IS STARTING DAY, DSTP IS LAST DAY TO PLOT
    IDST=DST
    TDAST =TDA(IDST)
C PLOT DAY TICKS
    XM=XPAGE
    YY=YPAGE+1.
    M=DST
50    CONTINUE
    CALL MOVETO_W(XM,YY,WXY)
    DUMMY = LINETO_W(XM,YY)
    DUMMY = LINETO_W(XM,YY+T1)
    DUMMY = LINETO_W(XM,YY-T1)
    DUMMY = LINETO_W(XM,YY)
C PLOT HOUR TICKS
    X=THR(1)+XM
    I=0
100   CONTINUE
    DUMMY = LINETO W(X,YY)
    DUMMY = LINETO_W(X,YY+T2)
    DUMMY = LINETO W(X,YY)
    I=I+1
    IF(I.GT.24) GO TO 101
    X=THR(I)+XM
    GO TO 100
101   CONTINUE
    M=M+1
    IXPLT=M-1.
    XD=TDA(M)-TDAST
    DUMMY = LINETO W(XD,YY)
    X=THR(8)+XM
    CALL MOVETO_W(X+.5,YY-1.7,WXY)
    CALL GETCURRENTPOSITION (POSITION)
    CALL SETTEXTPOSITION (POSITION.YCOORD/8, POSITION.XCOORD/8,
+                                POSITION)
    WRITE(STR, '(I3)')IXPLT
    CALL OUTTEXT(STR)
C START NEXT DAY
    XM=XD
    IF(M.LE.DSTP)GO TO 50
    YSCLINV=-YSCLINV
    TICK=-TICK
    ANNOT=-ANNOT
```

```
CALL MOVETO_W(XM,YM,WXY)
DO 20 YM = YM,YM+8.
DUMMY = LINETO_W(XM,YYM)
DUMMY = LINETO_W(XM+T1,YYM)
DUMMY = LINETO_W(XM-T1,YYM)
DUMMY = LINETO_W(XM,YYM)
IF(YYM.EQ.YM+8.)GO TO 20
DUMMY = LINETO_W(XM,YYM+.5)
DUMMY = LINETO_W(XM+T2,YYM+.5)
DUMMY = LINETO_W(XM-T2,YYM+.5)
DUMMY = LINETO_W(XM,YYM+.5)
20 CONTINUE
C RETURN TO MAIN
RETURN
END
SUBROUTINE STATCOD(ID2L, ID6L, STA, IUNIT)
CHARACTER*80 XSTA, STA
REWIND IUNIT
100 READ(IUNIT,10,END=90)JD2L,JD6L,XSTA
10 FORMAT(I2,1X,I6,1X,A)
IF(JD2L.EQ.ID2L.OR.JD6L.EQ.ID6L.OR.XSTA.EQ.STA)THEN
ID2L=JD2L
ID6L=JD6L
STA=XSTA
RETURN
ENDIF
GO TO 100
90 JD2L=99
JD6L=999999
STA='XXX'
RETURN
END

SUBROUTINE GRAPHICSMODE()
INCLUDE 'FGRAPH.FD'
RECORD /VIDEOCONFIG/ VDINFO
CALL GETVIDEOCONFIG(VDINFO)
dummy=setvideomode(6)
RETURN
END

SUBROUTINE ENDGRAPH()
INCLUDE 'FGRAPH.FD'
DUMMY = SETVIDEOMODE( $DEFAULTMODE )
RETURN
END
```

APPENDIX 15

ADDBASE.FOR

PROGRAM ADDBASEC
C VAX ADAPTATIONS
C REPEAT STATION* DEC 1986, MODIFIED BY L NEWITT
C OBSERVATORY * MAR 1988, " " F PLET
C
C THIS PROGRAM WAS WRITTEN BY R.L. COLES, NOVEMBER 1984
C AS A MAJOR ADAPTATION OF AMOSBL2. (CYBER)
C
C THE PRESENT VAX OBSERVATORY ADAPTATION COMMENCED FROM
C THAT PLATFORM. HOWEVER DUE TO CHANGES IN DATA INPUT
C FORMAT AND MONTHLY FRAMING, MANY ASPECTS OF THE EARLIER
C CYBER VERSION ARE IRRELEVANT. THEY HAVE BEEN SUBJECTED
C TO CUT-OFF SURGERY, REVISED OR REPLACED IN THIS VERSION
C WITHOUT ELIMINATION FOR THE MOMENT, IF MERELY TO RETAIN
C FUNCTIONAL CLUES IN ITS FORMER USE.
C (WITH AN EYE TO ANY NEEDS FOR RESURRECTION)
C THE CURRENT VERSION IS THEREBY LOADED WITH 'DEADWOOD'.
C A PROPER HOUSE-CLEANING COULD FOLLOW AT CONVENIENCE.
C * THE REPEAT STATION FUNCTION IS NOT ACTIVE HERE *
C * RETUNING WOULD TAKE ADVANTAGE OF THIS NEW BASE *
C
C.....THIS PROGRAM 1)ADDS BASELINE TO AMOS MINUTE VALUES
C.....I/O UNITS ARE AS FOLLOWS
C.....MUNIT--DISP, INPUT/OUTPUT--MASTER FILE, OUTPUT ONLY IN CREATE MODE
C.....INUNIT--DISP, INPUT-NEW INPUT OF MINUTE VALUES
C.....IBUNIT--DISP, SCRATCH-TEMPORARY FILE TO STORE BASELINE VALUES
C.....IBSUN--DISP, INPUT OF BASELINE VALUES FROM PROGRAM BASLIN
C
C
C.....NEW INPUT FILE MUST BE POSITIONED UNDER SCOPE CONTROL
C..... BEFORE EXECUTING THIS PROGRAM.
C
C.....KCARD = R FOR DHZ REPEAT STATION MODE
C
C
C.....IF INREF IS NON-ZERO CHECK IDENT OF MASTER AND NEW INPUT
C.....IF KEYD IS NON-ZERO CONTINUE PROCESSING DESPITE DAY GAP
C.....IF KEYDR IS NON-ZERO CONTINUE PROCESSING DESPITE DAY REVERSAL,
C..... OVERWRITE OLD DAY/S
C.....IF KEYHR IS NON-ZERO CONTINUE PROCESSING DESPITE HOUR REVERSAL,
C..... OVERWRITE OLD HOUR/S
C.....IF KEYER IS NON-ZERO NEW ID AND YEAR ARE READ FROM
C.....DATA CARD FOLLOWING CNTL CARD TYPE D
C
C.....IF KEYDL IS NON-ZERO DAYS TO BE MASKED OUT ARE READ FROM
C..... TYPE B PROG CNTL CARD
C.....IF KEYLH IS NON-ZERO READ FROM TYPE E PROG CNTL CARD, DAYS
C..... FOR WHICH LAST HOUR IN NEW INPUT IS NOT 23, THESE MUST
C..... NOT INCLUDE THE FIRST AND THE LAST DAYS OF THE NEW INPUT
C..... FILE.

C
C.....IF KEYMYR IS NON-ZERO 1ST MASTER WILL START AT ANY DAY
C..... WHEN THE NEW INPUT FILE STARTS. IT WILL NOT FILL IN THE
C..... STARTING DAYS OF THE YEAR WITH NULL VALUES.
C
C.....KOIN IS THE LOGICAL SWITCH FOR HOUR MATCH.
C.....KOIN2 IS THE LOGICAL SWITCH FOR DAY MATCH.
C.....KDLTE IS LOGICAL SWITCH TO DENOTE IF MINUTE VALUES
C..... FOR A DAY ARE TO BE DELETED.
C.....KDREV IS THE LOGICAL SWITCH TO REGISTER THAT DAY REVERSAL
C..... HAS BEEN OVERWRITTEN.
C.....KCHONLY IS THE SWITCH TO CONTINUE CHECKING DATA CARDS DESPITE
C..... ERRORS. JOB IS ABORTED AFTERWARDS.
C
REAL*8 ISTRT,IISTRV,IENDT,IENDV
REAL*8 INBASE,SLOPST,SLOPE
DIMENSION TEMP(1440)
DIMENSION IDDL(366),IEDL(366),ISDL(366)
DIMENSION INBASE(200,4),SLOPST(3),JBASC(3),MINXT(3),LPR(3)
DIMENSION KOMP(3),NDAY(366),NMON(366),LHD(366)
COMMON IDAT(1536,4),IHRD(1536,4),IO(148),KO(128), NCOMP
COMMON ITEMP(128),JTEMP(128)
COMMON /KEY/ INREF,KEYD,KEYDR,KEYHR,KEYER,KEYDL,KEYLH,KEYMYR,INUM
COMMON /BLK1/ INHR,INMIN,INDY
COMMON /CHON/KCHONLY
LOGICAL KOIN,KOIN2,KDLTE,KDREV
LOGICAL KCHONLY

CHARACTER*20 DOCFILE,INFILE,MSTFILE,BASFILE,ISTMN,YESNO
CHARACTER*3 ISNAME
CHARACTER*1 KOMP,OLDNEW
DATA KOMP/'X','Y','Z'/
RAD=180./3.1415926536
800 FORMAT (128I8)
1500 FORMAT (A)
PRINT 451
451 FORMAT(1H0,' ENTER NAME OF INPUT FILE--->')
READ(*,1500) INFILE
PRINT 452
452 FORMAT(1H0,' ENTER NAME OF OUTPUT FILE--->')
READ(*,1500) MSTFILE
OPEN(UNIT=6,FILE='OUTPUT.DOC',STATUS='NEW')
PRINT 301,INFILE
WRITE(6,301) INFILE
301 FORMAT(1H1,'.WKF FILE ----> ',A)
OPEN(UNIT=1,FILE=INFILE,STATUS='OLD')
PRINT 302,MSTFILE
WRITE(6,302) MSTFILE
302 FORMAT(1H0,'MASTER FILE ----> ',A)
OPEN(UNIT=2,FILE=MSTFILE,STATUS='NEW',RECL=1024)
NREF = 0
PRINT 453

```
453 FORMAT(1H0,' ENTER NAME OF BASELINE FILE--->')
      READ(*,1500) BASFILE
      PRINT 305,BASFILE
      WRITE(6,305) BASFILE
 305 FORMAT(1H0,'BASELINE FILE----> ',A)
      WRITE(6,306)
 306 FORMAT(1H1)
      OPEN(UNIT=99,FILE=BASFILE,STATUS='OLD')
C      OPEN(UNIT=33,FILE='STACODE.DAT',STATUS='OLD')
      OPEN(UNIT=20,STATUS='SCRATCH')
      OPEN(UNIT=21,STATUS='SCRATCH')
      OPEN(UNIT=22,STATUS='SCRATCH')
      DO 2 I=1,1440
      DO 2 J=1,4
 2 IDAT(I,J)=999999
      MUNIT=2
      KUNIT=2
      INUNIT=1
      IBUNIT=3
      IBSUN=99
      ISTUNIT=33
      MARK=0
      NEWDY=0
      KDAY=0
      KOIN=.FALSE.
      KOIN2=.FALSE.
      KDLTE=.FALSE.
      KCHONLY=.FALSE.
      KDREV=.FALSE.
C.....ENTER PROGRAM CONTROL DATA
C
C      ACCEPT 1500,KCARD
 7000 FORMAT(1H0,'ARE DATA FOR REPEAT STATIONS OR OBSERVATORIES?//,
&1H , 'ENTER R (REP) OR A (OBS)---->')
      KCARD = 1HA
C      READ(*,7001)KCARD
 7001 FORMAT(A)
C      PRINT 7002
 7002 FORMAT(1H , 'ENTER NUMBER OF FILES ON TREORG TAPE---->')
C      READ(*,*)INUM
      INUM = 1
C      PRINT 7003
 7003 FORMAT(1H0,'DO YOU WANT DEFAULT CONTROL PARAMETER? THESE ARE:',
      *//,' INDEF=0,KEYD=1,KEYDR=0,KEYHR=0,KEYER=0,KEYDL=0,KEYLH=0'
      &,//,' KEYMYR=0 FOR OBSERVATORIES, KEYMYR=1 FOR REPEAT STATIONS'
      %//,' ENTER Y OR N---->')
C      READ(*,7001)YESNO
C      IF(YESNO.EQ.'Y'.OR.YESNO.EQ.'y')THEN
      KEYD=1
      KEYDR=0
      KEYHR=0
      KEYER=0
```

```
KEYDL=0
KEYLH=0
KEYMYR=1
IF(KCARD.EQ.'R'.OR. KCARD.EQ.'r')KEYMYR=1
C   ELSE
C   PRINT 7004
7004 FORMAT(1H ,'ENTER INREF---->')
C   READ(*,*)INREF
C   PRINT 7005
7005 FORMAT(1H ,'ENTER KEYD---->')
C   READ(*,*)KEYD
C   PRINT 7006
7006 FORMAT(1H ,'ENTER KEYDR---->')
C   READ(*,*)KEYDR
C   PRINT 7007
7007 FORMAT(1H ,'ENTER KEYHR---->')
C   READ(*,*)KEYHR
C   PRINT 7008
7008 FORMAT(1H ,'ENTER KEYER---->')
C   READ(*,*)KEYER
C   PRINT 7009
7009 FORMAT(1H ,'ENTER KEYDL---->')
C   READ(*,*)KEYDL
C   PRINT 7010
7010 FORMAT(1H ,'ENTER KEYLH---->')
C   READ(*,*)KEYLH
C   PRINT 7011
7011 FORMAT(1H ,'ENTER KEYMYR---->')
C   READ(*,*)KEYMYR
C   ENDIF
4001 FORMAT(1H0,80X,40(1H*),/)
IF(KCARD.NE.'A'.AND.KCARD.NE.'R') GO TO 9970
CALL MESSG
WRITE(6,4001)
I=1
IF(KEYDL.EQ.0)GO TO 17
PRINT 7304
7304 FORMAT(1H0,'***** WARNING - KEYDL IS NO LONGER A VALID OPTION ****',/,
%1H ,'PROGRAM DOES NOT IMPLEMENT IT')
GO TO 17
16 READ(5,1015) KCARD,NDL,IDL(I),ISDL(I),IEDL(I)
1015 FORMAT(A1,4X,I2,3I6)
IF(KCARD.NE.'B') GO TO 9970
IF(NDL.LT.2) GO TO 17
IF(I.LT.2) JDL=NDL
I=I+1
IF(I.GT.JDL) GO TO 17
IF(KCARD.NE.'B') GO TO 9970
GO TO 16
17 CONTINUE
19 IF(KEYER.EQ.0)GO TO 35
PRINT 7300
```

```
7300 FORMAT(1H0,'KEYER IS NON-ZERO; ENTER NEW ID AND YEAR---->')
      READ(*,*)IDCOR,IYCOR
  35 IF(KEYLH.EQ.0)GO TO 45
      PRINT 7301
7301 FORMAT(1H0,'KEYLH IS NO-ZERO; ENTER TOTAL NUMBER OF DAYS FOR WHICH
&LAST HOUR IN INPUT IS NOT 23---->')
      READ(*,*)NLH
      PRINT 7302
7302 FORMAT(1H , 'ENTER ACTUAL DAY NUMBERS---->')
      READ(*,*)(LHD(I),I=1,NLH)
  45 READ(1,800,END=6) (IO(I),I=1,128)
      IDSTN = IO(1)
      IYR1 = IO(5)
      IDAY1 = IO(6)
      IHR1 = IO(7)
      BACKSPACE 1
      GO TO 70
  6 WRITE(6,2001)
      PRINT 2001
2001 FORMAT(1H ,20X,'UNSUCCESSFULL OPENING OF NEW INPUT FILE',
  1 5X,'SUGGEST USE SKIPF,TAPE1,1,17,B. SCOPE CNTL CARD',
  2 'BEFORE LDSET')
      STOP
  70 IF(KEYER.LT.1) GO TO 8
      IF(IDCOR.LT.1) GO TO 71
      WRITE(6,1003)IDSTN, IDCOR
      PRINT 2003, IDSTN, IDCOR
1003 FORMAT(1H0,T66,'IDENT ON NEW INPUT WAS ',I6,
  1', WILL BE REPLACED ON MASTER FILE BY ',I6)
      IDSTN=IDCOR
  71 IF(IYCOR.LT.1) GO TO 8
      WRITE(6,1004) IYR1,IYCOR
      PRINT 1004, IYR1,IYCOR
1004 FORMAT(1H ,T66,'YEAR ON NEW INPUT WAS ',I6,
  1', WILL BE REPLACED BY ',I6)
      IYR1=IYCOR
C.....ISNAME IS 3 CHAR IAGA ABBREVIATION.
  82 CONTINUE
  8 CALL SELSTN (IDSTN,ISNAME)
      IF(KCHONLY)CALL EREXT
      IF(INREF.NE.0)GO TO 87
  89 CONTINUE
C      IF A NEW MASTER CREATION, WRITE FIRST TREORG HEADER RECORD
CC      WRITE(MUNIT) IDSTN,IYR1,IDAD1,IHR1
C      TEMPORARY
      IF(IDAY1.LT.2) GO TO 400
C
C.....FILL IN FOR START OF THE YEAR
C
      IGY2=0
      I=IHR1-1
      IGY1=IDAD1-1
```

```
      WRITE(6,1009) ISNAME,IYR1,IGY1,I
      PRINT 1009,ISNAME,IYR1,IGY1,I
1009 FORMAT(1H , '--CAUTION-- ',A3,' YEAR',I5,' GAP FROM DAY 1 TO',
           1' DAY',I4,' HR',I3,' (INCLUSIVE)')
           IF(KEYMYR.EQ.0)GO TO 78
C.....IF KEYMYR NON-ZERO MASTER STARTS AT WHEREVER NEW INPUT STARTS
    74 WRITE(6,1010) IGY1,I
      PRINT 1010,IGY1,I
1010 FORMAT(1H ,11X,'MASTER FILES WILL START AFTER DAY',I4,' HR',
           1I3,'IGNORE 1ST ERROR IN PHASE 2')
           GO TO 234
C.....FILL START OF YEAR WITH NULL VALUES
    78 WRITE(6,1011)
      PRINT 1011
1011 FORMAT(1H ,11X,'ABOVE GAP WILL BE FILLED WITH NULL VALUES')
           GO TO 226
C
C.....READ LAST RECORD OF OLD MASTER
C
     87 CONTINUE
       IER=0
C     CALL TREBAK(MUNIT)
     IF (MARK.NE.0) GOTO 22
     CALL READBIN(MUNIT,IHRD,IER,INUM,MARK)
       LDAY = IDAY1
       CLOSE(UNIT=2,STATUS='KEEP')
       OPEN(UNIT=2,FILE=MSTFILE,STATUS='OLD',FORM='UNFORMATTED',
*ACCESS='APPEND')
C*     CALL TREBAK(MUNIT)
       GOTO 409
22 CALL READCOD(INUNIT,IHRD,IER,INUM)
       LDAY = IO(3)
20 CONTINUE
       IDSTN2=IO(1)
       IYR2=IO(2)
       ITDY2=IO(3)
       IHR2=IO(4)
       WRITE(6,1999) IO(1),IO(2),IO(3),IO(4)
       PRINT 1999, IO(1),IO(2),IO(3),IO(4)
1999 FORMAT(1H ,4I10,' AT 160')
   160 CONTINUE
C   160 IF(IDSTN.NE.IDSTN2) GO TO 9971
C   IF(IYR1.NE.IYR2) GO TO 9972
C   TEMPORARY
C
C.....COMPARE DAYS BETWEEN MASTER AND NEW INPUT
C
   210 IDTW=ITDY2
       IF(IDTW-IDAY1) 220,400,400
C   TEMPORARY
220 IF((IDAY1-IDTW).GT.62) GO TO 235
222 IGY2=ITDY2
```

```
IGY1=IDAY1-1
LH=IHR2+1
IF(IGY1.EQ.IGY2.AND.LH.EQ.24) GO TO 400
WRITE(6,2002) ISNAME,IYR2,IGY2,LH,IGY1
PRINT 2002, ISNAME,IYR2,IGY2,LH,IGY1
2002 FORMAT(1H ,'-CAUTION-- ',A3,' YEAR',I5,' GAP FROM DAY',I4,
1' HR',I3,
1' TO END OF DAY ',I3,/,11X,' FILLED WITH NULL VALUES')
225 CONTINUE
226 IGY2=IGY2+1
IF(IGY1.LT.IGY2) GO TO 400
DO 230 IJ=IGY2,IGY1
KDAY=KDAY+1
NDAY(KDAY)=IJ
C 30 CALL CLDATE(IJ,IYR1,MON,IDATE)
C      NMON(KDAY)=MON
IF=0
4993 FORMAT(13,4I2)
231 CONTINUE
C      CALL TRENUL(MUNIT,IDLSTN,IYR1,IJ)
C      TEMPORARY
230 CONTINUE
234 CONTINUE
232 IDRET=IGY1+1
GO TO 400
240 IF(KEYDR.EQ.0)GO TO 242
C
C.....FORWARD GAP IN FRONT OF NEW INPUT --PROCESSING CONTINUES
C
C
C.....NEW INPUT FILE STARTS AT EARLIER DATE THAN LAST DATE OF MASTER
C.....ABORT JOB IF KEYDR IS NOT ZERO.
242 PRINT 2003,ISNAME,IYR2,IDADY1,ITDY2
WRITE(6,2003)ISNAME,IYR2,IDADY1,ITDY2
2003 FORMAT(1H ,'ERROR..',A3,' YEAR',I5,
1'REVERSED SEQUENCE',/,1H , 'INPUT FILE STARTS AT DAY',I3,
2' MASTER STARTS DAY',I3,/,1H , 'PUNCH NON-ZERO AT COL.19 OF CARD ',
3'TYPE A TO OVERWRITE JOB ABORT')
CALL EREXT
C
C.....IF DAY REVERSE IS OVERWRITTEN, BACKSPACE OLDMASTER AND POSITION
C.....AT THE START OF THE DAY
C
245 CONTINUE
PRINT 2309 ,ISNAME,IYR2,IDADY1,ITDY2
WRITE(6,2309)ISNAME,IYR2,IDADY1,ITDY2
2309 FORMAT(1H , '--CAUTION-- ',A3,' YEAR',I5,
1'OLD MASTER DAYS OVERWRITTEN FROM ',I4,' TO ',I4)
247 CONTINUE
C*247 CALL TREBAK(MUNIT)
C*      CALL TREBAK(MUNIT)
      CALL READBIN(MUNIT,IHRD,IER,INUM,MARK)
```

```
LDAY = IDAY1
C
C.....BACKSPACE MASTER UNTIL FIRST INPUT DAY FOUND
C
250 IDSTN2=IO(1)
IYR2=IO(2)
ITDY2=IO(3)
IHR2=IO(4)
255 IF(ITDY2-IDAY1)248,260,247
260 CONTINUE
400 CONTINUE
INDY=IDAY1
IDRET=INDY
409 CALL RDRLN3(IBSUN, IDSTN)
IER=0
C..... READ NEW DAY
410 CONTINUE
CALL READCOD(INUNIT, IDAT, IER, INUM)
IF(IER.EQ.INUM) GO TO 9998
ITAR=IO(2)
ITDY=IO(3)
ITHR=IO(4)
LDAY = IO(3)
WRITE(6,449) IO(1),ITAR,ITDY,ITHR,KDAY,NEWDY
CPR PRINT 449, IO(1),ITAR,ITDY,ITHR,KDAY,NEWDY
449 FORMAT(1H ,6I10,' AT 449')
450 KDAY=KDAY+1
C.....NEWDY IS THE RUNNING COUNT OF DAYS ON THE NEW INPUT FILE
NEWDY=NEWDY+1
IF(KEYER.NE.0.AND.IYCOR.NE.0) ITAR=IYCOR
NDAY(KDAY)=ITDY
C CALL CLDATE(ITDY,ITAR,MON,IDATE)
C NMON(KDAY)=MON
NM=1
480 IF(NEWDY.LT.2) GO TO 490
ITDY2=ITDY
INMIN=NM
C
C
C.....DETECT GAP IN INPUT FILE, ISSUE PROPER MESSAGE ABORT FOR -VE GAP
547 IF(ITDY-IDRET)550,530,570
550 LPRINT=IDRET-1
IF(ITDY.EQ.LPRINT) GO TO 495
WRITE(6,4100) LPRINT,ITDY
PRINT 4100, LPRINT,ITDY
4100 FORMAT(40X,'--NOTE-JOB TERMINATED,REVERSE DAY SEQ IN INPUT FILE',
1' FROM DAY',I5,' TO',I5)
ITDY=LPRINT
KDAY=KDAY-1
GO TO 9998
C
C.....PUT NULL VALUES FOR MISSING DAYS IN INPUT FILE
```

C

570 LPRINT=IDRET
WRITE(6,4105) LPRINT,ITDY
PRINT 4105,LPRINT,ITDY

4105 FORMAT(1H , '--CAUTION--GAP IN INPUT FILE FROM DAY',I5,' TO DAY',
1I5,/,11X,' FILLED WITH NULL VALUES')
IDW=ITDY-1
DO 575 IJ=LPRINT, IDW

C CALL TRENUL(MUNIT, IDSTN, ITAR, IJ)
C TEMPORARY
575 CONTINUE

C

C

MISF=IDRET
MISL=ITDY-1
DO 595 IJ=MISF, MISL
NDAY(KDAY)=IJ

C CALL CLDATE(IJ, ITAR, MON, IDATE)
C NMON(KDAY)=MON
KDAY=KDAY+1

595 CONTINUE
NDAY(KDAY)=ITDY
GO TO 700

C

C

C

C.....READ FROM MASTER AND DETERMINE LAST MINUTE OF PREVIOUS MONTH

490 CONTINUE
IF(INREF.EQ.0) GO TO 530

C GOTO 530

C TEMPORARY

495 CONTINUE

6011 FORMAT(1H0,4I10)
C CALL TREBAK(MUNIT)
IF (MARK.EQ.0) CALL READBIN(MUNIT,IHRD,IER,INUM,MARK)
LDAY = IDAY1

C* CALL TREBAK(MUNIT)
DO 500 I=1,1440
J=1441-I
IF(IHRD(J,3).LT.80000) GO TO 510

500 CONTINUE

C.....LM IS THE LAST SIGNIFICANT MINUTE OF PREVIOUS MONTH

510 LM=J

C NM=LM+1

C.....TEST PRINT OF MERGING DAY HR MIN

PRINT 3011, IDAY1, NM
WRITE(6,3011) IDAY1, NM

3011 FORMAT(1H , '--NOTE-- DAY',14,' IS BEING APPENDED',
1' TO OLD MASTER STARTING AT MIN',I5)

C DO 520 I=1,LM
C DO 520 J=1,4
C 520 1DAT(1440,J)=IHRD(1440,J)

```
C      INMIN=LM+1
530  CONTINUE
      INMIN = 1
      NM = 1
C*** MODIFIED KERNEL AMOSBL2
C  USING TIMES FROM TREORG HEADER, CALCULATE INITIAL YEAR-MINUTE FOR EACH
      IF(NEWDY.EQ.1) ISTART = (IDAY1-1)*1440+INMIN
700  IF(NEWDY.GT.1) GO TO 710
710  CONTINUE
      IF(NEWDY.GT.1) ISTART = (ITDY-1)*1440 + 1
C  PRINT 712,NEWDY,ISTART,LDAY
      WRITE(6,712)NEWDY,ISTART,LDAY
712 FORMAT(1H , 'NEWDAY',I3,'    START ',I6,' DAY ',I3)
C
C  BASELINE CORRECTIONS WILL BE ADDED ONE COMPONENT AT A TIME, ONE DAY AT A
C
C  MAIN LOOP
C  LOGIC OF THIS SECTION HAS BEEN CHANGED FOR REPEAT STATIONS
711  CONTINUE
      DO 6900 I=1,3
      II=I
C  FOR REPEAT STATIONS, ALWAYS REVERSE COMPONENTS
C  IF(KCARD.EQ.1HR)GO TO 6901
C**** INVERT X AND Y FOR RES, ALE, MBC
C  IF(IDSTN.EQ.007298.OR.IDSTN.EQ.015265.OR.IDSTN.EQ.014241)
C  * GO TO 6901
C  MINH BASELINES ON IBUNIT =20, I = 1, ICPT = 1
C  MAJH BASELINES ON IBUNIT =21, I = 2, ICPT = 2
C  VERTZ BASELINES ON IBUNIT =22, I = 3, ICPT = 3
      IF(I.EQ.1) II=2
      IF(I.EQ.2) II=1
6902  IBUNIT = II+19
      INTIM = ISTART
C      NM = 1
      IMINT = 1
      IF(NEWDY.GT.1) GO TO 6730
C      GO TO 6700
C6901  IF(I.EQ.1) II=2
C      IF(I.EQ.2) II=1
C      GO TO 6902
6700  CONTINUE
C  READ BASELINE CARD IMAGE
      READ(IBUNIT,10,END=6800) ICPT,(INBASE(I,J),J=1,4),SLOPST(I)
10  FORMAT(I5,4F10.0,E16.10)
C  CHECK FOR CORRECT FILE - IF OK PRINT NEW BASELINE
      IF(ICPT.NE.II) GO TO 6810
      WRITE (6,4000) ICPT,(INBASE(I,J),J=1,4),SLOPST(I)
      PRINT 4000,ICPT,(INBASE(I,J),J=1,4),SLOPST(I)
4000 FORMAT(1H ,T35,'NEW BASELINE FOR CPT',I5,4F10.0,E16.10)
C  COMPUTE YEARMINUTES
      ISTRT = INBASE(I,1)
      IENDT = INBASE(I,3)
```

```
C      WRITE(6,4994) INBASE(I,1),INBASE(I,3)
C      PRINT 4994, .INBASE(I,1),INBASE(I,3)
IYR=ISTRT/10000000
ISTRT=ISTRT-IYR*10000000
IDY=ISTRT/10000
IHRMIN=ISTRT-IDY*10000
IHOUR=IHRMIN/100
IMIN=IHRMIN-IHOUR*100
INBASE(I,1)=(IDY-1)*1440+IHOUR*60+IMIN
IYR=IENDT/10000000
IENDT=IENDT-IYR*10000000
IDY=IENDT/10000
IHRMIN=IENDT-IDY*10000
IHOUR=IHRMIN/100
IMIN=IHRMIN-IHOUR*100
INBASE(I,3)=(IDY-1)*1440+IHOUR*60+IMIN
MINXT(I)=0
LPR(I)=1
C      WRITE(6,4994) ISTRT,IENDT,IYR, IDY, IHRMIN, IHOUR, IMIN
C      PRINT 4994, ISTRT, IENDT, IYR, IDY, IHRMIN, IHOUR, IMIN
4994 FORMAT(1H , 'HERE', 2F20.1,5I10)
6730 CONTINUE
C  PLACE BASELINE PARAMETERS IN TEMPORARY WORKING VARIABLES
  ISTRT = INBASE(I,1)
  ISTRV = INBASE(I,2)
  IENDT = INBASE(I,3)
  IENDV = INBASE(I,4)
  SLOPE=SLOPST(I)
C
6750 CONTINUE
C  CHECK THAT CURRENT TIME IS LESS THAN BASELINE LAST TIME
  IF(INTIM.GE.IENDT) GO TO 6700
C  IF BASELINE HAS EXPIRED, READ IN NEW BASELINE CARD IMAGE
C  LIST UNCORRECTED FIRST 5 MINUTE VALUES AFTER NEW BASELINE
C
  IF(LPR(I).LT.1) GO TO 6770
  IPHR = IMINT/60
  IPMINT = IMINT - IPHR*60
  IF(MINXT(I).GT.0) GO TO 6760
  WRITE(6,4995) ICPT
  PRINT 4995, ICPT
4995 FORMAT(1H ,T50,'FIRST 5 MINUTE VALUES AFTER NEW BASELINE',
1' FOR COMPONENT',I3,' WAS EFFECTIVE',//,T57,'DAY', ' HR',
2' MIN',T70,'ORIGINAL',T101,'CORRECTED')
6760 WRITE(6,5000) ITDY,IPHR,IPMINT, IDAT(IMINT,I)
  PRINT 5000, ITDY,IPHR,IPMINT, IDAT(IMINT,I)
5000 FORMAT(1H ,T52,I8,I3,I4,T70,I8)
6770 CONTINUE
  IF (MINXT(I).GT.4) MINXT(I)=0
C
C  COMPUTE AND ADD CORRECTION TO EACH MINUTE VALUE
C
```

```
IBASC1 = JBASC(I)
IBASC = ISTRV + SLOPE*(INTIM - ISTRT)
JBASC(I)=IBASC
IF(1BASC.EQ.IBASC1)GO TO 6772
PRINT 6771, IBASC,ISTRV,SLOPE,INTIM,ISTRT,I,LDAY
WRITE(6,6771) IBASC,ISTRV,SLOPE,INTIM,ISTRT,I,LDAY
6771 FORMAT(1H , 'BASE=',I4,' BASE1=',F6.1,' SLOPE= ',E10.4
      *,' MIN=',I7,' MIN1=',F8.0,' CPT=',I2,' DAY=',I4)
C CHECK FOR NULL VALUES
6772 IF(IDAT(IMINT,I).GT.90000) GO TO 6754
IF(KCARD.NE.'R')GO TO 6779
IF(I.NE.1)GO TO 6774
IF(IDAT(IMINT,2).GT.90000)GO TO 6754
IDAT(IMINT,I)=IDAT(IMINT,I)+IBASC
DAT2=IDAT(IMINT,2)
DAT1=IDAT(IMINT,I)
ALPHA1=ASIN(DAT2/DAT1)
IDAT(IMINT,I)=IDAT(IMINT,I)/COS(ALPHA1)
GO TO 6755
6774 IF(I.NE.2)GO TO 6779
IF(IDAT(IMINT,1).GT.90000)GO TO 6754
DAT2=IDAT(IMINT,I)
DAT1=IDAT(IMINT,I)
ALPHA=ASIN(DAT2/DAT1)*RAD*60.
IDAT(IMINT,I)=ALPHA+IBASC
GO TO 6755
6779 IDAT(IMINT,I)=IDAT(IMINT,I)+IBASC
GO TO 6755
6754 IDAT(IMINT,I)=999999
6755 CONTINUE
C LIST CORRECTED FIRST 5 MINUTE VALUES
IF(LPR(I).LT.1) GO TO 6780
WRITE(6,5001) IDAT(IMINT,I)
PRINT 5001, IDAT(IMINT,I)
5001 FORMAT(1H+,T101,I8)
6780 CONTINUE
C INCREMENT COUNTERS
MINXT(I) = MINXT(I) + 1
IF(MINXT(I).GT.4) LPR(I)=0
INTIM = INTIM + 1
IMINT = IMINT + 1
C PRINT 5002, INTIM,IMINT
C WRITE(6,5002) INTIM,IMINT
C5002 FORMAT(1H , 'INTIM ',I5,'IMINT ',I5)
C CHECK FOR END OF DAY IN CURRENT COMPONENT
IF(IMINT.LE.1440) GO TO 6750
IMINT = 1
C PROCEED TO NEXT COMPONENT IF 1.LT.3
6900 CONTINUE
C ALL THREE COMPONENTS FOR DAY HAVE BEEN CORRECTED
6910 CONTINUE
C RESET COUNTERS AND SWITCHES
```

```
IMINT = 1
MINXT(1)=0
MINXT(2)=0
MINXT(3)=0
KOIN=.FALSE.
IDRET=ITDY+1
C
C.....INSERT MODULE HERE FOR MINUTE VALUE INSERTIONS
C
C.....WRITE BASELINE CORRECTED DATA ON BINARY MASTER FILE
C..... FIRST HOUR TO BE OUTPUT IS IHR1
C.....LAST HOUR TO BE OUTPUT IS ITHR
950 IF(KEYDL.EQ.0)GO TO 970
C
C THIS OPTION IS NO LONGER USED
C
GO TO 970
C
960 KDLTE=.FALSE.
DO 968 J=1,JDL
IF(ITDY.NE.IDDL(J)) GO TO 968
KDLTE=.TRUE.
IDLS=ISDL(J)*60 + 1
IDLE=IEDL(J)*60
DO 961 I=IDLS, IDLE
DO 961 IJK=1,4
961 IDAT(I,IJK)=999999
WRITE(6,5998) ITDY,ISDL(J),IEDL(J)
PRINT 5998, ITDY,ISDL(J),IEDL(J)
5998 FORMAT(1H ,--CAUTION--MINUTE VALUES ARE MASKED OUT ',
1'FOR DAY ',13,' HOURS ',12,' TO ',12)
GO TO 970
968 CONTINUE
970 IFHR=IHR1+1
C.....FORCE LAST HOUR TO BE 23 IF THIS DAY WAS SPECIFIED ON CD TYPE E
IF(KEYLH.EQ.0) GO TO 977
DO 975 I=1,NLH
IF(NDAY(KDAY).NE.LHD(I)) GO TO 975
ITHR=23
GO TO 977
975 CONTINUE
977 IF(KDAY.GT.1) IFHR=1
IF(.NOT.KOIN2) IFHR=1
ILHR=ITHR+1
IO(1)=IDSTN
IO(2)=ITAR
IO(3)=ITDY
C
C
C CALL TREFIL(MUNIT, IDSTN, ITAR, ITDY, ILHR, ISNAME)
CALL WRITCOD(KUNIT,IER,INUM)
C
```

C.....DAY BLOCK BASELINE CORRECTED
983 CONTINUE
C
C.....LOOP BACK TO READ ANOTHER DAY FROM NEW INPUT FILE
C
GO TO 410
C
C.....ERROR MESSAGES
15 WRITE(6,2000)
PRINT 2000
2000 FORMAT(20X,'--JOB ABORTED--INCORRECT PRE-POSITIONING OF MASTER',
'1',SCOPE BACKSPACE ON BINARY FILE DID NOT WORK')
STOP
C.....
C.....SUSPECT THE INPUT FILE DOES NOT CONTAIN DATA FOR DESIRED MONTH
C
235 IF(KEYD.NE.0)GO TO 222
237 WRITE(6,2021) ISNAME,IYR2,IDAY1,ITDY2
PRINT 2021, ISNAME,IYR2,IDAY1,ITDY2
2021 FORMAT(1H , '--ERROR--',A3,' YEAR',I5,' INPUT STARTS AT DAY',I4,
'1' MASTER ENDS AT DAY ',I3,/,1H ,'.SUSPECT, WRONG INPUT FILE',/
21H , ' PUNCH NON-ZERO AT COL.25 OF CARD TYPE A ',
3'TO OVERWRITE JOB ABORT')
CALL EREXT
248 PRINT 2004
WRITE(6,2004)
2004 FORMAT(1H ,20X,'JOB ABORTED --DIFFICULTY IN BACKSPACING MASTER',
1/30X,'OR OLD MASTER TIME SEQ WAS WRONG')
CALL EREXT
9964 WRITE(6,8997) MUNIT
PRINT 8997, MUNIT
8997 FORMAT(45X,' ..JOB ABORTED-I/O ERROR ON MINUTE MASTER')
CALL EREXT
9970 WRITE(6,8999) KCARD
PRINT 8999, KCARD
8999 FORMAT(45X,'--JOB ABORTED-CARD TYPE IS ',A1,', EITHER IT IS'
1/,55X,' IN WRONG SEQUENCE OR DOES NOT RELATE TO A KEY')
CALL EREXT
9971 WRITE(6,9000) IDSTN2, IDSTN
PRINT 9000, IDSTN2, IDSTN
9000 FORMAT(45X,'...JOB ABORTED--UNMATCHED OLD AND NEW IDENT',
1' OLD IS',I6,' NEW IS',I7)
CALL EREXT
9972 WRITE(6,9001) IDSTN, IYR2, IYR1
PRINT 9001, IDSTN, IYR2, IYR1
9001 FORMAT(20X,'--JOB ABORTED--STATION ',I6,
1' UNMATCHED YEAR BETWEEN OLD AND NEW,OLD IS',I7, 'NEW IS',I7)
CALL EREXT
CALL SETSCAN
GO TO 82
9998 WRITE(6,9006)ISNAME,NDAY(1),NDAY(KDAY),IYR1
PRINT 9006, ISNAME, NDAY(1), NDAY(KDAY), IYR1

```
9006 FORMAT(1H0,'STATION ',A3,30X,'--END OF JOB--',
 1' DAY ',I3,' TO DAY ',I3,',FOR YEAR',I5,
 2', WAS ADDED TO MASTER FILE')
  GO TO 9999
6800 WRITE(6,6801) IBUNIT
  PRINT 6801, IBUNIT
6801 FORMAT(1H , 'PREMATURE EOF ON UNIT',I3,' DO YOU HAVE',
 1' SUFFICIENT ABSOLUTES ON THE FILE ?')
  STOP
6810 WRITE(6,6811) ICPT,I
  PRINT 6811, ICPT,I
6811 FORMAT(1H , 'INVALID BASELINE CARD',I3,' INSTEAD OF',I3)
9999 CONTINUE
  CLOSE(UNIT=6,STATUS='KEEP')
  CLOSE(UNIT=2,STATUS='KEEP')
  STOP
  END
  SUBROUTINE TREBAK(MUNIT)
C   BACKSPACES A TREORG FILE BY 5 RECORDS, I.E. ONE DAY
10 FORMAT(1H0,'IX= ',I3)
  PRINT 10, IX
  DO 100 I=1,5
  PRINT 10, IX
  BACKSPACE MUNIT
  PRINT 10, IX
  IX = IX+1
100 CONTINUE
  PRINT 10, IX
  READ(MUNIT) IA,IB,IC,ID
  PRINT 200, IA,IB,IC,ID
  WRITE(6,200) IA,IB,IC,ID
200 FORMAT(1H0,'BACKSPACED RECORD IS ',10I8)
  RETURN
  END
  SUBROUTINE TRENUL(MUNIT, IDST, IYR, IDAY, ILHR, ISNAME)
C   NULLS THE DATA FOR ONE DAY
  DIMENSION IDATA(1440),IDUM(100),IO(148)
  DO 10 I=1,1440
10 IDATA(I)=999999
  WRITE(MUNIT) IDST,IYR,IDAD,ILHR
  DO 30 J=1,4
30 WRITE(MUNIT) (IDATA(I),I=1,1440)
  RETURN
  END
  SUBROUTINE TREFIL(MUNIT, IDST, IYR, IDAY, ILHR, ISNAME)
  CHARACTER*3 ISNAME
C   WRITES ONE DAY OF DATA IN TREORG FORMAT
  COMMON IDAT(1536,4),IHRD(1536,4),IO(148), NCOMP
  WRITE(MUNIT) (IO(I),I=1,148)
  WRITE(6,100) IO
C   PRINT 100, IO
100 FORMAT(1H ,10I8,' TREFIL')
```

```
DO 10 J=1,NCOMP
10 WRITE(MUNIT) (IDAT(I,J),I=1,1440)
      WRITE(6,6000) IYR, IDAY, ILHR
CPR  PRINT 6000, IYR, IDAY, ILHR
      DO 11 J=1,4
      WRITE(6,100) (IDAT(I,J),I=1,10)
11  WRITE(6,100) (IDAT(I,J),I=1431,1440)
CC  PRINT 6000, IDST,IYR, IDAY, ILHR
6000 FORMAT(1H ,3I10,' TREFIL')
      WRITE(6,6001) ISNAME,IYR, IDAY, ILHR
CPR  PRINT 6001, ISNAME,IYR, IDAY, ILHR
6001 FORMAT(1H , 'STATION ',A3,' YEAR',I5,' DAY',I4,
1' WRITTEN ON BASELINE-CORRECTED FILE TO HOUR',I3)
      RETURN
      END
      SUBROUTINE SETSCAN
COMMON /CHON/ KCHONLY
LOGICAL KCHONLY
KCHONLY=.TRUE.
PRINT 1000
WRITE(6,1000)
1000 FORMAT(' --SCANNING OF DATA CARDS CONTINUES-----')
      RETURN
      END
      SUBROUTINE RDBLN3(IBSUN, IDSTN)
REAL*8 ISTRT,ISTRV,IENDT,IENDV,SLOPE
CHARACTER*1 TYP
C READ IN BASELINE DATA, CONVERT TIMES TO YEARMINUTES
C WRITE PARAMETERS FOR EACH COMPONENT ON SEPARATE FILE
C MINH ON IBUNIT =20 ICPT = 1
C MAJH ON IBUNIT =21 ICPT = 2
C VERTZ ON IBUNIT =22 ICPT = 3
      5 CONTINUE
      READ(99,10,END=900) ISTN,ICPT,ISTRT,ISTRV,IENDT,IENDV,SLOPE
      *,TYP
10  FORMAT(I6,I4,2(F10.0,F12.3),E16.1,5X,A5)
      IF(ISTN.NE.IDSTN) GO TO 950
C WRITE ON SCRATCH FILES
      IBUNIT=ICPT+19
      WRITE(IBUNIT,11) ICPT,ISTRT ,ISTRV,IENDT ,IENDV,SLOPE
11  FORMAT(I5,4F10.0,E16.10)
      GO TO 5
900 CONTINUE
      REWIND 20
      REWIND 21
      REWIND 22
      WRITE(6,20)
      PRINT 20
20  FORMAT(1HO,'BASELINE VALUES HAVE BEEN READ IN')
      RETURN
950 WRITE(6,21) ISTN, IDSTN
      PRINT 21, ISTN, IDSTN
```

```
21 FORMAT(1H0,'MISMATCH IN STATION ID ON BASELINE FILE',2I10)
      RETURN
      END
      SUBROUTINE EREXT
      DIMENSION DUMMY(2)
11   FORMAT(30HIGNORE ERROR MODE, SEE LISTING)
      PRINT 11
      WRITE(6,11)
      FF=DUMMY(700000)**2
      RETURN
      END
      SUBROUTINE MESSG
      COMMON/KEY/ INREF,KEYD,KEYDR,KEYHR,KEYER,KEYDL,KEYLH,KEYMYR,INUM
      CALL GETDAT(JYR,JMON,JD)
      CALL GETTIM(JHR,JMIN,JSEC,JFRAC)
      WRITE(6,1100) JYR,JMON,JD,JHR,JMIN
      PRINT 1100, JYR,JMON,JD,JHR,JSEC
1100 FORMAT(1H1,50X,'AMOS BASELINE CORRECTION',//,
      140X,'RUN DATE ',I4,2I3,' RUN TIME ',3I3,//)
      WRITE(6,1001)INUM,INREF,KEYD,KEYDR,KEYHR,KEYER,KEYDL,KEYLH,KEYMYR
      PRINT 1001, INUM,INREF,KEYD,KEYDR,KEYHR,KEYER,KEYDL,KEYLH,KEYMYR
1001 FORMAT(25X,'FIRST PROGRAM CONTROL CARD IS-'13I6,/)
      WRITE(6,1000)
      WRITE(6,1000)
      PRINT 1000
      PRINT 1000
      GOTO 1115
      WRITE(6,1101)
      PRINT 1101
1101 FORMAT(1H ,50X,'-----FOLLOWING CONTROLS ACTIVE-----',//)
      IF(INREF.NE.0) WRITE(6,1102)
      IF(INREF.NE.0)PRINT1102
      IF(KEYD.NE.0) WRITE(6,1103)
      IF(KEYD.NE.0)PRINT 1103
      IF(KEYDR.NE.0) WRITE(6,1104)
      IF(KEYDR.NE.0)PRINT 1104
      IF(KEYHR.NE.0) WRITE(6,1105)
      IF(KEYHR.NE.0)PRINT 1105
      IF(KEYER.NE.0) WRITE(6,1107)
      IF(KEYER.NE.0)PRINT 1107
      IF(KEYDL.NE.0) WRITE(6,1111)
      IF(KEYDL.NE.0)PRINT 1111
      IF(KEYLH.NE.0) WRITE(6,1112)
      IF(KEYLH.NE.0)PRINT 1112
      IF(KEYMYR.NE.0) WRITE(6,1114)
      IF(KEYMYR.NE.0)PRINT 1114
      WRITE(6,1000)
      PRINT 1000
1000 FORMAT(1H0,49X,40(1H*),//)
1102 FORMAT(40X,'OLD MASTER ATTACHED')
1103 FORMAT(40X,'PROCESS DESPITE DAY GAP ON NEW INPUT FILE')
1104 FORMAT(40X,'PROCESS DESPITE DAY REVERSAL BETWEEN MASTER',
```

```
1' AND NEW INPUT')
1105 FORMAT(40X,'PROCESS DESPITE HOUR REVERSAL BETWEEN MASTER',
1' AND NEW INPUT')
1107 FORMAT(40X,'READ OVERRIDING ID AND YEAR FROM CARD TYPE D')
1111 FORMAT(40X,'HOUR BLOCKS TO BE MASKED OUT ARE GIVEN',
1' ON TYPE B CARD/S')
1112 FORMAT(40X,'READ FROM PROG CARD TYPE E,DAYS WHERE',
1' LAST HOUR TO BE FORCED TO 23')
1114 FORMAT(40X,'FIRST MASTER WILL START WHEREVER NEW INPUT '
1'FILE STARTS')
1115 RETURN
    END
    SUBROUTINE READCOD(INUNIT,KDAT,IER,INUM)
    COMMON IDAT(1536,4),IHRD(1536,4),IO(148),KO(128),NCOMP
    COMMON ITEMP(128),JTEMP(128)
    DIMENSION KDAT(1536,4)
    IER = 0
    WRITE(6,700)
CPR  PRINT 700
700 FORMAT (1H0,'ENTER READCOD')
800 FORMAT (128I8)
2 READ(INUNIT,800,END=95) (KO(I),I=1,128)
    NCMP =KO(14)+KO(15)+KO(16)+KO(17)+KO(18)+KO(19)+KO(20)
    IF(NCOMP.GT.4)WRITE(6,'(A)') ' TO MANY COMPONENTS'
    WRITE(6,1999) KO(1),KO(5),KO(6),KO(7)
    READ(INUNIT,800,END=95) ITEMP
    READ(INUNIT,800,END=95) JTEMP
C     WRITE(6,200) ITEMP
CC   PRINT 200, ITEMP
    IO(1) = KO(1)
    IO(2) = KO(5)
    IO(3) = KO(6)
    IO(4) = KO(7)
    DO 3 IT=1,128
3 IO(IT+4) = ITEMP(IT)
    DO 4 JT=1,16
4 IO(JT+132) = JTEMP(JT)
C     WRITE(6,200) IO
CC   PRINT 200, IO
200 FORMAT(1H ,10I8,' READCOD')
    DO 10 J=1,NCOMP
    K2 = 0
    DO 11 IREC=1,12
    K1 = K2+1
    K2 = K1+127
11 READ (INUNIT,800,END=55) (KDAT(L,J),L=K1,K2)
    IF (J.EQ.4) GOTO 10
    DO 20 I=1,1440
        IF (IDAT(I,J).EQ.999999) KDAT(I,J)=9999990
20 IDAT(I,J)=IDAT(I,J)/10
C     WRITE(6,1999) IDAT(I,J),IO(6)
10 CONTINUE
```

```
      WRITE(6,1999) IO(1),IO(2),IO(3),IO(4),KDAT(1,1),KDAT(1,3)
C      PRINT 1999, IO(1),IO(2),IO(3),IO(4),KDAT(1,1),KDAT(1,3)
1999 FORMAT(1H ,6I10,' READCOD')
      RETURN
95 CONTINUE
      IER = IER+1
C      IF(IER.EQ.INUM) GO TO 50
      PRINT 100, INUNIT
      WRITE(6,100) INUNIT
100 FORMAT(' END OF FILE FOUND IN S/R READCOD ON UNIT',I2,
1' - LOOKING FOR ANOTHER FILE')
      READ(INUNIT,800,END=50) (IO(I),I=1,128)
      PRINT 120
      WRITE(6,120)
120 FORMAT(' NEXT FILE FOUND, PROCEEDING')
      GO TO 2
50 WRITE(6,105) INUNIT
      PRINT 105, INUNIT
105 FORMAT(' END OF REQUESTED DATA IN S/R READCOD FROM UNIT',I2)
      RETURN
55 WRITE(6,110) INUNIT
      PRINT 110, INUNIT
110 FORMAT(' INCOMPLETE DAY RECORDS FOUND IN S/R READCOD ON UNIT',I2)
      STOP
      END
      SUBROUTINE WRITCOD(KUNIT,IER,INUM)
      COMMON IDAT(1536,4),IHRD(1536,4),IO(148),KO(128),NCOMP
      COMMON ITEMP(128),JTEMP(128)
      WRITE(6,700)
700 FORMAT (1HO,'ENTER WRITCOD')
800 FORMAT (128I8)
      WRITE(KUNIT,800) (KO(I),I=1,128)
      WRITE(6,1999) KO(5),KO(6),KO(7)
1999 FORMAT(1H ,6I10,' WRITCOD')
      WRITE(KUNIT,800) (ITEMP(I),I=1,128)
      WRITE(KUNIT,800) (JTEMP(I),I=1,128)
      DO 10 J=1,NCOMP
      IF (J.EQ.4) GOTO 21
      DO 20 I=1,1440
      IDAT(I,J)=IDAT(I,J)*10
      IF (IDAT(I,J).EQ.9999990) IDAT(I,J)=999999
20 CONTINUE
21 K2 = 0
      DO 11 IREC=1,11
      K1 = K2+1
      K2 = K1+127
11 WRITE(KUNIT,800) (IDAT(L,J),L=K1,K2)
      DO 12 L=1441,1536
12 IDAT(L,J) = 999999
      WRITE(KUNIT,800) (IDAT(L,J),L=1409,1536)
10 CONTINUE
      WRITE(6,1999) KO(5),KO(6),KO(7),IDAT(1,1),IDAT(1,3)
```

```
    RETURN
95 CONTINUE
    WRITE(6,100) KUNIT
100 FORMAT(' END OF FILE FOUND IN S/R WRITCOD ON UNIT',I2)
    STOP
    END
    SUBROUTINE READBIN(MUNIT,KDAT,IER,INUM,MARK)
    COMMON IDAT(1536,4),IHRD(1536,4),IO(148),NCOMP
    DIMENSION KDAT(1536,4)
    IER = 0
    WRITE(6,300)
    PRINT 300
300 FORMAT (1H0,'ENTER READBIN')
    WRITE(6,200) MARK,INUM
    IF (MARK.NE.0) GOTO 2
    READ(MUNIT,END=95) (IO(I),I=1,4)
    WRITE(6,200) (IO(I),I=1,4)
200 FORMAT(1H0,10I8,' READBIN')
2   LL1 = IO(1)
    LL2 = IO(2)
    LL3 = IO(3)
    LL4 = IO(4)
    READ(MUNIT,END=95) (IO(I),I=1,148)
    WRITE(6,200) (IO(I),I=1,148)
    DO 10 J=1,NCOMP
10  READ(MUNIT,END=95) (KDAT(I,J),I=1,1440)
    DO 11 J=1,NCOMP
      WRITE(6,200) (KDAT(I,J),I=1,10)
11  WRITE(6,200) (KDAT(I,J),I=1431,1440)
    GOTO 2
95 CONTINUE
    IER = IER+1
    IF(IER.EQ.INUM) GO TO 50
    WRITE(6,100) MUNIT,IER
100 FORMAT(' END OF FILE FOUND IN S/R READBIN ON UNIT',I2,
1' END CODE',I3,' LOOKING FOR NEW HEADER')
    WRITE(6,102) LL1
    PRINT 102, LL1
    WRITE(6,101) LL2,LL3,LL4
    PRINT 101, LL2,LL3,LL4
101 FORMAT(' LAST DAY FOUND ON PREVIOUS MASTER IS      ',3I5)
102 FORMAT(' STATION IDENT FOR PREVIOUS MASTER IS      ',I7)
    PRINT 103, (KDAT(1440,J),J=1,4)
    WRITE(6,103) (KDAT(1440,J),J=1,4)
103 FORMAT(' LAST VALUES OF THIS DAY(AT 1440) ARE      ',4I10)
    DO 104,J=1,NCOMP
    DO 104,I=1,1440
104 IDAT(I,J) = KDAT(I,J)
    MARK = 1
    RETURN
C    TEMPORARY TRIAL FOR E-O-F
    READ(MUNIT,END=95) IO
```

```
      WRITE(6,120)
120 FORMAT('HEADER FOUND, PROCEEDING')
      GO TO 2
50  WRITE(6,105) MUNIT
      PRINT 105, MUNIT
105 FORMAT(' END OF REQUESTED DATA IN S/R READBIN FROM UNIT',I2)
      WRITE(6,102) LL1
      PRINT 102, LL1
      WRITE(6,101) LL2,LL3,LL4
      PRINT 101, LL2,LL3,LL4
      MARK = 1
      RETURN
55  WRITE(6,110) MUNIT
      PRINT 110, MUNIT
110 FORMAT(' INCOMPLETE DAY RECORDS FOUND IN S/R READCOD ON UNIT',I2)
      STOP
      END
      SUBROUTINE SELSTN (ID,ISTN)
      CHARACTER*3 ISTN
      ISTN=' '
      IF (ID.EQ.21255.OR.ID.EQ.25255) ISTN='CBB'
      IF (ID.EQ.31266) ISTN='FCC'
      IF (ID.EQ.35247) ISTN='MEA'
      IF (ID.EQ.45284) ISTN='OTT'
      IF (ID.EQ.41237) ISTN='VIC'
      IF (ID.EQ.42307) ISTN='STJ'
      IF (ID.EQ.15265) ISTN='RES'
      IF (ID.EQ.14241) ISTN='MBC'
      IF (ID.EQ.35282) ISTN='PBQ'
      IF (ID.EQ.26264) ISTN='BLC'
      IF (ID.EQ.28246) ISTN='YKC'
      IF (ID.EQ.40265) ISTN='WHS'
      IF (ID.EQ.07298) ISTN='ALE'
      IF (ID.EQ.21270) ISTN='PEB'
      IF (ID.EQ.27268) ISTN='RIT'
      IF (ID.EQ.29266) ISTN='EKP'
      IF (ID.EQ.32266) ISTN='BKC'
      IF (ID.EQ.34272) ISTN='FSV'
      IF (ID.EQ.34265) ISTN='GIM'
      IF (ID.EQ.34262) ISTN='TMP'
      IF (ID.EQ.36265) ISTN='ISL'
      IF (ID.EQ.40263) ISTN='GLN'
      IF (ID.EQ.21278) ISTN='IGL'
      RETURN
      END
```

APPENDIX 16

MHVCHECK.FOR

PROGRAM MHVCHECK

C
C VAX VERSION

C MODIFIED BY L NEWITT DEC, 1986

C MODIFIED BY G. JANSEN VAN BEEK APRIL.1988 FOR BATCH SUBMISSION
C JUNE.1988 FOR IAGA FILE WRITING
C JUNE.1990 FOR CLOSURE OF FILES
C JULY.1990 FOR ONE MHV BASE PER DAY

C
C THIS PROGRAM CALCULATES MEAN HOURLY VALUES, DAILY MEANS,HOURLY
C RANGES AND RANGE SUMS FROM THE 1-MINUTE FILE.

C INPUT FILE IS IN *.WKF FORMAT.

C OPTION 1 - OUTPUT FILES OF MHV AND MHR ON TAPE IN IAGA FORMAT

C OPTION 2 - PRINTER PLOT OF DAILY MEANS FOR CHECKING PURPOSES

C OPTION 3 - FULL LISTING OF MHV AND MHR FOR CHECKING PURPOSES

C
C INUM IS THE NUMBER OF FILES ON THE INPUT TAPE (OR FILE).

C IT IS NORMALLY 1, BUT MAY BE GREATER THAN 1 WHEN EXAMINING

C SPOOLED TREORG TAPES

COMMON IO(128),IDAT(1440,4),IMN(25),IMR(25),
1 IQ(62),ID(62),NMON(366),NDATE(366),ISUMR(366),ISUMD(366),
2 IM(366,25),NDAY(366),NMIN(366)
COMMON/COMP/ KOMP,IYR
COMMON/NAME/ IDSTN,ISNAME,ISTNM
DIMENSION IDAMN(366), ITMP1(128), ITMP2(128)
CHARACTER*3 CMR
CHARACTER*1 YESNO, KOMP(3), KOM, IQD(366)
CHARACTER*80 INFILE, ISTNM, ISNAME, FILE1
DATA KOMP/'X','Y','Z'/
INUNIT=1
IOPT0=0
IOPT1=0
IOPT2=0
IOPT3=0

C

C OPEN FILES

C

OPEN(UNIT=31,FILE='QDFILE.DAT',STATUS='OLD')
OPEN(UNIT=10,FILE='STACODE.DAT',STATUS='OLD')

OPEN(UNIT=6,FILE='MHV.DOC',STATUS='NEW')

DO 10 K=21,26

10 OPEN(UNIT=K,STATUS='SCRATCH',FORM='UNFORMATTED')

PRINT 202

202 FORMAT(' ENTER NAME OF INPUT FILE--->')

READ(*,201)INFILE

201 FORMAT(A)

OPEN(UNIT=1,FILE=INFILE,STATUS='OLD',ACCESS='SEQUENTIAL')

C

C READ IN INSTRUCTIONS - OPTIONS REQUIRED AND START DAY

C

C VARIOUS INPUT OPTIONS HAVE BEEN SET TO REPEAT STATION DEFAULT
C

INUM = 1
PRINT 203

203 FORMAT(' ENTER STACODE, YEAR AND START DAY--->')
READ(*,*)IDSTN,IYR,IDAEST
YESNO = 'Y'
IF(YESNO.EQ.'Y'.OR.YESNO.EQ.'y')IOPT3=1
YESNO = 'N'
IF(YESNO.EQ.'Y'.OR.YESNO.EQ.'y')IOPT2=1
YESNO = 'N'
IF(YESNO.EQ.'Y'.OR.YESNO.EQ.'y')IOPT1=1
YESNO = 'N'
IF(YESNO.EQ.'Y'.OR.YESNO.EQ.'y')IOPT0=1

C

C

C READ IN DATA FROM *.WKF BASELINE CORRECTED TAPE
IDAY=IDAEST

C READ FIRST HEADER
READ(INUNIT,800,END=900) (IO(I),I=1,128)

800 FORMAT(128I8)
NCOMP=IO(14)+IO(15)+IO(16)+IO(17)+IO(18)+IO(19)+IO(20)
IF(NCOMP.GT.4)WRITE(6,'(A)') ' TO MANY COMPONENTS'
IF(IO(1).NE.IDSTN) GO TO 910
IF(IO(5).NE.IYR) GO TO 920
REWIND INUNIT
CALL STATCOD(ID2,IDSTN,ISTNM,ISNAME,10)
WRITE(CMR,'(A3)')ISTNM
FILE1='IAGAMIN.'//CMR
IF(IOPT0.EQ.1)OPEN(30,FILE=FILE1,STATUS='NEW',RECL=1464)

100 CALL TRERED(INUNIT,IO,IDAT,IER,INUM,ITMP1,ITMP2,NCOMP)
IF(IER.EQ.INUM) GO TO 200
IF(IO(6).LT.IDAY) GO TO 100
IDAY=IO(6)
CALL CLDATE>IDAY,IYR,MON,IDATE,IO)
LDAY=IDAY
NDAY(LDAY)=IDAY
DO 50 IC=1,3
LUNIT=IC+20
MUNIT=IC+23
CALL MRCALC(IC, IDAY)
WRITE(LUNIT) IDAY,MON, IDATE, (IMN(I), I=1,25), ISUMD(IDAY)
WRITE(MUNIT) IDAY,MON, IDATE, (IMR(I), I=1,25), ISUMR(IDAY)

50 CONTINUE
IF(IOPT0.EQ.1)CALL OUTPT(IO,ITMP1,ITMP2, IDAT)
GO TO 100

200 CALL QDASS(IDAST,LDAY,IOD)
DO 500 KC=1,6
KUNIT=KC+20
REWIND KUNIT
DO 400 LD=IDAEST,LDAY
READ(KUNIT,END=930) NDAY(LD),NMON(LD),NDATE(LD),

```
1 (IM(LD,J),J=1,25),ISUMR(LD)
NMIN(LD)=0
IDAMN(LD)=IM(LD,25)
400 CONTINUE
IB=0
KCC=KC
IF(KC.GT.3) THEN
  KCC=KC-3
ELSE
  CALL MHVBAS(IDAST,LDAY)
ENDIF
KOM=KOMP(KCC)
IF(IOPT1.NE.0) CALL IAGTAP(KC, IDAST, LDAY, KCC, IQD, ISTNM,
IYR,KOMP)
IF(IOPT2.NE.0) CALL GRPH(ISTNM,IYR,NMON,NDATE,IB,KC, IDAMN, IDAST,
LDAY,KOM,IQD,NDAY)
IF(IOPT3.NE.0) THEN
  WRITE(6,'(1H1)')
  CALL LIST(KC,KOM, IDAST, LDAY, IQD, ISTNM, IYR)
ENDIF
500 CONTINUE
WRITE(6,1002)
1002 FORMAT(1H0,'END OF JOB')
CLOSE(UNIT=6)
STOP
900 WRITE(6,1003)
1003 FORMAT(1H0,'ERROR IN OPENING FILE')
STOP
910 WRITE(6,1004)
1004 FORMAT(1H0,'ERROR IN STATION ID')
STOP
920 WRITE(6,1005)
1005 FORMAT(1H0,'ERROR IN YEAR')
STOP
930 WRITE(6,1006)
1006 FORMAT(1H0,'ERROR IN END OF SCRATCH FILE')
STOP
END
SUBROUTINE MRCALC(IC, IDAY)
C.....COMPUTE MHV AND RANGES
DIMENSION IDMN(24)                                     007270
COMMON IO(128),IDAT(1440,4),IMN(25),1MR(25),
1 IQ(62),ID(62),NMON(366),NDATE(366),ISUMR(366),ISUMD(366),
2 IM(366,25),NDAY(366),NMIN(366)
MIND=90000                                           007280
MISIN=0                                              007290
MAXD=-90000                                         007300
DO 1300 I=1,24                                       007310
NMISS=0
NHIT=0
ISUM=0
JKS=(I-1)*60 + 1                                     007330
```

JKF=JKS+59 007340
DO 1250 J=JKS,JKF 007350
IF(NMISS.GT.4) GO TO 1130 007360
ITMP=IDAT(J,IC)
IF(ITMP.GT.80000) GO TO 1150 007370
GO TO 1140 007380
1130 IMN(I)=99999
IMR(I)=9999
C.....INSERT WRITE(UNIT,7000)KOMP(IC),NDAY(KDAY),I TO 007410
C.....PRINT FOR WHICH DAY AND HOUR NO HMV OR RANGE WERE COMPUTED 007420
C.....FROM DATA IN NEW INPUT FILE. REWIND UNIT AT END OF LGO STEP 007430
C.....AND COPY ON TO OUTPUT TO GET THE PRINT OUT 007440
GO TO 1300 007450
7000 FORMAT(1H8,10X,' MISSING HMV+ RANGE FOR COMP',A1,
1'DAY',I4,' HR',I2)
1140 NHIT=NHIT + 1 007480
ISUM=ISUM + ITMP 007490
NMISS=0 007500
IF(NHIT.GT.1) GO TO 1148 007510
MAXC=ITMP 007520
MINC=ITMP 007530
1148 MAXC=MAX0(ITMP,MAXC) 007540
MINC=MIN0(ITMP,MINC) 007550
MAXD=MAX0(ITMP,MAXD) 007560
MIND=MIN0(ITMP,MIND) 007570
GO TO 1250 007580
1150 NMISS=NMISS + 1 007590
MISIN=MISIN+1 007600
1250 CONTINUE 007610
FM=ISUM/NHIT 007620
IMN(I)=FM+SIGN(.5,FM)
IMR(I)=9999
DIFF=IDIM(MAXC,MINC)
SUR=DIFF/10. 007660
IMR(I)=SUR+SIGN(.5,SUR)
C.....IF ANY MINUTE MISSING ADD 9000 TO RANGE
IF(NHIT.LT.60) IMR(I)=IMR(I)+9000
C.....IF MORE THAN 10 MINUTES MISSING AND RANGE GT NRMAX*10, SET RANGE TO 9999
IF(IMR(I).GT.NRMAX.AND.NHIT.LT.50) IMR(I)=9999
1300 CONTINUE 007680
C 007690
C.....COMPUTE DAILY RANGE, SET TO NULL IF ANY MINUTE MISSING 007700
IMR(25)=9999
IF(MISIN.NE.0) GO TO 1301 007720
DIFF=IDIM(MAXD,MIND)
SUR=DIFF/10. 007730
IMR(25)=SUR+SIGN(.5,SUR)
1301 CONTINUE 007750
C
C.....COMPUTE MEAN OF THE HOURLY MEANS
C.....AND DAILY SUM OF THE HOURLY RANGES
C

1390	CONTINUE	008700
	ISUM=0	008720
	NHR=0	
	ISUR=0	
	NHIT=0	
	DO 1400 I=1,24	008730
	IF(IMN(I).GT.80000) GO TO 1395	
	NHR=NHR + 1	008750
	ISUM=ISUM+IMN(I)	
1395	IF(IMR(I).GT.9998) GOTO 1400	
	IF(IMR(I).GT.9000) THEN	
	ISUR=ISUR+IMR(I)-9000	
	ELSE	
	ISUR=ISUR+IMR(I)	
	ENDIF	
	NHIT=NHIT+1	
1400	CONTINUE	008790
	IF(ISUM.NE.0) GO TO 1420	008800
	IMN(25)=99999	
	GO TO 1435	008820
1420	FM= FLOAT(ISUM) / NHR	008830
	IMN(25)=FM+SIGN(.5,FM)	
	IF(NHR.NE.24) IMN(25)=99999	
1435	IF(NHIT.EQ.24) THEN	
	ISUMR(IDAY)=ISUR	
	ELSE	
	ISUMR(IDAY)=9999	
	ENDIF	
	IF(ISUMR(IDAY).GT.9999) ISUMR(IDAY)=9999	
C.....		
C..		
1500	CONTINUE	008880
1510	CONTINUE	008910
	RETURN	
	END	
	SUBROUTINE QDASS(IDAST,LDAY,IQD)	
C.....	ASSIGNING Q OR D TO ALL DAYS	007800
	COMMON IO(128),IDAT(1440,4),IMN(25),IMR(25),	
1	IQ(62),ID(62),NMON(366),NDATE(366),ISUMR(366),ISUMD(366),	
2	IM(366,25),NDAY(366),NMIN(366)	
	CHARACTER*1, IQD(366)	
C....	READ 5 VALUES EACH PER MONTH OF QUIET AND DISTURBED DAYS FROM EACH RECORD	
41	IQDK=5	001190
	READ(31,* ,END=50)(IQ(I),I=1,IQDK),(ID(I),I=1,IQDK)	
C....	READ MORE Q AND D RECORDS WHEN MORE THAN 1 MONTH PROCESSED	
42	ISTK=IQDK+1	001250
	IQDK=IQDK+5	001270
	READ(31,* ,END=50) (IQ(I),I=ISTK,IQDK),(ID(I),I=ISTK,1ODK)	
	GO TO 42	
50	CONTINUE	
	CLOSE(31,STATUS='KEEP')	
	IQDK=IQDK-5	

```
CALL SEQDAY(IQDK)
1302 DO 1310 J=IDAST,LDAY
    IQD(J)=' '
    DO 1310 I=1,IQDK
        IF(NDAY(J).NE.IQ(I)) GO TO 1305
        IQD(J)='Q'
        GO TO 1310
1305 IF(NDAY(J).NE.ID(I)) GO TO 1310
    IQD(J)='D'
1310 CONTINUE
    RETURN
    END
    SUBROUTINE IAGTAP(KC, IDAST, LDAY, KCC, IQD, ISTNM, IYR,
    &KOMP)
C.....PRODUCING IAGA BINARY TAPE FOR MHV AND RANGES
    COMMON           IO(128),IDAT(1440,4),IMN(25),IMR(25),
    1 IQ(62),ID(62),NMON(366),NDATE(366),ISUMR(366),ISUMD(366),
    2 IM(366,25),NDAY(366),NMIN(366)
    CHARACTER*3 CMHV, CMR
    CHARACTER*80 FILE1, FILE2, ISTNM
    CHARACTER*1,IQD(366), KOMP(3)
C
1720 MHVUN=KC+12
    MRUN=KC+15
    IF(KC.LE.3)THEN
        WRITE(CMHV,'(A3)')ISTNM
        FILE1= 'IAGAMHV'//KOMP(KC)//'. '//CMHV
        OPEN(UNIT=MHVUN,FILE=FILE1,STATUS='NEW',RECL=120)
    ELSE
        WRITE(CMR,'(A3)')ISTNM
        IKC=KC-3
        FILE2= 'IAGARG'//KOMP(IKC)//'. '//CMR
        OPEN(UNIT=MRUN,FILE=FILE2,STATUS='NEW',RECL=120)
    ENDIF
    DO 1750 MM=IDAST,LDAY
        NMIN(MM)=NMIN(MM)/100
        IO(2)=IYR-1900
        IO(3)=NMON(MM)
        IO(5)=NDATE(MM)
        IF(KC.LT.4) THEN
            IO(7)=NMIN(MM)
            DO 1740 M=1,24
                J2=7+M
1740        IO(J2)=IM(MM,M)
            IO(32)=IM(MM,25)
            WRITE(MHVUN,900) ISTNM,IO(2),IO(3),KOMP(KCC),IO(5),IQD(MM),
            &(IO(I),I=7,32)
        900 FORMAT(A3,2I2,A1,12,2X,A1,3X,26I4)
        ELSE
            DO 1840 M=1,24
                J1=6+M
1840        IO(J1)=IM(MM,M)
```

```
    IO(31)=ISUMR(MM)
    IO(32)=IM(MM,25)
    WRITE(MRUN,910) ISTNM,IO(2),IO(3),KOMP(KCC),IO(5),IQD(MM),
    &(IO(I),I=7,32)
910 FORMAT(A3,2I2,A1,I2,2X,A1,1X,26I4,2X)
    ENDIF
1750 CONTINUE
    WRITE(6,100)KOMP(KCC)
100 FORMAT(1H , 'COMPONENT ',A1,' WRITTEN IN IAGA FORMAT')
    RETURN
    END
    SUBROUTINE GRPH(ISTN,IYR,MO,IDTE,IBAS,IFL,1DAT,1DAST,N,KOM,IQDD,
* NDAY)
    DIMENSION MO(366),IDTE(366),IDAT(366),LINE(120),NDAY(366)
    CHARACTER*5 MR
    CHARACTER*80 ISTN
    CHARACTER*1, IQDD(366),KOM
C..... FIND MEAN OF IDAT
    JJ=0
    JJ=0
    LDIV=2
    LRAN=121
    MR='MEAN '
    JYR=IYR-1900
    IF(IFL.LT.4) GO TO 9
    IBAS=0
    LDIV=5
    LRAN=300
    MR='RANGE'
9   ISUM=0
    DO 10 J=1DAST,N
    IF(IFL.GT.3) GO TO 11
    IF(IDAT(J).GT.70000) GO TO 10
    GO TO 12
11  IF(IDAT(J).GT.9990) GO TO 10
12  JJ=JJ + 1
    ISUM=ISUM + IDAT(J)
10  CONTINUE
    MEAN=ISUM / JJ
C.....
    LLLOW=MEAN-LRAN
    LTOP= MEAN+LRAN
    DO 20 J=1DAST,N
    IF(IFL.GT.3) GO TO 13
    IF(IDAT(J).GT.70000) GO TO 20
    GO TO 14
13  IF(IDAT(J).GT.9990) GO TO 20
14  IDAT(J)= (IDAT(J)-LLLOW) / LDIV
20  CONTINUE
C.....WRITE HEADER
    WRITE(6,2100) ISTN,MR,KOM,IBAS,LDIV
2100 FORMAT(1H1,60X,'STATION',3X,A3,,
```

```
1      50X,'DAILY ',A5,' GRAPH FOR COMPONENT ',A1,/,  
152X,'ADD',I6,' GAMMAS TO SCALE VALUES',  
2/,53X,'SCALE IS',I3,' GAMMAS PER COLUMN')  
    WRITE(6,2200) LLOW,MEAN,LTOP  
2200 FORMAT(T11,I5,T70,I5,T129,I5)  
    WRITE(6,2300)  
2300 FORMAT(1H , 'YR  MON DAT',1X,120('1'))  
    DO 200 J=IDAST,N  
    DO 40 K=1,120  
40 LINE(K) = 1H  
    INDX=IDAT(J)  
    IF(INDX.GT.120) INDX=120  
    IF(INDX.LT.1) INDX=1  
    LINE(1) = 1HI  
    LINE(60) = 1HI  
    LINE(120) = 1HI  
    LINE(INDX) = 1H*  
    IF(IQDD(J).EQ.'Q') LINE(INDX)=1HQ  
    IF(IQDD(J).EQ.'D') LINE(INDX)=1HD  
    IF(IFL.GT.3) GO TO 42  
    IF(IDAT(J).GT.70000) LINE(INDX)=1H/  
42 IF(IDAT(J).EQ.9999) LINE(INDX)=1H/  
    WRITE(6,2400) JYR,NDAY(J),MO(J),IDTE(J),IODE(J),(LINE(K),K=1,120)  
2400 FORMAT(1H ,I2,I3,1X,I2,1X,I2,A1,120A1)  
200 CONTINUE  
    WRITE(6,2300)  
    WRITE(6,2200) LLOW,MEAN,LTOP  
    WRITE(6,'(1H1)')  
    RETURN  
END  
SUBROUTINE TRERED(MUNIT,IO,IDAT,IER,INUM,ITMP1,ITMP2,NCOMP)  
DIMENSION IDAT(1440,4),ITMP1(128),ITMP2(128)  
DIMENSION IO(128),IN(1568)  
IER=0  
2 READ(MUNIT,800,END=95,ERR=200) (IO(I),I=1,128)  
3 CALL DEFINE(NC,NREC,IO)  
800 FORMAT(128I8)  
FF=IO(22)  
READ(MUNIT,800,END=55,ERR=200)(ITMP1(I),I=1,128)  
READ(MUNIT,800,END=55,ERR=200)(ITMP2(I),I=1,128)  
5 DO 10 J=1,NCOMP  
    CALL FILARA(NREC,IOS,MUNIT,IN,IO)  
    IF(IOS.NE.0)GOTO 55  
    DO 10 I=1,1440  
    IF(J.EQ.4) THEN  
        IDAT(I,J)=IN(I)  
        IF(IDAT(I,J).GT.800000)IDAT(I,J)=999999  
    ELSE  
        IDAT(I,J)=IN(I)/FF  
    ENDIF  
10 CONTINUE  
    RETURN
```

```
95 CONTINUE
  IER=IER+1
  IF(IER.EQ.INUM) GO TO 50
  WRITE(6,100) MUNIT,IER
100 FORMAT(' ENDOF FILE FOUND IN S/R TRERED ON UNIT',I5,
  1' END CODE',I3,' LOOKING FOR NEW HEADER')
  READ(MUNIT,800,END=95) (IO(I),I=1,128)
  WRITE(6,120)
120 FORMAT(' HEADER FOUND, PROCEEDING')
  GO TO 3
  50 WRITE(6,105) MUNIT
105 FORMAT(' END OF REQUESTED DATA IN S/R TRERED FROM UNIT',I5)
  RETURN
  55 WRITE(6,110) MUNIT
110 FORMAT(' INCOMPLETE DAY RECORDS FOUND IN S/R TRERED ON UNIT',I5)
  STOP
200 WRITE(6,201) (IO(I),I=1,10)
201 FORMAT(1H0,10I10)
  END
  SUBROUTINE CLDATE (DAY,YR,MNTH,DATE,IO)          010680
C                                                 010690
C.....THIS SUBROUTINE DETERMINES THE DATE OF A GIVEN SEQUENTIAL 010700
C.....DAY OF AN YEAR                                         010710
C.....DAY,YR,AND DATE MUST BE DECLARED INTEGERS IN THE MAIN PROGRAM 010720
C.....DAY IS THE SEQUENTIAL DAY OF THE YEAR YR --BOTH INPUT      010730
C.....TO THE SUBROUTINE                                         010740
C.....MNTH IS THE OUTPUT MONTH IN INTEGER                      010760
C.....DATE IS THE OUTPUT DATE IN NUMERIC                         010770
C
  INTEGER DATE,DAY,YR                                         010780
  DIMENSION IA(12),IO(128)
  CHARACTER*5 MOTH(12)
  DATA MOTH/'JAN. ','FEB. ','MAR. ','APR. ','MAY ','JUNE ',
  1'JULY ','AUG. ','SEPT. ','OCT. ','NOV. ','DEC. '/          010800
  DATA IA/31,28,31,30,31,30,31,31,30,31,30,31/               010810
  INDLP=0                                                       010820
C                                                 010830
C
C.....LEAP YEAR TEST                                         010840
C
  LT=YR-100*(YR/100)                                         010850
  IF(LT)30,30,40                                              010860
30  LT=YR-400*(YR/400)                                         010890
  IF(LT)50,45,50                                              010900
40  LT=YR-4*(YR/4)                                            010910
  IF(LT)50,45,50                                              010920
45  INDLP=1                                                    010930
50  IF(INDLP-1)57,55,57                                         010940
55  IA(2)=29                                                   010950
  GO TO 60                                                   010960
57  IA(2)=28                                                   010970
60  ICOL=0                                                    010980
  DO 65 I=1,12                                              010990
```

```
ICOL=ICOL+IA(I)          011000
IF(ICOL.GE.DAY)GO TO 70  011010
65 CONTINUE               011020
70 DATE=DAY-(ICOL-IA(I)) 011030
  MNTH=I                  011040
  IF(MNTH.NE.IO(4)) GOTO 100
  IF(DATE.NE.IO(3)) GOTO 100
  RETURN
100 WRITE(6,110)
110 FORMAT(1H0,'Error in month and day specification in the data',
*' file.')
  WRITE(6,120)(IO(I),I=1,13)
120 FORMAT(1H , 'Error header reads',13I8)
  RETURN                   011050
  END                      011060
  SUBROUTINE LIST(KC,KOM,IDASt,LDAY,IQD,ISTNm,IYR)
C* PRINTS OUT MHV OR MHR
C
COMMON           IO(128),IDAT(1440,4),IMN(25),IMR(25),
1 IQ(62),ID(62),NMOn(366),NDATE(366),ISUMR(366),ISUMD(366),
2 IM(366,25),NDAY(366),NMIN(366)
CHARACTER*1 IQD(366),KOM
CHARACTER*80 ISTNm
C
I2YR=IYR-1900
DO 100 LD=IDASt,LDay
WRITE(6,'(1X)')
WRITE(6,1001) ISTNm,I2YR,NDAY(LD),NMOn(LD),NDATE(LD),KOM,
1 NMIN(LD),(IM(LD,J),J=1,12)
1001 FORMAT(1H ,A3,I3,I4,I3,I3,1X,A1,2X,I5,2X,12I4)
  WRITE(6,1002)(IM(LD,J),J=13,25)
1002 FORMAT(28X,13I4)
100 CONTINUE
RETURN
END
SUBROUTINE MHVBAS(IDAST,KDAY)
C.....COMPUTING BASE VALUES AND SUBTRACTING THEM FROM HMV      009030
COMMON/NAME/ IDSTn,ISNAME,ISTNm
COMMON/COMP/ KOMP(3),IYR
COMMON           IO(128),IDAT(1440,4),IMN(25),IMR(25),
1 IQ(62),ID(62),NMOn(366),NDATE(366),ISUMR(366),ISUMD(366),
2 IM(366,25),NDAY(366),NMIN(366)
C
DO 1600 M=IDASt,KDAY
MIN=IM(M,25)
DO 1555 MM=1,24                                         009074
MIN=MIN0(MIN,IM(M,MM))
1555 CONTINUE
MIN=MIN-500
ORD= FLOAT(MIN)/1000.
NMIN(M)=IFIX(ORD+SIGN(.5,ORD))*1000
ISUMD(M) =0
009076
009080
009090
```

```
DO 1600 MM=1,25
IF(IM(M,MM).GT.80000) GO TO 1590
IM(M,MM)=IM(M,MM)-NMIN(M)
IF(MM-25)1580,1600,1580
1580 ISUMD(M)=ISUMD(M)+IM(M,MM)
GO TO 1600
1590 IM(M,MM)=9999
1600 CONTINUE
IF(ISUMD(M).EQ.0) ISUMD(M)=9999
009180
C
RETURN
END
SUBROUTINE STATCOD(ID2L, ID6L, STA, STA10, IUNIT)
CHARACTER*80 STA, STA10, XSTA, XSTA10
REWIND IUNIT
100 READ(IUNIT,10,END=90)JD2L,JD6L,XSTA,XSTA10
10 FORMAT(I2,1X,I6,1X,A3,1X,A10)
IF(JD2L.EQ.ID2L.OR.JD6L.EQ.ID6L.OR.XSTA.EQ.STA
*.OR.XSTA10.EQ.STA10)THEN
ID2L=JD2L
ID6L=JD6L
STA=XSTA
STA10=XSTA10
CLOSE(IUNIT,STATUS='KEEP')
RETURN
ENDIF
GO TO 100
90 ID2L=99
ID6L=999999
STA='XXX'
STA10='XXXXXXXXXX'
CLOSE(IUNIT,STATUS='KEEP')
RETURN
END
SUBROUTINE DEFINE(NC,NREC,ID)
C
C G. JANSEN VAN BEEK 1987.335
C
DIMENSION ID(128)
C
C SUBROUTINE DEFINES THE NUMBER OF COMPONENTS AND THE NUMBER
C OF RECORDS PER COMPONENT THAT EXIST FOR THE DATA TIME INTERVAL
C FOUND IN THE DATA HEADER ARRAY "ID"
C
C CALCULATE THE NUMBER OF COMPONENTS PRESENT
C
NC=0
DO 10 I=14,20
IF(ID(I).NE.0)NC=NC + 1
10 CONTINUE
C
C CALCULATE THE NUMBER OF RECORDS PER COMPONENT
```

C

```
MMDAY=ID(7)*60 + ID(8)
NMDAY=ID(10)*60 + ID(11)
MNDIF=NMDAY-MMDAY + 1
NREC=MNDIF/128.
IF(MOD(MNDIF,128).NE.0) NREC=NREC+1
RETURN
END
SUBROUTINE FILARA(NREC,IOS,MUNIT,IN,ID)

C G. JANSEN VAN BEEK 1987.335
C
C DIMENSION IN(1568),NN(128),ID(128)
C
C SUBROUTINE READS THE DATA FOR ONE COMPONENT DAY INTO THE 1440 DATA
C POINT ARRAY "XIN"
C
NSTRRT=ID(7)*60 + ID(8)
DO 20 K=1,NREC
READ(MUNIT,900,IOSTAT=IOS) (NN(I),I=1,128)
IF (IOS.NE.0) RETURN
900 FORMAT(128I8)
M=1
DO 10 L=NSTRRT,NSTRRT+127
IN(L)=NN(M)
M=M + 1
10 CONTINUE
NSTRRT=NSTRRT + 128
20 CONTINUE

C
C
RETURN
END
SUBROUTINE SEQDAY(1QDK)

C G. JANSEN VAN BEEK VERSION 119.1988
C
C THIS PROGRAM TAKES THE DAY AND MONTH AND CONVERTS IT INTO SEQUENTIAL
C DAY OF THE YEAR FOR THE QUIET AND DISTURBED DAY LISTS
C
COMMON IO(128),IDAT(1440,4),IMN(25),IMR(25),
1 IQ(62),ID(62),NMON(366),NDATE(366),ISUMR(366),ISUMD(366),
2 IM(366,25),NDAY(366),NMIN(366)

C
DIMENSION IA(12)
DATA IA/31,28,31,30,31,30,31,31,30,31,30,31/
INDLP=0

C.....LEAP YEAR TEST
C
IYR=IO(5)
LT=IYR-100*(IYR/100)
```

```
IF(LT)30,30,40
30 LT=IYR-400*(IYR/400)
   IF(LT)50,45,50
40 LT=IYR-4*(IYR/4)
   IF(LT)50,45,50
45 INDLP=1
50 IF(INDLP-1)57,55,57
55 IA(2)=29
   GO TO 60
57 IA(2)=28
C
60 MNTH=0
   ICOL=0
   DO 220 J=1,IQDK,5
   ISTA=J
   ISTO=J+4
   MNTH=MNTH+1
   IF(MNTH.GT.1) ICOL=ICOL+IA(MNTH-1)
210 DO 220 I=ISTA,ISTO
   IQ(I)=IQ(I)+1COL
   ID(I)=ID(I)+ICOL
220 CONTINUE
   RETURN
END
SUBROUTINE OUTPT(ID,ITMP1,ITMP2,IDAT)
C
C      VERSION 1988.159    G. JANSEN VAN BEEK
C
DIMENSION ID(128),ITMP1(128),ITMP2(128)
DIMENSION IDAT(1440,4)
C
C      SUBROUTINE TO OUTPUT THE MINUTE INFORMATION FOR ONE DAY INTO AN
C      IAGA ASCII FILE
C
C      IAGA STATION NUMBER          = ID(1)
C      YEAR                         = ID(5)
C      SEQUENTIAL DAY OF THE YEAR   = ID(6)
C      HOUR OF THE DAY             = IHR
C
DO 10 I=1,24
IHR=I-1
IEND=I*60
ISTART=IEND-59
WRITE(30,800)ID(1),ID(5),ID(6),IHR,((IDAT(MIN,1C),1C=1,4),
*MIN=ISTART,IEND)
800 FORMAT(244I6)
10 CONTINUE
C
RETURN
END
```

APPENDIX 17

FFPLT.FOR

```
CC      PROGRAM FFPLT
C
C version for plotting F-F* values to pc screen
C based on wkfplt.for and adapted by L. Newitt, Sept, 1990
C compiled using MS-Fortran with graphics
C
INCLUDE 'FGRAPH.FI'
INCLUDE 'FGRAPH.FD'
CHARACTER*20 INFILE,NAME
CHARACTER*40 CHARS
DIMENSION BUF(1),IDAT(128),DAT(128),ICOM(8),VAL(1536)
CHARACTER*1 ICOMP(8),COMP,BX
CHARACTER*2 STR1
RECORD /XYCOORD/ XY
RECORD /WXYCOORD/WXY
DATA ICOMP/'X','Y','Z','D','H','F','I','*'/
C
C PLOT 'GRAMS FROM *.WKF TYPE FILES AT VARIABLE MM/HR ON ZETA
C PLOTTER FOR DISPLAY PURPOSES
C
C VERSION 358.1987 G. JANSEN VAN BEEK
C
WRITE(*,'(1H$,A)') 'Input file -----> '
READ(*,880) INFILE
880 FORMAT(A)
OPEN(15,FILE=INFILE,STATUS='OLD',ACCESS='SEQUENTIAL')
WRITE(*,'(1H$,A)') 'Start day, end day and scale value -----> '
READ(*,*) KDAY,KFDAY,XSCALE
ITMSCL = 10
IBX=0
IDA = KDAY
IF(XSCALE.EQ.0.0)XSCALE=20
FSC = XSCALE
FIN=0.0
KOUNT=0
XB=0.0
NFILE=1
YB=0.0
IBOX=0
SCALE=25.4*XSCALE
C
C KDAY=INITIAL STARTING DAY AFTERWARDS THE CURRENT DAY BEING
C PROCESSED
C KFDAY=FINAL DAY THAT IS REQUESTED TO BE PLOTTED
C IDA=START DAY THAT IS REQUESTED TO BE PLOTTED
C SMPRT=NUMBER OF DATA POINT PER HOUR
C ISMPRT=SAMPLE RATE IN SECONDS
C
10 read(15,800,IOSTAT=IOS) (IDAT(I),I=1,128)
IF(IOS.LT.0) GOTO 999
IF(IOS.GT.0) GOTO 16
YB=0.0
```

```
C.....WRITE(*,'(1H0,13I8)')(IDAT(I),I=1,13)
    II=1
    DO 11 JJ=14,21
    ICOM(II)=IDAT(JJ)
    II=II+1
11 CONTINUE
    KOUNT = KOUNT + 1
    FYR = IDAT(5)
    IDAY = IDAT(6)
    DTSCl=IDAT(22)
    DIDENT = IDAT(2)
    SMPRT = 3600/IDAT(13)
    ISMPRT = IDAT(13)
1000 IBG = IDAT(7)*3600 + IDAT(8)*60 + IDAT(9)
    IEN = IDAT(10)*3600 + IDAT(11)*60 + IDAT(12)
    NUM = (IEN-IBG+IDAT(13))/IDAT(13)
C
C      IBG=time of first data point in seconds of the day
C      IEN=time of the last data point in seconds of the day
C      NUM=number of data points per component until the end of the day
C      IRR=number of records per component until the end of the day
C      NC=number of components in the data file
C      ISTRT=sequential number of the first data point of the day
C      DTSCl=data multiplication factor
C      NMAX=sequential number of the last data point of the day
C
800 format(128I8)
    NC = 0
    DO 1010 I=14,21
    IF(IDAT(I).NE.0) NC=NC+1
1010 CONTINUE
    READ(15,810,ERR=16,END=15) IBUF
810 FORMAT(I8)
    KOUNT = KOUNT + 1
    READ(15,810,ERR=16,END=15) IBUF
    KOUNT=KOUNT+1
    IRR = NUM/128
    N = MOD(NUM,128)
    IF(N.NE.0) IRR=IRR+1
    ISTRT = IBG/ISMPRT
    NMAX = IEN/ISMPRT
    GO TO 20
15 FIN=5.0
    GO TO 25
16 KOUNT = KOUNT + 1
    WRITE(*,5992)KOUNT
5992 FORMAT(1H0,'ERROR IN RECORD NO. ',I5)
    WRITE(*,'(1H0,13I8)')(IDAT(I),I=1,13)
    WRITE(*,'(1H ,13F7.0)') (DAT(IN),IN=1,13)
    WRITE(*,'(1H ,A,I10)')'READ ERROR NO. ',IOS
    GOTO 9999
20 CONTINUE
```

```
21 IF(IDAY.LT.KDAY) GO TO 250
    IF(IDAY.GT.KFDAY) GOTO 999
25 continue
    GOTO 257
250 DO 255 I=1,NC
    DO 255 J=1,IRR
        READ(15,810,ERR=16,END=15) IBUF
        KOUNT = KOUNT + 1
255 CONTINUE
    GOTO 10
C
C      YD=MEAN POSITION OF COMPONENT TRACE
C      XB,YB=PLOT ORIGIN OF EACH 'GRAM'
C      X,Y=CURRENT PEN COORDINATES
C      ENTR=TOTAL LENGTH OF THE COMPONENT TRACE FOR A FULL DAY
C
C.....DATA OBTAINED FOR ONE FULL DAY
C*****ESTABLISH THE DISTANCE IN INCHES BETWEEN EACH DATA POINT
C*****FINC=THE PLOTTING DISTANCE/HOUR DIVIDED BY THE NO. OF
SAMPLES/HOUR
257 FINC = ITMSCL/(SMPRT*25.4)
    ENTR = FINC*86400/ISMPRT
C
C CALCULATES F-F* VALUES FOR 1 DAY AND WRITES TO SCRATCH FILE
C
OPEN (UNIT=20,STATUS='SCRATCH',RECL=1024)
DO 499 N=1,1536
499 VAL(N)=0.
    DO 500 K=1,4
        MM=1
        DO 500 J=1,IRR
            READ(15,800,END=15)(IDAT(I),I=1,128)
            DO 510 II=1,128
                DAT(II)=IDAT(II)
                DAT(II)=DAT(II)/10.
                IF(K.EQ.1)VAL(MM)=DAT(II)*DAT(II)
                IF(K.EQ.2)VAL(MM)=VAL(MM)+DAT(II)*DAT(II)
                IF(K.EQ.3)VAL(MM)=VAL(MM)+DAT(II)*DAT(II)
                IF(K.EQ.4)THEN
                    DAT(II)=DAT(II)-SQRT(VAL(MM))
                    IDAT(II)=DAT(II)*10.
                    IF (ABS(IDAT(II)).GT.10000)IDAT(II)=999999
                ENDIF
                MM=MM+1
510 CONTINUE
    IF(K.EQ.4)WRITE(20,800)(IDAT(I),I=1,128)
500 CONTINUE
    REWIND 20
    CALL GRAPHICSMODE()
    CALL SETVIEWORG(10,10,XY)
    DUMMY = SETWINDOW(.TRUE.,-1.,-1.,11.5,10.)
C      CALL POS(IBOX,XB,YB,IBX)
```

```
X=XB
Y=YB
INCT=3.0
YD=YB+INCT
FDY=FLOAT(IDAY)
DO 400 J=1,1
OPEN(UNIT=12,STATUS='SCRATCH')
C.....WORKING SPACE FOR EACH COMPONENT TRACE IS +/- 2 INCHES
YMX=YD+2.5
YMN=YD-2.5
IF(YMN.LT.-1.0) YMN=-1.0
IF(YMX.GT.10.) YMX=10.
NAV=0
TOT=0.0
C.....WRITE(*,'(1H0,10I8)') IBG,IEN,NUM,NMAX
C.....WRITE(*,'(1H0,10I8)') NC,IRR
DO 50 I=1,IRR
C.....ASSUME THE DATA SCALE (IDAT(22)) IS SET TO 10
READ(20,800,ERR=16,END=15) (IDAT(IN),IN=1,128)
820 FORMAT(128F8.1)
C.....WRITE(*,'(1H ,13F7.0)') (DAT(IN),IN=1,13)
KOUNT = KOUNT + 1
DO 50 II=1,128
DAT(II)=IDAT(II)/DTSCALE
WRITE(12,840) DAT(II)
IF(IDAT(II).GT.800000) GO TO 50
NAV=NAV+1
TOT=TOT+DAT(II)
50 CONTINUE
REWIND 12
IF(NAV)65,65,53
53 FMN=TOT/NAV
X = XB
XS=X-0.2
LW=0
CALL SYMK(XS,YD,FMN,LW)
IT=0
DO 531 JJ=1,7
IF(ICOM(JJ).NE.0)IT=IT+1
IF(IT.EQ.J) GOTO 532
531 CONTINUE
JJ=8
532 COMP=ICOMP(6)
YDD=YD+0.2
CALL MOVETO_W(XS,YD,WXY)
CALL GETCURRENTPOSITION(XY)
CALL SETTEXTPOSITION(XY.YCOORD/8+2,XY.XCOORD/8,XY)
CALL OUTTEXT(COMP)
IF(NUM.EQ.NMAX) GOTO 530
X = XB + (ISTRRT)*FINC
GOTO 535
530 CONTINUE
```

```
535 READ(12,840,END=400) FV
840 FORMAT(F10.3)
55 IF(FV.GT.80000.) FV=FMN
    Y=(FV-FMN)/SCALE + YD
    IF(Y.GT.YMX) Y=YMX
    IF(Y.LT.YMN) Y=YMN
    Y1=Y
    CALL MOVETO_W(X,Y,WXY)
    READ(12,840,END=400)FV
    DO 60 II=ISTRRT+1,NMAX
    X=X+FINC
    IF(FV.GT.80000.)THEN
    CALL MOVETO_W(X,Y,WXY)
    ELSE
    Y=(FV-FMN)/SCALE+YD
    IF(Y.LT.YMN) GO TO 58
    IF(Y.GT.YMX) GO TO 59
    GO TO 581
58 Y=YMN
    GO TO 581
59 Y=YMX
581 CONTINUE
    DUMMY=LINETO_W(X,Y)
    ENDIF
    Y1=Y
    IF(II.LT.NMAX+1)READ(12,840,END=400)FPLUS
    IF(FPLUS.GT.80000.)CALL MOVETO_W(X,Y,WXY)
    IF(FV.GT.80000..AND.FPLUS.LT.88000.) THEN
        XPLUS=X+FINC
        FPLUS1=(FPLUS-FMN)/SCALE + YD
        CALL MOVETO_W(XPLUS,FPLUS1,WXY)
    ENDIF
    FV=FPLUS
60 CONTINUE
    XS=X+0.2
    IF(NMAX.NE.86400/ISMPRT) XS=XB+ENTR+0.2
    LW=1
    CALL SYMK(XS,YD,FMN,LW)
    YDD=YD+0.2
    CALL MOVETO_W(XS,YDD,WXY)
    CALL GETCURRENTPOSITION(XY)
    CALL SETTEXTPOSITION(XY.YCOORD/8+2,XY.XCOORD/8+5,XY)
    CALL OUTTEXT(COMP)
    GO TO 69
65 X=XB + NMAX*FINC/2.0
    chars = 'NO DATA'
C
    X=XB
69 XNX=2.25
    IF(IBX.EQ.1) XNX=1.0
    YD=YD+XNX
    CLOSE (12,STATUS='DELETE')
```

```
CLOSE (20,STATUS='DELETE')
400 CONTINUE
C.....INITIALIZE ARRAY
Y=YB+0.4
X = XB
CALL MOVETO_W(X,Y,WXY)
C.....WRITE(*,'(1H ,A)')'Start to plot the hour tiks.'
DO 450 J=1,24
CALL TIKS(X,Y,0.1,0.0)
IF(MOD(J-1,6).NE.0) GO TO 452
FJ = FLOAT(J-1)
IF(FJ.EQ.24.OR.FJ.EQ.0) GO TO 452
CALL MOVETO_W(X,Y-0.2,WXY)
CALL GETCURRENTPOSITION(XY)
CALL SETTEXTPOSITION(XY.YCOORD/8+3,XY.XCOORD/8+2,XY)
IFJ=FJ
WRITE(STR1,'(I2)')IFJ
CALL OUTTEXT(STR1)
CALL MOVETO_W(X,Y,WXY)
452 CONTINUE
X=X+FINC*SMPRT
DUMMY = LINETO_W(X,Y)
450 CONTINUE
C.... WRITE(*,'(1H A)')'End of hour tik plotting.'
KDAY=IDAY
IF(FIN.NE.0.0) GO TO 999
IBOX=IBOX+1
CALL SETTEXTPOSITION(26,5,XY)
PRINT 151, IDAY
151 FORMAT(1H0,' DAY',I4,' PLOTTED, PRESS ANY KEY TO CONTINUE--->')
READ(*,'(A)')ICONT
CALL ENDGRAPH()
GO TO 10
999 CONTINUE
X = 1
9999 CONTINUE
CLOSE(12,STATUS='DELETE')
WRITE(*,'(1H ,A)') 'Plot file name is "WKFPLT.PLT".'
CALL ENDGRAPH()
END
SUBROUTINE POS(IBX,X,Y,IB)
IF(IBX.LT.1) RETURN
XINC=1.0
IF(IB.EQ.0) GOTO 10
IF(MOD(IBX,2).NE.0) GOTO 50
10 X=XINC
Y=0.0
RETURN
50 Y=5.0
IF(IBX.EQ.1)XINC=1.5
X=XINC
RETURN
```

```
END
SUBROUTINE TIKS(XM,YM,HI,ANG)
INCLUDE 'FGRAPH.FD'
RECORD /WXYCOORD/WXY
RECORD /XYCOORD/XY
X=XM
Y=YM
IF(X.NE.999..OR.Y.NE.999.) GO TO 40
CALL GETCURRENTPOSITION(XY)
40 CONTINUE
X1=X
Y1=Y+HI
Y2=Y-HI
IF(ABS(ANG).NE.90.) GO TO 50
S=SIGN(1.,ANG)
X1=X+HI*S
Y1=Y
50 CALL MOVETO_W(X1,Y1,WXY)
DUMMY = LINETO_W(X1,Y2)
CALL MOVETO_W(X,Y,WXY)
RETURN
END
SUBROUTINE SYMK(XT,YT,FMN,KEY)
INCLUDE 'FGRAPH.FD'
RECORD /WXYCOORD/ WXY
RECORD /XYCOORD/POSITION
CHARACTER*5, STR
CALL MOVETO_W(XT,YT,WXY)
DUMMY = LINETO_W(XT+0.1,YT)
DUMMY = LINETO_W(XT-0.1,YT)
DUMMY = LINETO_W(XT,YT)
DUMMY = LINETO_W(XT,YT+0.1)
DUMMY = LINETO_W(XT,YT-0.1)
DUMMY = LINETO_W(XT,YT)
YS=YT+0.1
IF(KEY.LT.1) RETURN
CALL MOVETO_W(XT+0.2,YS,WXY)
CALL GETCURRENTPOSITION(POSITION)
CALL SETTEXTPOSITION(POSITION.YCOORD/8+3,POSITION.XCOORD/8+5,
*POSITION)
IFMN =FMN
WRITE(STR,'(I5)') IFMN
CALL OUTTEXT(STR)
RETURN
END
SUBROUTINE GRAPHICSMODE()
INCLUDE 'FGRAPH.FD'
RECORD /VIDEOCONFIG/ VDINFO
CALL GETVIDEOCONFIG(VDINFO)
DUMMY = SETVIDEOMODE(6)
RETURN
END
```

-A17- 9-

```
SUBROUTINE ENDGRAPH()
INCLUDE 'FGRAPH.FD'
DUMMY = SETVIDEOMODE( $DEFAULTMODE)
RETURN
END
```