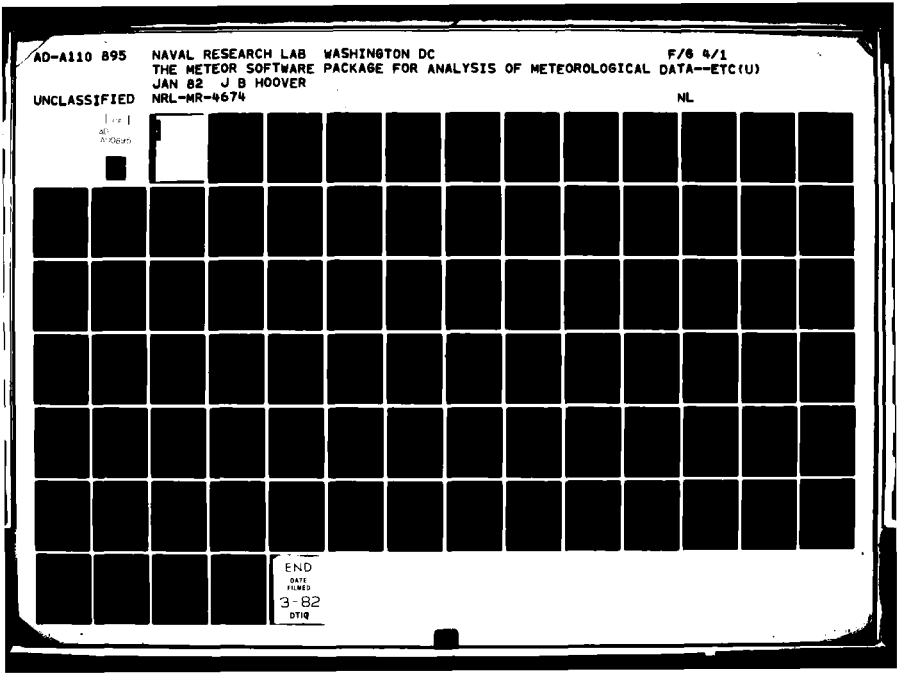
AD-A110 895 NAVAL RESEARCH LAB WASHINGTON DC F/6 4/1
THE METEOR SOFTWARE PACKAGE FOR ANALYSIS OF METEOROLOGICAL DATA--ETC(U)
JAN 82 J B HOOVER
UNCLASSIFIED NRL-MR-4674 NL



END
DATE FILMED
3-82
DTIC

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: NRL Memorandum Report 4674 — 2. GOVT ACCESSION NO. — 3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): THE METEOR SOFTWARE PACKAGE FOR ANALYSIS OF METEOROLOGICAL DATA	5. TYPE OF REPORT & PERIOD COVERED: Report on a continuing NRL problem.
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): John B. Hoover	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Research Laboratory, Washington, D.C. 20375	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61153N; 43-1130-0-2
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE: January 27, 1982
	13. NUMBER OF PAGES: 88
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

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

18. SUPPLEMENTARY NOTES

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

Meteorology
Data acquisition
Data processing
FORTRAN
Computer data processing
collecting and analyzing

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

The METEOR software package was developed at NRL to aid in the collection and analysis of meteorological data. It consists of four FORTRAN programs and is currently in use on a DEC System-10. All four programs were designed for ease of use and require a minimal amount of effort on the part of the user.

The programs in this package are capable of sorting data according to source, creating large data files, processing these files, and producing printed and plotted output as required.

(Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Continued)

—Versatility was a prime objective in creating these programs. They have been written in a modular fashion so that functions may easily be added or changed. In addition, the user may select from a variety of options within each program, thus allowing the software to be tailored to specific requirements at execution time. This is accomplished through the use of a command file which provides instructions to the various programs.

One ~~of this~~ programs is interactive and conducts a dialog with the user in order to ascertain what data is to be processed and which functions are to be performed. The output of this program is the command file described above.

The remaining three programs require no interaction and may be used in a batch mode. Each of these produces error messages whenever unexpected conditions are encountered. Insofar as possible, these messages were intended to be self-explanatory. If more details are required, each program also contains a complete list of error messages and an explanation of each.

This report includes descriptions of the programs, examples of the required inputs, and copies of typical program outputs. Complete source listings are also provided in the appendices.

Acknowledgements

The author would like to thank Katherine Schwarz and Joseph Liu for their assistance in writing programs METCLC and METPLT during their participation in the American University/Naval Research Laboratory Research Apprentice Program.

Elizabeth Hill and Jay Oberfield of the Research Computation Division were instrumental in debugging the programs.

This work was performed while the author was a National Research Council Research Associate at NRL.

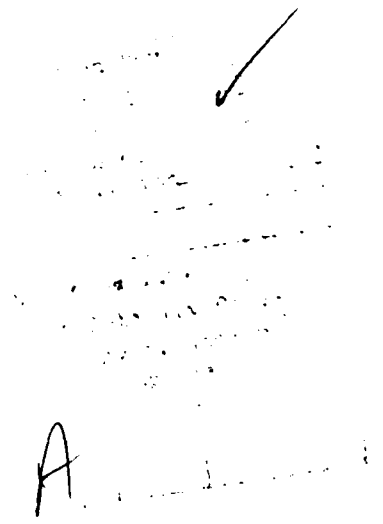


Table of Contents

List of Figures and Tables

The METEOR Software Package for Analysis of Meteorological Data

I. INTRODUCTION

A requirement for many experiments in environmental chemistry is that extensive meteorological records must be maintained. Frequently, simultaneous collection of several different types of data (temperature, humidity, pressure, etc.) at regular intervals (minutes to hours) over rather long, continuous periods (days or weeks) is necessary. The use of modern data loggers greatly simplifies the acquisition of data and often provides for storage and transfer on computer-readable media, such as magnetic tape. Off-line analysis involves conversion of the data to physically meaningful units, calculation of derived quantities, and presentation of the results in formats which are convenient for the user's purposes. Due to the large quantities of data, these steps are usually very time-consuming.

METEOR is a FORTRAN program package which is designed to alleviate many of these difficulties. It provides software for inputting data files, searching out relevant portions of these files, processing data, and generating printed or plotted output.

The goal has been to maximize the generality of the program while minimizing the demands on the operator. The former goal has been addressed by the liberal use of subroutines and functions, making program expansion and alteration a simple matter of inserting new or updated program modules.

The requirement that the package be easy to use has led to the development of an auxiliary, interactive program which aids in the creation of a command file. This file then controls the execution of the primary programs. Currently, there are three primary programs (METSRT, METCLC, and METPLT) and one auxiliary program (METINP) in the METEOR package. The relationship among these programs is illustrated in Figure 1.

METSRT (METeorological data SoRTing program) is responsible for input of raw data, selection of the data required for analysis, preliminary processing (conversion of units and scaling, where necessary), testing of the data for various error conditions, listing of selected data in tabular form, and creation of a file containing all processed data.

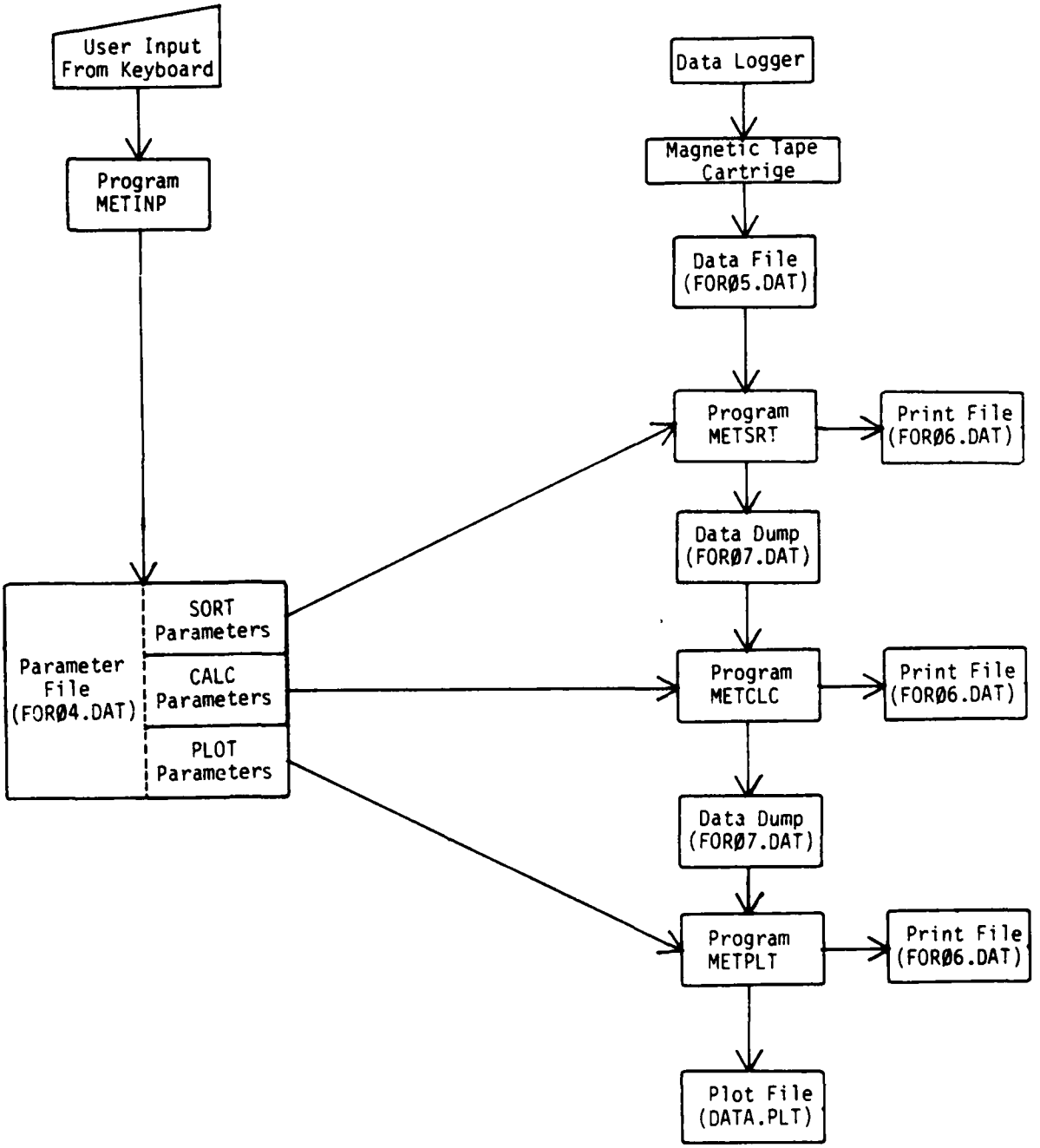
METCLC (METeorological data CaLCulation program) then operates on the output of METSRT, calculating the values of various derived quantities (total moisture loading, for example). These results are added to the file of processed data and may also be printed out, again in tabular form.

METPLT (METeorological data PLoTting program), the last of the primary programs, reads the data file produced by the previous programs and generates selected graphs.

It is possible to string these programs together and to process the data, from raw data input to finished plots, in one continuous batch job. However, the sequential design of the programs, with intermediate outputs, was intended to allow easy operator intervention in the event that bad data is encountered.

Manuscript submitted October 30, 1981.

Figure 1
Relationship of the programs and files
in the METEOR program package.



The auxiliary program, METINP, (METeorological parameter INPut program), is interactive and is intended to be run from a CRT terminal. This program requests information regarding the specific functions in METSRT, METCLC, and METPLT that are desired and constructs a control file in the appropriate format. This file is displayed line-by-line for operator verification before being written onto the disk. Errors may easily be corrected at this point.

The METEOR package is currently running on a DEC System-10 and the I/O file names discussed below are those used by the TOPS-10 operating system. No major problems are expected to arise if the programs are transfered to another system (in fact, earlier versions of METSRT and METPLT were run on a Texas Instruments Advanced Scientific Computer). However, METINP, because it is interactive, will not operate properly in a batch processing environment.

II. METSRT OPERATION

For reference during the following discussion, a complete listing of the METSRT source code is given in Appendix A.

METSRT may be logically divided into input, processing, and output sections. The input, and, to some extent, the processing sections of the program must be tailored to the characteristics of the data source. METSRT was originally written specifically for use with a Fluke Model 2240B data logger, having the analog data output format shown in Table 1. A digital data format (Table 1) is also available, but is not currently in use. Other formats would necessitate changes to the search and input routines and possibly to the error flagging subroutine. We are presently making alterations in order that the output of a newly constructed data acquisition system may be processed by METSRT.

Four I/O files (and four different logical devices) are involved:

1) FORØ4.DAT (device 4) contains parameters which control program execution.
2) FORØ5.DAT (device 5) is the input data file.
3) FORØ6.DAT (device 6) is a tabular output for printing.
4) FORØ7.DAT (device 7) contains all data and parameters and is intended to be read by subsequent programs.

Initially, METSRT reads the control file (FORØ4.DAT), which specifies the dates of interest, the specific types of data which are to be processed, the desired format for printed output, and the units for both input and output. Print switches may be set to select channels for which data is to be listed. In addition, other parameters, pertaining to METCLC and METPLT, may be present. These parameters, if present, are ignored by METSRT. An example of the control file is given in Table 2.

A subroutine (SEARCH) is then called which searches the data file (FORØ5.DAT) for the first data set and reads the time and date header. Any data set which is dated prior to the specified initial time is rejected and the search continues until one of the following conditions is met:

1) An End of File (EOF) is read.

Table 1
Data Logger Format

	Column Number																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Data Set Header	Y X	d f	d f	d f	: f	h f	h f	:	m	m	:	s	s				
Analog Data	A	n	n	n	l	s	a	a	a	a	a	a	u	u	e	e	
Digital Data	D	a	a	∅	d	d	d	d	d	d	d	d	d				

(Underlined characters are those which are always present)

Data Set Header: ddd = Day code (Julian date)
hh = hours
mm = minutes
ss = seconds
ffffff = fixed data (the last four digits are used to represent the year)

Analog Data: nnn = channel number
l = limit alarm;
May take the following values: "> " = upper limit exceeded
"< " = lower limit exceeded
" " = data within limits
s = sign of data
aaaaaa = analog data value, including a decimal point
uu = data units;
May take the following values: " C" = degrees centigrade
" F" = degrees Fahrenheit
" V" = volts
"MV" = millivolts
"**" = error (see error conditions below)
ee = error condition;
May take the following values: "OL" = overload
"BT" = broken thermocouple

Digital Data: aa = data address
∅ = space
ddddddddd = digital data

Table 2
Parameter file (FORØ4.DAT) required to process a portion of the data from the 1981 cruise of the USNS HAYES.

```
SORT PARAMETERS
HAYES 1981 CRUISE
 10
1981  29 IHU
1981  29 IHU
 40 OUT.TEMP   C  DEG C    2X,F6.1,2X,   1
 27 PWR.SUP.   V  VOLTS    2X,F7.3,1X,   0
 26 POS.PWR.   V  VOLTS    2X,F7.3,1X,   0
 22 OUT.TEMP   V  DEG C    2X,F6.1,2X,   1
 21 REL.HUM.   V PER CENT  2X,F6.0,2X,   0
 20 PRESSURE  MV  TORR     2X,F6.1,2X,   0
 15 SHP.HEAD  MV  DEGREES  2X,F6.0,2X,   1
 14 SHP.SPD.  MV  KNOTS    2X,F6.0,2X,   1
 13 WND.DIR.   V  DEGREES  2X,F6.0,2X,   1
 12 WND.SPD.   V  KNOTS    2X,F6.0,2X,   1

CALC PARAMETERS


WIND SUBROUTINE
 12 REL.SPD.   KNOTS    2X,F6.0,2X,   0
 13 REL.DIR.   DEGREES  2X,F6.0,2X,   0
 14 SHP.SPD.   KNOTS    2X,F6.0,2X,   0
 15 SHP.HEAD   DEGREES  2X,F6.0,2X,   0
100 ABS.SPD.   KNOTS    2X,F6.0,2X,   1
101 ABS.DIR.   DEGREES  2X,F6.0,2X,   1


MOIST SUBROUTINE
 40 OUT.TEMP   DEG C    2X,F6.1,2X,   0
 20 PRESSURE   TORR     2X,F6.1,2X,   0
 21 REL.HUM.  PER CENT  2X,F6.0,2X,   0
102 H2O VAP     PPMV    2X,F6.0,2X,   1


 22 OUT.TEMP   DEG C    2X,F6.1,2X,   0
 20 PRESSURE    TORR    2X,F6.1,2X,   1
 21 REL.HUM.  PER CENT  2X,F6.0,2X,   1
  0                                   1
```

2) A data set within the specified time window is found.
3) A data set having a date later than the last desired time is encountered.

Cases 1) and 3) cause appropriate error messages to be printed and execution terminates. It is assumed that time monotonically increases between data sets.

In case 2), a data input routine (DATAIN) is executed. Data is read from the current data set and compared with the list of desired inputs, as specified by the original parameter file. If a match is found, the data is stored in an array for further processing; otherwise it is ignored. In either case, data input continues until the required data values have been read. In the event that the start of a new data set is encountered before input of the current set is completed, an abnormal exit from the data input routine occurs and an error message is printed.

Regardless of the mode of exit, the search routine is invoked to locate the next data set. As before, the header is tested and any of the three conditions previously mentioned will halt the search. This time, however, case 2) causes a repeat of the data input routine and case 3) causes a data processing function (MANIP) to be performed. Case 1) still produces an error message, but continues to the data processing step rather than terminating the program.

The data input routine also tests for the following error conditions to the FLAG subroutine:

1) Broken sensor.
2) Overloaded sensor.
3) Value exceeds upper set point.
4) Value below lower set point.

All of these conditions are indicated by flags which are present in the original data from the data logger. These tests may be tailored to other data formats by alteration of the FLAG subroutine.

The data storage array is organized as a two dimensional matrix in which the columns contain data obtained from a particular channel and the rows correspond to different data sets (different times). MANIP accesses a cross reference matrix and determines, for each input channel, the column in which the corresponding data has been stored. This column is processed in accordance with the function specified for that input channel and the resulting value is replaced in the data array.

In general, the function will be different for each type of sensor and will have been chosen so as to convert the data logger output (typically a voltage) into a value of an appropriate physical quantity having the desired units. During this processing step the units of the input quantity (as read from the original data file) are compared with the expected input units (as given in the parameter file) and an error message is produced if a disagreement is found. This message identifies the date, time, and channel for which the error was detected and also shows what units were actually found.

After processing of the data is completed, the output subroutine (DATOUT) is called.

Table 3 shows a sample of the input data obtained from the data logger. The resulting error and warning messages appear in an output file, FORØ6.DAT. To this, DATOUT appends a tabular listing of data from the selected input channels, as shown in Table 4. The year, Julian date, and time are listed in the left columns. For each selected channel, a column of data will be produced having a heading which gives the channel number, sensor identification, and the units. A maximum of twelve channels of data may appear across a line printer page. If more channels were requested, additional pages will be produced, each having the date and time on the left and appropriate column headings across the top.

In addition to selection of the data to be listed, the user formats the output by providing FORTRAN-type format specifications, as desired. A total of ten characters (including spaces) should be specified for each channel which is to be printed.

An additional file, FORØ7.DAT, is also produced by METSRT. This file, which contains all of the processed data plus reference information needed by subsequent programs, was intended to be read only by computer. The format was chosen for compactness and few concessions to human readability have been made. A sample of this output is shown in Table 5.

III. METCLC OPERATION

METCLC, listed in Appendix B, reads both the control file and file FORØ7.DAT and calculates values for the following derived quantities:

1) Absolute wind velocity, expressed as a wind speed (knots) and bearing (degrees referenced to true north).
2) Atmospheric moisture loading, with water vapor concentration in ppm by volume.

For each calculation, the name and channel number for each input and output channel is printed in the summary listing, FORØ6.DAT (Table 6). Any problems encountered (missing channels, for example) are also listed at this point.

The calculated values are then stored in the data array and are available for output in both a tabular form and as an array intended to be read by subsequent programs. As before, any combination of these results may be selected for listing by setting the appropriate print switches. The print formats are specified by the user. An example of this listing is given in Table 7.

To provide versatility and make future alterations simple, calculations have been implemented in separate subroutines.

In general, there may be several alternate sources of data for these subroutines (multiple anemometers or hygrometers may have been used, for

Table 3

An excerpt from the USNS HAYES cruise data file (FORØ5.DAT).
The entire file is over 200 pages long when printed.

```
 10 + 1.730 V
 11 + 1.320 V
 12 +0.1114 V
 13 +0.4721 V
 14 + 11.040V
 15 +173.048V
 20 + 13.430V
 21 + 3.320 V
 22 +2.3804 V
 23 +0.0000 V
 24 -0.0004 V
 25 -0.0003 V
 26 +10.040 V
 27 -14.230 V
 28 + 0.178 V
 29 - 0.000 V
 40 +  17.0 C
 41 +  13.0 C

 10 + 1.730 V
 11 + 1.330 V
 12 +0.1470 V
 13 +0.1077 V
 14 + 11.040V
 15 +173.040V
 20 + 13.440V
 21 + 3.030 V
 22 +2.3038 V
 23 +0.0000 V
 24 -0.0004 V
 25 -0.0003 V
 26 +10.004 V
 27 -14.121 V
 28 + 0.178 V
 29 + 0.000 V
 40 +  17.7 C
 41 +  13.3 C

 10 + 1.713 V
 11 + 1.329 V
 12 +0.0991 V
 13 +0.1030 V
 14 + 11.040V
 15 +173.078V
 20 + 13.078V
 21 + 3.334 V
 22 +2.3330 V
 23 +0.0000 V
 24 -0.0004 V
 25 -0.0004 V
 26 +10.040 V
 27 -14.173 V
 28 + 0.178 V
 29 + 0.000 V
 40 +  17.4 C
 41 +  13.1 C
```

Table 4
The METSRT listing (FORØ6.DAT) generated using the parameters and data file shown in Tables 2 and 3. No warnings or error messages were produced by this data, so that portion of the output is not shown.

[illegible] 1981 CRUISE
DATA FOR THE PERIOD JAN 29, 1981 THROUGH FEB 29, [illegible]

	CHANNEL 40 OUT.TEMP DEG C	CHANNEL 22 CUT.TEMP DEG C	CHANNEL 15 SHP.HEAD DEGREES	CHANNEL 14 SHP.SPD. KNOTS	CHANNEL 13 WND.DIR DEGREES	CHANNEL 12 WND.SPD. KNOTS
1981						
29						
0	17.6	[illegible]	179.	1.	274.	10.
100	17.7	[illegible]	179.	1.	263.	13.
200	17.4	[illegible]	179.	1.	269.	9.
300	16.8	17.4	179.	1.	271.	8.
400	16.2	17.2	179.	1.	271.	8.
500	16.1	16.9	179.	1.	279.	6.
600	15.5	[illegible]	179.	1.	283.	[illegible]
700	14.9	15.9	179.	1.	285.	7.
800	14.5	15.5	179.	1.	283.	8.
900	14.8	15.7	179.	1.	273.	14.
1000	14.8	15.7	179.	1.	283.	9.
1100	17.5	[illegible]	179.	1.	287.	15.
1200	17.4	[illegible]	179.	0.	243.	3.
1300	16.7	17.7	179.	0.	244.	16.
1400	16.9	17.7	179.	0.	252.	14.
1500	16.8	17.9	179.	0.	252.	14.
1600	16.7	17.7	179.	0.	227.	12.
1700	16.7	17.4	179.	0.	235.	12.
1800	16.7	17.5	179.	0.	251.	15.
1900	16.6	17.6	179.	0.	223.	11.
2000	16.4	17.3	179.	0.	237.	10.
2100	16.4	16.3	179.	0.	139.	8.
2200	16.2	17.5	179.	0.	179.	8.
2300	16.3	16.5	179.	0.	152.	9.

Table 5

Part of the METSRT data dump (FORØ7.DAT) produced from the inputs shown above. This data serves as the input to METCLC.

```
 40  24
 1988  10 CHU
 1988  20 THU
  40 PLT.TEMP  DEG C    2X,12.4,2X,
  37 PRO.PRE.  MBARS    2X,F7.3,2X,
  26 DEW.TEMP  DEG C    2X,F7.2,2X,
  33 CON.TEMP  DEG C    2X,F7.2,2X,
  21 TEL.HUM.  PER CENT 2X,F6.0,2X,
  20 EEEEEEEE  FFFFF    2X,F6.1,2X,
  16 EEE.EEE.  EEEEEE   2X,F6.0,2X,
  14 PRE.PRE.  MMMMM    2X,F6.0,2X,
  13 ###.###.  EEEEEE   2X,F6.0,2X,
  12 ###.###.  MMMMM    2X,F6.0,2X,
 1988  10    0 1988  10  40 1988  10  200 1988  10  400 1988  10  600
 1988  10  800 1988  10 1000 1988  10 1200 1988  10 1400 1988  10 1600
 1988  10 1800 1988  10 2000 1988  10 2200 1988  10 2400 1988  11    0
CHANNEL NUMBER   40
 .1760000E+02  .1770000E+02  .1774000E+02  .1769000E+02  .1752000E+02  .1764000E+02
 .1755000E+02  .1740000E+02  .1758000E+02  .1748000E+02  .1745000E+02  .1758000E+02
 .1771000E+02  .1767000E+02  .1763000E+02  .1768000E+02  .1767000E+02  .1767000E+02
 .1767000E+02  .1771000E+02  .1764000E+02  .1764000E+02  .1752000E+02  .1760000E+02
CHANNEL NUMBER   37
-.1407800E+02 -.1411700E+02 -.1407300E+02 -.1412300E+02 -.1415700E+02 -.1407800E+02
-.1419400E+02 -.1405400E+02 -.1413300E+02 -.1406100E+02 -.1413300E+02 -.1407900E+02
-.1400300E+02 -.1397700E+02 -.1401400E+02 -.1413500E+02 -.1413300E+02 -.1400300E+02
-.1406700E+02 -.1407700E+02 -.1401600E+02 -.1415500E+02 -.1411000E+02 -.1400500E+02
CHANNEL NUMBER   26
 .1760180E+02  .1760020E+02  .1760200E+02  .1756670E+02  .1753890E+02  .1760760E+02
 .1760260E+02  .1760260E+02  .1759870E+02  .1760100E+02  .1753940E+02  .1760090E+02
 .1760710E+02  .1760120E+02  .1760050E+02  .1759670E+02  .1753490E+02  .1759900E+02
 .1760400E+02  .1760400E+02  .1760500E+02  .1760010E+02  .1753250E+02  .1760110E+02
CHANNEL NUMBER   33
 .1748374E+02  .1758384E+02  .1749344E+02  .1776445E+02  .1723562E+02  .1763489E+02
 .1764477E+02  .1758446E+02  .1754634E+02  .1756689E+02  .1757302E+02  .1760533E+02
 .1760844E+02  .1758832E+02  .1758667E+02  .1776444E+02  .1755573E+02  .1759467E+02
 .1758174E+02  .1758467E+02  .1758225E+02  .1776444E+02  .1752003E+02  .1765200E+02
CHANNEL NUMBER   21
 .7652000E+02  .7670000E+02  .7664000E+02  .7662000E+02  .7642000E+02  .7696000E+02
 .7697000E+02  .7694000E+02  .7690000E+02  .7694000E+02  .7653000E+02  .7702400E+02
 .7676000E+02  .7684000E+02  .7678000E+02  .7684000E+02  .7653000E+02  .7726000E+02
 .7668000E+02  .7682000E+02  .7687000E+02  .7686000E+02  .7651000E+02  .7700000E+02
CHANNEL NUMBER   20
 .7659774E+02  .7664764E+02  .7654922E+02  .7659752E+02  .7644447E+02  .7694450E+02
 .7690460E+02  .7684862E+02  .7684702E+02  .7659222E+02  .7644336E+02  .7704356E+02
 .7660448E+02  .7664837E+02  .7664777E+02  .7661780E+02  .7641451E+02  .7681796E+02
 .7667740E+02  .7667387E+02  .7667884E+02  .7667304E+02  .7641426E+02  .7674417E+02
CHANNEL NUMBER   16
 .1778400E+02  .1778200E+02  .1778570E+02  .1775800E+02  .1773802E+02  .1778580E+02
 .1778570E+02  .1778400E+02  .1778570E+02  .1773550E+02  .1773402E+02  .1778540E+02
 .1778534E+02  .1778530E+02  .1778550E+02  .1774560E+02  .1773600E+02  .1778590E+02
 .1778400E+02  .1778200E+02  .1778600E+02  .1776820E+02  .1773550E+02  .1778520E+02
CHANNEL NUMBER   14
 .1400400E+01  .1400400E+01  .1400400E+01  .1400300E+01  .1102400E+01  .1400300E+01
 .1400300E+01  .1400300E+01  .1400300E+01  .1400400E+01  .1102300E+01  .1400300E+01
 .6300000E-01  .6300000E-01  .6300000E-01  .6800000E-01  .3300000E-01  .6300000E-01
 .6300000E-01  .6300000E-01  .6300000E-01  .6400000E-01  .3400000E-01  .6300000E-01
```

Table 6
The METCLC summary (FORØ6.DAT) which results from the parameter file shown in Table 2 and the data set of Table 5.

WIND SUBROUTINE

ITERATION NUMBER	INPUT: REL WND SPD # NAME	REL WND DIR # NAME	ABS SHP SPD # NAME	ABS SHP DIR # NAME	OUTPUT: ABS WND SPD # NAME	ABS WND DIR # NAME
1	12 REL. SPD	13 REL.DIR.	14 SHP.SPD.	15 SHP.HEAD	100 ABS.SPD.	101 ABS.DIR.

WATER VAPOR SUBROUTINE

ITERATION NUMBER	INPUT: TEMPERATURE # NAME	AIR PRESSURE # NAME	ABS HUMIDITY # NAME	OUTPUT: WATER VAPOR # NAME
1	40 OUT.TEMP	20 PRESSURE	41 REL.HUM.	102 H2O VAP.
2	22 OUT.TEMP	20 PRESSURE	41 REL.HUM.	900 MOIS 2

Table 7

The METCLC data listing (FORØ6.DAT) from the same run which produced Table 6.

[illegible] 1981 CRUISE

DATA FOR THE PERIOD THU 29, 1981 THROUGH THU 29, 1981

	CHANNEL 21 REL.HUM. PER CENT	CHANNEL 20 PRESSURE TORR	CHANNEL 100 ABS.SPD. KNOTS	CHANNEL 101 ABS.DIR. DEGREES	CHANNEL 102 H2O VAP. PPMV	CHANNEL 900 HCIS 2 PPMV
1981						
29						
0	77.	758.8	10.	84.	13426.	.16E+05
100	77.	758.9	13.	81.	13532.	.16E+05
200	78.	759.0	9.	80.	13442.	.16E+05
300	77.	759.2	8.	82.	14701.	.16E+05
400	78.	759.4	8.	82.	14434.	.15E+05
500	78.	759.4	6.	88.	14269.	.15E+05
600	79.	759.5	7.	93.	13957.	.15E+05
700	80.	759.5	6.	94.	13540.	.14E+05
800	80.	759.5	8.	91.	13231.	.14E+05
900	80.	759.9	11.	89.	13332.	.14E+05
1000	81.	760.0	9.	95.	13540.	.14E+05
1100	74.	760.0	15.	81.	14632.	.15E+05
1200	74.	760.0	3.	60.	14354.	.15E+05
1300	73.	760.0	16.	62.	13939.	.15E+05
1400	73.	761.2	14.	70.	14020.	.15E+05
1500	73.	761.9	14.	51.	13940.	.15E+05
1600	73.	761.5	12.	46.	13932.	.15E+05
1700	73.	761.7	12.	54.	13704.	.14E+05
1800	72.	761.7	15.	49.	13527.	.14E+05
1900	72.	761.7	11.	41.	13586.	.15E+05
2000	69.	761.8	10.	26.	12782.	.14E+05
2100	69.	761.3	8.	7.	12767.	.14E+05
2200	68.	761.4	8.	349.	12516.	.14E+05
2300	71.	761.4	9.	340.	12378.	.13E+05

2 FUNCTION CALLS

example). Accordingly, there is provision for user specification of the inputs for each calculation. In fact, the same calculations can be repeated with different combinations of inputs and the results may be listed for comparison.

Absolute wind velocity is calculated by vector addition of the absolute velocity of the sensor platform (ship) and the relative wind velocity. Subroutine input and output vectors are in the form of a magnitude and a direction. It is assumed that the direction is in degrees from true north (for absolute bearings) or degrees clockwise from the platform velocity vector (for relative bearings) and that speeds are in knots.

Atmospheric moisture loading is calculated[1] as

$$[H_2O] = H \frac{P_s(T_a)}{P_a} 10^4$$

where $[H_2O]$ = water vapor concentration (ppmv); H = relative humidity (%); P_a = ambient pressure (mb); $P_s(T_a)$ = saturation vapor pressure (mb) at ambient temperature T_a (°K).

The saturation vapor pressure may be obtained from

$$P_s(T_a) = P_0 \exp\left[\sum_{n=1}^{4} C_n t^n\right]$$

where P_0 = 1013.25 mb; C_1 = 13.3185; C_2 = -1.9760; C_3 = -0.6445; C_4 = -0.1299 and t is given by

$$t = 1 - \frac{373.15}{T_a}$$

with T_a = ambient temperature (°K).

Inputs to this subroutine are assumed to be in units of torr, per cent, and degrees centigrade for pressure, relative humidity, and temperature, respectively. The output is the water vapor concentration in parts per million by volume.

IV. METPLT OPERATION

METPLT is designed to plot selected subsets of the data contained in FORØ7.DAT according to the specifications given in file FORØ4.DAT. The actual details of generating a plot file are handled by DISSPLA, a package of FORTRAN-callable subroutines provided by Integrated Software Systems

1. G.J. McRae, APCA Journal, 30(4), 394 (1980).

Corporation. Basically, METPLT provides the data needed by the DISSPLA routines. Appendix C contains the source code for METPLT, but not for any of the DISSPLA software.

The parameter file (Table 2) is searched until the "PLOT PARAMETERS" section is found and the title to be used on the output is read. Next, the data file is read and stored in memory. In the event that either of these files is missing, or if the "PLOT" section is not found, an error message will be written into FORØ6.DAT.

The next parameters to be read specify the number of days for which the data is to be plotted on a single page (NDAYS) and the dates of the first and last data which is to be plotted. The set of all plotted data for an NDAYS-long period is referred to as a plot set. Typically, the length of a plot set is seven days so that the plotted output will have one week of data per page. There may be more than NDAYS between the initial and final dates specified, in which case multiple plot sets will be produced. Each plot set may itself involve several pages of output since there are normally only three graphs per page.

We must still specify which data is to be plotted and how it is to be plotted. This is done by providing sets of parameters which define each axis for each plot. Subroutines XAXIN and YAXIN are responsible for reading and storing these parameters.

METPLT first searches for the set of parameters which describes the desired X-axis, then it looks for corresponding Y-axis parameters. For each X-axis, there may be multiple Y-axis specifications so that several different graphs may easily be generated using the same independent variable.

Each axis specification consists of the following nine parameters:

1) Channel number.
2) Channel name.
3) Channel units.
4) Minimum value.
5) Incremental value.
6) Maximum value.
7) Threshold value.
8) Hysteresis parameter.
9) Axis type.

The channel number tells the program which data is to be plotted on the specified axis. In the event that a channel number of zero is given, METPLT will use time, rather than data values, for that axis. In this case, the axis will be labeled with the Julian dates and each day will be labeled at 1200 and 2400 hours. For purposes of axis specification, however, times must be given in minutes.

The channel name and units are used to produce a label for the axis. Minimum, maximum, and incremental values are needed in order to calculate the scale.

Threshold and hysteresis parameters provide increased control over the plotted output. If the value of any coordinate is below the corresponding threshold, plotting of the point will be suppressed. The hysteresis parameters allow points to be suppressed if they lie within a specified "dead band" surrounding the most recently plotted point. Note that, when hysteresis is set to zero on any axis, the dead band area will also be zero and all points will be plotted. In order to prevent this, any of the hysteresis parameters may be set to a negative value and will then be ignored.

The axis type parameter allows the user to select either a linear Y-axis (type = LIN) or a "vector" Y-axis. Only linear X-axes are permitted in the current version of METPLT.

In the vector mode, the two channels representing R- and Θ-components are designated by type = RVEC and type = AVEC, respectively. They are used to generate a vector quantity which is then displayed as an arrow of the appropriate length and direction. The tip of the arrowhead is located at the corresponding X-coordinate for the quantity. For reference, a short arrow (not of unit length) is drawn in the zero degree direction and a scale is provided on the Y-axis. This scale, the axis name, and the axis units are those given for the vector magnitude channel.

The vector plotting mode is useful for representing quantities such as wind direction.

Provisions have been incorporated into METPLT allowing easy program enhancement to include other types of axes, such as logarithmic scales or vectors expressed as X- and Y-components.

Error messages are written into FORØ6.DAT by XAXIN or YAXIN if the axis type is not defined or (in the case of vector axes) if the two types are not self consistent.

At this point, subroutine SETUP determines the first and last dates for the next plot set and searches the ITIME array to locate the corresponding rows in the data matrix. If they cannot be found, an error message is produced and the plot set is skipped. Assuming that the rows have been located, LOAD copies the appropriate data into the X- and Y-arrays needed by the actual curve drawing routines. During the loading process, each datum is checked to see if it is invalid (the character string "----"), if it is below the threshold, or if the hysteresis criterion is not met. In the first two cases, the point will not be plotted and a message to this effect will appear in the plot summary listing (FORØ6.DAT). In the third case, a message will also be printed but the point will not be suppressed unless the hysteresis test fails for all axes.

After all of the graphs on a page have been completed, a page caption is added. The title (specified in the parameter file) is written across the top and a subtitle, giving the initial and final dates of the plot set, appears below the title.

Examples of the printed and plotted output from METPLT are shown in Tables 8 and 9.

Additional pages are produced as necessary in order to graph all of the data in the first plot set. Further plot sets will then be created, each one starting on the Julian date following the end of the previous set.

V. METINP OPERATION

METINP is the only interactive program in the METEOR package. A listing of the FORTRAN program is given in Appendix D and an example of a terminal session is shown in Appendix E. METINP uses this dialog to construct the parameter file, FORØ4.DAT.

Initially, the user is asked to identify the program for which a parameter file is desired. The answer to this question is used to select one of three major subroutines: SORTIN, CALCIN, or PLOTIN. These produce control files for METSRT, METCLC, and METPLT, respectively.

In the case of CALCIN, additional information is requested regarding the specific type of calculation required. Depending on the response, either WNDIN (for wind velocity calculations) or MSTIN (for moisture loading) will be called.

In order to minimize the size of the program, each parameter input routine utilizes the same set of I/O subroutines. Subroutine TTYIN writes a prompter message, reads the user's response, and stores the answer. Subroutine FILE then formats the answer as required for that specific line of the parameter file.

To make these two subroutines more generally useful, they operate on data arrays. For example, the prompter character string and the corresponding input format are contained in arrays PROMPT and FORMIN, respectively, while the user response is stored in INARAY. Each call to TTYIN or to FILE may therefore be tailored to specific requirements by passing the appropriate arrays as arguments in the subroutine call.

After all required information has been obtained, subroutine CHECK writes the parameters to the TTY in exactly the form in which they will appear in the final parameter file. If any changes are required, subroutine EDIT allows the old line to be overwritten by a new one, which is then displayed. When all lines have been verified, the complete set is written onto the disk. If further input is desired, the entire process may be repeated, either for another function for the same program or for a different program.

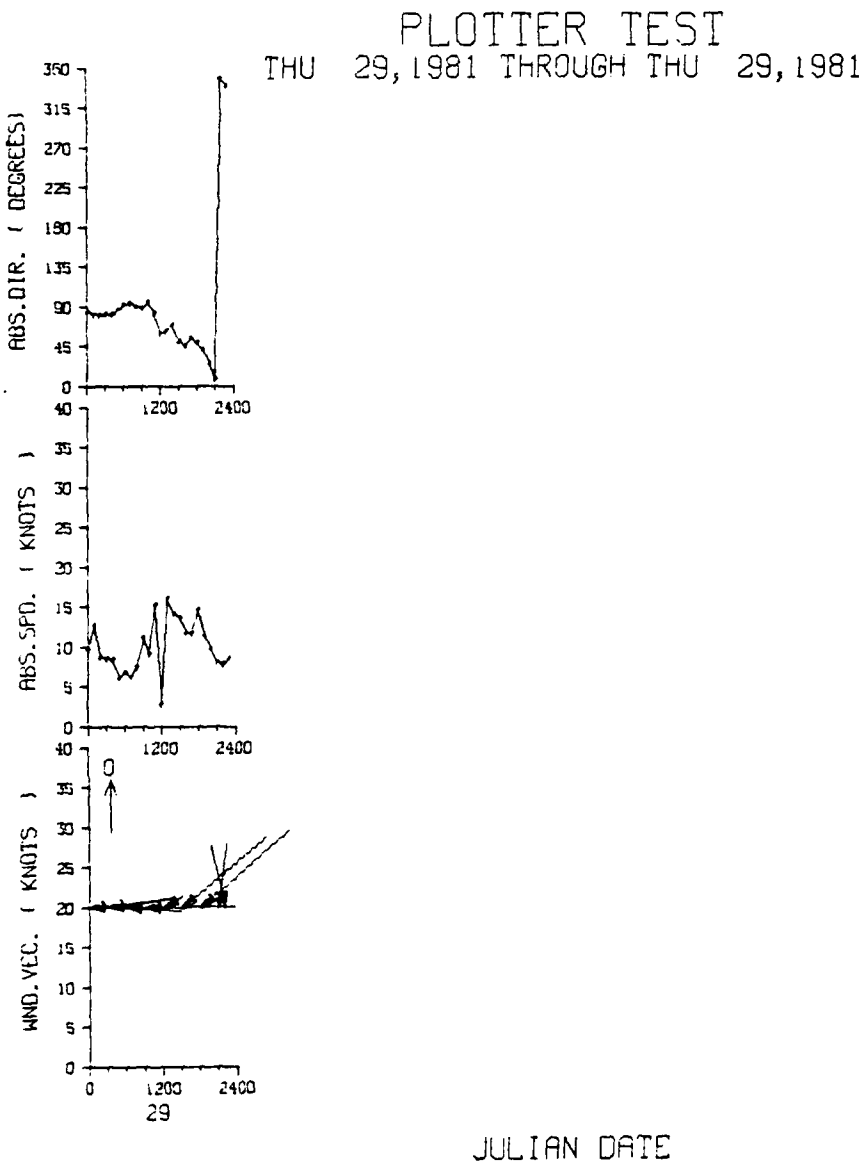
Since METSRT is the first program to be used in data analysis, it is assumed that a new file FORØ4.DAT will be required whenever a METSRT control file is to be created. For this reason, SORTIN causes a new disk file to be opened and any existing file with the name FORØ4.DAT will be destroyed. Some care must therefore be exercised to ensure that the current file (if one exists) has been saved under another name before a new file is opened.

Table 8

The METPLT summary (FORØ6.DAT) resulting from the parameters and data of Tables 2 and 5, respectively.

```
                                                                  SAVES 1991 values.
                                                  Uses the DEFAULT for  x,1991 labels and  y,1991

SUMMARY OF PLOT SET NUMBER  1


PLOT NUMBER  1: WIND VEC.  VS TIME

                  X  CHANNEL =   0; INTERCEPT =     0.; MYSTEP      =  100.
                  Y  CHANNEL =  100; INTERCEPT =     3.; MYSTEP      =  0.0000
                  Z2 CHANNEL =  101; INTERCEPT =     0.; MYSTEP      =  0.0
                  TYPE = VECTOR


1981  2y  100: POINT SUPPRESSED --
X  =  .000000E+02 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .013975E+02 IS WITHIN   30. OF PREVIOUS VALUE


1981  2y  200: POINT SUPPRESSED --
X  =  .100000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .790029E+02 IS WITHIN   30. OF PREVIOUS VALUE


1981  2y  300: POINT SUPPRESSED --
X  =  .200000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .013274E+02 IS WITHIN   30. OF PREVIOUS VALUE


1981  2y  500: POINT SUPPRESSED --
X  =  .300000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .078E6.1E+02 IS WITHIN   30. OF PREVIOUS VALUE


1981  2y  700: POINT SUPPRESSED --
X  =  .400000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .993E9+6E+02 IS WITHIN   30. OF PREVIOUS VALUE


1981  2y  800: POINT SUPPRESSED --
X  =  .400000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .993E95E+02 IS WITHIN   30. OF PREVIOUS VALUE


1981  2y 1000: POINT SUPPRESSED --
X  =  .500000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .993919E+02 IS WITHIN   30. OF PREVIOUS VALUE


1981  2y 1100: POINT SUPPRESSED --
X  =  .600000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .981+0E+02 IS WITHIN   30. OF PREVIOUS VALUE


1981  2y 1300: POINT SUPPRESSED --
X  =  .700000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .000115E+02 IS WITHIN   30. OF PREVIOUS VALUE


1981  2y 1400: POINT SUPPRESSED --
X  =  .800000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .701+0E2E+02 IS WITHIN   30. OF PREVIOUS VALUE


1981  2y 1600: POINT SUPPRESSED --
X  =  .900000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .000134E+02 IS WITHIN   30. OF PREVIOUS VALUE


1981  2y 1700: POINT SUPPRESSED --
X  =  .100000E+04 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .000+04E+02 IS WITHIN   30. OF PREVIOUS VALUE


1981  2y 1900: POINT SUPPRESSED --
X  =  .110000E+04 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .981+03E+02 IS WITHIN   30. OF PREVIOUS VALUE


1981  2y 2000: POINT SUPPRESSED --
X  =  .120000E+04 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .000711+8E+02 IS WITHIN   30. OF PREVIOUS VALUE


1981  2y 2300: POINT SUPPRESSED --
X  =  .130000E+04 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .000+95E+03 IS WITHIN   30. OF PREVIOUS VALUE


PLOT NUMBER  2: ANG. SEC.  VS TIME

                  X  CHANNEL =   0; INTERCEPT =     0.; MYSTEP      =  0.
                  Y  CHANNEL =  100; INTERCEPT =     0.; MYSTEP      =  0.0000
                  TYPE = LINEAR


PLOT NUMBER  3: AUS. DIR.  VS TIME

                  X  CHANNEL =   0; INTERCEPT =     0.; MYSTEP      =  0.
                  Y  CHANNEL =  101; INTERCEPT =     0.; MYSTEP      =  0.0000
                  TYPE = LINEAR
```

Table 9
METCLC plotted output corresponding to the above summary.



METCLC and METPLT both require the output of METSRT, so it is assumed, whenever control files for these programs are requested, that the METSRT parameter file already exists. Accordingly, the new parameters are appended to the existing file, which is not lost.

VI. SUMMARY

METEOR provides a coordinated set of programs which can read data tapes, locate specified types of data, test for a wide range of error conditions, calculate values of several derived quantities, and produce both printed and plotted output, all under control of a user-created command file. Existing functions may be selected as required and new functions may be added with relative ease.

Although originally intended to process meteorological data, this software package should be equally applicable to any situation in which large quantities of diverse data are acquired over long time periods.

In many cases,data may be collected on several different data logger systems simultaneously. For these situations, it would be advantageous to be able to merge the resulting data files. Other possible improvements include addition of statistical, curve smoothing, and cross-correlation capabilities in METCLC and provision for better control over plot size and shape in METPLT.

We expect that other users may find it necesssary to revise the METEOR programs to meet their special requirements. It is hoped that the documentation provided will prove to be sufficient for this purpose. Any comments or suggestions regarding alterations to or extension of these programs will be welcomed.

Appendix A
Listing of Program METSRT
(Version 2.0)

```
C     PROGRAM RECOUT
C
C     RECSET VERSION 3.0   DATED 1981
C
C     JOHN HOOVER
C     CODE 4330
C     U.S. NAVAL RESEARCH LABORATORY
C     WASHINGTON, D.C.  20375
C
C     PROGRAM INPUT DATA FILE(S) FROM DIGITAL RECORDING OSCILLOSCOPES, EDITS THE DATA
C     ACCORDING TO THE DATA SPECIFICATIONS SUPPLIED BY THE USER, AND
C     OUTPUTS THE RESULTING INFORMATION IN A DATA FILE.  ANY OR ALL
C     OF THE DATA MAY ALSO BE PRINTED.
C
      PROGRAM PARAMETER S;
C
C        MCHAN=MAXIMUM POSSIBLE NUMBER OF DATA ITEMS (ie CHANNELS, NDATA=500)
C        NCHAN=MAXIMUM POSSIBLE NUMBER OF DATA CHANNELS PER SET (CHECKED BY
C              RCHAN=25
C
C     TEST=NAME OF THE PARAMETER FILE DESIRED ASSIGNED (TITLE=MYDATA
C          FOR THIS RECORD)
C
C     INPUT VARIABLES FOR PARAMETER FILE:
C
C        TITLE=DESIRED TITLE OF OUTPUT DATA
C        NCHAN=NUMBER OF DATA CHANNELS PER SET FOR ALL CHANNELS
C              (FORMAT=1X,I2)
C        IYEAR=INITIAL YEAR
C        IDATE=INITIAL JULIAN DATE
C        ILAST=INITIAL LAST DAY OF WEEK
C        LEAPF,LDAYS,LDAY =FLAG TO CHECK FOR LEAP YEARS,
C              (FORMAT=1A,2X,I3,2(/1A,I6,1A,I3,1A,I3))
C
C        IPRF  =PARAM WHICH CHOOSE PRINTOUT OPTIONS (FORMAT=1A,DATA ITEM,1X,
C              'TOPRAM LOCATION METHOD FILE DATA SERIAL (CHECKS NUMBER').
C        NNAME =PARAM OF NAMES OF OUTPUT VAR SHOULD BE=1
C        NUM   =PARAM OF CONTINUING THE PICKING FOR SUCCESSIVE EE DATA
C        IPRINT=PARAM CHOOSING VARIOUS PRINTOUTS FOR SUCCESSIVE DATA SETS
C              (FORMAT=1A,I2,1A,I###,1A,A2,1A,I###,1A,I###,1A,I2)
C
C
C     DATA INPUT :
C
C        DATA  =ARRAY INTO WHICH ALL DATA ARE TO BE READ.  DATA IS ARRANGED
C               IN TIME SEQUENCE FOR EACH CHANNEL, WITH THE END OF CHANNEL
C               (1,K)=FIRST INPUT CHANNEL
C        TITLE =ARRAY HOLDING THE TITLE, WHICH IS READ FIRST FROM DATA SET.
C               THE TITLE CORresponds TO THE USER OF THE DATA MODIFIER.
C
C
C*****************************************************************************
C                              FILE STRUCTURE
C                              ---- ---------
C
C
```

```
RECSET  RECSET.FOR     FORTRAN V.04(021) VAX   26-JUN-81       13:17   PAGE  2
C
C     FILE NAME          TYPE              USED              WRITE
C     ---------          ----              ----              -----
C     FOR04.DAT     FA:ANALYZE ABPOS       Status            *****
C     FOR05.DAT       DATA INPUT       STATUS,STATUS          *****
C     FOR06.DAT     FA:FORMATTED LIST        ****          DATAIN,DATOUT,
C                                                          SEARCH,RPMTR
C     FOR07.DAT       DATA OUTPUT            ****          DATOUT
C
C
C*****************************************************************************
C                              ERROR MESSAGES
C                              ----- --------
C
C        <NOTE: Y=YEAR,J=DATE,H=HOUR,MIN=MINUTE,SEC=SECOND,VALUE,I=INTEGER>
C        <      MECHANICAL BUGGER, MAY CONTINUE WITH ERROR          >
C
C              MESSAGE                  ACTION                  MEANING
C              -------                  ------                  -------
C     NO PARAMETER FILE FOUND           Status          FILE FOR04.DAT WAS MISSING;
C                                                       IN TO EXECUTION ENABLED MESSAGE
C                                                       PROGRAM COULD BE FOUND
C
C     NO DATA FILE FOUND                Status          FILE FOR05.DAT WAS NOT FOUND
C
C     NO DATA SET FOUND                 Status          NO DATA FOR THE REQUESTED
C                                                       YEAR AND DATES COULD BE FOUND.
C
C     EOF DURING SEARCH: Y,D,H          Status          EOF WAS ENCOUNTERED WHILE
C                                                       SEARCHING FOR NEXT DATA SET.
C                                                       LAST READ Y,D, AND H ARE GIVEN
C
C     EOF DURING DATA READ: Y,D,H       Status          EOF WAS ENCOUNTERED WHILE AT-
C     -------------------------                         TEMPTING TO READ DATA Y,D, OR
C                                                       H.  WHEN TRIED Y,D, AND H ARE
C                                                       GIVEN AND THE LAST DATA SET
C                                                       RECORDED (FROM)
C
C     SYSTEM RESET AFTER H HOURS OF     No Stop       A RESET OF SYSTEM WAS ENCOUNTERED
C     DAY J,Y. DATA RECORDED UNTIL H.                 WHICH CAUSED END OF DATA SET.
C                                                     ANOTHER DATA SET IS EXPECTED
C                                                     BUT NO MORE DATA WERE RECORDED
C                                                     FOR THIS DAY.  THE LAST VALID
C                                                     DATA WERE J,H, AND Y ARE
C                                                     GIVEN
C
C     ACQUISITION RESUMED AT H HOURS    No Stop       A DATA SET WAS NOT
C     ON DAY J,Y.                                     FOUND BECAUSE SYSTEM CLOCK
C                                                     RESET OCCURRED BEFORE
C                                                     ACQUISITION OF DATA.  NO MORE
C                                                     DATA WERE RECORDED AFTER
C                                                     THIS TIME.
C
C     MORE THAN 1 DATA FILE FOUND       Status          NUMBER OF DATA SETS AVAILABLE
C     DATA BEYOND H HOURS ON DAY        Stop            EXCEEDED MAXIMUM
C     J,Y will be lost.
C     LAST DATA RECORDED AT H
C     HOURS ON J,Y
C
C     DUPLICATE DATA FOUND FOR          Status          TWO DATA SETS WERE FOUND
C     CHANNEL N AT H HOURS J ON         Stop            FOR SAME CHANNEL NUMBER AND
```

```
00113  C     LAY D,E:
00114  C     ------------------------------
00115  C
00116  C     ILLEGAL ZERO=D AT H HOURS ON      WARNING
00117  C     DAY D,Y:
00118  C     ------------------------------
00119  C
00120  C
00121  C     END OF DATA SET FOR H HOU         WARNING
00122  C     ON DAY D,Y: H CHANGED TO=0
00123  C
00124  C
00125  C     UNCORRECTABLE DATA R.             ERROR
00126  C     H HOURS ON DAY D,Y:
00127  C     ------------------------------
00128  C
00129  C     Y,D,H CHANNEL 4: WIND SPEED       FATAL
00130  C     EXCEEDED
00131  C
00132  C     Y,D,H CHANNEL 4: D MIN SPEED      FATAL
00133  C     EXCEEDED
00134  C
00135  C     Y,D,H CHANNEL 4: UNRECOGNIZED     FATAL
00136  C     CHARACTERS: FOUND=__
00137  C
00138  C
00139  C
00140  C     Y,D,H CHANNEL 4: SPEED            *FATAL
00141  C     ERROR
00142  C
00143  C     Y,D,H CHANNEL 4: WIND DI          *FATAL
00144  C     IS OK
00145  C
00146  C     Y,D,H CHANNEL 4: SPEED            *FATAL
00147  C     OVERLOAD
00148  C
00149  C     Y,D,H CHANNEL 4: SPEED            *FATAL
00150  C     IN RANGE
00151  C
00152  C     Y,D,H CHANNEL 4: UNRECOGNIZED     FATAL
00153  C     CHARACTERS: FOUND=__
00154  C
00155  C
00156  C     ILLEGAL CHANNEL NUMBER 4          WARNING
00157  C     (ISTATION=I)
00158  C
00159  C
00160  C
00161  C     UNSPECIFIED ERROR=NO F=           WARNING
00162  C     CHANNEL 4 (ISTATION=I)
00163  C
00164  C
00165  C
00166  C    *Y,D,H CHANNEL 4: UNEXPLICTED      *FATAL
00167  C     UNITS (__) FOUND
00168  C
```



```
00169  C     Y,D,H CHANNEL 4: CORRECT          *WARNING  CORRECT UNITS WERE FOUND FOR A
00170  C     UNITS (__) FOUND                            CHANNEL PREVIOUSLY FLAGGED AS
00171  C                                                 INCORRECT
00172  C
00173  C     Y,D,H CHANNEL 4: HUMIDITY         HUMIDA    HUMIDITY WAS GREATER THAN 100.0
00174  C     OUT OF RANGE (4);                           PERCENT OR LESS THAN 0.0
00175  C     CORRECTED TO X                              PERCENT
00176  C
00177  C     Y,D,H CHANNEL 25: W.D.            WARN      VOLTAGE FOR THE WIND DIRECTION
00178  C     VOLTAGE OUT OF RANGE (X V)                  SENSOR WAS NOT IN THE RANGE
00179  C                                                 5.00 +- 0.03 VOLTS
00180  C
00181  C     Y,D,H CHANNEL 26: ECU.            ERROR     RECEIVED POWER SUPPLY VOLTAGE
00182  C     REC. SUPPLY OUT OF RANGE                    WAS NOT IN THE RANGE 15.0
00183  C     (X V)                                       +-0.5 VOLTS
00184  C
00185  C     Y,D,H CHANNEL 27: STA.            ERROR     RECEIVED POWER SUPPLY VOLTAGE
00186  C     REC. SUPPLY OUT OF RANGE                    WAS NOT IN THE RANGE 15.0
00187  C     (X V)                                       +-0.5 VOLTS
00188  C
00189  C    *THESE CONDITIONS EXIST FOR ALL SUBSEQUENT DATA READ FOR THE AFFECTED CHANNEL
00190  C     UNTIL OTHERWISE INDICATED
00191  C
00192  C     ********************************************************************
00193  C
00194  C           !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
00195  C           !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
00196  C           !!                                                        !!
00197  C           !!  N     Nx    0000    xIIxxxx   xxxxIIx1   !!
00198  C           !!  Nn    nN   000000   IIxIxxxx   xxxxIIIE   !!
00199  C           !!  NNN   Nn   00   00     xI       xx       !!
00200  C           !!  NNNI  Nn   0I   0x     Ix       xx       !!
00201  C           !!  Nn N  Nn   0x   xx     Ix       xxxxIx   !!
00202  C           !!  Nn  x Nn   0x   0x     II       xxxxIx   !!
00203  C           !!  NN  NNNN   0x   0I     Ix       xx       !!
00204  C           !!  Nn   NnN   0x   xI     xI       xx       !!
00205  C           !!  Nn    Nx    000000     Ix       xxxxxIIx  !!
00206  C           !!  NN     N     0000      xI       xxxxIIIx  !!
00207  C           !!                                                        !!
00208  C           !!  Nxx-FORTRAN-xxx-Nxxxxxxxx-xx-xxxx-xxxxxx  !!
00209  C           !!  xxx IN ALL xxxxxxxxxx-xxxx-xx-xxxxxxx-xx  !!
00210  C           !!  xx xxxx xxx xxxx xxxx-xx-xxxxxxx-xx  !!
00211  C           !!  xxx xxxx xxxx.                                        !!
00212  C           !!                                                        !!
00213  C           !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
00214  C           !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
00215  C
00216  C    ERRORS WILL CAUSE THE FOLLOWING ACTIONS:
00217  C    MESSG, IMAIN, CHAIN, WARN, WARN.000, ERROR, SEVERE, FATAL *, ***,
00218  C    ***W, ENDW, ***E
00219  C
00220  C    WHEN NEW ERROR CONDITIONS ENCOUNTERED ARE ADDED AS "ERRORS", THEY MUST ALSO
00221  C    BE INCLUDED IN THE "SYSTEM.MAP" DIRECTIVE.
00222  C
00223  C     ********************************************************************
00224  C
```

```
MAIN.  MEINIT.FOR     FORTRAN V.5A(643) /AI  dd-mmm-yy       15:17  PAGE 1


00001          SUBROUTINE INIT
00002      C
00003      C     SUBROUTINE INITIALIZES IDENT AND DATA ARRAYS,
00004      C     DATA TO ----, UNIT ARRAY, ABPAT TO ZEROES, AND ASSIGNS DATA
00005      C     ABPAT COLUMN NUMBERS FOR EACH INPUT DATASET
00006      C
00007          PARAMETER NDATA = 300,NCHAN = 20
00008          COMMON IDENT(NCHAN),IDENT2(NCHAN),
00009          COMMON /ABPAT/ DATA(NDATA,NCHAN),
00010          COMMON /U/ UNIT(NDATA,NCHAN,2),
00011          DATA DASH,BLANK/'----','    '/
00012          DO 10 J=1,NCHAN
00013      C
00014      C     INITIALIZE ENTRY READ ARRAYS
00015      C
00016          IEPERT(J) = 0
00017          IEPSUC(J) = 0
00018      C
00019      C     INITIALIZE COLUMN NUMBERS IN PERIPHERAL ARRAY
00020      C
00021          ISEP(J,4) = J
00022      C
00023      C     INITIALIZE DATA ARRAY
00024      C
00025          DO 10 I=1,NDATA
00026          DATA(I,J) = DASH
00027      C
00028      C     INITIALIZE UNIT TO BLANKS
00029      C
00030          DO 10 K=1,2
00031       10 UNIT(I,J,K) = BLANK
00032          RETURN
00033          END
```

```
SEARCH  MAINSI.FOR     FORTRAN V.5A(643) /AI  dd-mmm-yy       15:17  PAGE 1


00001          SUBROUTINE SEARCH(NTIME,*)
00002      C
00003      C     SUBROUTINE SEARCHS FOR NEW READS THE TIME AND DATE HEADER ON
00004      C     A DATA SET
00005      C
00006          DIMENSION DADLIN (20)
00007          COMMON /M/ MYEAR,MDATE,MHOUR,MIN,MSEC,DUMMY
00008          DATA Y/'Y   '/
00009      C
00010      C     LOOK FOR DATE AND TIME (START OF DATA SET)
00011      C
00012       10 READ (5,20,END=50) TEST
00013       20 FORMAT (A1)
00014          IF (TEST .NE. Y) GO TO 10
00015      C
00016      C     SAVE PREVIOUS YEAR, DATE, AND TIME
00017      C
00018          MYEAR = MYEAR
00019          MDATE = MDATE
00020          NTIME = NTIME
00021      C
00022      C     READ NEW DATE, TIME, AND YEAR
00023      C
00024          BACKSPACE 5
00025          READ (5,30,END=50,ERR=70) MDATE,MHOUR,MIN,MSEC,MYEAR
00026       30 FORMAT (8X,I3,4X,I2,1X,I2/8X,I6)
00027          NTIME = 100 * MHOUR + MIN
00028      C
00029      C     CHECK FOR SYSTEM RESET
00030      C
00031          IF (MDATE .EQ. 0) GO TO 100
00032          IF (IFLAG .EQ. 0) RETURN
00033          IFLAG = 0
00034          WRITE (6,40) NTIME,MDATE,MYEAR
00035       40 FORMAT (1X'ACQUISITION RESUMED AT ',I4,' HOURS ON DAY ',I3,',',I4)
00036          RETURN
00037      C
00038      C     IF END OF FILE WAS ENCOUNTERED
00039      C
00040       50 WRITE (6,60) MYEAR,MDATE,NTIME
00041       60 FORMAT (1X,'EOF DURING SEARCH: ',I4,1X,I3,1X,I4)
00042          RETURN 1
00043      C
00044      C     IF READ ERROR OCCURRED
00045      C
00046       70 READ (5,80) DADLIN
00047       80 FORMAT (20A4)
00048          WRITE (6,90) MYEAR,MDATE,NTIME,DADLIN
00049       90 FORMAT (1X'ERROR DURING SEARCH: ',I4,1X,I3,1X,I4/1X,20A4)
00050          GO TO 10
00051      100 IF (IFLAG .EQ. 1) GO TO 10
00052          WRITE (6,110) NTIME,MDATE,MYEAR
00053      110 FORMAT (1X'SYSTEM RESET AFTER ',I4,' HOURS ON DAY ',I3,
00054         1 ',',I4,'. DATA SET(S) SKIPPED.')
00055          IFLAG = 1
00056          GO TO 10
00057          END
```

```
DATAIN  [illegible]      FORTRAN V1.0(021) /NI  [illegible]      15:17  [illegible]

      SUBROUTINE DATCHK(N,*N1,*N2,*)
C
C     SUBROUTINE READS, CHECKS, AND STORES ONE DATA
C
      PARAMETER MDATA=300,MCHAN=20
      DIMENSION DAQLIN(20),DATDAT(7),IDAT(MDATA)
      COMMON IPRIN(5*MCHAN),ICHAN(MCHAN),[illegible]
      COMMON /ABFAB/ DATA(MDATA,MCHAN),[illegible]
      COMMON /II/ ILATE,ITIME,IDAY,ILAT,[illegible]
      COMMON /N/ NYEAR,NDATE,NHOUR,NMIN,[illegible]
      COMMON /D/ DAQ(IDATA,MCHAN,2),[illegible]
      DATA A,I,DAYNM/'A   ','I   ','SUN ','MON ','TUE ','WED ','THU ',
     1 'FRI ','SAT '/
C
C     CLEAR CHANNEL COUNT, INCREMENT ROW NUMBER, AND TEST FOR ROW
C     LIMIT EXCEEDED
C
      NCOUNT = 0
      NRO = NRO + 1
      IF (NROW .LE. MDATA) GO TO 40
C
C     IF LIMIT EXCEEDED, DECREMENT ROW NUMBER AND RETURN TO CALLING PROGRAM
C
      NRO = NRO - 1
      N = NDATA
      WRITE (6,30) N,N,NDATE,NHOUR,(DAQ(NRO,J),
     1 IIME(NROW,2),ITIME(NROW,3)
   30 FORMAT (1X,'MORE THAN ',I3,' DATA SETS FOUND'/1X,'DATA AT ',I4,
     1 ' HOURS ON DAY ',I3,',',I2,',',I2,' AND ',I2,' HOURS WILL BE IGNORED'/
     2 1X,'LAST DATA ACCEPTED AT ',I2,' HOURS ON DAY ',I3,',',I2)
      N = 1
      RETURN
C
C     SET LDATE AND LYEAR TO LAST VALUES READ
C
      LDATE = ITIME(NROW,2)
      LYEAR = ITIME(NROW,1)
      N = 1
C
C     FIND INITIAL DAY OF MEAN IN THE ARRAY OF DAY NAMES
C
   40 IF (DAYNAM(N) .EQ. IDAY) GO TO 50
      N = N + 1
      GO TO 40
C
C     FIND NAME OF LAST DAY
C
   50 I = LDATE - IDATE
      N = (I - (I/7)*7) + N
      IF (N .GT. 7) N = N - 7
      IF (N .LT. 1) N = N + 7
      IDAY = DAYNAM(N)
      PRINT 1
C
C     CLEAR MCHANNEL FOUND FLAG
C
      DO 60 I=1,MCHAN
   60 IREAD(I) = 0
```

```
DATAIN  MICHIT.ROB      FORTRAN V1.0(021) /NI  [illegible]      15:17  [illegible]

C
C     STORE YEAR, DATE, AND TIME
C
      ITIME(NROW,1) = NYEAR
      ITIME(NROW,2) = NDATE
      ITIME(NROW,3) = 100 * NHOUR + MIN
C
C     TEST FOR END OF DATA SET
C
   70 READ (5,70,END=80) TEST
   70 FORMAT (A1)
C
C     TEST FOR ABNORMAL DATA
C
      IF (ITEST .EQ. A) GO TO 90
C
C     TEST FOR START OF NEXT DATA BLOCK
C
      IF (ITEST .EQ. I) GO TO 150
C
C     IF NOT VALID DATA, WRITE ERROR MESSAGE
C
      BACKSPACE 5
      READ (5,75) NADDIN
   75 FORMAT (20A4)
      WRITE (6,80) ITIME(NROW,3),NDATE,NYEAR,NADDIN
   80 FORMAT (A,'ILLEGAL RECORD AT ',I4,' HOURS ON DAY ',I3,',',
     1 I4,'I'/1X,20A4)
      GO TO 60
C
C     IF VALID DATA, NORMAL RECORD
C
   90 BACKSPACE 5
      READ (5,100,ERR=190) ICHAN,IDAY,NDATE,NYEAR,A
  100 FORMAT (1X,I3,A1,I7.6,1X,A2)
C
C     COMPARE WITH ACTIVE CHANNEL LIST
C
      I = 1
  110 IF (ICHAN .EQ. IPRT(I,1)) GO TO 120
      I = I + 1
      IF (I - MCHAN) 110,190,60
  120 MCOL = IPRT(I,4)
C
C     CHECK FOR DUPLICATE DATA SETS AND WRITE ERROR MESSAGE
C
      IF (IREAD(MCOL) .EQ. 0) GO TO 140
      WRITE (6,130) ICHAN,IIME(NROW,3),NDATE,NYEAR,A,ICHAR,
     1 ITEST,LATUM,UNITS,I1
  130 FORMAT (A,'DUPLICATE DATA FOUND FOR CHANNEL ',I3,' AT ',I4,
     1 ' HOURS ON DAY ',I3,',',I2,',',I2,':'/
     2 1X,A1,1X,I2,A1,I7.6,2A2)
      GO TO 60
C
C     TEST FOR ERROR CONDITIONS AND SET FLAGS
```

```
      1000 CALL PLAY(ICHAN,TEST,FC,BROW,BCOL)
C
C     STORE RAW DATA (IF BC CHOSEN WAS DIFFERENT AND IN...)
C
      IF (IERROR(BCOL) .EQ. 0 .AND. IERROR.(...) ...) GO
     * DATA(BROW,BCOL) = DATUM
      UNIT(BROW,BCOL,1) = UNIT
C
      SET "CHANNEL POWER" FLAG, INCREMENT ...
C     "ALL CHANNELS READ"
C
      IREAD(BCOL) = 1
      NCOUNT = NCOUNT + 1
      IF (NCOUNT - NCHAN) 10,170,170
C
C     BEAT DATA OUT ONTO THE...
C
  170 CALL ERASEFACE
      WRITE (6,190)
  190 FORMAT (8X,'ESC TO ...
     1 ' : ',13,' CHANNELS READ')
C
C     SET "DATA POWER" ...
C
      IF (NCOUNT .LT. 0) N = N-1
      GOTO 1000
C
C     END OF DATA INPUT, ...
C
      CALL ...
      WRITE(5,200) JUNK
  200 FORMAT(20A4)
      WRITE(6,210)IF...
  210 FORMAT(8,'ENCODE ...
     *13,',',13,',',F14.2,20A4)
      GO TO 60
      END
```

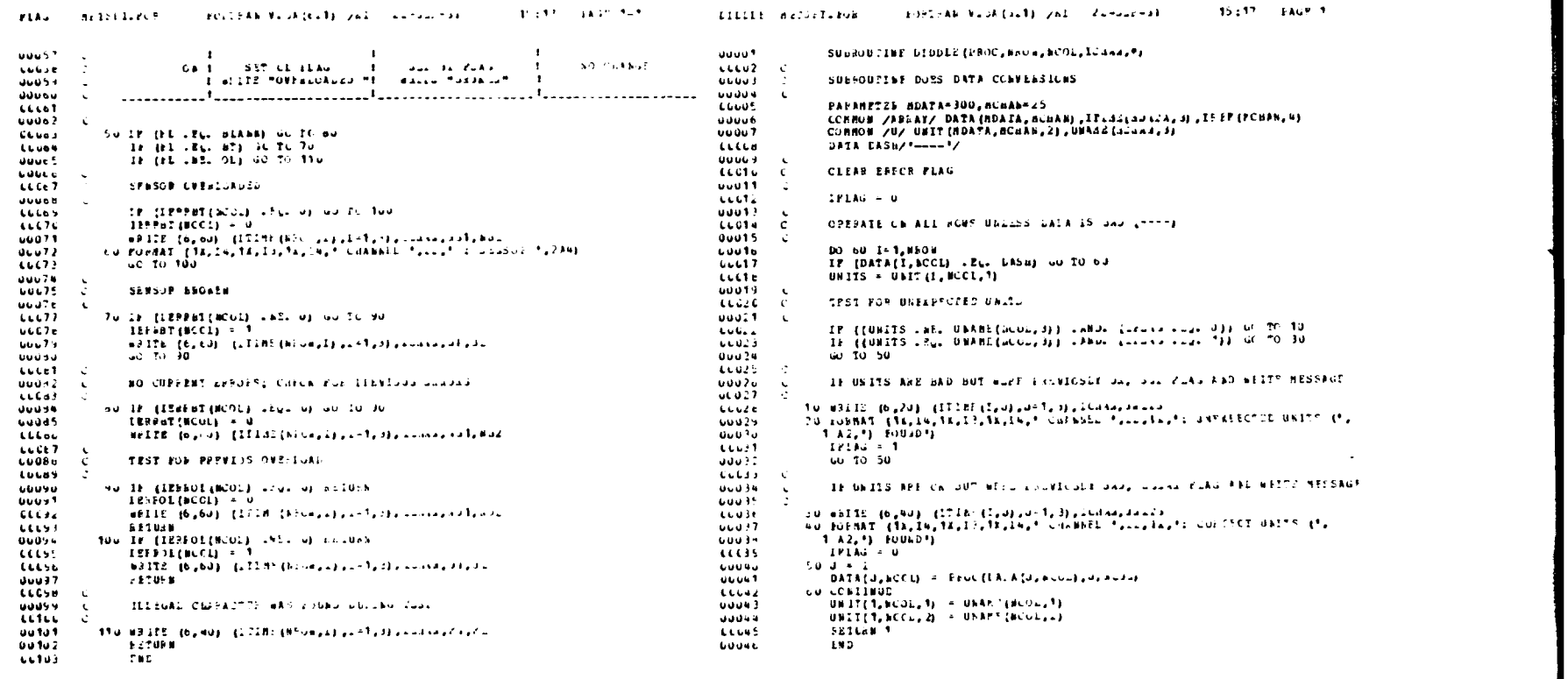
```
C
C     SUBROUTINE PARM1
C
C     SUBROUTINE CONVERTS DATA TO ...
C
      PARAMETER NDATA=...,NCHAN=...
      COMMON /APPAR/ DATA(NDATA,NCHAN),...,IERR(NCHAN,6)
      COMMON /B/ BETA...,BVOLT...,BCOUR,...,BUNIT...
      COMMON /U/ UNIT(NDATA,NCHAN,2),UNAME...
      EXTERNAL A1000,A100,PRESST,HUMID0,...,FREQV,FVOLT,DIFF0,FV0,...
     1 FREQV,FVOLT,DIFF0,FV0,...
C
C     CYCLE THROUGH ALL CHANNELS
C
      DO 770 I=1,NCHAN
C
C     LOOK UP CHANNEL AND ...
C
      ICHAN = IREF(I,1)
      NCOL = IREF(I,4)
C
C     SELECT THE PROPER FUNCTION
C
      GO TO (10,10,10,10,10,10,10,10,10,10,...,100,110,120,130,140,150,160,170,
     *180,190,200,210,220,230,240,250,260,270,280,290,300,310,320,330,340,350,
     *10,10,10,10,10,10,360,370) ICHAN
C
C     ICHAN OUT OF RANGE; ...
C
      WRITE (6,690) ICHAN,I
  690 FORMAT (1H,'ILLEGAL CHANNEL NUMBER ',...,' (ICHAN=I3.3M + ',I2,')')
      GO TO 770
C
C     NO FUNCTION HAS BEEN DEFINED FOR THIS CHANNEL
C
   10 WRITE (6,700) ICHAN,I
  700 FORMAT (1H,'UNSPECIFIED ...ECTED FOR CHANNEL ',I2,' (ICHANGE = ',
     1 I2,')')
      GO TO 770
  100 CALL DIDDLE(A10,BROW,BCOL,ICHAN,671)
  110 CALL DIDDLE(WINDBTS,BROW,BCOL,ICHAN,671)
  120 CALL DIDDLE(WIND2,BROW,BCOL,ICHAN,671)
  130 CALL DIDDLE(WIND10,BROW,BCOL,ICHAN,671)
  140 CALL DIDDLE(EQUALC,BROW,BCOL,ICHAN,671)
  150 CALL DIDDLE(A1000,BROW,BCOL,ICHAN,671)
  160 CALL DIDDLE(HUMID0,BROW,BCOL,ICHAN,671)
  200 CALL DIDDLE(PRESST,BROW,BCOL,ICHAN,671)
  210 CALL DIDDLE(HUMID1,BROW,BCOL,ICHAN,671)
  220 CALL DIDDLE(TEMP1,BBOW,BCOL,ICHAN,671)
  230 CALL DIDDLE(A100,BROW,BCOL,ICHAN,671)
  240 CALL DIELLE(DIFF,BROW,BCOL,ICHAN,671)
  250 CALL DIDDLE(DIFV1,BROW,BCOL,ICHAN,671)
  260 CALL DILDLE(POSV,BROW,BCOL,ICHAN,671)
  270 CALL DIDDLE(FREQV,BROW,BCOL,ICHAN,671)
  300 CALL DIELLE(FVOLT,BBOW,BCOL,ICHAN,671)
  710 CONTINUE
  720 RETURN
      END
```

```
DATOUT  WRITSFT.MOR     MORTRAN V.5A(561) /AS   22-JUL-81       15:17   PAGE 1

00001          SUBROUTINE DATOUT(NCHAN,NBUFF,ATITLE)
00002      C
00003      C     SUBROUTINE PRINTS OUT THE RESULTS IN SPECIFIED FORMATS
00004      C
00005          PARAMETER NDATA=100,NCHAN=25
00006          DIMENSION OUT1(80),OUT2(80),BUFFER(1:,:)
00007          COMMON /PRINT/NCHAN),IPRINT(NCHAN),
00008          COMMON /ARRAY/ DATA(NDATA,NCHAN),
00009          COMMON /F/FORM(NCHAN,3)
00010          COMMON /II/ IDATE,II,A3,IDAY,IDATE,
00011          COMMON /O/ UNIT(NDATA,NCHAN,2),UNAME(NCHAN,3)
00012          DATA C1,C2,F11,F12,F13,A3,A4,1A3H/'    ','   ',' (I3',',I4,',
00013         *)   ',' A3,',' A4,','----'/
00014      C
00015      C     DUMP ALL DATA (WHICH EXISTS (FORMAT.NE.0)) FOR EACH CHANNEL
00016      C
00017          WRITE (7,90) ATITLE,NCHAN,NBUFF,IYEAR,IDATE,IDAY,
00018        90 FORMAT (1H1,5X,5A5/2(A13)/(1X,I4,1X,I3,1X,A4))
00019        95 WRITE (7,20) ((IFORM(I,J),J=1,3),(UNAME(I,J),J=1,4),
00020         1 (FORM(I,J),J=1,3),I=1,NCHAN)
00021        20 FORMAT (1X,13,1X,2A4,1X,2A4,1X,3A4)
00022           WRITE (7,60) ((IFORM(I,J),J=1,3),I=1,NCHAN)
00023        30 FORMAT (6(1X,I4,1X,I3,1X,I4))
00024           DO 45 J=1,NCHAN
00025           WRITE (7,35) IBEP(J,1)
00026        35 FORMAT (1X,'CHANNEL NUMBER ',I3)
00027           WRITE (7,60) (DATA(I,J),I=1,NBUF)
00028        40 FORMAT (6(1X,F12.6))
00029        45 CONTINUE
00030      C
00031      C     LOCATE COLUMNS OF DATA ARRAY WHICH ARE TO BE PRINTED OUT; FIND
00032      C     THE CORRESPONDING CHANNEL NUMBERS, NAMES, AND UNITS.
00033      C     J=OLD COLUMN NUMBER
00034      C     K=NEW COLUMN NUMBER
00035      C
00036          NPRINT = 0
00037          J = 1
00038          K = 1
00039        50 IF (J - NCHAN) 70,70,80
00040      C
00041      C     PACK INFORMATION OF CHANNELS TO BE PRINTED INTO PRINT OUT LOCATIONS
00042      C     OF REFERENCE ARRAY
00043      C
00044        70 IF (IPRINT(J) .EQ. 0) GO TO 90
00045           UNAME(K,1) = UNAME(J,1)
00046           UNAME(K,2) = UNAME(J,2)
00047           IBEP(K,1) = IBEP(J,1)
00048           IBEP(K,2) = IBEP(J,2)
00049           IBEP(K,3) = IBEP(J,3)
00050           IBEP(K,4) = IBEP(J,4)
00051           FORM(K,1) = FORM(J,1)
00052           FORM(K,2) = FORM(J,2)
00053           FORM(K,3) = FORM(J,3)
00054           NPRINT = NPRINT + 1
00055           K = K + 1
00056        80 J = J + 1
```

```
DATOUT  WRITSFT.MOR     MORTRAN V.5A(561) /AS   22-JUL-81       15:17   PAGE 1-1

00057          GO TO 50
00058      C
00059      C     SET UP TO PRINT 8 COLUMNS (MAX=10)
00060      C
00061        85 NFIRST = 1
00062        90 IPRINT = 8-1
00063           IF (IPRINT .GT. 12) IPRINT = 12
00064           NLAST = NFIRST + IPRINT - 1
00065      C
00066      C     WRITE PAGE TITLE
00067      C
00068           WRITE (6,100) ITITLE,IDAY,IDATE,IYEAR,NBUFF,NPRINT
00069        100 FORMAT (1H1,53X,5A5/38X,'DATA FOR THE NUMBER ',
00070          1 A3,1X,13,',',26,' INFORMATION ',A3,1X,A3,',',40//)
00071      C
00072      C     WRITE CHANNEL NUMBERS, CHANNEL NAMES, AND UNITS
00073      C
00074           WRITE (6,110) (C1,C2,I=1,IPRINT)
00075        110 FORMAT (12A,12(2X,2A4))
00076           WRITE (6,120) (IBEP(I,1),I=NFIRST,NLAST)
00077        120 FORMAT (12X,12(3X,I3,4X))
00078           WRITE (6,110) (IBEP(I,2),IBEP(I,3),I=NFIRST,NLAST)
00079           WRITE (6,110) (UNAME(I,1),UNAME(I,2),I=NFIRST,NLAST)
00080      C
00081      C     CONSTRUCT OUTPUT FORMAT ARRAY "OUT1"
00082      C
00083           OUT1(1) = F11
00084           OUT1(2) = F12
00085           L = 3
00086           DO 140 J=NFIRST,NLAST
00087           DO 140 K=1,3
00088           OUT1(L) = FORM(J,K)
00089        140 L = L + 1
00090           OUT1(L) = F13
00091           LMAX = L
00092      C
00093      C     WRITE YEAR, DATE (WHEN DIFFERENT FROM THE PREVIOUS VALUES),
00094      C     TIME, AND DATA
00095      C
00096           JYEAR = 0
00097           JDATE = 0
00098           DO 190 I=1,NBUF
00099      C
00100      C     COPY OUTPUT FORMAT ARRAY INTO "OUT2"
00101      C
00102           DO 145 L = 1,LMAX
00103        145 OUT2(L) = OUT1(L)
00104           K = 1
00105      C
00106      C     LOAD DATA INTO BUFFER AND LOOK FOR THE DATA (-----) ; IF FOUND,
00107      C     ALTER CORRESPONDING LOCATION IN "OUT2"
00108      C
00109           DO 150 J=NFIRST,NLAST
00110           BUFFER(K) = DATA(I,IBEP(J,4))
00111           IF (BUFFER(K) .EQ. DASH) GO TO 150
00112           OUT2(3*K) = X3
```

```
DATOUT  METSPT.FOR      FORTRAN V.5A(6a1) /Al   22-Dec-81        15:17   PAGE 1-2

00113          OUT2(3*K + 1) = A4
00114          OUT2(3*K + 2) = A3
00115      150 K = K + 1
00116          IF (ITIME(I,1) .EQ. JYEAR) GO TO 170
00117          WRITE (6,160) ITIME(I,1)
00118      160 FORMAT (1X,I4)
00119          JYEAR = ITIME(I,1)
00120      170 IF (ITIME(I,2) .EQ. JDATE) GO TO 190
00121          WRITE (6,180) ITIME(I,2)
00122      180 FORMAT (5X,I3)
00123          JDATE = ITIME(I,2)
00124  C
00125  C       WRITE DATA ACCORDING TO "OUT2" AS REQUIRED ABOVE
00126  C
00127      190 WRITE (6,OUT2) ITIME(I,3),(SUBREE(J),J=1,NCHAR2)
00128          NPRINT = NPRINT - 1PRINT
00129          NFIRST = NLAST + 1
00130          IF (NPRINT .GT. 0) GO TO 90
00131          RETURN
00132          END
```

```
FLAG    METSPT.FOR      FORTRAN V.5A(6a1) /Al   22-Dec-81        15:17   PAGE 1

00001          SUBROUTINE FLAG(ICHAR,IOUT,FI,BLANK,ASSG)
00002  C
00003  C       SUBROUTINE TESTS FOR UPPER OR LOWER ASSIGN BACKSPACE ARE PLUS BROKEN
00004  C       OR OVERLOADED SENSORS. WARNING MESSAGES ARE ASSIGNED AS NECESSARY.
00005  C
00006          PARAMETER MDATA=300,MCHAR=20
00007          COMMON /ARRAY/ DATA(MDATA,MCHAR),ITIME(MDATA,3),IFREE(MCHAR,4)
00008          COMMON IPRINT(MCHAR),IOUT(MCHAR),ASSIG(MCHAR)
00009          DATA BLANK,PLESS,PMORE,O1,O2/'    ','<  ','>  ','O1  ','O2  '/
00010          DATA U1,U2,I1,I2/'U1  ','U2  ','I1  ','I2  '/
00011          DATA C1,C2,U*,U2/'OVER','LOAD','U1  ','U2  '/
00012          DATA T1,F1/'TEST','FLAG'/
00013          DATA NB1,NB2,NO1,NO2/'1B O','*A  ','2B O','*A  '/
00014  C
00015  C       TEST FOR UPPER OR LOWER LIMIT EXCEEDED
00016  C
00017          IF (TEST .EQ. BLANK) GO TO 30
00018          IF (TEST .EQ. PMORE) GO TO 20
00019          IF (TEST .NE. PLESS) GO TO 30
00020  C
00021  C       LOWER LIMIT EXCEEDED
00022  C
00023          WRITE (6,10) (ITIME(NFCH,I),I=1,3),CHARA,O1,O2
00024       10 FORMAT (1X,I4,1X,I3,1X,I4,' CHANNEL ',A4,' : ',A1,A4,
00025         1 ' LIMIT EXCEEDED.')
00026          GO TO 50
00027  C
00028  C       UPPER LIMIT EXCEEDED
00029  C
00030       20 WRITE (6,10) (ITIME(NFCH,I),I=1,3),CHARA,O1,O2
00031          GO TO 50
00032  C
00033  C       ILLEGAL CHARACTER WAS FOUND DURING TEST
00034  C
00035       30 WRITE (6,40) (ITIME(NFCH,I),I=1,3),CHARA,O1,TEST
00036       40 FORMAT (1X,I4,1X,I3,1X,I4,' CHANNEL ',A4,' : ',
00037         1 'UNEXPECTED CHARACTER : ',A4,' = ',A4)
00038  C
00039  C       TEST FOR BROKEN OR OVERLOADED SENSOR FLAGS SET AND MESSAGES PRINTED
00040  C       ACCORDING TO THE FOLLOWING MATRIX:
00041  C
00042  C                                                  CHANGE CONDITION
00043  C                  !        OL          !        BT          !        OK
00044  C   ---------------!--------------------!--------------------!--------------------
00045  C                  !                    !                    !
00046  C                  !                    !   NO CHANGE IN FLAG!
00047  C                  !    NO CHANGE       !   WRITE "IN RANGE" !    CLR OL FLAG
00048  C                  !                    !   NO CHANGE IN FLAG!  WRITE "IN RANGE"
00049  C                  !                    !   WRITE "BROKEN"   !
00050  C                  !--------------------!--------------------!--------------------
00051  C                  !                    !                    !
00052  C                  !    CLR BT FLAG     !                    !
00053  C                  !  WRITE "NOT BROKEN"!                    !    CLR BT FLAG
00054  C                  !    SET OL FLAG     !   NO CHANGE        !  WRITE "NOT BROKEN"
00055  C                  !  WRITE "OVERLOADED"!                    !
00056  C                  !--------------------!--------------------!--------------------
```



```
      SUBROUTINE DIDDLE(IPROC,NROW,NCOL,ICHAN,*)
C
C     SUBROUTINE DOES DATA CONVERSIONS
C
      PARAMETER NDATA=300,NCHAN=25
      COMMON /ABRAY/ DATA(NDATA,NCHAN),IFLAG(NDATA,3),IERP(PCHAN,4)
      COMMON /U/ UNIT(NDATA,NCHAN,2),UNAME(NCHAN,3)
      DATA DASH/'-----'/
C
C     CLEAR ERROR FLAG
C
      IFLAG = 0
C
C     OPERATE ON ALL ROWS UNLESS DATA IS BAD (-999)
C
      DO 60 I=1,NROW
      IF (DATA(I,NCOL) .EQ. DASH) GO TO 60
      UNITS = UNIT(I,NCOL,1)
C
C     TEST FOR UNEXPECTED UNITS
C
      IF ((UNITS .NE. UNAME(NCOL,3)) .AND. (UNITS .NE. '   ')) GO TO 10
      IF ((UNITS .EQ. UNAME(NCOL,3)) .AND. (UNITS .NE. '   ')) GO TO 30
      GO TO 50
C
C     IF UNITS ARE BAD BUT WERE PREVIOUSLY BAD, SET FLAG AND WRITE MESSAGE
C
   10 WRITE (6,20) (IFLAG(I,J),J=1,3),ICHAN,NCOL
   20 FORMAT (1X,14,1X,13,1X,14,' CHANNEL ',I3,': UNEXPECTED UNITS (',
     1 A2,') FOUND')
      IFLAG = 1
      GO TO 50
C
C     IF UNITS ARE OK BUT WERE PREVIOUSLY BAD, RESET FLAG AND WRITE MESSAGE
C
   30 WRITE (6,40) (IFLAG(I,J),J=1,3),ICHAN,NCOL
   40 FORMAT (1X,14,1X,13,1X,14,' CHANNEL ',I3,': CORRECT UNITS (',
     1 A2,') FOUND')
      IFLAG = 0
   50 J = 1
      DATA(J,NCOL) = PROC(IA,A(J,NCOL),J,ICHAN)
   60 CONTINUE
      UNIT(1,NCOL,1) = UNAME(NCOL,1)
      UNIT(1,NCOL,2) = UNAME(NCOL,2)
      RETURN 1
      END
```

```
DIV10   METSET.FOR      FORTRAN V.5A(621) /AI   22-xxx-xx       15:17   PAGE 1

00001          FUNCTION DIV10(DATUM,IROW,ICOL)
00002    C
00003    C     FUNCTION CONVERTS MILLIVOLTS TO ENGLISH UNITS & SCALE FACTOR
00004    C     OF ONE TENTH. (NEW CHAN. 14)
00005    C
00006          DIV10 = DATUM * .1
00007          RETURN
00008          END
```

```
EQUAL   METSET.FOR      FORTRAN V.5A(621) /AI   22-xxx-xx       15:17   PAGE 1

00001          FUNCTION EQUAL(DATUM,IROW,ICOL)
00002    C
00003    C     FUNCTION APPLIES A SCALE FACTOR OF 1.0 (CHAN. 30-32,40,41)
00004    C
00005          EQUAL = DATUM
00006          RETURN
00007          END
```

```
X10     METSET.FOR      FORTRAN V.5A(621) /AI   22-xxx-xx       15:17   PAGE 1

00001          FUNCTION X10(DATUM,IROW,ICOL)
00002    C
00003    C     FUNCTION CONVERTS VOLTS TO MILLIVOLTS AND APPLIES A SCALE
00004    C     FACTOR OF 10!. (NEW CHAN. 10,11)
00005    C
00006          X10 = DATUM * 10.
00007          RETURN
00008          END
```

```
EQUALC  METSET.FOR      FORTRAN V.5A(621) /AI   22-xxx-xx       15:17   PAGE 1

00001          FUNCTION EQUALC(DATUM,IROW,ICOL)
00002    C
00003    C     FUNCTION APPLIES A SCALE FACTOR OF 1.0 AND CORRECTS RANGE
00004    C     DC=EQUALCC-360 TO RANGE DC=EQUALCC+360 (NEW CHAN. 15)
00005    C
00006          EQUALC = DATUM
00007       10 IF (EQUALC .LT. 360.) RETURN
00008          EQUALC = EQUALC - 360.
00009          GO TO 10
00010          END
```

```
X1000   METSET.FOR      FORTRAN V.5A(621) /AI   22-xxx-xx       15:17   PAGE 1

00001          FUNCTION X1000(DATUM,IROW,ICOL)
00002    C
00003    C     FUNCTION CONVERTS VOLTS TO MILLIVOLTS AND APPLIES A SCALE
00004    C     FACTOR OF ONE. (OLD CHAN. 17,19)
00005    C
00006          X1000 = DATUM * 1000.
00007          RETURN
00008          END
```

```
MNEATS  METSET.FOR      FORTRAN V.5A(621) /AI   22-xxx-xx       15:17   PAGE 1

00001          FUNCTION MNEATS(DATUM,IROW,ICOL)
00002    C
00003    C     FUNCTION CONVERTS VOLTS TO MILLIVOLTS AND APPLIES A SCALE
00004    C     FACTOR OF ONE TENTH AND CONVERTS MPH TO KNOTS. (NEW CHAN. 12)
00005    C
00006          MNEATS = DATUM * 0.869
00007          RETURN
00008          END
```

```
X100    METSET.FOR      FORTRAN V.5A(621) /AI   22-xxx-xx       15:17   PAGE 1

00001          FUNCTION X100(DATUM,IROW,ICOL)
00002    C
00003    C     FUNCTION CONVERTS VOLTS TO MILLIVOLTS AND APPLIES A SCALE
00004    C     FACTOR OF ONE TENTH. (OLD CHAN. 13)
00005    C
00006          X100 = DATUM * 100.
00007          RETURN
00008          END
```

```
PRESS1  METSET.FOR      FORTRAN V.5A(621) /AI   22-xxx-xx       15:17   PAGE 1

00001          FUNCTION PRESS1(DATUM,IROW,ICOL)
00002    C
00003    C     FUNCTION TESTS UNITS, CONVERTS VOLTS TO MILLIVOLTS WHEN
00004    C     NECESSARY, AND APPLIES THE EQN. P=(29.47/30.0)*(1V)*749
00005    C     TORR. (CHAN. 20)
00006    C
00007          PARAMETER MDATA=300,MCHAN=20
00008          COMMON /U/ UNIT(MDATA,MCHAN,2),UNAME(xxxxx,3)
00009          DATA V/'V   '/
00010          IF (UNIT(IROW,ICOL,1) .EQ. V) DATUM = 1000 * DATUM
00011          PRESS1 = .982 * DATUM * 749.
00012          RETURN
00013          END
```

```
HUMID0  MEISP1.FOR      FORTRAN V.5A(041) /AI   22-JUL-91        1:17   PAGE 1


00001          FUNCTION HUMID0(DATUM,IFCW,ICCL)
00002      C
00003      C     FUNCTION CONVERTS VOLTS TO MILLIVOLTS, APPLIES A CALIBRATION
00004      C     FACTOR, AND CALLS "HUMCOR" FOR VALUES GREATER THAN 1.0 (CHAN. 14)
00005      C
00006          HUMID0 = DATUM * 1000.
00007          IF ((HUMID0 .LT. 0.) .OR. (HUMID0 .GT. 1000.))
     1     1  CALL HUMCOR(HUMID0,IFCW,ICCL)
00008          RETURN
00009          END
```

```
HUMID1  MEISP1.FOR      FORTRAN V.5A(041) /AI   22-JUL-91        1:17   PAGE 1


00001          FUNCTION HUMID1(DATUM,IFCW,ICCL)
00002      C
00003      C     FUNCTION APPLIES THE EQN. H=700*(V) TO VOLTAGES FROM 0 TO
00004      C     VALUES GREATER THAN 1000. (CHAN. 15)
00005      C
00006          PARAMETER MDATA=500,MCHAN=20
00007          COMMON /ARRAY/ DATA(MDATA,MCHAN),IFREQ(MCHAN,4)
00008          HUMID1 = 20. * DATUM
00009          IF ((HUMID1 .LT. 0.) .OR. (HUMID1 .GT. 1000.))
     1     1  CALL HUMCOR(HUMID1,IFCW,ICCL)
00010          RETURN
00011          END
00012
```

```
TEMP1   MEISA1.FOR      FORTRAN V.5A(041) /AI   22-JUL-91        15:17   PAGE 1


00001          FUNCTION TEMP1(DATUM,IFCW,ICCL)
00002      C
00003      C     FUNCTION APPLIES EQN. T=400*(V)-50 AND CONVERTS TO DEG. C (CHAN 22)
00004      C
00005          TEMP1 = 5./9. * (400. * DATUM - 62.)
00006          RETURN
00007          END
```

```
DIF1    MEISP1.FOR      FORTRAN V.5A(041) /AI   22-JUL-91        1:17   PAGE 1


00001          FUNCTION DIF1(DATUM,IFCW,ICCL)
00002      C
00003      C     FUNCTION APPLIES EQN. D=1000*(V);OUTPUT IS 0-5V;0-1000*(V).
00004      C     -1.000<=V<=0. (OLD CHAN. 24)
00005      C
00006          DIF1 = 1000. * DATUM
00007          IF (DATUM .LT. 0) DIF1 = 1000. * DATUM
00008          RETURN
00009          END
```

```
DIF2    MEISP1.FOR      FORTRAN V.5A(041) /AI   22-JUL-91        15:17   PAGE 1


00001          FUNCTION DIF2(DATUM,IFCW,ICCL)
00002      C
00003      C     FUNCTION CONVERTS VOLTS TO MILLIVOLTS, APPLIES A CALIBRATION
00004      C     FACTOR OF ONE, AND CORRECTS RANGE OF -1.000<=V<=0.000 TO RANGE
00005      C     0<=DIF2<=1000. (NEW CHAN. 13)
00006      C
00007          DIF2 = 1000. * DATUM
00008     10   IF (DIF2 .LT. 500.)  RETURN
00009          DIF2 = DIF2 - 500.
00010          GO TO 10
00011          END
```

```
DIFV1   MEISP1.FOR      FORTRAN V.5A(041) /AI   22-JUL-91        15:17   PAGE 1


00001          FUNCTION DIFV1(DATUM,IFCW,ICCL)
00002      C
00003      C     FUNCTION CHECKS VOLTAGE OF MIRO DIFFERENCE CHECK SUPPLY AND
00004      C     ISSUES A WARNING IF IT LIES OUTSIDE THE RANGE 3.00<V<5.00 V.
00005      C     (OLD CHAN. 25)
00006      C
00007          PARAMETER MDATA=500,MCHAN=20
00008          COMMON /ARRAY/ DATA(MDATA,MCHAN),IFREQ(MCHAN,4)
00009          DIFV1 = DATUM
00010          IF ((DATUM .LT. 3.0) .OR. (DATUM .GT. 5.0))
     1     1  WRITE (6,10) (ICHAR(ICCW,I),I=1,3),(ICHAR(ICCL,I),I=1,3),DATUM
00012     10   FORMAT (1X,I4,1X,I3,1X,3A1,' CHANNEL ',A3,
     1     1  ': M.D. VOLTAGE OUT OF RANGE (',F7.4,' V)')
00013          RETURN
00014          END
```

```
VOSV    MEISP1.FOR      FORTRAN V.5A(041) /AI   22-JUL-91        15:17   PAGE 1


00001          FUNCTION VOSV(DATUM,IFCW,ICCL)
00002      C
00003      C     FUNCTION CHECKS VOLTAGE OF THE POSITIVE STRAIN SUPPLY AND ISSU
00004      C     A WARNING IF IT LIES OUTSIDE THE RANGE 4.75<V<5.25 (CHAN. 26)
00005      C
00006          PARAMETER MDATA=500,MCHAN=20
00007          COMMON /ARRAY/ DATA(MDATA,MCHAN),IFREQ(MCHAN,4)
00008          VOSV = DATUM
00009          IF ((DATUM .LT. 4.5) .OR. (DATUM .GT. 5.5))
     1     1  WRITE (6,10) (ICHAR(ICCW,I),I=1,3),(ICHAR(ICCL,I),I=1,3),DATUM
00011     10   FORMAT (1X,I4,1X,I3,1X,3A1,' CHANNEL ',A3,
     1     1  ': VOS. EXP. SUPPLY OUT OF RANGE (',F7.4,' V)')
00013          RETURN
00014          END
```

```
PRESW   METSEL.FOR      FORTRAN V.5A(041) /RI   22-Sep-81       15:17   PAGE 1


00001         FUNCTION PRESW(DATUM,IECH,ICOL)
00002   C
00003   C     FUNCTION CHECKS VOLTAGE OF THE BAROMETRIC PRESS SUPPLY AND IF FOUND
00004   C     A WARNING IS ISSUED AND THE VALUE RETURNED=-1.0 (CHAN. 27)
00005   C
00006         PARAMETER MDATA=300,NCHAN=15
00007         COMMON /ARRAY/ DATA(MDATA,NCHAN),ITIME(MDATA,3),IZERO(NCHAN,4)
00008         PRESW = DATUM
00009         IF ((DATUM .GT. -17.0) .AND. (DATUM .LT. -17.2))
00010        1 WRITE (6,10) (ITIME(IECH,I),I=1,3),DATUM
00011       10 FORMAT (1X,I4,'/',I2,'/',I2,' CHANNEL ',I2,
00012        1 ': NEG. PRES. SUPPLY OUT OF RANGE (',F6.2,' V)')
00013         RETURN
00014         END
```

```
HUMCOR  METSEL.FOR      FORTRAN V.5A(041) /RI   22-Sep-81       15:17   PAGE 1


00001         SUBROUTINE HUMCOR(DATUM,IECH,ICOL)
00002   C
00003   C     SUBROUTINE ISSUES A WARNING AND CORRECTS THE DATUM IF HUMIDITY IS FOUND
00004   C     TO BE LESS THAN 0.0 PERCENT OR GREATER THAN 100.0 PERCENT.
00005   C
00006         PARAMETER MDATA=300,NCHAN=15
00007         COMMON /ARRAY/ DATA(MDATA,NCHAN),ITIME(MDATA,3),IZERO(NCHAN,4)
00008         DATA ZERO,HUNDRD/0.0,100.0/
00009         IF (DATUM .GE. ZERO) GO TO 20
00010         WRITE (6,10) (ITIME(IECH,I),I=1,3),DATUM,ZERO
00011       10 FORMAT (1X,I4,'/',I2,'/',I2,' CHANNEL ',I2,
00012        1 ': HUMIDITY OUT OF RANGE (',F6.0,'). CORRECTED TO ',F6.0)
00013         DATUM = ZERO
00014       20 IF (DATUM .LE. HUNDRD) GO TO 30
00015         WRITE (6,10) (ITIME(IECH,I),I=1,3),DATUM,HUNDRD
00016         DATUM = HUNDRD
00017       30 RETURN
00018         END
```

Appendix B
Listing of Program METCLC
(Version 1.0)

```
      PROGRAM METCLC
C
C     METCLC VERSION 1.0   12/11/80
C
C     JOHN MOORE
C     CODE 4330
C     U.S. NAVAL RESEARCH LABORATORY
C     WASHINGTON, D.C.  20375
C
C
C     PROGRAM READS "METSPI" OUTPUT FILE AND CALCULATES ABSOLUTE
C     WIND VELOCITY, WATER VAPOR LOADING, AND OZONE CONCENTRATION.
C     THIS DATA IS WRITTEN INTO A FILE DESIGNED TO BE READ BY
C     PROGRAM "METPLT".  ANY OR ALL OF THE DATA MAY ALSO BE
C     PRINTED.
C
C     PROGRAM PARAMETERS:
C
C        MDATA=MAXIMUM POSSIBLE NUMBER OF DATA SETS
C              (CURRENTLY, MDATA=300)
C        MCHAN =MAXIMUM POSSIBLE NUMBER OF DATA CHANNELS PER SET
C              (CURRENTLY, MCHAN=20)
C
C     TEST-NAME OF THE PARAMETER FILE DESIRED (TEST="CALC"
C          FOR THIS PROGRAM)
C
C     TITLE-HEADING TO BE USED AS OUTPUT PAGE (STARTS AT 5A5)
C
C     WORD-KEYWORD TO SELECT SPECIFIC SUBROUTINE (SEE BELOW FOR LIST OF
C          VALID KEYWORDS) (FORMAT=A4)
C
C     INPUT VARIABLES AND FORMATS:
C
C        NCHAN =NUMBER OF DATA CHANNELS PER SET FOR CURRENT RUN
C        NPCH=NUMBER OF SETS IN THE DATA ARRAY
C        MXINUM=NEXT AVAILABLE DEFAULT CHANNEL NUMBER
C        IYEAR=INITIAL YEAR
C        IDATE=INITIAL JULIAN DATE
C        IDAY=INITIAL DAY OF WEEK
C        LYEAR,LDATE,LDAY REFER TO LAST (YEAR,DATE,DAY
C              (FORMAT=3(I3),,(/4,10,4,13,4,A3))
C
C     DATA  INPUT AND FORMATS:
C
C        IREF  =ARRAY WHICH CROSS REFERENCES CHANNEL NUMBER, DATA TYPE,
C               AND STORAGE LOCATION WITHIN THE DATA ARRAY (COLUMN NUMBER)
C        UNAME =ARRAY OF NAMES OF OUTPUT DATA
C        FORM  =ARRAY CONTAINING THE FORMATS FOR PRINTING RESULTS
C               (FORMAT=4(3,A,A4,A,2A4,,A4))
C
C        ITIME =ARRAY HOLDING THE YEAR,DATE, AND TIME FOR EACH
C               DATA SET.  THE HOURS CORRESPOND TO THE HOUR OF THE
C               DATA RECORD.
C               (FORMAT=(I,14,A,13,A,I4)
C
C        DATA  =ARRAY INTO WHICH ALL DATA IS PLACED.  DATA IS ARRANGED
```



```
C               IN ROWS (CORRESPONDING TO SUCCESSIVE TIMES) AND COLUMNS
C               (CORRESPONDING TO DIFFERENT DATA CHANNELS).
C               (FORMAT=6(X,E12.6))
C
C
C     KEYWORDS:
C
C     FUNCTIONS WITHIN "METCLC" ARE SELECTED BY USE OF
C     KEYWORDS.  CURRENTLY, THE LEGAL KEYWORDS ARE:
C        "WIND " = SELECT VECTOR ADDITION SUBROUTINE FOR CALCULATION
C                  OF ABSOLUTE WIND VELOCITIES.
C        "MOIST" = SELECT FUNCTION TO CALCULATE ATMOSPHERIC WATER
C                  VAPOR LOADINGS.
C
C     EACH SUBROUTINE IS RESPONSIBLE FOR SETUP OF PARAMETERS SPECIFIC TO
C     THAT FUNCTION.  SEE COMMENTS WITHIN THE SUBROUTINES FOR DETAILS.
C
C*****************************************************************************
C
C                                 ERROR MESSAGES
C                                 ----- --------
C
C                        <NOTE:  I = CHANNEL NUMBER>
C
C                MESSAGE                 SUBJECT                 MEANING
C                -------                 -------                 -------
C     NO PARAMETER FILE FOUND            METCLC      FILE PCSGT.DAT WAS MISSING OR
C                                                    NO SECTION LABELED "CALC
C                                                    PARAMETERS" COULD BE FOUND.
C
C     NO DATA FILE FOUND                 METCLC      FILE PCSGT.DAT WAS NOT
C                                                    FOUND
C
C     MAXIMUM NUMBER OF CHANNELS         MAXX        A SUBTRACT WAS CALLED WHICH
C     WOULD BE EXCEEDED BY I                         REQUIRED ADDITIONAL COLUMNS IN
C     CURRENT NUMBER OF CHANNELS= J                  THE DATA ARRAY FOR ITS OUTPUT,
C     FURTHER ITERATIONS CANCELLED                   AND THE ARRAY AS DIMENSIONED
C                                                    BY PARAMETER "MCHAN" DID NOT
C                                                    HAVE SUFFICIENT SPACE.  PARAM-
C                                                    ETER STATEMENTS IN "METCLC"
C                                                    AND AN "ASSIGN" SHOULD BE
C                                                    CHANGED BEFORE EXECUTING THE
C                                                    PROGRAM AGAIN.
C
C     REQUESTED CHANNEL (N) DOES NOT     SUBTRACT    CHANNEL N IS NOT AVAILABLE FOR
C     EXIST. ITERATION ABORTED.                      SUBTRACTION OUTPUT
C
C*****************************************************************************
C
C     ALL "PARAMETER" STATEMENTS MUST BE CHANGED IF THERE ARE MORE THAN 20
C     CHANNELS OR 300 DATA SETS!
C
C     PROGRAM UNITS CURRENTLY USING PARAMETER STATEMENTS:
C     METCLC, INIT, DATAIN, WACK, LAYOUT, SMOOTH, PSTOOL, SETUP, ADDWND,
C     WNDCLC, TSTDAT, MOIST, VAICLC
C
      PARAMETER MDATA=300,MCHAN=20
```

```
00113         DIMENSION ITITLE(5)
00114         COMMON IPRINT(MCHAN)
00115         COMMON /ASRAY/ DATA(MDATA,MCHAN),IF...(MDATA,3),...EF(MCHAN,4)
00116         COMMON /F/ FOPN(MCHAN,3)
00117         COMMON /I1/ IDATE,IYEAR,IDAY,LDATE,...AY
00118         COMMON /B/ MCHAN,NFCN,NAINIK
00119         COMMON /U/ UNASE(MCHAN,2)
00120         COMMON /MAT/ MTCHAN(MCHAN),MICOL(MCHAN
00121         EXTERNAL REVAL
00122         DATA CALC/'CALC'/
00123         OPEN (UNIT=6,DEVICE='DSK',ACCESS='...',...='FORT6')
00124         OPEN (UNIT=4,DEVICE='DSK',ACCESS='...',...='...',
00125        1 ERR=110)
00126         OPEN (UNIT=7,DEVICE='DSK',ACCESS='...',...='FORT7',
00127        1 ERR=130)
00128   C
00129   C     LOOK FOR "CALC" SECTION OF FORT4.DAT
00130   C
00131    10 READ (4,20,END=110) WORD
00132    20 FORMAT (A5)
00133       IF (WORD .NE. CALC) GO TO 10
00134       READ (4,30) ITITLE
00135    30 FORMAT (15A5)
00136       CALL INIT
00137       KOUNT = 0
00138       CALL DATAIN(ITITLE)
00139    40 READ (4,50,END=60) WORD
00140    50 FORMAT (X,A5)
00141       WRITE (6,55)
00142    55 FORMAT (//)
00143       GO TO (60,60,60,60,60,60) ALLVAL(WORD)
00144    60 CALL DATOUT(MCHAN,NBCW,ITITLE)
00145       WRITE (6,70) KOUNT
00146    70 FORMAT (/X,I3,' FUNCTION CALLS')
00147       STOP 'PROGRAM EXECUTES SUCCESSFULLY'
00148    80 CALL ABSEND(040)
00149       KOUNT = KOUNT + 1
00150       GO TO 40
00151    90 CALL NOISY(040)
00152       KOUNT = KOUNT + 1
00153       GO TO 40
00154   110 WRITE (6,120)
00155   120 FORMAT (X'NO PARAMETER FILE FOUND')
00156       STOP 'NO PARAMETER FILE FOUND'
00157   130 WRITE (6,140)
00158   140 FORMAT (X'NO DATA FILE FOUND')
00159       STOP 'NO DATA FILE FOUND'
00160       END
```



```
00001         SUBROUTINE INIT
00002   C
00003   C     SUBROUTINE INITIALIZES DATA ...
00004   C     DATA ARRAY COLUMN NUMBERS FOR EACH ...
00005   C
00006         PARAMETER MDATA=300,MCHAN=15
00007         COMMON /ASRAY/ DATA(MDATA,MCHAN),...(MDATA,3),IPREF(MCHAN,4)
00008         DATA DASH/'---- '/
00009         DO 10 J=1,MCHAN
00010   C
00011   C     INITIALIZE COLUMN NUMBERS IN REFERENCE ARRAY
00012   C
00013         IREF(J,4) = J
00014   C
00015   C     INITIALIZE DATA ARRAY
00016   C
00017         DO 10 I=1,MDATA
00018      10 DATA(I,J) = DASH
00019         RETURN
00020         END
```

```
DATAIN  MERGEC.FOR      FORTRAN V4.5A(0a1) /Ka   22-Jul-81       15:16   PAGE 1

00001          SUBROUTINE DATAIN(ITITLE)
00002      C
00003      C     SUBROUTINE READS IN PARAMETERS, AND DATA CORRESPONDING TO EACH
00004      C     DATA SET, AND  THE DATA ARRAY, ALL FROM FILE FOR07.
00005      C
00006          PARAMETER MDATA=300,MCHAN=25
00007          DIMENSION ITITLE(5)
00008          COMMON IFREF(MCHAN)
00009          COMMON /ARRAY/ DATA(MDATA,MCHAN),IFIX(MDATA,3),IREF(MCHAN,4)
00010          COMMON /R/ FORM(MCHAN,3)
00011          COMMON /II/ ILATE,IYEAR,IDAY,KLATE,KYEAR,KDAY
00012          COMMON /U/ UNAME(MCHAN,2)
00013          COMMON /M/ MCHAN,MPCM,NAINUB
00014          DATA DASH,IBLANK/'----- ','      '/
00015      C
00016      C     CHECK TO SEE IF A NEW PAGE HEADING HAS BEEN READ IN
00017      C
00018          ISKIP = 0
00019          DO 10 I=1,5
00020       10 IF (ITITLE(I) .NE. IBLANK) ISKIP = 1
00021          IF (ISKIP .EQ. 1) GO TO 30
00022      C
00023      C     USE OLD PAGE HEADING IF A NEW ONE HAS NOT BEEN READ
00024      C
00025          READ (7,20) ITITLE
00026       20 FORMAT (5A5)
00027          GO TO 35
00028       30 SKIP RECORD 7
00029      C
00030      C     READ PARAMETERS OUTPUT BY THE PREVIOUS PROGRAM
00031      C
00032       35 READ (7,40) NCHAN,NPCM,NAINUB,(IFIX(...),...),KYEAR,
00033         1 ILATE,IDAY
00034       40 FORMAT (3(5X,I3),2(/A,I4,A,I3,A,A5))
00035      C
00036      C     INITIALIZE DEFAULT CHANNEL NUMBER IF NO VALUE WAS SPECIFIED
00037      C
00038          IF (NAINUB .EQ. 0) NAINUB = 300
00039      C
00040      C     READ IN NAMES OF CHANNELS, UNITS, AND FORMATS
00041      C
00042          READ (7,50) ((IREF(I,J),J=1,3),(UNAME(...),J=1,...),
00043         1 (FORM(I,J),J=1,3),I=1,NCHAN)
00044       50 FORMAT (5X,I3,14,2A4,5X,...A4,1X,3A4)
00045      C
00046      C     READ IN TIME ARRAY
00047      C
00048          READ (7,60) ((IFIX(I,J),J=1,3),I=1,MDATA)
00049       60 FORMAT (6(I4,I4,5X,I3,1X,I4))
00050      C
00051      C     READ IN DATA
00052      C
00053          DO 80 J=1,NCHAN
00054          SKIP RECORD 7
00055          READ (7,70) (DATA(I,J),I=1,NPCM)
00056       70 FORMAT (6(14,F12.6))

DATAIN  MERGEC.FOR      FORTRAN V4.5A(0a1) /Ka   22-Jul-81       15:16   PAGE 1-1

00057      C
00058      C     LOOK FOR E-FORMAT REPRESENTATION OF "-----" AND REPLACE WITH THE
00059      C     A-FORMAT REPRESENTATION
00060      C
00061          DO 80 I=1,NPCM
00062       80 IF (DATA(I,J) .EQ. .37DA91E+10) DATA(I,J) = DASH
00063          RETURN
00064          END
```

```
@EXCLC  MEXCLC.FOR      FORTRAN V.02(021) /A1   2[illegible]-31        15:19   PAGE 1-2


00113         DIMENSION ITITLE(5)
00114         COMMON IPRINT(MCHAN)
00115         COMMON /ASRAY/ DATA(MDATA,MCHAN),IF[illegible]A,3),IREF(MCHAN,4)
00116         COMMON /E/ FORM(MCHAN,3)
00117         COMMON /I1/ IDATE,IYEAR,IDAY,LDATE,[illegible]
00118         COMMON /B/ MCHAN,NFCN,NAINIR
00119         COMMON /U/ UMASK(MCHAN,2)
00120         COMMON /MAT/ MTCHAR(MCHAN),MICOL(MC[illegible]
00121         EXTERNAL AERVAL
00122         DATA CALC/'CALC'/
00123         OPEN (UNIT=6,DEVICE='DSK',ACCESS='[illegible]',FILE='FORO06')
00124         OPEN (UNIT=4,DEVICE='DSK',ACCESS='[illegible]',FILE='FORO04',
00125        1 ERR=110)
00126         OPEN (UNIT=7,DEVICE='DSK',ACCESS='[illegible]',FILE='FORO07',
00127        1 ERR=130)
00128   C
00129   C     LOOK FOR "CALC" SECTION OF FORO04.DAT
00130   C
00131      10 READ (4,20,END=110) WORD
00132      20 FORMAT (5A5)
00133         IF (WORD .NE. CALC) GO TO 10
00134         READ (4,30) ITITLE
00135      30 FORMAT (5A5)
00136         CALL INIT
00137         KOUNT = 0
00138         CALL DATAIN(ITITLE)
00139      40 READ (4,50,END=60) WORD
00140      50 FORMAT (A5)
00141         WRITE (6,55)
00142      55 FORMAT (//)
00143         GO TO (40,60,90,60,40) AERVAL(WORD)
00144      60 CALL DATOUT(MCHAN,ABCW,ITITLE)
00145         WRITE (6,70) KOUNT
00146      70 FORMAT (/X,I3,' FUNCTION CALLS')
00147         STOP 'PROGRAM EXECUTES SUCCESSFULLY'
00148      80 CALL ABSMAD(40)
00149         KOUNT = KOUNT + 1
00150         GO TO 40
00151      90 CALL NOISY(40)
00152         KOUNT = KOUNT + 1
00153         GO TO 40
00154     110 WRITE (6,120)
00155     120 FORMAT (X'NO PARAMETER FILE FOUND')
00156         STOP 'NO PARAMETER FILE FOUND'
00157     130 WRITE (6,140)
00158     140 FORMAT (X'NO DATA FILE FOUND')
00159         STOP 'NO DATA FILE FOUND'
00160         END

@AIN.   MEXCLC.FOR      FORTRAN V.02A(021) /A1   2[illegible]-31        15:19   PAGE 1


00001         SUBROUTINE INIT
00002   C
00003   C     SUBROUTINE INITIALIZES DATA [illegible] AND ADDRESS
00004   C     DATA ARRAY COLUMN NUMBERS FOR EACH [illegible]
00005   C
00006         PARAMETER MDATA=300,MCHAN=25
00007         COMMON /ASRAY/ DATA(MDATA,MCHAN),IF[illegible]A,3),IREF(MCHAN,4)
00008         DATA DASH/'---- '/
00009         DO 10 J=1,MCHAN
00010   C
00011   C     INITIALIZE COLUMN NUMBERS IN REFERE[illegible]
00012   C
00013         IREF(J,4) = J
00014   C
00015   C     INITIALIZE DATA ARRAY
00016   C
00017         DO 10 I=1,MDATA
00018      10 DATA(I,J) = DASH
00019         RETURN
00020         END
```

```
WARN    METCLC.FOB      FORTRAN V.5A(024) /Al   22-Sep-81       15:16   PAGE 1


00001          SUBROUTINE WARN(NEWCOL,*)
00002    C
00003    C      SUBROUTINE WARNS THE USER IF AN ADDITION OF "NEWCOL" COLUMNS WILL
00004    C      EXCEED THE AVAILABLE SPACE IN THE DATA ARRAY.
00005    C
00006          PARAMETER MDATA=300,MCHAN=25
00007          COMMON /B/ NCHAN,NSCd,MATNUM
00008          N = NCHAN * NEWCOL - NCHAN
00009          IF (N .LE. 0) RETURN
00010          WRITE (6,10) N,NCHAN
00011       10 FORMAT (1X,'MAXIMUM NUMBER OF CHANNELS WOULD BE EXCEEDED BY ',
     1  I3/10X,'CURRENT NUMBER OF CHANNELS = ',I3/
     2  10X,'FURTHER ITERATIONS CANCELLED')
00014          RETURN 1
00015          END
```

```
DATOUT  METCLC.FOB      FORTRAN V.5A(024) /Al   22-Sep-81       15:16   PAGE 1


00001          SUBROUTINE DATOUT(NCHAN,NROW,IITITLE)
00002    C
00003    C      SUBROUTINE PRINTS OUT THE RESULTS IN SPECIFIED FORMATS
00004    C
00005          PARAMETER MDATA=300,MCHAN=25
00006          DIMENSION OUT1(40),OUT2(40),LUPERR(10),...(5)
00007          COMMON IPRINT(MCHAN)
00008          COMMON /ABRAY/ DATA(MDATA,MCHAN),...(...,5),IREF(MCHAN,4)
00009          COMMON /E/POWR(MCHAN,3)
00010          COMMON /II/ IDATE,ITEAR,IDAY,LDATE,...,...
00011          COMMON /U/ UNAME(MCHAN,2)
00012          DATA C1,C2,FL1,FL2,FB1,B3,A4,DASH/'....','....',' ',' (8X',',I4,',
00013         *')   ',' 3X,',' A4,',','----'/
00014    C
00015    C      DUMP ALL DATA ONTO DISK (POSU7.DAT) FOR REFERENCE
00016    C
00017          CLOSE (UNIT=7,FILE='POSU7',DISPOSE='...')
00018          OPEN (UNIT=7,DEVICE='DSK',ACCESS='SEQOUT',FILE='POSU7')
00019          WRITE (7,10) ITITLE,NCHAN,NROW,MATNUM,...,LDATE,IDAY,
     1 LYEAR,LDATE,LDAY
00021       10 FORMAT (8595/3(X,I4),2(/A,I4,X,I3,A,A))
00022          WRITE (7,20) ((IREF(I,J),J=1,3),(UNAME(...),J=1,2),
     1 (POWR(I,J),J=1,3),I=1,NCHAN)
00023       20 FORMAT (X,I3,X,2A4,X,2A4,X,3A4)
00024          WRITE (7,30) ((IPRINT(I,J),J=1,3),I=1,...)
00025       30 FORMAT (6(1X,I4,1X,I3,1X,I4))
00026          DO 45 J=1,NCHAN
00027          WRITE (7,35) IREF(J,1)
00028       35 FORMAT (1X,'CHANNEL NUMBER ',I3)
00029          WRITE (7,40) (DATA(I,J),I=1,NROW)
00030       40 FORMAT (6(1X,F12.6))
00031       45 CONTINUE
00032    C
00033    C      LOCATE COLUMNS OF DATA ARRAY WHICH ARE TO BE PRINTED OUT; FIND
00034    C      THE CORRESPONDING CHANNEL NUMBERS, NAMES, AND DATES.
00035    C      J=OLD COLUMN NUMBER
00036    C      K=NEW COLUMN NUMBER
00037    C
00038          NPRINT = 0
00039          J = 1
00040          K = 1
00041       60 IF (J - NCHAN) 70,70,85
00042    C
00043    C      PACK INFORMATION BE CHANNELS TO BE PRINTED INTO FIRST K LOCATIONS
00044    C      OF THE REFERENCE ARRAY
00045    C
00046       70 IF (IPRINT(J) .EQ. 0) GO TO 80
00047          UNAME(K,1) = UNAME(J,1)
00048          UNAME(K,2) = UNAME(J,2)
00049          IREF(K,1) = IREF(J,1)
00050          IREF(K,2) = IREF(J,2)
00051          IREF(K,3) = IREF(J,3)
00052          IREF(K,4) = IREF(J,4)
00053          PORM(K,1) = PORM(J,1)
00054          PORM(K,2) = PORM(J,2)
00055          PORM(K,3) = PORM(J,3)
00056
```

```
DATA1  ARTICLE.FOR      FORTRAN V.5A(621) /A1   22-...-81       15:18   PAGE 1-1

00057          NPRINT = NPRINT + 1
00058          K = K + 1
00059       80 J = J + 1
00060          GO TO 60
00061  C
00062  C       SET UP TO PRINT K COLUMNS (K<=12)
00063  C
00064       85 NFIRST = 1
00065       90 LPRINT = NPRINT
00066          IF (LPRINT .GT. 12) LPRINT = 12
00067          NLAST = NFIRST + LPRINT - 1
00068  C
00069  C       WRITE PAGE TITLE
00070  C
00071          WRITE (6,100) ITITLE,(LAB(I,LEAFE,IY(I,...,...,LABEL,LBREAK
00072       100 FORMAT (1HY,53A,5A5/3HA,'DATA FOR THE PERIOD ',
00073          1 A3,X,I3,',',I0,' THROUGH ',A3,X,I3,',',...//)
00074  C
00075  C       WRITE CHANNEL NUMBERS, CHANNEL NAMES, AND DATES
00076  C
00077          WRITE (6,110) (C1,C2,I=1,LPRINT)
00078       110 FORMAT (12X,12(2X,2A4))
00079          WRITE (6,120) (IDAY(I,1),I=NFIRST,NLAST)
00080       120 FORMAT (12X,12(3A,I3,4A)/)
00081          WRITE (6,130) (IDAY(I,2),I=IDAY(I,3),...,NLAST)
00082          WRITE (6,130) (UNAME(I,1),UNAME(I,2),...,NLAST)
00083  C
00084  C       CONSTRUCT OUTPUT FORMAT ARRAY "OUT"
00085  C
00086          OUT1(1) = FL1
00087          OUT1(2) = FL2
00088          L = 3
00089          DO 140 J=NFIRST,NLAST
00090          DO 140 K=1,3
00091          OUT1(L) = FORM(J,K)
00092       140 L = L + 1
00093          OUT1(L) = FR1
00094          LMAX = L
00095  C
00096  C       WRITE YEAR, DATE (WHEN DIFFERENT FROM THE PREVIOUS VALUES),
00097  C       TIME, AND DATA
00098  C
00099          JYEAR = 0
00100          JDATE = 0
00101          DO 190 I=1,NROW
00102  C
00103  C       COPY OUTPUT FORMAT ARRAY INTO "OUT2"
00104  C
00105          DO 145 L = 1,LMAX
00106       145 OUT2(L) = OUT1(L)
00107          K = 1
00108  C
00109  C       LOAD DATA INTO BUFFER AND LOOK FOR BAD DATA (-999) ; IF FOUND
00110  C       ALTER CORRESPONDING LOCATION IN "OUT2"
00111  C
00112          DO 150 J=NFIRST,NLAST

DATOUT  ARTICLE.FOR      FORTRAN V.5A(621) /A1   22-...-81       15:18   PAGE 1-2

00113             BUFFER(K) = DATA(I,IBACK(J,4))
00114             IF (BUFFER(K) .NE. DASH) GO TO 150
00115             OUT2(3*K) = X3
00116             OUT2(3*K + 1) = A4
00117             OUT2(3*K + 2) = X3
00118       150 K = K + 1
00119             IF (ITIME(I,1) .EQ. JYEAR) GO TO 170
00120             WRITE (6,160) ITIME(I,1)
00121       160 FORMAT (1X,I4)
00122             JYEAR = ITIME(I,1)
00123       170 IF (ITIME(I,2) .EQ. JDATE) GO TO 180
00124             WRITE (6,160) ITIME(I,2)
00125       180 FORMAT (5X,I3)
00126             JDATE = ITIME(I,2)
00127       190 WRITE (6,OUT2) ITIME(I,3),(BUFFER(J),J=1,...)
00128             NPRINT = NPRINT - LPRINT
00129             NFIRST = NLAST + 1
00130             IF (NPRINT .GT. 0) GO TO 90
00131             RETURN
00132             END
```

```
KEYVAL  MTCLC.FOR      FORTRAN 9.5A(b.1) /A1  22-Jul-81       15:18   PAGE 1


00001          FUNCTION KEYVAL(WORD)
00002    C
00003    C     FUNCTION CHECKS THE INPUT WORD AND RETURNS A NUMBER AS FOLLOWS:
00004    C                 WORD           KEYVAL
00005    C                 ----           ------
00006    C                 "    "            5
00007    C                 "PLOT "           4
00008    C                 "HOIST"           3
00009    C                 "WIND "           2
00010    C                  OTHER            1
00011    C
00012          DATA WIND,WHOIST,PLOT,BLANK/'WIND ','HOIST','PLOT ','    '/
00013          KEYVAL = 1
00014          IF (WORD .EQ. WIND) KEYVAL = 2
00015          IF (WORD .EQ. WHOIST) KEYVAL = 3
00016          IF (WORD .EQ. PLOT) KEYVAL = 4
00017          IF (WORD .EQ. BLANK) KEYVAL = 5
00018          RETURN
00019          END
```

```
MARKCL  MTCLC.FOR      FORTRAN 9.5A(b.1) /A1  22-Jul-81       15:18   PAGE 1


00001          SUBROUTINE MARKCL(J,LU,UNIT,NCHAN)
00002    C
00003    C     SUBROUTINE CREATES NEW DATA RECORDS WITH STATUS FOR MAP, PLOT, AND HOIST
00004    C
00005          PARAMETER NDATA=300,NCHANS=5
00006          DIMENSION NSTRNG(2),UNIT(2)
00007          COMMON IFREE(NCHAN)
00008          COMMON /F/ FORM(NCHAN,2)
00009          COMMON /U/ UNAME(NCHAN,2)
00010          COMMON /ARRAY/ DATA(NDATA,6,NCHAN),ITEMP(NCHAN,4),IFREF(NCHAN,4)
00011          COMMON /N/ NCHAN,NEW,NAINUM
00012          COMMON /MAT/ MTCHAN(NCHAN),MTCOL(NCHAN)
00013          COMMON /F/ NAME(NCHAN,2),OUTU(NCHAN,2),FORM(NCHAN,2),MTFINT(NCHAN)
00014          DATA E,E1/'ES.2','A'/
00015    C
00016    C     INCREMENT NUMBER OF CHANNELS
00017    C
00018          NCHAN = NCHAN + 1
00019    C
00020    C     IF DESIRED CHANNEL NUMBER IS UNASSIGNED, ASSIGN NEXT SEQU ENTRY
00021    C     CHANNEL NUMBER AND INCREMENT NAINUM
00022    C
00023          IF (MTCHAN(J) .NE. 0) GO TO 10
00024          MTCHAN(J) = NAINUM
00025          NAINUM = NAINUM + 1
00026    C
00027    C     UPDATE IFEF WITH NEW CHANNEL NUMBER (UNUSED OUTSIDE WERE SET AT
00028    C     INITIALIZATION)
00029    C
00030       10 MTCOL(J) = NCHAN
00031          IFEF(NCHAN,1) = MTCHAN(J)
00032    C
00033    C     ASSIGN CHANNEL NAME (DEFAULT NAME IS "space" OR WHERE "FORM;" IS THE
00034    C     NAME OF THE REQUEST TO FUNCTION AND "AM" IS A SUBJECT NUMBER)
00035    C
00036          ENCODE(10,20,NSTRNG) ID,NOUN
00037       20 FORMAT (A5,1X,I2,X)
00038          IFEF(NCHAN,3) = NSTRNG(1)
00039          IFEF(NCHAN,4) = NSTRNG(2)
00040    C
00041    C     ASSIGN DEFAULT VALUES FOR OUTPUT UNLESS ADD FORMAT, SET IFFINT=0
00042    C
00043          UNAME(NCHAN,1) = UNIT(1)
00044          UNAME(NCHAN,2) = UNIT(2)
00045          FORM(NCHAN,1) = E1
00046          FORM(NCHAN,2) = E
00047          FORM(NCHAN,3) = E1
00048          IFFINT(NCHAN) = 0
00049    C
00050    C     OVER RIDE DEFAULT VALUES WITH SPECIFIED VALUES, IF ANY
00051    C
00052          CALL SETUP(J,MTCOL(J))
00053          RETURN
00054          END
```

```
PRECOL  RETCLC_POS      /CONTRAN V.3A(0.1) /AT  22-DEC-81        PAGE 1

00001          SUBROUTINE PRECOL(NPRINT,NLAST,*)
00002    C
00003    C     SUBROUTINE LOCATE 3 COLUMNS FOR SPECIFIED CHANNEL NUMBERS AND
00004    C     PRINTS ERROR MESSAGE IF ANY ARE OUT RANGE
00005    C
00006          PARAMETER MDATA=300,MCHAN=25
00007          COMMON /ARRAY/ DATA(MDATA,MCHAN),XXXXX(MDATA,3),IPREF(MCHAN,4)
00008          COMMON /B/ NCHAN,NSEC,NCHOU
00009          COMMON /MAT/ MTCOL(MCHAN),NTCOL(MCHAN)
00010          IERROR = 0
00011          DO 40 J=NPRINT,NLAST
00012          N = J
00013          MTCOL(J) = 0
00014    C
00015    C     SEARCH FOR REQUESTED CHANNEL NUMBER IN TABLE AND SAVE CORRESPONDING
00016    C     COLUMN NUMBER
00017    C
00018          DO 10 I=1,NCHAN
00019       10 IF (IREF(I,1) .EQ. MTCHAN(J)) MTCOL(J) = IREF(I,4)
00020    C
00021    C     IF NO CHANNEL WAS FOUND, SET ERROR FLAG AND WRITE ERROR MESSAGE
00022    C
00023          IF (MTCOL(J) .NE. 0) GO TO 30
00024          IERROR = 1
00025          WRITE (6,20) MTCHAN(J)
00026       20 FORMAT (1X,'REQUESTED CHANNEL (',I3,') DOES NOT EXIST.'/
00027         1 1X'ITERATION ABORTED.')
00028          GO TO 40
00029    C
00030    C     OVER RIDE DEFAULTS (PREVIOUS NAME, UNITS, AND PRINT SWITCH)
00031    C     IF NEW VALUES ARE SPECIFIED
00032    C
00033       30 CALL SETUF(N,MTCOL(N))
00034       40 CONTINUE
00035    C
00036    C     IF CHANNEL WAS MISSING, ABORT CURRENT EXECUTION AND PROCEED TO THE NEXT
00037    C
00038          IF (IERROR .EQ. 1) RETURN 1
00039          RETURN
00040          END
```

```
SETUP   RETCLC_POS      /CONTRAN V.3A(0.1) /AT  22-DEC-81        PAGE 1

00001          SUBROUTINE SETUP(I,ICOL)
00002    C
00003    C      SUBROUTINE SETS VALUES FOR CHANNEL NAME,XXXXXX UNITS, OUTPUT FORMAT,
00004    C      AND PRINT SWITCH
00005    C
00006          PARAMETER MDATA=300,MCHAN=25
00007          COMMON IPRINT(MCHAN)
00008          COMMON /ARRAY/ DATA(MDATA,MCHAN),XXXXX(MDATA,3),IPREF(MCHAN,4)
00009          COMMON /U/ UNAME(MCHAN,2)
00010          COMMON /F/ FORM(MCHAN,3)
00011          COMMON /B/ NAME(MCHAN,2),OUTU(MCHAN,2),XXXX(MCHAN,3),NPRINT(MCHAN)
00012          DATA IBLANK,BLANK/'    ','    '/
00013    C
00014    C     REPLACE DEFAULT NAME
00015    C
00016          IF ((NAME(I,1) .EQ. IBLANK) .AND. (XXXXX(I,2) .EQ. IBLANK))
00017         1 GO TO 10
00018          IREF(ICOL,2) = NAME(I,1)
00019          IREF(ICOL,3) = NAME(I,2)
00020    C
00021    C     REPLACE DEFAULT OUTPUT UNITS
00022    C
00023       10 IF ((OUTU(I,1) .EQ. BLANK) .AND. (XXXXX(I,2) .EQ. BLANK)) GO TO 20
00024          UNAME(ICOL,1) = OUTU(I,1)
00025          UNAME(ICOL,2) = OUTU(I,2)
00026    C
00027    C     REPLACE DEFAULT OUTPUT FORMAT
00028    C
00029       20 IF ((OUTF(I,1) .EQ. BLANK) .AND. (XXXXX(I,2) .EQ. BLANK) .AND.
00030         1 (OUTF(I,3) .EQ. BLANK)) GO TO 30
00031          FORM(ICOL,1) = OUTF(I,1)
00032          FORM(ICOL,2) = OUTF(I,2)
00033          FORM(ICOL,3) = OUTF(I,3)
00034    C
00035    C     SET PRINT SWITCH
00036    C
00037       30 IPRINT(ICOL) = NPRINT(I)
00038          RETURN
00039          END
```

```
ABSWND  METRIC.PCF      FORTRAN V..A(0&1) /AI   ..-....-..      15:5*   PAGE 1

00001         SUBROUTINE ABSWND(*)
00002   C
00003   C     TO ABSOLUTE WIND VELOCITY CALCULATIONS
00004   C
00005         PARAMETER (NDATA=300,NCHAN=2)
00006         DIMENSION UNIT#1(2),UNIT#2(2)
00007         COMMON /DATA/ DATA(NDATA,NCHAN),ITCOL(NCHAN)
00008         COMMON /R/ NAM_(NCHAN,2),UNIT#(NCHAN,2),FACT(NCHAN,3),PIPARM(NCHAN)
00009         DATA ID,UNIT#1,UNIT#2,UNIT#/'WIND ',' KTS','KTS ',' DEG',
00010        1 'FEES','_____'/
00011   C
00012   C     #FILE READING FOR USER'S INPUT
00013   C
00014         WRITE (6,10) (UNIT#(...),...=1,2)
00015      10 FORMAT (60X,'#ESS  UNIT#(1,A1'/1X,A5,'___#(1,A1'/1X,A5,'____',/,
00016        1 3X,2A5,'___UNIT#(2,A1'/1X,2A5/4X,'#EL#A...',4X'AN #ND #ED ',
00017        2 8X'#EL #N)#IS ','#X'#D# UNI# #ED ',8X'#D#'## ### ### ',8X,
00018        3 'ABS WNE ',3X,'#E# #ND #A#'/2X'#ANGLE',2(3X'#',8X,'#ANG',
00019        4 8X)//)
00020   C
00021   C     #ECIEV ROOM TO EXPAND DATA ARRAY ADDS ##### IF SPACE NOT AVAILABLE
00022   C
00023         ITEM = 0
00024      20 CALL #APB(2,67J)
00025         ITEP = ITEM + 1
00026   C
00027   C     READ FUNCTION PARAMETERS
00028   C
00029         READ (5,30,END=70) (#CHAN(J),(#ANG...,#,#,J=1,...),(OUTU(I,J),
00030        1 J=1,2),(OUTP(I,J),J=*,3),#ESIN(I),...=*,#)
00031      30 FORMAT (8,I3,F,2A4,X,2A4,A,3A4,A,I1)
00032   C
00033   C     LOCATE INPUT DATA
00034   C
00035         CALL FRECO.(7,*,640)
00036   C
00037   C     PARAM# DATA #R#AY; UP#AIE ##EF, NCHAN, AND #AIAM
00038   C
00039         CALL MARCOL(5,ID,UNIT#1,ITEP)
00040         CALL MARCOL(6,ID,UNIT#2,,1EP)
00041   C
00042   C     DO CALCULATIONS
00043   C
00044         CALL MMJCIC(ITEP)
00045   C
00046   C     LOOK FOR NEXT FUNCTION
00047   C
00048      40 READ (8,50,END=60) WORD
00049      50 FORMAT (1X,A5)
00050         IF (KEYVAL(WORD) .EQ. 5) GO TO 40
00051         BACKSPACE 4
00052         IF (KEYVAL(WORD) .EQ. 1) GO TO 20
00053         RETURN
00054      60 BACKSPACE 4
00055         RETURN
00056      70 RETURN 1
00057         END
```

```
WABSIC  METRIC.PCF      FORTRAN V..A(0&1) /AI   ..-....-..      15:5*   PAGE 1

00001         SUBROUTINE WABSIC(NCHAN)
00002   C
00003   C     SUBROUTINE CALCULATES ABSOLUTE WIND #####.## FROM SHIP VELOCITY
00004   C     AND TRE RELATIVE WIND VELOCITY.  ######## WIND DIRECTION IS
00005   C     RECORDED AS THE DIRECTION FROM which the wind comes, BUT
00006   C     SHIP DIRECTION IS RECORDED AS THE DIRECTION IN which THE SHIP
00007   C     IS #EADING.
00008   C
00009         PARAMETER (NDATA=300,NCHAN=2)
00010         COMMON /ARRAY/ DATA(NDATA,NCHAN),ITCOL(NCHAN),IFREF(NCHAN,4)
00011         COMMON /N/ NCHAN,NPOW,MAINUN
00012         COMMON /WAC/ WICHAP(NCHAN),WICOL(NCHAN)
00013         DATA PI/3.141594/
00014         DO 40 I=1,NPOW
00015         J = 1
00016   C
00017   C     CHECK FOR INVALID DATA, SET OUTPUTS TO ****** IF FOUND
00018   C
00019         CALL TOTAL(IFREF,J,4)
00020         IF (IFREF .GT. 0) WE TO 20
00021         SPEED = DATA
00022         ANGLE = DATA
00023         GO TO 30
00024   C
00025   C     CALCULATE ABSOLUTE ANGLES IN RADIANS
00026   C
00027      20 SHANG = DATA(I,WICOL(4)) * PI/180.
00028         #ELANG = DATA(I,WICOL(2)) * PI/180.
00029         WNDANG = SHPANG + #ELANG + PI
00030   C
00031   C     CALCULATE COMPONENTS OF ABSOLUTE WIND #######
00032   C
00033         XVEL = DATA(I,WICOL(3))*SIN(SHPANG)+DATA(I,WICOL(1))*SIN(WNDANG)
00034         YVEL = DATA(I,WICOL(3))*COS(SHPANG)+DATA(I,WICOL(1))*COS(WNDANG)
00035   C
00036   C     CONVERT TO SPEED AND DIRECTION
00037   C
00038         SPEED = SQRT(XVEL*XVEL + YVEL*YVEL)
00039         ANGLE = (ATAN2(YVEL,XVEL) * 180./PI)  + 360.
00040         IF (ANGLE .LT. 0.) ANGLE = ANGLE + 360.
00041   C
00042   C     STORE RESULTS IN DATA ARRAY
00043   C
00044      30 DATA(I,WICOL(5)) = SPEED
00045      40 DATA(I,WICOL(6)) = ANGLE
00046   C
00047   C     OUTPUT LIST OF CHANNELS USED FOR THIS CALCULATION
00048   C
00049         WRITE (6,50) NOUNT,(ISEP(WICOL(I),I=1,4),
00050        1 (IREF(WICOL(2),I),I=1,3),(IREF(WICOL(1),I),I=1,3),
00051        2 (IREF(WICOL(4),I),I=1,3),(IREF(WICOL(3),I),I=1,3),
00052        3 (IREF(WICOL(6),I),I=1,3)
00053      50 FORMAT (4X,12,2X,6(4A,I3,A,2A4,4A))
00054         RETURN
00055         END
```

```
ISTEAT  M:\CALC.FOR      FORTRAN V.5A(021) /A1   22-Dec-84       15:36   PAGE 1


00001         SUBROUTINE ISTEAT(IP,P,I,J)
00002   C
00003   C     SUBROUTINE TESTS FOR BAD DATA (-999) AND SETS P(I,J) IF
00004   C     IT IS DETECTED
00005   C
00006         PARAMETER MDATA=300,MCHAR=25
00007         COMMON /BRAY/ DATA(MDATA,MCHAR),IP(MDATA,3),P(MDATA,4)
00008         COMMON /BA1/ MCHAR(MCHAR),BLOCK(MCHAR)
00009         DATA DASH/'-----'/
00010         IERROR = 0
00011         DO 10 K=1,M
00012      10 IF (DATA(I,MCHAR(J)) .EQ. DASH) RETURN = 1
00013         RETURN
00014         END
```

```
MOIST   M:\CALC.FOR      FORTRAN V.5A(021) /A1   22-Dec-84       15:36   PAGE 1


00001         SUBROUTINE MOIST(*)
00002   C
00003   C     DO MOISTURE LOADING CALCULATIONS
00004   C
00005         PARAMETER MDATA=300,MCHAR=25
00006         DIMENSION UNITS(2)
00007         COMMON /BA1/ MCHAR(MCHAR),BLOCK(MCHAR)
00008         COMMON /B/ NAME(MCHAR,4),OUTU(MCHAR,2),OUTF(MCHAR,3),REPINT(MCHAR)
00009         DATA ID,BLANK,UNITS,BLINE2/'MOIST',' ',' ','  ','  ','_____'/
00010   C
00011   C     WRITE HEADING FOR OUTPUT LISTING
00012   C
00013         WRITE (6,10) (ULINE,I=1,3)
00014      10 FORMAT(50X,'WATER VAPOR DIFFUSION'/I2A,A4,3A,'___INPUT___',4A5,
00015        1 2A4,'___OUTPUT___'/4X,'ITERATION',2A,'___MOISTURE___',4X,'AIR PRPS',
00016        2 'SURF',8X,'REL HUMIDITY',2A4,'WATER VAPOR '/2A,'NUMBER',3(5X,
00017        3 'P',4A,'NAME',4X),25X'P',4A'NAME'//)
00018   C
00019   C     VERIFY ROOM TO EXPAND DATA ARRAY. ADJUST FUNCTION IF SPACE NOT AVAILABLE
00020   C
00021         ITER = 0
00022      20 CALL WARN(1,&70)
00023         ITER = ITER + 1
00024   C
00025   C     READ FUNCTION PARAMETERS
00026   C
00027         READ (4,30,END=70) (MCHAR(I),(NAME(J,I),J=1,4),(OUTU(I,J),J=1,2),
00028        1 (OUTF(I,J),J=1,3),REPINT(I),I=1,M)
00029      30 FORMAT(1X,I3,1X,2A4,1X,2A4,1X,3A4,1X,I5)
00030   C
00031   C     LOCATE INPUT DATA
00032   C
00033         CALL FNDCOL(1,3,&40)
00034   C
00035   C     EXPAND DATA ARRAY; UPDATE LOCK, BLOCK, VARIABLE
00036   C
00037         CALL WARCOL(4,ID,UNITS,ITER)
00038   C
00039   C     DO CALCULATIONS
00040   C
00041         CALL VAPCIC(ITER)
00042   C
00043   C     LOOK FOR NEXT FUNCTION
00044   C
00045      40 READ (4,50,END=60) WORD
00046      50 FORMAT (1A,A5)
00047         IF (KEYVAL(WORD) .EQ. 3) GO TO 40
00048         BACKSPACE 4
00049         IF (KEYVAL(WORD) .EQ. 1) GO TO 20
00050         RETURN
00051      60 BACKSPACE 4
00052         RETURN
00053      70 RETURN 1
00054         END
```

```
VAPCLC  WETCLC.FOR      FORTRAN V.5A(625) /A1   24-Dec-81       15:18   PAGE 1

00001          SUBROUTINE VAPCLC(ROUND)
00002    C
00003    C     SUBROUTINE CALCULATES WATER VAPOR PRESSURE IN THE
00004    C     ATMOSPHERE
00005    C
00006          PARAMETER NDATA=300,NCHAN=25
00007          COMMON /ARRAY/ DATA(NDATA,NCHAN),ITIME(NDATA,3),IREF(NCHAN,4)
00008          COMMON /B/ NCHAN,NPOW,NAMOUN
00009          COMMON /BAT/ BCHAR(NCHAN),NICOL(NCHAN)
00010          DATA DASH/'-----'/
00011          FACTOR = 25./19.
00012          DO 30 I=1,NPOW
00013          K = I
00014    C
00015    C     CHECK FOR BAD DATA; SET OUTPUT TO "-----" IF FOUND
00016    C
00017          CALL TSTDAT(IREF(NOR,K,3)
00018          IF (IREF(K .EQ. 0) GO TO 10
00019          VAPOR = DASH
00020          GO TO 20
00021    C
00022    C     CONVERT TEMP TO KELVINS, PRESS TO MILLIBARS
00023    C
00024       10 TEMP = DATA(I,NICOL(1)) + 273.15
00025          PRESS = DATA(I,NICOL(2))*FACTOR
00026    C
00027    C     CALCULATE WATER VAPOR CONCENTRATION
00028    C
00029          T = 1.0 - 373.15/TEMP
00030          EXPON = (((-0.1299*T)-0.6445)*T-1.9760)*T+11.3165)*T
00031          SAT = 1013.25*EXP(EXPON)
00032          VAPOR = 10000.*(SAT/PRESS)*DATA(I,NICOL(3))
00033    C
00034    C     STORE RESULTS IN DATA ARRAY
00035    C
00036       20 DATA(I,NICOL(4)) = VAPOR
00037       30 CONTINUE
00038    C
00039    C     OUTPUT LIST OF CHANNELS USED IN THIS CALCULATION
00040    C
00041          WRITE(6,40) NCHAN,(IREF(NICOL(1),4),I=1,3),(IREF(NICOL(2),I),
00042         1 I=1,3), (IREF(NICOL(3),I),I=1,3),(IREF(NICOL(4),I),I=1,3)
00043       40 FORMAT (4X,I2,2X,3(4A,I3,A,2A4,4A),4A,3(A,2A4))
00044          RETURN
00045          END
```

Appendix C
Listing of Program METPLT
(Version 3.2)

```
C*****
C
C     MAIN PROGRAM
C
C     METPLT VERSION 3.00  (JULY 1990)
C
C     JOHN HOOVER
C     CODE 4230
C     U.S. NAVAL OCEANOGRAPHIC LABORATORY
C     NSTL, MS  39529
C
C     PROGRAM INPUTS AN ARRAY OF DATA (IN ONE OF SEVERAL FILE FORMATS)
C     AND PLOTS ONE OR MORE GRAPHS OF THE DATA VERSUS TIME.
C
C     DECLARE PARAMETERS:
C
      PARAMETER (NDATA=MAXIMUM POSSIBLE NUMBER OF DATA SETS (ONE STATION, ONE DATE=100)
     &          ,NCHAN=MAXIMUM POSSIBLE NUMBER OF DATA CHANNELS PER SET (ONE STATION,
     &                 ONE DAY=25)
C
C     INPUTS AND FORMATS:
C
C     CONTROL PARAMETERS AND DATA READ FROM VARIOUS DATA FILES WHICH PARAMETERIZE
C     THE FILE PCRU4.DAT, WHICH WAS CREATED BY PROGRAM METDIC.  DATA IS READ
C     FROM FILE PCRU7.DAT, WHICH IS THE DATA DATA FILE PROGRAMS METSET AND
C     METCLC.
C
C     PCRU7.DAT:
C
C        NCHAN=NUMBER OF CHANNELS READ (MAXIMUM OF 9 COUNTING IN THE DATA ARRAY)
C        NROW=NUMBER OF ROWS IN THE DATA ARRAY
C        METNUM=METAT NUMBER AVAILABLE TO ASSOCIATE TO A NEW CHANNEL
C        IYEAR=INITIAL YEAR IN DATA SET
C        IDATE=INITIAL DATE IN DATA SET
C        IDAY=INITIAL DAY IN DATA SET
C        LYEAR,LDATE,LDAY REFER TO THE LAST YEAR,DATE, AND DAY IN THE DATA SET
C             FORMAT = 3(I3),2(/A4,A3,A4)
C
C        OTITLE=OLD PAGE HEADING TO BE USED FOR LABEL.
C             FORMAT = A24A1
C
C
C
C        IREF=ARRAY WHICH CROSS REFERENCES VARIABLE NUMBER, NAME, AND
C             COLUMN IN DATA ARRAY
C        UNAME=ARRAY CONTAINING THE NAMES OF THE DIFFERENT UNITS
C        FORM=FORMATS USED FOR SUBSET OF DATA ARRAYS
C             FORMAT = 4I3,2A4,A4,A1A4
C        IFIME=ARRAY CONTAINING THE YEAR, BEGIN DATE, AND HOUR FOR EACH SET OF
C              DATA.  EACH ROW CONTAINS ID FOR DATA PERIOD.
C             FORMAT = 6(A2B,A13,A14)
C        DATA=ARRAY OF DATA VALUES.  EACH COLUMN CONTAINS TO ONE INPUT CHANNEL,
C             EACH ROW CONTAINS DATA FOR ONE DATA PERIOD (WHICH IS DEFINED IN
C             THE CORRESPONDING ROW OF "IFIME"). DATA IS READ COLUMN BY COLUMN.
C             FORMAT = 6(E,F12.6)
C
C     PCRU4.DAT:
C
```



```
C        TITLEL=NEW PAGE HEADING TO BE USED FOR LABEL
C             FORMAT = A24A1
C        (NOTE: IF 'TITLEL' IS NON-BLANK, IT SUPERSEDES 'OTITLE')
C
C        NDAYS=NUMBER OF DAYS TO BE PLOTTED ON EACH GRAPH
C        IPYEAR=INITIAL YEAR TO BE PLOTTED
C        IPDATE=INITIAL DATE TO BE PLOTTED
C        IPDAY=INITIAL DAY OF WEEK TO BE PLOTTED
C        IPYEAR,IPDATE,IPDAY REFER TO THE SAME YEAR,DATE & DAY TO BE PLOTTED
C             FORMAT = I12,2(/A14,A13,A14)
C
C        NUMX=CHANNEL NUMBER FOR X-AXIS DATA
C        NAMEX=X-AXIS NAME
C        UNITX=X-AXIS UNITS
C        XMIN=MINIMUM VALUES FOR SCALING X-AXIS
C        XINC=DESIRED X-AXIS INCREMENTS
C        XMAX=MAXIMUM VALUES FOR SCALING X-AXIS
C        THLDX=X-THRESHOLD VALUE
C        DPLX=X-AXIS HYSTERESIS PARAMETER (IF = 0, THIS PARAMETER IS NOT TESTED)
C        TYPEX=X-AXIS TYPE SPECIFICATION
C        THERE ARE SIMILAR PARAMETERS FOR THE Y- AND Y2-AXES (Y=MAGNITUDE;
C        FOR VECTOR PLOTS, Y2=DIRECTION)
C             FORMAT = I13,X1A4,A2A4,5(A5,A4)
C
C*****
C                                  INPUT MESSAGES
C                                  ----- --------
C
C     <Y-YEAR,D=DATE,W=DAY OF WEEK,T=TIME,A=VARIABLE(X,Y,OR Y2)>
C          <P=PARAMETER,A=DATA VALUE,_____=ERROR CHARACTERS>
C
C     MESSAGE                    STATUS      MEANING
C     -------                    ------      -------
C     NO PARAMETER FILE FOUND    STOPS       FILE PCRU4.DAT WAS NOT FOUND
C                                            OR NO SECTION LABELED *PLOTM
C                                            COULD BE FOUND.
C
C     NO DATA FILE FOUND         STOPS       FILE PCRU7.DAT WAS NOT FOUND
C
C     INITIAL YEAR REQUESTED IS NOT   STOPS  IYEAR WAS < IYEAR
C     FOUND. FIRST DATE WILL BE = DAY
C     D,Y
C
C     INITIAL DATE REQUESTED IS NOT   STOPS  IPDATE WAS < IDATE
C     FOUND. FIRST DATE WILL BE = DAY
C     D,Y
C
C     FINAL YEAR REQUESTED IS NOT     STOPS  LPYEAR WAS > LYEAR
C     FOUND. LAST DATE WILL BE = DAY
C     D,Y
C
C     FINAL DATE REQUESTED IS NOT     STOPS  LPDATE WAS > LDATE
C     FOUND. LAST DATE WILL BE = DAY
C     D,Y
C
C     W IS NOT FOUND IN DAY NAME LIST  STOPS   W IS NOT ONE OF THE SEVEN
```

```
00113   C                                                  ...AL NAMES ("SUN ","MON ",
00114   C                                                  "TUE ","WED ","THU ","FRI ",
00115   C                                                  "SAT ").
00116   C
00117   C     "_____" IS NOT A VALID X-AXIS      ******    CURRENTLY, ONLY A LINEAR X-
00118   C     TYPE.  PLOT IS SKIPPED                       AXIS MAY BE SPECIFIED ("TYP
00119   C                                                  ...LIN).
00120   C
00121   C     "_____" IS NOT A VALID Y-AXIS      ******    LEGAL Y-AXIS TYPES ARE:
00122   C     TYPE.  PLOT IS SKIPPED                       LINEAR (TYPE = LIN)
00123   C                                                  LOGIOF (TYPE = LOG10 OR AVPC)
00124   C
00125   C     "_____" AND "_____" ARE NOT        ******    FOR VECTOR AXES, ONE SPECIF-
00126   C     COMPATIBLE TYPES.  PLOT SKIPPED.             ICATION MUST BE "AVPC" AND
00127   C                                                  THE OTHER "AVPC" (MAGNITUDE
00128   C                                                  AND DIRECTION, RESPECTIVELY)
00129   C
00130   C     INITIAL JULIAN DATE (M,D) IS       ******    DATE ON WHICH PLOT SET IS TO
00131   C     NOT FOUND                                    BEGIN WAS NOT FOUND IN ARRAY
00132   C     PLOT SET IS SKIPPED.
00133   C                                                  "TITLE"
00134   C
00135   C     FINAL JULIAN DATE (M,D) IS         ******    DATE ON WHICH PLOT SET IS TO
00136   C     NOT FOUND                                    END WAS NOT FOUND IN "TITLE"
00137   C     PLOT SET IS SKIPPED.
00138   C
00139   C     NUM WAS NOT FOUND IN "PLOT"--      ******    THE CHANNEL SPECIFIED FOR
00140   C     PLOT NUMBER N IS SKIPPED                     THE AXIS WAS NOT FOUND.
00141   C                                                  "NAME" (MAY BE "NUMX","NUMY",
00142   C                                                  OR "NUMY2") TELLS WHICH AXIS
00143   C
00144   C     Y,D,T: DATA IS INVALID             ******    A DATA VALUE OF "____" WAS
00145   C                                                  FOUND.  POINT NOT PLOTTED.
00146   C
00147   C     Y,D,T: A-K IS < THRESHOLD          ******    A DATA VALUE BELOW THRESHOLD
00148   C                                                  WAS FOUND.  POINT NOT PLOTTED
00149   C
00150   C     Y,D,T: POINT SUPPRESSED --         ******    DATA VALUE DOES NOT MEET
00151   C     A-K IS WITHIN P OF PREVIOUS                  THE HYSTERESIS CRITERION.
00152   C     VALUE                                        POINT WILL STILL BE PLOTTED
00153   C                                                  UNLESS ALL TESTS FAIL.
00154   C
00155   C     NOTE: OTHER ERROR MESSAGES MAY BE ...ATED BY THE "DISSPLA" PACKAGE.
00156   C
00157   C     ***************************************************************
00158   C
00159         PARAMETER MDATA=300,MCHAN=25
00160         DIMENSION ITITLE(5),TITLPL(25)
00161         REAL NAMEX,NAMEY,NAMEY2
00162         COMMON NUMX(MCHAN),NAMEX(2,MCHAN),UNITX(2,MCHAN),XMIN(MCHAN),
00163        1 XINC(MCHAN),XMAX(MCHAN),THLDX(MCHAN),SCALX(MCHAN),TYPEX(MCHAN),
00164        2 NUMY(MCHAN),NAMEY(2,MCHAN),UNITY(2,MCHAN),YMIN(MCHAN),
00165        3 YINC(MCHAN),YMAX(MCHAN),THLDY(MCHAN),SCALY(MCHAN),TYPEY(MCHAN),
00166        4 NUMY2(MCHAN),NAMEY2(2,MCHAN),UNITY2(2,MCHAN),YMIN2(MCHAN),
00167        5 YINC2(MCHAN),YMAX2(MCHAN),THLDY2(MCHAN),SCALY2(MCHAN),
00168        6 TYPEY2(MCHAN)
```



```
00169         COMMON /ARRAY/ DATA(MDATA,MCHAN),IF...(MDATA,3),IREF(MCHAN,4)
00170         COMMON /DAY/ WANDAY(7),NUMDAY,DAYS..
00171         COMMON /REAL/ FORM(MCHAN,3),UNAME(5...,4)
00172         COMMON /IL/ IDATE,IYEAR,IDAY,LDATE,...AR,...AY
00173         COMMON /M/ MCHAN,MROW,MXTNUM,MDAYS
00174         COMMON /P/ IPDATE,IPYEAR,IPDAY,LPLAT...,LPDAY
00175         COMMON /JK/ JPDATE,JPYEAR,JPDAY,KPDA...,KPDAY
00176         COMMON /INTPLT/ IPLOT,LPLOT,NPLOT
00177         COMMON /RC/ IROW,KROW,LROW,ICOLK,IC...
00178         COMMON /ALPHA/ XSTRNG(9),YSTRNG(9)
00179         COMMON /PCINT/ XARRAY(MDATA),YMAG(...,MDATA)
00180         COMMON /X/ X00,X12,X24,XDATE
00181         COMMON /SIZE/ XSIZE,YSIZE
00182         DATA BLANK,DOLLAR/'  ','$'/
00183         DATA WANDAY /'SUN','MON','TUE','WED','THU','FRI','SAT'/
00184         DATA DAYSIZ,HSIZE,YSIZE,XGAP,DASH,P...
00185        1 / 1.375,9.625,3.0,0.2,'----','PLOT'/
00186         OPEN (UNIT=6,DEVICE='DSK',ACCESS='S...',FILE='FOR006')
00187         OPEN (UNIT=4,DEVICE='DSK',ACCESS='S...',FILE='FORD4',FPP=330)
00188         OPEN (UNIT=7,DEVICE='DSK',ACCESS='S...',FILE='FOR007',FPP=350)
00189   C
00190   C     LOOK FOR "PLOT" SECTION OF FORDN.DA.
00191   C
00192        10 READ (4,20,END=330) WORD
00193        20 FORMAT (1A5)
00194           IF (WORD .NE. PLOT) GO TO 10
00195   C
00196   C     INITIALIZE ARRAY "TITLPL"
00197   C
00198           DO 25 I=1,25
00199           TITLPL(I) = BLANK
00200        25 CONTINUE
00201           READ (4,30) (TITLPL(I),I=1,24)
00202        30 FORMAT (24A1)
00203   C
00204   C     READ FOR07.DAT DATA
00205   C
00206           CALL DATAIN(TITLPL)
00207   C
00208   C     ENCODE TITLE STRING
00209   C
00210           N = 25
00211        35 N = N - 1
00212           IF (N .EQ. 0) GO TO 40
00213           IF (TITLPL(N) .EQ. BLANK) GO TO 35
00214        40 ENCODE (N+1,45,ITITLE) (TITLPL(I),I=1,...,...)
00215        45 FORMAT (25A1)
00216   C
00217   C     READ DATES TO BE PLOTTED AND NUMBER OF DATA/PLOT
00218   C
00219           READ (4,50) NDAYS,IPYEAR,IPDATE,IPDA...,...DATE,IEDAY
00220        50 FORMAT(XI2/XI4,XI3,XA4/XI4,XI3,XA4)
00221   C
00222   C     CONFIRM PRESENCE OF DATA FOR THE REQUESTED DATES
00223   C
00224   C     CHECK INITIAL YEAR
```

```
NETPLT  NETPLT.FOR      FORTRAN V.5A(021) /Ax  [illegible]        d:22    PAGE 1-4


00225   C
00226         IF (IPYEAR .GE. IYEAR) GO TO 80
00227         WRITE (6,55) IDAY,IDATE,IYEAR
00228      55 FORMAT (X'INITIAL YEAR REQUESTED IS BEFORE FIRST. FIRST DATE WILL ',
00229        1 'BE ',A4,' DAY ',I3,',',I4)
00230         IPYEAR = IYEAR
00231         IPDATE = IDATE
00232         IPDAY = IDAY
00233         GO TO 80
00234   C
00235   C     CHECK INITIAL DATE
00236   C
00237      60 IF (IPDATE .GE. IDATE) GO TO 80
00238         WRITE (6,70) IDAY,IDATE,IPYEAR
00239      70 FORMAT (X'INITIAL DATE REQUESTED IS BEFORE FIRST. FIRST DATE WILL ',
00240        1 'BE ',A4,' DAY ',I3,',',I4)
00241         IPDATE = IDATE
00242         IPDAY = IDAY
00243   C
00244   C     CHECK FINAL YEAR
00245   C
00246      80 IF (LPYEAR .LE. LYEAR) GO TO 100
00247         WRITE (6,90) LDAY,LDATE,LYEAR
00248      90 FORMAT (X'FINAL YEAR REQUESTED IS BEYOND. LAST DATE WILL BE ',
00249        1 A4,' DAY ',I3,',',I4)
00250         LPYEAR = LYEAR
00251         LPDATE = LDATE
00252         LPDAY = LDAY
00253         GO TO 120
00254   C
00255   C     CHECK FINAL DATE
00256   C
00257     100 IF (LPDATE .LE. LDATE) GO TO 120
00258         WRITE (6,110) LDAY,LDATE,LYEAR
00259     110 FORMAT (X'FINAL DATE REQUESTED IS BEYOND. LAST DATE WILL BE ',
00260        1 A4,' DAY ',I3,',',I4)
00261         LPDATE = LDATE
00262         LPDAY = LDAY
00263   C
00264   C     INITIALIZE SYSTEM
00265   C
00266     120 CALL INIT (6370)
00267   C
00268   C     LOOK FOR "XAXIS" AND READ PARAMETERS
00269   C
00270     130 READ (5,140,END=180) WORD
00271     140 FORMAT (2A5)
00272         IF (KEYVAL(WORD) .NE. 2) GO TO 150
00273     150 CALL XAXIN(6130)
00274   C
00275   C     LOOK FOR "YAXIS" AND READ PARAMETERS
00276   C
00277     160 READ (5,140,END=180) WORD
00278         GO TO (160,150,170,160) KEYVAL(WORD)
00279     170 CALL YAXIN (6160)
00280     180 NPLOT = NPLOT - 1


NETPLT  NETPLT.FOR      FORTRAN V.5A(021) /Ax  [illegible]        d:22    PAGE 1-5

00281         IF (NPLOT .LE. 0) GO TO 370
00282   C
00283   C     SET LENGTH OF X-AXIS
00284   C
00285         XSIZE = NDAYS * DAYSIZ
00286         IF (XSIZE .GT. MSIZE) XSIZE = MSIZE
00287   C
00288   C     INITIALIZE PLOTTER AND PLOTTER SOFTWARE
00289   C
00290         CALL CALCMP
00291         CALL INCH30
00292         CALL PAGE(11.0,11.0)
00293         CALL MARKER(5)
00294         CALL SCLPIC(0.4)
00295         CALL INTAXS
00296         CALL YAXANG(0.0)
00297         CALL NOCHEK
00298         CALL XTICKS(4)
00299         CALL HEIGHT(0.10)
00300         X00 = -XINT(0) /2.0
00301         X12 = (DAYSIZ - XINT(1200))/2.0
00302         X24 = XSIZE - XINT(2400)/2.0
00303         CALL RESET('HEIGHT')
00304         XDATE = X12 + 1.0
00305   C
00306   C     SET UP LIMITING DATES AND FOR NUMBERS FOR THE CURRENT PLOT SET.
00307   C
00308         NSET = 0
00309     190 NSET = NSET + 1
00310         CALL SETUP(NSET,TITLPL,6320)
00311         KPLOT = NPLOT
00312   C
00313   C     DO PLOTS IN SETS OF THREE
00314   C
00315         IPLOT = 0
00316     200 LPLOT = NPLOT
00317         IF (LPLOT .GT. 3) LPLOT = 3
00318         CALL PHYSOR(1.0,1.0)
00319         DO 310 I=1,LPLOT
00320         L = I
00321         IPLOT = IPLOT + 1
00322         CALL LOAD (IPLOT,6310)
00323   C
00324   C     SET UP X-AXIS TITLE
00325   C
00326         IF (ICOLX) 230,210,230
00327     210 ENCODE (19,220,XSTRNG)
00328     220 FORMAT (19X)
00329         GO TO 250
00330     230 ENCODE (19,240,XSTRNG) NAMEX(1,IPLOT),NAMEX(2,IPLOT),
00331        1 UNITX(1,IPLOT),UNITX(2,IPLOT)
00332     240 FORMAT (2A4,' (',2A4,')')
00333   C
00334   C     SET UP Y-AXIS TITLE (USE NAME AND UNITS ASSOCIATED WITH "YMAG").
00335   C
00336     250 ENCODE (19,240,YSTRNG) NAMEY(1,IPLOT),NAMEY(2,IPLOT),
```

```
METPLT  METPLT.FOR      FORTRAN V.5A(023) /AI  [illegible]        [illegible]   PAGE 1-6

00337          1 UNITY(1,IPLOT),UNITY(2,IPLOT))
00338       C
00339       C      SPECIFY SUBPLOT SIZE AND GENERATE AXES [illegible]
00340       C
00341            CALL TITLE(0,0,XSTRNG,19,YSTRNG,19,[illegible])
00342            IF (ICOLX .EQ. 0) CALL ABSNUM
00343            GO TO (310,260,310,310,310,270,310) [illegible](IPLOT))
00344       260  CALL STRPIT (IPLOT,6300)
00345       270  CALL VECPIT (IPLOT,6300)
00346       300  IF (ICOLX .EQ. 0) CALL [illegible] (2)
00347            CALL RESET ('ENDSUM')
00348            CALL ENDGR (IPLOT)
00349            CALL OREL(0.0,YSIZE + YSAV)
00350       310  CONTINUE
00351            CALL PHYSOR(0.,0.)
00352            CALL APPA2D(11.,11.)
00353            CALL TEXT(ITITLE)
00354            IF (ICOLX .EQ. 0) CALL TIMES
00355            CALL ENDPL(0)
00356            KPLOT = KPLOT - LPLOT
00357            IF (KPLOT .GT. 0) GO TO 200
00358       320  IF (KPDATE .LT. LPDATE) GO TO [illegible]
00359            CALL DONEPL
00360            STOP 'PROGRAM EXECUTES SUCCESSFULLY'
00361       330  WRITE(6,340)
00362       340  FORMAT (X'NO PARAMETER FILE FOUND')
00363            STOP 'NO PARAMETER FILE FOUND'
00364       350  WRITE(6,360)
00365       360  FORMAT (X'NO DATA FILE FOUND')
00366            STOP 'NO DATA FILE FOUND'
00367       370  STOP
00368            END
```

```
DATAIN  METPLT.FOR      FORTRAN V.5A(023) /AI  [illegible]        [illegible]   PAGE [illegible]

CCC01            SUBROUTINE DATAIN(OTITLE)
00002       C
00003       C      SUBROUTINE READS IN DATA SET PARAMETERS, AND [illegible] CORRESPONDING TO EACH
00004       C      DATA SET, AND THE DATA ARRAY FROM [illegible].DAT.
00005       C
CC006            PARAMETER MDATA=300,MCHAN=20
00007            DIMENSION OTITLE(25)
00008            COMMON NUMX(MCHAN),NAMEA(2,MCHAN),[illegible],XMIN(MCHAN),
CCC09           1 XINC(MCHAN),XMAX(MCHAN),[illegible],TYPEX(MCHAN),
CC010           2 NUMY(MCHAN),NAMEY(2,MCHAN),UNITY(2,[illegible],YMIN(MCHAN),
00011           3 YINC(MCHAN),YMAX(MCHAN),[illegible],TYPEY(MCHAN),
CC012           4 NUMY2(MCHAN),NAMEY2(2,MCHAN),[illegible],YMIN2(MCHAN),
CC013           5 YINC2(MCHAN),YMAX2(MCHAN),[illegible],TYPEY2(MCHAN),
00014           6 TYPEY2(MCHAN)
00015            COMMON /ARRAY/ DATA(MDATA,MCHAN),[illegible],IREF(MCHAN,4)
CC016            COMMON /REAL/ FORM(MCHAN,3),UNAME([illegible]
00017            COMMON /I1/ IDAY,IYEAR,LDAY,LYEAR,[illegible]
00018            COMMON /N/ NCHAN,NROW,NXTNUM,NDATA
CCC19            DATA CASH,BLANK/'---- ','     '/
00020       C
00021       C      CHECK TO SEE IF A NEW PAGE HEADING [illegible]
00022       C
00023            ISKIP = 0
00024            DO 10 I=1,25
00025       10   IF (OTITLE(I) .NE. BLANK) ISKIP = 1
00026            IF (ISKIP .EQ. 1) GO TO 30
00027       C
00028       C      USE OLD PAGE HEADING IF A NEW ONE [illegible]
00029       C
00030            READ (7,20) (OTITLE(I),I=1,24)
00031       20   FORMAT (X24A1)
00032            GO TO 40
00033       30   SKIP RECORD 7
00034       C
00035       C      READ PARAMETERS OUTPUT BY THE [illegible]
00036       C
00037       40   READ (7,50) NCHAN,NROW,NXTNUM,IYEAR,[illegible],LYEAR,
00038           1 LDATE,LDAY
00039       50   FORMAT (3(X13),2(/X14,A3,XA3))
00040       C
00041       C      READ IN CHANNEL NAMES, UNITS, AND [illegible]
00042       C
00043            READ (7,60) ((IREF(I,J),J=1,3),(UNAME([illegible]),J=1,2),(FORM(I,J),
00044           1 J=1,3),I=1,NCHAN)
00045       60   FORMAT (X13,X2A4,X2A4,X3A4)
00046       C
00047       C      READ IN TIME ARRAY
00048       C
CC049            READ (7,70) ((ITIME(I,J),J=1,3),I=1,[illegible]
CC050       70   FORMAT (6(X14,X13,X14))
00051       C
00052       C      READ IN DATA
00053       C
CC054            DO 90 J=1,NCHAN
00055            SKIP RECORD 7
CC056            READ (7,80) (DATA(I,J),I=1,NROW)
```

```
DATAIN  METPLT.FOR      FORTRAN V.5A(b.1) /A1  [illegible]        8:22    PAGE 1-1

00057       80 FORMAT (b(1X12.6))
00058   C
00059   C       LOOK FOR E-FORMAT REPRESENTATION OF "-----" AND REPLACE WITH THE
00060   C       A-FORMAT REPRESENTATION
00061   C
00062           DO 90 I=1,NROW
00063       90  IF (DATA(I,3) .EQ. .375d+13*1b) DATA(I,3) = DASH
00064           RETURN
00065           END
```

```
INIT    METPLT.FOR      FORTRAN V.5A(b.1) /A1  [illegible]        8:22    PAGE 1

00001           SUBROUTINE INIT(*)
00002   C
00003   C       SUBROUTINE INITIALIZES POINTERS, COUNTERS, AND DATE TO FIRST PLOTTED
00004   C       VALUES.
00005   C
00006           COMMON /P/ IPDATE,IPYEAR,IPDAY,IPDA[illegible],NUMDAY
00007           COMMON /JN/ JPDATE,JPYEAR,JPLAY,NJ[illegible],NJDAY
00008           COMMON /PC/ IPOW,KPOW,LSOW,ICOLS,IC[illegible]
00009           COMMON /INTPLT/ IPLOT,[illegible]
00010           COMMON /DAY/ NAMDAY(7),NUMDAY,DAYS[illegible]
00011   C
00012   C       INITIALIZE "END OF LAST PLOT" DATE AND [illegible]
00013   C
00014           KPYEAR = IPYEAR
00015           KPDATE = IPDATE - 1
00016           KPDAY = IPDAY
00017   C
00018   C       INITIALIZE "LAST ROW" POINTER AND NUMBER OF PLOTS
00019   C
00020           LROW = 1
00021           NPLOT = 1
00022   C
00023   C       FIND INITIAL PLOT DAY IN DAY NAME [illegible]
00024   C
00025           NUMDAY = 0
00026       10  IF (NAMDAY(NUMDAY+1) .NE. IPDAY) RETURN
00027           NUMDAY = NUMDAY + 1
00028           IF (NUMDAY .LE. 6) GO TO 10
00029           WRITE (6,20) IPDAY
00030       20  FORMAT (1X,"'",A3,"' IS NOT FOUND IN DAY NAME [illegible]")
00031           RETURN 1
00032           END
```

```
KEYVAL  GETPLT.FOR      FORTRAN V.5A(042) /AL   8-JUL-81        8:22    PAGE 1


00001         FUNCTION KEYVAL(WORD)
00002   C
00003   C      FUNCTION CHECKS THE INPUT WORD AND RETURNS A NUMBER AS FOLLOWS:
00004   C
00005   C               WORD            KEYVAL
00006   C              ------           ------
00007   C              "    "             4
00008   C              "XAXIS"            3
00009   C              "YAXIS"            2
00010   C               OTHER             1
00011   C
00012         DATA XAXIS,YAXIS,BLANK/'XAXIS','YAXIS','    '/
00013         KEYVAL = 1
00014         IF (WORD .EQ. XAXIS) KEYVAL = 3
00015         IF (WORD .EQ. YAXIS) KEYVAL = 2
00016         IF (WORD .EQ. BLANK) KEYVAL = 4
00017         RETURN
00018         END
```

```
IPTYPE  GETPLT.FOR      FORTRAN V.5A(042) /AL   8-JUL-81        8:22    PAGE 1


00001         FUNCTION IPTYPE(WORD)
00002   C
00003   C      FUNCTION CHECKS THE INPUT WORD AND RETURNS A NUMBER AS FOLLOWS:
00004   C
00005   C               WORD            IPTYPE
00006   C              ------           ------
00007   C              "AVEC "            7
00008   C              "BVEC "            6
00009   C              "YVEC "            5
00010   C              "XVEC "            4
00011   C              "LOG  "            3
00012   C              "LIN  "            2
00013   C               OTHER             1
00014   C
00015         DATA BLIN,BLOG,XVEC,YVEC,BVEC,AVEC/'LIN  ','LOG  ','XVEC ',
00016        1 'YVEC ','BVEC ','AVEC '/
00017         IPTYPE = 1
00018         IF (WORD .EQ. BLIN) IPTYPE = 2
00019         IF (WORD .EQ. BLOG) IPTYPE = 3
00020         IF (WORD .EQ. XVEC) IPTYPE = 4
00021         IF (WORD .EQ. YVEC) IPTYPE = 5
00022         IF (WORD .EQ. BVEC) IPTYPE = 6
00023         IF (WORD .EQ. AVEC) IPTYPE = 7
00024         RETURN
00025         END
```

```
XAXIS   GETPLT.FOR      FORTRAN V.5A(042) /AL   8-JUL-81        8:22    PAGE 1


00001         SUBROUTINE XAXIS(*)
00002   C
00003   C      SUBROUTINE HANDLES INPUT OF X-AXIS PARAMETERS
00004   C
00005         PARAMETER MDATA=300,MCHAN=20
00006         COMMON NUMX(MCHAN),NAMEX(2,MCHAN),UNITX(2,MCHAN),XMIN(MCHAN),
00007        1 XINC(MCHAN),XMAX(MCHAN),INTEX(MCHAN),XDATA(MDATA),TYPEX(MCHAN),
00008        2 NUMY(MCHAN),NAMEY(2,MCHAN),UNITY(2,MCHAN),YMIN(MCHAN),
00009        3 YINC(MCHAN),YMAX(MCHAN),INTEY(MCHAN),YDATA(MDATA),TYPEY(MCHAN),
00010        4 NUMY2(MCHAN),NAMEY2(2,MCHAN),UNITY2(2,MCHAN),YMIN2(MCHAN),
00011        5 YINC2(MCHAN),YMAX2(MCHAN),INTEY2(MDATA),YDAT2(MCHAN),
00012        6 TYPEY2(MCHAN)
00013         COMMON /INTPLT/ IPLOT,NPLOT,NPLOT
00014   C
00015   C      READ X-AXIS PARAMETERS
00016   C
00017        10 READ (4,20,END=60,ERR=60) NUMX(NPLOT),XMIN(1,NPLOT),
00018         1 NAMEX(2,NPLOT),UNITX(1,NPLOT),XMIN(NPLOT),
00019         2 XINC(NPLOT),XMAX(NPLOT),INTEX(NPLOT),XDATA(NPLOT),TYPEX(NPLOT)
00020        20 FORMAT (I3,2A4,2A4,5(F5.0),A5)
00021           GO TO (30,50,30,30,30,30,30) IFIX(XDATA(NPLOT))
00022        30 WRITE (6,40) TYPEX(NPLOT)
00023        40 FORMAT (1X,"",A5,"" IS NOT A VALID X-AXIS INPUT UNIT IS SKIPPED.")
00024           GO TO 10
00025        50 RETURN
00026        60 RETURN 1
00027           END
```

```
VAXIB   METPLT.PCB      FORTRAN V.5A(621) /AL   8-JUL-81        8:22    PAGE 1


00001         SUBROUTINE VAXIN(*)
00002   C
00003   C     SUBROUTINE HANDLES INPUT OF X-AXIS PARAMETERS
00004   C
00005         PARAMETER MDATA=300,MCHAN=25
00006         COMMON NUMX(MCHAN),NAMEX(2,MCHAN),UNITX(2,MCHAN),XMIN(MCHAN),
00007        1 XINC(MCHAN),XMAX(MCHAN),TMLDX(MCHAN),DELX(MCHAN),TYPEX(MCHAN),
00008        2 NUMY(MCHAN),NAMEY(2,MCHAN),UNITY(2,MCHAN),YMIN(MCHAN),
00009        3 YINC(MCHAN),YMAX(MCHAN),TMLDY(MCHAN),DELY(MCHAN),TYPEY(MCHAN),
00010        4 NUMY2(MCHAN),NAMEY2(2,MCHAN),UNITY2(2,MCHAN),YMIN2(MCHAN),
00011        5 YINC2(MCHAN),YMAX2(MCHAN),TMLDY2(MCHAN),DELY2(MCHAN),
00012        6 TYPEY2(MCHAN)
00013         COMMON /IN=PLT/ IFLG ,...,NPLOT
00014   C
00015   C     READ FIRST X-AXIS PARAMETERS
00016   C
00017        10 READ (4,20,END=150,ERR=150) NUMX(NPLOT),(NAMEX(I,NPLOT),I=1,2),
00018          1 NAMEX(2,NPLOT),UNITX(1,NPLOT),UNITX(2,NPLOT),XMIN(NPLOT),
00019          2 XINC(NPLOT),XMAX(NPLOT),TMLDX(NPLOT),DELX(NPLOT),TYPEX(NPLOT)
00020        20 FORMAT (I3,2A4,1X2A4,3(F5.0),A4)
00021   C
00022   C     CHECK TYPE OF PLOT REQUESTED
00023   C
00024         GO TO (30,50,30,30,30,60,90) INDEX(TYPEX(NPLOT),'*')
00025   C
00026   C     TYPE IS UNDEFINED - WRITE ERROR MESSAGE
00027   C
00028        30 WRITE (6,40) TYPEY(NPLOT)
00029        40 FORMAT (1X'*',A5,'* IS NOT A VALID X-AXIS PARAM. PLOT IS SKIPPED.')
00030           GO TO 110
00031   C
00032   C     TYPE = "LIN" - LOOK FOR NEXT X-AXIS
00033   C
00034        50 CALL XCOPY
00035           GO TO 110
00036   C
00037   C     TYPE = "BVEC" - LOOK FOR MATCHING TYPE = "AVEC"
00038   C
00039        60 READ (4,20,END=150,ERR=150) NUMY2(NPLOT),(NAMEY2(I,NPLOT),I=1,2),
00040          1 NAMEY2(2,NPLOT),UNITY2(1,NPLOT),UNITY2(2,NPLOT),YMIN2(NPLOT),
00041          2 YINC2(NPLOT),YMAX2(NPLOT),TMLDY2(NPLOT),DELY2(NPLOT),
00042          3 TYPEY2(NPLOT)
00043   C
00044   C     IF TYPE = "AVEC" IS NOT FOUND, WRITE ERROR MESSAGE
00045   C
00046           IF (IPTYPE(TYPEY2(NPLOT)) .EQ. 7) GO TO 90
00047           WRITE (6,70) TYPEY(NPLOT),TYPEY2(NPLOT)
00048        70 FORMAT (1X'*',A5,'* AND *',A5,'* ARE NOT COMPATIBLE TYPES. PLOT',
00049          1 ' SKIPPED.')
00050           BACKSPACE 4
00051           GO TO 110
00052        80 CALL XCOPY
00053           GO TO 110
00054   C
00055   C     TYPE = "AVEC" - SAVE "AVEC" PARAMETERS AND LOOK FOR MATCHING
00056   C                     TYPE = "BVEC"
```

```
VAXIB   METPLT.FOR      FORTRAN V.5A(621) /AL   8-JUL-81        8:22    PAGE 1-1


00057   C
00058        90 NUMY2(NPLOT) = NUMY(NPLOT)
00059           NAMEY2(1,NPLOT) = NAMEY(1,NPLOT)
00060           NAMEY2(2,NPLOT) = NAMEY(2,NPLOT)
00061           UNITY2(1,NPLOT) = UNITY(1,NPLOT)
00062           UNITY2(2,NPLOT) = UNITY(2,NPLOT)
00063           YMIN2(NPLOT) = YMIN(NPLOT)
00064           YINC2(NPLOT) = YINC(NPLOT)
00065           YMAX2(NPLOT) = YMAX(NPLOT)
00066           TMLDY2(NPLOT) = TMLDY(NPLOT)
00067           DELY2(NPLOT) = DELY(NPLOT)
00068           TYPEY2(NPLOT) = TYPEY(NPLOT)
00069           READ (4,20,END=150,ERR=150) NUMX(NPLOT),(NAMEX(I,NPLOT),
00070          1 NAMEX(2,NPLOT),UNITX(1,NPLOT),UNITX(2,NPLOT),XMIN(NPLOT),
00071          2 XINC(NPLOT),XMAX(NPLOT),TMLDX(NPLOT),DELX(NPLOT),TYPEY(NPLOT)
00072   C
00073   C     IF TYPE = "BVEC" IS NOT FOUND, WRITE ERROR MESSAGE
00074   C
00075           IF (IPTYPE(TYPEY(NPLOT)) .EQ. 6) GO TO 100
00076           WRITE (6,70) TYPEY(NPLOT),TYPEY2(NPLOT)
00077           BACKSPACE 4
00078           GO TO 110
00079       100 CALL XCOPY
00080           GO TO 110
00081   C
00082   C     LOOK FOR MORE Y-AXIS PARAMETERS
00083   C
00084       110 READ (4,120,END=150) NUMX
00085       120 FORMAT (I4)
00086           GO TO (130,140,10,110) KEYVAL(NUMX)
00087       130 BACKSPACE 4
00088           GO TO 10
00089       140 BACKSPACE 4
00090       150 RETURN 1
00091           END
```

```
MAIN.   RETPLT.FOR      FORTRAN V.5A(021) /AI   [illegible]        8:22    PAGE 1


00001         SUBROUTINE XCOPY
00002   C
00003   C     SUBROUTINE COPIES DEFAULT X-AXIS PARAMETERS
00004   C
00005         PARAMETER NDATA=300,NCHAN=25
00006         COMMON NUMX(NCHAN),NAMEX(2,NCHAN),[illegible],XZERO(NCHAN),
00007        1 XINC(NCHAN),XMAX(NCHAN),TMLDA(NCHAN),[illegible],TYPEX(NCHAN),
00008        2 NUMY(NCHAN),NAMEY(2,NCHAN),UNITY(2,[illegible]),YZERO(NCHAN),
00009        3 YINC(NCHAN),YMAX(NCHAN),TMLDY(NCHAN),[illegible],TYPEY(NCHAN),
00010        4 NUMY2(NCHAN),NAMEY2(2,NCHAN),UNITY2(2,[illegible]),YZERO2(NCHAN),
00011        5 YINC2(NCHAN),YMAX2(NCHAN),TMLDY2(NCHAN),[illegible],
00012        6 TYPEY2(NCHAN)
00013         COMMON /INTPLT/ IPLOT,LPLOT,NPLOT
00014         NUMX(NPLOT+1) = NUMX(NPLOT)
00015         NAMEX(1,NPLOT+1) = NAMEX(1,NPLOT)
00016         NAMEX(2,NPLOT+1) = NAMEX(2,NPLOT)
00017         UNITX(1,NPLOT+1) = UNITX(1,NPLOT)
00018         UNITX(2,NPLOT+1) = UNITX(2,NPLOT)
00019         XMIN(NPLOT+1) = XMIN(NPLOT)
00020         XINC(NPLOT+1) = XINC(NPLOT)
00021         XMAX(NPLOT+1) = XMAX(NPLOT)
00022         TMLDX(NPLOT+1) = TMLDX(NPLOT)
00023         DELX(NPLOT+1) = DELX(NPLOT)
00024         TYPEX(NPLOT+1) = TYPEX(NPLOT)
00025         NPLCT = NPLOT + 1
00026         RETURN
00027         END
```

```
SETUP   RETPLT.FOR      FORTRAN V.5A(021) /AI   [illegible]        8:22    PAGE 1


00001         SUBROUTINE SETUP(NSET,TITLPL,*)
00002   C
00003   C     SUBROUTINE SETS THE FINAL JULIAN DATE (AND DAY OF THE WEEK) FOR THE
00004   C     CURRENT PLOTS.  ROW NUMBERS FOR THE BEGINNING AND FINAL DATES ARE
00005   C     ALSO FOUND.
00006   C
00007         PARAMETER NDATA=300,NCHAN=25
00008         DIMENSION TITLPL(25)
00009         COMMON /ARRAY/ DATA(NDATA,NCHAN),IF[illegible](NDATA,3),IF2F(NCHAN,4)
00010         COMMON /DAY/ NAMDAY(7),NUMDAY,DAYS(7)
00011         COMMON /IL/ IDATE,IYEAR,IDAY,LDATE,[illegible]
00012         COMMON /N/ NCHAN,NROW,MAXNUM,NDAYS
00013         COMMON /P/ IPDATE,IPYEAR,IPDAY,LPDATE,[illegible]
00014         COMMON /JK/ JPDATE,JPYEAR,JPDAY,KPDATE,[illegible]
00015         COMMON /RC/ IROW,KROW,LROW,ICOLK,ICOLL,[illegible]
00016         DATA I1,I2,F1,F2 /' IMI','TIAL','    ',' [illegible]'/
00017   C
00018   C     SET FIRST DATE AND DAY EQUAL TO THE [illegible] AND DAY AFTER THE END OF THE
00019   C     PREVIOUS PLOT SET.
00020   C
00021         JPDATE = KPDATE + 1
00022         NUMDAY = NUMDAY + 1
00023         IF (NUMDAY .GT. 7) NUMDAY = NUMDAY - 7
00024         JPDAY = NAMDAY(NUMDAY)
00025         JPYEAR = KPYEAR
00026         IF ([illegible]) NDAYS = LPDATE - JPDATE + 1
00027   C
00028   C     SET LAST DATE AND DAY EQUAL TO "NDAYS" AFTER FIRST DATE AND DAY
00029   C
00030         KPDATE = JPDATE + NDAYS - 1
00031         NUMDAY = NUMDAY + NDAYS - [(NDAYS/7)*7] - 1
00032         IF (NUMDAY .GT. 7) NUMDAY = NUMDAY - 7
00033         IF (NUMDAY .LT. 1) NUMDAY = NUMDAY + 7
00034         KPDAY = NAMDAY(NUMDAY)
00035         KPYEAR = JPYEAR
00036   C
00037   C     WRITE PAGE HEADING FOR OUTPUT SUMMARY
00038   C
00039         WRITE (6,10) TITLPL,JPDAY,JPDATE,JP[illegible],KPDAY,KPDATE,KPYEAR,NSET
00040        10 FORMAT (1H1,53X,25A1/40X'DATA FOR THE [illegible] ',A3,XI3,',',I4,
00041       1 ' THROUGTH ',A3,XI3,',',I4//X'SUMMARY OF PLOT SET NUMBER ',I3//)
00042   C
00043   C     FIND THE ROW NUMBER FOR THE BEGINNING OF THE FIRST JULIAN DATE WHICH IS
00044   C     TO BE PLOTTED.
00045   C
00046         IROW = LROW
00047        20 IF (ITIME(IROW,2) .GE. JPDATE) GO TO 40
00048         IROW = IROW + 1
00049         IF (IROW .LE. NROW) GO TO 20
00050         WRITE (6,30) I1,I2,JPDATE
00051        30 FORMAT (1X,2A4,' JULIAN DATE (',I3,') [illegible]'/
00052       1 X'PLOT SET IS SKIPPED')
00053         RETURN 1
00054   C
00055   C     FIND THE ROW NUMBER FOR THE END OF THE [illegible] JULIAN DATE WHICH IS TO BE
00056   C     PLOTTED.
```

```
SETUP   RETPLT.FOR     FORTRAN V.5A(bx1) /A...  ...:...:..        8:32    PAGE 1-9

00057   C
00058      40 LBOW = IBOW
00059      50 IF (ITIME(LPOW,2) .GT. XPDATE) GO TO 60
00060         LBOW = LBOW + 1
00061         IF (LBOW .LE. NPOW) GO TO 50
00062         IF (ITIME(LPOW-1,2) .EQ. XPDATE) GO TO 60
00063         WRITE (6,30) P1,P2,XPDATE
00064         RETURN 1
00065      60 LBOW = LBOW - 1
00066         RETURN
00067         END
```

```
LOAD    RETPLT.FOR     FORTRAN V.5A(bx1) /AI    ...:...:..        8:22    PAGE 1

00001         SUBROUTINE LOAD (IPLOT,*)
00002   C
00003   C     SUBROUTINE LOCATES THE APPROPRIATE COLUMNS IN THE DATA ARRAY, TESTS
00004   C     THIS DATA AND LOADS THE X- AND Y-ARRAYS WITH ACCEPTABLE DATA IS
00005   C     FOUND.  IF ICOLX = 0, THEN "ITIME" NAMED LABEL "DATA" IS USED FOR THE X-
00006   C     AXIS.
00007   C
00008         PARAMETER NDATA=300,NCHAN=25
00009         DIMENSION TYPE1(2),TYPE2(2)
00010         REAL NAMEX
00011         COMMON NUM(NCHAN),NAMEX(2,NCHAN),UNITX(2,NCHAN),XMIN(NCHAN),
00012        1 XINC(NCHAN),XMAX(NCHAN),TMLDX(NCHAN),...(NCHAN),TYPEX(NCHAN),
00013        2 NUMY(NCHAN),NAMEY(2,NCHAN),UNITY(2,...),YMIN(NCHAN),
00014        3 YINC(NCHAN),YMAX(NCHAN),TMLDY(NCHAN),...(NCHAN),TYPEY(NCHAN),
00015        4 NUMY2(NCHAN),NAMEY2(2,NCHAN),UNITY2(...),YMIN2(NCHAN),
00016        5 YINC2(NCHAN),YMAX2(NCHAN),TMLDY2(...),...(NCHAN),
00017        6 TYPEY2(NCHAN)
00018         COMMON /N/ NCHAN,NPCH,MAXNUM,NDAYS
00019         COMMON /ARRAY/ DATA(NDATA,NCHAN),I...(...,3),IFLAG(NCHAN,4)
00020         COMMON /JK/ JPDATE,JPTIME,JPDAY,JMOA...,JADAY
00021         COMMON /BC/ IBOW,NPCH,LBOW,ICOLX,...
00022         COMMON /POINT/ XARRAY(NDATA),YMAG(...(NDATA)
00023         DATA X,1,Y2,MX,MY,MY2,ITYPE1,ITYPE2,...'A ',' Y ','Y2',
00024        1 'NUMX ','NUMY ','NUMY2','LINEAR   ',...,'TIME'/
00025   C
00026   C     FIND DATA MATRIX COLUMN NUMBERS
00027   C
00028         ICOLX = -1
00029         ICOLY = -1
00030         ICOLY2 = -1
00031         DO 10 I=1,NCHAN
00032         IF (NUMX(IPLOT) .EQ. IABS(I,1)) ICOLX = I
00033         IF (NUMY(IPLOT) .EQ. IABS(I,1)) ICOLY = I
00034         IF (NUMY2(IPLOT) .EQ. IABS(I,1)) ICOLY2 = I
00035      10 CONTINUE
00036         IF (NUMX(IPLOT) .NE. 0) GO TO 20
00037   C
00038   C     IF X-AXIS IS TIME, RESET XMIN,XINC,XMAX, AND BASIS TO DEFAULTS
00039   C
00040         ICOLX = 0
00041         XMIN(IPLOT) = 0
00042         XINC(IPLOT) = 720
00043         XMAX(IPLOT) = NDAYS * 1440
00044         IF ((NAMEX(1,IPLOT) .EQ. BLANK) .AND. (NAMEX(...,IPLOT) .EQ.
00045        1 BLANK)) NAMEX(1,IPLOT) = 'M
00046   C
00047   C     SET ITYPE = 1 FOR LINEAR; ITYPE = 2 FOR ...
00048   C
00049      20 ITYPE = 1
00050         KEY = IPTYPE(TYPEX(IPLOT))
00051         IF ((KEY .EQ. 0) .OR. (KEY .EQ. 8)) ITYPE = 2
00052   C
00053   C     WRITE SUMMARY FOR CURRENT PLOT
00054   C
00055         WRITE (6,30) IPLOT,(NAMEX(I,IPLOT),...(...,IPLOT),I=1,2)
00056      30 FORMAT (//X'PLOT NUMBER ',I3,': ',2A4,' vs ',2A4/)
```

```
LOAD    METPLT.FOR      FORTRAN V.5A(021) /A1   [illegible]          8:22    PAGE 1-1


00057   C
00058   C       WRITE X-CHANNEL PARAMETERS
00059   C
00060           IF (DELX(IPLOT) .GT. 0) WRITE (6,50) XCHAN(IPLOT),THLDX(IPLOT),
00061          1 DELX(IPLOT)
00062           IF (DELX(IPLOT) .LT. 0) WRITE (6,50) XCHAN(IPLOT),THLDX(IPLOT)
00063   C
00064   C       WRITE Y-CHANNEL PARAMETERS
00065   C
00066           IF (DELY(IPLOT) .GT. 0) WRITE (6,60) YCHAN(IPLOT),THLDY(IPLOT),
00067          1 DELY(IPLOT)
00068           IF (DELY(IPLOT) .LT. 0) WRITE (6,60) YCHAN(IPLOT),THLDY(IPLOT)
00069       50 FORMAT (10X,A2,' CHANNEL = ',I2,'; CHANNEL NO = ',F5.0,
00070          1 '; HYSTERESIS = ',F5.0)
00071       60 FORMAT (10X,A2,' CHANNEL = ',I2,'; CHANNEL NO = ',F5.0,
00072          1 '; HYSTERESIS IS TOO LOW')
00073   C
00074   C       CHECK PLOT TYPE
00075   C
00076           GO TO (60,90) ITYPE
00077   C
00078   C       LINEAR PLOT
00079   C
00080       60  WRITE (6,70) (TYPE(I),I=1,2)
00081       70  FORMAT (10X,'TYPE = ',2A5//)
00082           GO TO 90
00083   C
00084   C       WRITE Y2-CHANNEL PARAMETERS FOR VECTOR PLOT
00085   C
00086       80  IF (DELY2(IPLOT) .GE. 0) WRITE (6,50) Y2CHAN(IPLOT),
00087          1 THLDY2(IPLOT),DELY2(IPLOT)
00088           IF (DELY2(IPLOT) .LT. 0) WRITE (6,50) Y2CHAN(IPLOT),
00089          1 THLDY2(IPLOT)
00090           WRITE (6,70) (TYPE2(I),I=1,2)
00091       90  IF ((ICOLX .LT. 0) .OR. (ICOLY .LT. 0)) GO TO 150
00092           IF ((ICOLY2 .LT. 0) .AND. (ITYPE .EQ. 2)) GO TO 150
00093           KOUNT = 1
00094           DO 140 KROW=IROW,LROW
00095           NPLTX = 1
00096           NPLTY = 1
00097           NPLTY2 = 1
00098           IF (ICOLX) 110,100,110
00099   C
00100   C       X-AXIS IS TIME -- CONVERT READING TO SECONDS.
00101   C
00102      100  FNEWX = (ITIME(KROW,2) - ITIME(1,2)) * 3600 + (ITIME(KROW,3)/100)
00103          1 + 60 * (ITIME(KROW,3) - ((ITIME(KROW,3)/100) * 100))
00104           GO TO 120
00105   C
00106   C       X-AXIS IS DATA FROM COLUMN "ICOLX"
00107   C
00108      110  FNEWX = DATA(KROW,ICOLX)
00109   C
00110   C       TEST FOR ACCEPTABLE X DATA
00111   C
00112      120  CALL CHECK (1,FNEWX,OLDX,THLDX(IPLOT),XCHAN(IPLOT),NPLTX,
00113          1 &140)
00114           FNEWY = DATA(KROW,ICOLY)
00115   C
00116   C       TEST FOR ACCEPTABLE Y DATA (MAGNITUDE)
00117   C
00118           CALL CHECK (2,FNEWY,OLDY,THLDY(IPLOT),YCHAN(IPLOT),NPLTY,
00119          1 &140)
00120   C
00121   C       SEE IF A VECTOR PLOT WAS REQUESTED
00122   C
00123           IF (ITYPE .NE. 2) GO TO 130
00124           FNEWY2 = DATA(KROW,ICOLY2)
00125   C
00126   C       TEST FOR ACCEPTABLE Y2 DATA (DIRECTION)
00127   C
00128           CALL CHECK (3,FNEWY2,OLDY2,THLDY2(IPLOT),Y2CHAN(IPLOT),
00129          1 NPLTY2,&140)
00130       130 IF ((NPLTX .EQ. 1) .AND. (NPLTY .EQ. 1) .AND. (NPLTY2 .EQ. 1))
00131          1 CALL MEST (ITYPE,FNEWX,FNEWY,FNEWY2,XCHAN(IPLOT),DELY(IPLOT),
00132          2 DELY2(IPLOT),&140)
00133   C
00134   C       LOAD ARRAYS IFF ACCEPTABLE DATA WAS FOUND FOR X AND Y (AND Y2, IN
00135   C       THE CASE OF VECTOR PLOTS). UPDATE "OLD" VALUES FOR COMPARISON AND
00136   C       INCREMENT ROW COUNTER.
00137   C
00138           XARRAY(KOUNT) = FNEWX
00139           OLDX = FNEWX
00140           YMAG(KOUNT) = FNEWY
00141           OLDY = FNEWY
00142           YDIR(KOUNT) = FNEWY2
00143           OLDY2 = FNEWY2
00144           KOUNT = KOUNT + 1
00145       140 CONTINUE
00146           KROW = KOUNT - 1
00147           RETURN
00148   C
00149   C       WRITE ERROR MESSAGES IF ANY CHANNEL NUMBERS WERE NOT FOUND
00150   C
00151       150 IF (ICOLX .LT. 0) WRITE (6,160) XA,XCHAN
00152           IF (ICOLY .LT. 0) WRITE (6,160) YA,YCHAN
00153           IF ((ICOLY2 .LT. 0) .AND. (ITYPE .EQ. 2)) WRITE (6,160) YY2,IPLOT
00154       160 FORMAT (2A5,' WAS NOT FOUND IN "DATA" -- PLOT NUMBER ',
00155          1 I3,' IS SKIPPED')
00156           RETURN 1
00157           END
```

```
CHECK   METPLT.FOR      FORTRAN V.5A(b21) /A1   8-JUL-91        8:22    PAGE 1


00001         SUBROUTINE CHECK (J,DATNEW,DATOLD,THRESH,DELTA,NOPLOT,*)
00002   C
00003   C     SUBROUTINE TESTS FOR INVALID DATA ("-----"), VALUES < THRESHOLD,
00004   C     AND VALUES WITHIN A RANGE "DELTA" OF THE PREVIOUS VALUE.
00005   C
00006         PARAMETER MDATA=300,MCHAN=25
00007         DIMENSION XYZ(3)
00008         COMMON /ARRAY/ DATA(MDATA,MCHAN),ITIME(MDATA,3),IPEF(MCHAN,4)
00009         COMMON /BC/ IBOM,KBOM,LBOM,ICOLX,ICOLY,ICOLZ
00010         DATA DASH,XYZ/'----- ',' X ',' Y ',' Z '/
00011         IF (DATNEW .NE. DASH) GO TO 20
00012         WRITE (6,10) (ITIME(KBOM,I),I=1,3)
00013      10 FORMAT (2I4,I3,2X,'DATA IS INVALID')
00014         RETURN 1
00015      20 IF (DATNEW .GE. THRESH) GO TO 40
00016         WRITE (6,30) (ITIME(KBOM,I),I=1,3),XYZ(J),DATNEW
00017      30 FORMAT (2I4,I3,2X,A5,F12.6,' IS < THRESHOLD')
00018         RETURN 1
00019      40 IF (K .EQ. 1) GO TO 60
00020         IF (DELTA .LT. 0) RETURN
00021         IF ((DATNEW .LE. (DATOLD - DELTA)) .AND. (DATNEW .GE.
00022        1 (DATOLD + DELTA))) GO TO 70
00023         RETURN
00024      60 DATOLD = DATNEW
00025      70 NOPLOT = 0
00026         RETURN
00027         END
```

```
MYST    METPLT.FOR      FORTRAN V.5A(b21) /A1   8-JUL-91        8:22    PAGE 1


00001         SUBROUTINE MYST(ITYPE,XMEAN,SMEAN,PMEAN,PSDEV,QSDEV,QDELTY,DELTY2,*)
00002   C
00003   C     SUBROUTINE PRINTS MYSTERIOUS MESSAGES & VALUES AS SUPPRESSED HOPE TO
00004   C     FAILURE OF THE MYSTERIOUS TEST
00005   C
00006         PARAMETER MDATA=300,MCHAN=25
00007         COMMON /ARRAY/ DATA(MDATA,MCHAN),ITIME(MDATA,3),IPEF(MCHAN,4)
00008         COMMON /BC/ IBOM,KBOM,LBOM,ICOLX,ICOLY,ICOLZ
00009         DATA X,Y,Z/'X  = ','Y  = ','Z  = '/
00010         WRITE (6,10) (ITIME(KBOM,I),I=1,3)
00011      10 FORMAT (2I4,I3,2X,'MYSTERIOUS SUPPRESSED ---')
00012         IF (DELTX .GE. 0) WRITE (6,20) X,PMEAN,QSDEV
00013         IF (DELTY .GE. 0) WRITE (6,20) Y,PMEAN,QSDEV
00014         IF ((ITYPE .EQ. 2) .AND. (DELTY2 .GE. 0)) WRITE (6,20) Z,PMEAN,
00015        1 DELTY2
00016      20 FORMAT (2X,A5,F12.6,' IS WITHIN ',F6.3,' OF PREVIOUS VALUE')
00017         WRITE (6,30)
00018      30 FORMAT (/)
00019         RETURN 1
00020         END
```

```
TEXT    METPLT.FOR      FORTRAN V.5A(b21) /A1   8-JUL-91        8:22    PAGE 1


00001         SUBROUTINE TEXT(ITITLE)
00002   C
00003   C     SUBROUTINE WRITES PAGE TITLE AND SUBTITLE
00004   C
00005         PARAMETER MDATA=300,MCHAN=25
00006         DIMENSION ITITLE(5)
00007         COMMON /ARRAY/ DATA(MDATA,MCHAN),ITIME(MDATA,3),IPEF(MCHAN,4)
00008         COMMON /JK/ JPDATE,JPYEAR,JPDAY,KPDATE,KPYEAR,KPDAY
00009         COMMON /BC/ IBOM,KBOM,LBOM,ICOLX,ICOLY,ICOLZ
00010         COMMON /ALPHA/ ISTRNG(9),XSTRNG(9)
00011   C
00012   C     WRITE PAGE TITLE WITH 0.3" CHARACTERS.
00013   C
00014         CALL HEIGHT(0.3)
00015         STRNGL = XMESS(ITITLE,100)
00016         X = (11. - STRNGL)/2.0
00017         CALL MESSAG(ITITLE,100,X,10.0)
00018   C
00019   C     SET UP STRING FOR SUBTITLE
00020   C
00021         ENCODE(36,10,XSTRNG)JPDAY,ITIME(JBOM,1),ITIME(JBOM,2),KPDAY,
00022        1 ITIME(LBOM,2),ITIME(LBOM,1)
00023      10 FORMAT (A3,I3,',',I4,' THROUGH ',A3,I3,':',I2,':',I2)
00024   C
00025   C     WRITE SUBTITLE WITH 0.2" CHARACTERS.
00026   C
00027         CALL HEIGHT(0.2)
00028         STRNGL = XMESS(XSTRNG,100)
00029         X = (11. - STRNGL)/2.0
00030         CALL MESSAG(XSTRNG,100,X,10.3)
00031         CALL RESET('HEIGHT')
00032         RETURN
00033         END
```

```
MAIN.   METPLT.FOR      FORTRAN V.5A(b21) /A1   8-JUL-91        8:22    PAGE 1


00001         SUBROUTINE TIMES
00002   C
00003   C     SUBROUTINE WRITES THE JULIAN DATES AND A LABEL ON THE TIME AXIS.
00004   C
00005         COMMON /JK/ JPDATE,JPYEAR,JPDAY,KPDATE,KPYEAR,KPDAY
00006         COMMON /DAY/ NAMDAY(7),NUMDAY,DAYSIZ
00007         COMMON /X/ X00,X12,X24,XDATE
00008         CALL HEIGHT(0.2)
00009         STRNGL = XMESS('JULIAN DATE',11)
00010         X = (11. - STRNGL)/2.0
00011         CALL MESSAG('JULIAN DATE',11,X,0.1)
00012         CALL RESET('HEIGHT')
00013         X = XDATE
00014         DO 20 NDATE=JPDATE,KPDATE
00015         I = NDATE
00016         CALL INTNO(I,X,0.5)
00017         X = X + DAYSIZ
00018      20 CONTINUE
00019         RETURN
00020         END
```

```
EX      HSTPLT.FOR      FORTRAN V.5A(621) /AI   8-JUL-81        8:22    PAGE 1


CCC01         SUBROUTINE EX(I)
00002  C
CCC03  C      SUBROUTINE WRITES TIMES ON THE X-AXIS
00004  C
CC005         COMMON /X/ X00,X12,X24,XDATE
00006         COMMON /B/ NCHAN,NBOX,NATBOX,NDAYS
CCC07         COMMON /DAY/ NABDAY(7),NUMDAY,DAYSIZ
CCC08         XSTART = X00
00009         XMID = X12
CC010         CALL WEIGHT(0.10)
CC011         DO 20 J=1,NDAYS
00012         IF ((I .EQ. 1) .OR. (J .NE. 1)) CALL XLABEL (J,XSTART,-0.25)
00013         CALL INTNO(1200,XMID,-0.25)
00014         XSTART = XSTART + DAYSIZ
00015      20 XMID = XMID + DAYSIZ
00016         CALL INTNO(2400,X24,-0.25)
CC017         CALL RESET('WEIGHT')
00018         RETURN
CC019         END
```

```
VECPLT  HSTPLT.FOR      FORTRAN V.5A(621) /AI   8-JUL-81        8:22    PAGE 1

00001         SUBROUTINE VECPLT (IPLOT,*)
CCC02  C
00003  C      SUBROUTINE DRAWS VECTORS WITH MAGNITUDES AND DIRECTIONS DEFINED BY
00004  C      YMAG AND YDIR
CCC05  C
00006         PARAMETER NDATA=300,NCHAN=25
CCC07         COMMON NUMX(NCHAN),NAMEX(2,NCHAN),NUMIX(2,NCHAN),XBIN(NCHAN),
CCC08        1 XINC(NCHAN),XMAX(NCHAN),YMLOX(NCHAN),YLOX(NCHAN),TYPEX(NCHAN),
00009        2 NUMY(NCHAN),NAMEY(2,NCHAN),NMITY(2,NCHAN),YBIN(NCHAN),
00010        3 YINC(NCHAN),YMAX(NCHAN),YMLOY(NCHAN),YLOY(NCHAN),TYPEY(NCHAN),
CC011        4 NUMX2(NCHAN),NAMEX2(2,NCHAN),NMITX2(2,NCHAN),XBIN2(NCHAN),
00012        5 XINC2(NCHAN),XMAX2(NCHAN),YMLOX2(NCHAN),YLOX2(NCHAN),
00013        6 TYPEX2(NCHAN)
00014         COMMON /BC/ IBOW,NBOW,LBOW,ICOLR,IDOLR,IDOLR2
00015         COMMON /POINT/ XARRAY(NDATA),YMAG(NDATA),YDIR(NDATA)
00016         COMMON /SIZE/ XSIZE,YSIZE
00017  C
00018
00019  C      CALCULATE SCALE FACTORS IN INCHES/UNIT
00020  C
00021         XUNITS = XSIZE/(XMAX(IPLOT) - XMIN(IPLOT))
00022         YUNITS = YSIZE/(YMAX(IPLOT) - YMIN(IPLOT))
00023  C
00024  C      DEFINE AXIS SYSTEM
00025  C
00026         CALL GPAF(XMIN(IPLOT),XINC(IPLOT),XMAX(IPLOT),XAIN(IPLOT),
00027        1 YINC(IPLOT),YMAX(IPLOT))
00028  C
CC029  C      PLOT AND LABEL A SMALL VECTOR (POINTING UP TO INDICATE ZERO DEGREES)
CC030  C
00031         CALL VECTOR(0.2,2.2,0.2,2.7,1221)
CC032         CALL RESSAG('0',1,0.15,2.75)
00033  C
00034  C      SET UP VECTORS FOR ALL SETS OF DATA VALUES
CC035  C
00036         XTO = XSIZE * 0.5
CC037         RADIAN = 3.14159/180.0
00038         DO 170 I=1,NBOW
00039  C
CC040  C      CALCULATE COMPONENTS OF VECTOR
00041  C
CC042         X = YMAG(I)*SIN(YDIR(I)*RADIAN)
CC043         Y = YMAG(I)*COS(YDIR(I)*RADIAN)
00044  C
00045  C      DETERMINE ABSOLUTE LOCATIONS (IN INCHES) FOR VECTOR END POINTS
CC046  C
00047         XTO = XARRAY(I) * XUNITS
CC04E         XPROB = XTO + (X * XUNITS)
CC049         YPROB = YTO + (Y * YUNITS)
00050  C
CC051  C      DRAW VECTORS
00052  C
00053    170 CALL VECTOR(XPROB,YPROB,XTO,YTO,1221)
CC054         RETURN 1
00055         END
```

```
SIELI  MEIPLI.#OR      #UBPIAB W.'A(1,1) /A1   ------##        M:2#    ##,#*

00001          SUBROUTINE S*#PLT(IFLCT,*)
00002    C
00003    C     SUBPOUTINP PLOTS EPAJ #J. AAP:AI #.I.....# A#J%.
00004    C
00005          PARAMETER MDATA=300,M.AA=15
00006          COMMON BUBX(MCHAN),BAR.A(2,M.HAN),U#.....1.HAN),XMIN(MCHAN),
00007         1 XIBC(MCHAN),XMAX(MCHAN),YMIN(MCHAN),.....(M.HAN),TYPEX(MCHAN),
00008         2 BUBY(MCHAN),BARPY(2,MCHAN),UNITY(2,.....),YMIN(MCHAN),
00009         3 YIBC(MCHAN),YMAX(MCHAN),YMIN1(M.HAN),.....,M.HAN),TYEPY(MCHAN),
00010         4 BUBX2(MCHAN),BARPY2(2,M.HAN),UBAI(2,.....),XMIN2(MCHAN),
00011         5 XIBC2(MCHAN),YMAX2(MCHAN),YMIN2(M.HAN),.....(M.HAN),
00012         6 TYPEY2(MCHAN)
00013          COMMON /PC/ I#Oa,ARCa,.....,.....,.....
00014          COMMON /POINP/ XARRAY(MDATA),YMA.(M.....,M.....1,MDATA)
00015    C
00016    C     DEFINE AXIS SYSTEM
00017    C
00018          CALL GPAP(XMIN(IFLCT),XIBC(IFLCT),XMAX(.....),XMIN(IFLCT),
00019         1 XIBC(IFLCT),YMAX(IFLCT))
00020    C
00021    C     D3Ab GPAPH
00022    C
00023          CALL CURVE(XARRAY,YMA.,ARCa,1)
00024          RETURN 1
00025          END
```

Appendix D
Listing of Program METINP
(Version 1.2)

```
MAIN.   METINP.FOR      FORTRAN V.5A(621) /AI   10-JUL-81       16:26   PAGE 1

00001         PROGRAM METINP
00002   C
00003   C     METINP VERSION 1.2   SEPT.1981
00004   C
00005   C     JOHN HOOVER
00006   C     CODE 8330
00007   C     U.S. NAVAL RESEARCH LABORATORY
00008   C     WASHINGTON, D.C.   20375
00009   C
00010   C
00011   C     PROGRAM INPUTS PARAMETERS FROM THE KEY AND PRODUCES A FORMATTED CONTROL
00012   C     FILE (FOR06.DAT) FOR THE METEOR PROGRAM PACKAGE
00013   C
00014   C     PROGRAM PARAMETERS:
00015   C     MLINE=MAXIMUM POSSIBLE NUMBER OF LINES OF INSTRUCTIONS FOR ANY ONE
00016   C           PROGRAM.  (CURRENTLY,MCHAN=28,CHANGE MLINE TO MLINE=25 IN "METSET")
00017   C
00018   C     KEYWORDS:
00019   C     THE PROGRAM (AND FUNCTION WITHIN A PROGRAM) FOR WHICH A CONTROL FILE IS
00020   C     TO BE CONSTRUCTED IS SELECTED BY USE OF KEYWORDS.  CURRENTLY, THE LEGAL
00021   C     KEYWORDS ARE-
00022   C
00023   C     "SORT "=SELECT PROGRAM "METSRT"
00024   C     "CALC "=SELECT PROGRAM "METCLC"
00025   C             "WIND "=FUNCTION TO CALCULATE GEOSTROPHIC WIND VELOCITY
00026   C             "MOIST"=FUNCTION TO CALCULATE WATER VAPOR LOADING
00027   C     "PLOT "=SELECT PROGRAM "METPLT"
00028   C
00029         DATA SORT,CALC,PLOT,EXIT,YES,NO/'SORT ','CALC ','PLOT ','EXIT ',
00030        1 'YES  ','NO   '/
00031        10 WRITE (5,30)
00032        30 FORMAT (3(1H1/),X,'PARAMETERS FOR WHICH PROGRAM ',
00033           1 '(SORT,CALC,PLOT)?: ',$)
00034        40 READ (5,50) CHOICE
00035        50 FORMAT (A5)
00036        60 IF (CHOICE .EQ. YES) GO TO 10
00037           IF (CHOICE .EQ. NO) GO TO 9999
00038           IF (CHOICE .EQ. EXIT) GO TO 9999
00039           IF (CHOICE .EQ. SORT) CALL SORTIN(*70)
00040           IF (CHOICE .EQ. CALC) CALL CALCIN(*70,*80,*90,CHOICE)
00041           IF (CHOICE .EQ. PLOT) CALL PLOTIN(*70)
00042           CALL ILLEGAL(*40,CHOICE)
00043        70 WRITE (5,80)
00044        80 FORMAT (3(1H1/),X,'PARAMETERS FOR ANALYSIS PROGRAM: ',$)
00045           GO TO 40
00046        90 WRITE (5,100)
00047       100 FORMAT (3(1H1/),X,'"SORT " WILL DESTROY EXISTING FILE. BYPASS',
00048           1 ' CONFIRM: ',$)
00049           GO TO 40
00050      9999 STOP
00051           END
```

```
SORTIN  METINP.FOR      FORTRAN V.5A(621) /AI   10-JUL-81       16:26   PAGE 1

00001         SUBROUTINE SORTIN(*)
00002   C
00003   C     SUBROUTINE INPUTS 'METSRT' PARAMETERS
00004   C
00005         PARAMETER MLINE=28
00006         INTEGER SRTNUM(13),SRTSPN(2,MLINE)
00007         COMMON IWARAY(5,20),LINE(MLINE,15),ADDRAY(MLINE),FORCUT(MLINE,15)
00008         DIMENSION SRTPRT(3,13),SRTPRT(13),IFILE(3)
00009         DATA SRTPRT/'# OF CHANNELS  ','YEAR            ','JULIAN DATE    ',
00010        1 'DAY OF WEEK     ','YEAR            ','JULIAN DATE    ',
00011        2 'DAY OF WEEK     ','CHANNEL NUMBER  ','CHANNEL NAME   ',
00012        3 'INPUT UNITS     ','OUTPUT UNITS    ','OUTPUT FORMAT  ',
00013        4 'PRINT SWITCH    '/
00014         DATA SRTPRT/'I3','I4','A3','A3','I4','A3','A3','I3','2A4','A2',
00015        1 '2A4','3A4','I1'/
00016         DATA SRTSPN(1,1),SRTSPN(2,1),SRTSPN(1,2),SRTSPN(2,2),SRTSPN(1,3),
00017        1 SRTSPN(2,3)/1,1,2,4,5,7/
00018         DATA SRTNUM,TITLE/8*1,2,1,2,3,1,'SORT '/
00019         N = MLINE - 3
00020         DO 10 I=4,MLINE
00021         SRTSPN(1,I) = 4
00022         SRTSPN(2,I) = 13
00023        10 CONTINUE
00024         WRITE (5,20)
00025        20 FORMAT (3(1H1/),X'INITIALIZE METSRT PARAMETERS:'/)
00026   C
00027   C     INITIALIZE IWARAY
00028   C
00029         CALL INIT (1,13)
00030   C
00031   C     GET TITLE FOR OUTPUT
00032   C
00033         CALL PAGE(ITITLE)
00034   C
00035   C     WRITE KEYWORD AND PAGE TITLE
00036   C
00037         OPEN (UNIT=4,DEVICE='DSK',ACCESS='SEQOUT')
00038         WRITE (4,25) ITITLE
00039        25 FORMAT (X'SORT PARAMETERS'/4,9A5)
00040         CLOSE (UNIT=4,DEVICE='DSK',ACCESS='SEQOUT')
00041   C
00042   C     GET NUMBER OF CHANNELS
00043   C
00044        30 CALL TTYIN (SRTPRT,SRTPRT,SRTNUM,1,1)
00045   C
00046   C     WRITE ERROR MESSAGE IF TOO MANY CHANNELS WERE REQUESTED
00047   C
00048         IF (IWARAY(1,1) .LE. 8) GO TO 50
00049         WRITE (5,40) IWARAY(1,1),8
00050        40 FORMAT (X'NCHAN = ',I3,': EXCEEDS MAXIMUM ALLOWABLE VALUE (',I3,')'/
00051           1 X'PLEASE ENTER NEW VALUE')
00052         CALL INIT (1,1)
00053         GO TO 30
00054        50 CALL FILE (SRTPRT,SRTNUM,1,1,1)
00055         WRITE (5,60)
00056        60 FORMAT (/1X,'INITIAL DATE:'/)
```

```
SCSTIN  METINP.SOB      FORTRAN V.5A(621) /A1   16-JUL-81       16:26   PAGE 1-1


00057   C
00058   C     GET YEAR, DATE, AND DAY FOR INITIAL DATA SET
00059   C
00060         CALL TTYIN (SRTPRT,SRTPRT,SRTNUM,2,4)
00061         CALL FILE (SRTPRT,SRTNUM,2,2,4)
00062         WRITE (5,70)
00063      70 FORMAT (/1X,'FINAL DATE:'/)
00064   C
00065   C     GET YEAR, DATE, AND DAY FOR FINAL DATA SET
00066   C
00067         CALL TTYIN (SRTPRT,SRTPRT,SRTNUM,5,7)
00068         CALL FILE (SRTPRT,SRTNUM,3,5,7)
00069         WRITE (5,80)
00070      80 FORMAT (/1X'CHANNEL VARIABLES:'/)
00071         DO 90 I=1,INARAY(1,1)
00072   C
00073   C     GET PARAMETERS FOR EACH CHANNEL
00074   C
00075         CALL TTYIN (SRTPRT,SRTPRT,SRTNUM,6,13)
00076         CALL FILE (SRTPRT,SRTNUM,3*I,6,13)
00077   C
00078   C     REINITIALIZE INARAY
00079   C
00080         CALL INIT (8,13)
00081      90 CONTINUE
00082         N = INARAY(1,1) * 3
00083         OPEN (UNIT=4,DEVICE='DSK',ACCESS='ASSOCIATE')
00084         CALL CHECK (SRTPRT,SRTPRT,SRTNUM,SRTDAY,4,...,...)
00085         CLOSE (UNIT=4,DEVICE='DSK',ACCESS='ASSOCIATE')
00086         RETURN 1
00087         END
```

```
CALCIN  METINP.SOB      FORTRAN V.5A(621) /A1   16-JUL-81       16:26   PAGE 1


00001         SUBROUTINE CALCIN(*,*,*,CHOICE)
00002   C
00003   C     SUBROUTINE INPUTS 'SETCLC' PARAMETERS
00004   C
00005         INTEGER CICNUM(5)
00006         DIMENSION CLCPRT(3,5),CLCPRT(5),ITIT...(3)
00007         DATA CLCPRT/'CHANNEL NUMBER ','CHANNEL NAME  ',
00008        1 'OUTPUT UNITS   ','OUTPUT FORMAT  ','CALC SWITCH   '/
00009         DATA CLCPRT/'I3','2A4','2A4','3A4','I1'/
00010         DATA CLCNUM/1,2,2,3,1/
00011         DATA WIND,RHOIST,EXIT,YES,NNO,PLOT,SORT
00012        1 /'WIND ','RHOIST','EXIT','YES  ','NO   ','PLOT ','SORT '/
00013         CALL INIT(1,13)
00014   C
00015   C     GET TITLE FOR OUTPUT
00016   C
00017         CALL PAGE(ITITLE)
00018   C
00019   C     WRITE KEYWORD AND PAGE TITLE
00020   C
00021         OPEN (UNIT=4,DEVICE='DSK',ACCESS='ASSOCIATE')
00022         WRITE (4,5) ITITLE
00023       5 FORMAT (/X'CALC PARAMETERS'/X,5A5)
00024         CLOSE (UNIT=4,DEVICE='DSK',ACCESS='ASSOCIATE')
00025      10 WRITE (5,20)
00026      20 FORMAT (3(1H1/)X'PARAMETERS FOR WHICH CALCULATION',
00027        1 ' (WIND,RHOIST)?: '$)
00028      30 READ (5,40) CHOICE
00029      40 FORMAT (A5)
00030      45 IF (CHOICE .EQ. YES) GO TO 10
00031         IF (CHOICE .EQ. NNO) RETURN 1
00032         IF (CHOICE .EQ. EXIT) RETURN 2
00033         IF (CHOICE .EQ. WIND)
00034        1  CALL WNDIN(&50,&45,CLCPRT,CLCPRT,...,...)
00035         IF (CHOICE .EQ. RHOIST)
00036        1  CALL RSTIN(&50,&45,CLCPRT,CLCPRT,...,...)
00037         IF (CHOICE .EQ. PLOT) RETURN 2
00038         IF (CHOICE .EQ. SORT) RETURN 3
00039         CALL ILEGAL(&30,CHOICE)
00040      50 WRITE (5,60)
00041      60 FORMAT (3(1H1/)X,'PARAMETERS FOR ANOTHER CALCULATION: '$)
00042         GO TO 30
00043         END
```

```
WNDIN   SETINP.FOR      FORTRAN V.5A(621) /AI   1[illegible]-31        16:26   PAGE 1


00001         SUBROUTINE WNDIN(*,*,WNDFMT,WNDFNT,[illegible],CHOICE)
00002   C
00003   C     SUBROUTINE INPUTS 'WIND' PARAMETERS
00004   C
00005         PARAMETER NLINE=28
00006         INTEGER WNDNUM(5),WNDSPN(2,NLINE)
00007         COMMON IWARAY(5,20),LINE(NLINE,15),[illegible],PRNOUT(NLINE,15)
00008         DIMENSION WNDFMT(3,5),WNDFNT(5),HEAD1(3,3),HEAD2(2,2)
00009         DATA TITLE,HEAD1,HEAD2/'WIND ',' RELATIVE WIND ',
0000A        1 ' ABSOLUTE SHIP ',' ABSOLUTE WIND ','[illegible]  ','DIRECTION '/
00011         DO 5 I=1,NLINE
00012         WNDSPN(1,I) = 1
00013         WNDSPN(2,I) = 5
00014       5 CONTINUE
00015         OPEN (UNIT=4,DEVICE='DSK',ACCESS='A[illegible]')
00016         WRITE (4,10)
00017      10 FORMAT (/1X'WIND SUBROUTINE')
00018         WRITE (5,20)
00019      20 FORMAT (3(1H1/),1X'INITIALIZE ABSOLUTE [illegible] VELOCITY PARAMETERS:'/)
00020         L = 1
00021         DO 50 I=1,3
00022         DO 50 K=1,2
00023   C
00024   C     INITIALIZE IWARAY
00025   C
00026         CALL INIT (1,5)
00027         WRITE (5,40) (HEAD1(J,I),J=1,3),(HEAD2(J,I),J=1,2)
00028      40 FORMAT (5A5)
00029         CALL TTYIN(WNDFMT,WNDFNT,WNDNUM,1,5)
00030         CALL FILE (WNDFMT,WNDNUM,L,1,5)
00031         CALL FIX(L)
00032         L = L + 1
00033      50 CONTINUE
00034         CALL CHECK (WNDFMT,WNDFNT,WNDNUM,WN[illegible],5,[illegible])
00035         WRITE (4,60)
00036      60 FORMAT (/)
00037         CALL REPEAT(630,670,TITLE,CHOICE)
00038         CLOSE (UNIT=4,DEVICE='DSK',ACCESS='A[illegible]')
00039         RETURN 1
00040      70 RETURN 2
00041         END
```

```
HSTIN   HSTINP.FOR      FORTRAN V.5A(621) /AI   1[illegible]-31        16:26   PAGE 1


00001         SUBROUTINE HSTIN(*,*,HSTFMT,HSTFNT,[illegible],CHOICE)
00002   C
00003   C     SUBROUTINE INPUTS 'HOIST' PARAMETERS
00004   C
00005         PARAMETER NLINE=28
00006         DIMENSION HSTNUM(5),HEAD(3,4),HSTSPN(2,[illegible])
00007         COMMON IWARAY(5,20),LINE(NLINE,15),[illegible],PRNOUT(NLINE,15)
00008         REAL HSTFMT(3,5),HSTFNT(5)
00009         DATA TITLE,HEAD/'HOIST','AIR TEMPERATURE','AIR PRESSURE   ',
0000A        1 'REL. HUMIDITY  ','H2O VAPOR CONC.'/
00011         DO 5 I=1,NLINE
00012         HSTSPN(1,I) = 1
00013         HSTSPN(2,I) = 5
00014       5 CONTINUE
00015         OPEN (UNIT=4,DEVICE='DSK',ACCESS='A[illegible]')
00016         WRITE (4,10)
00017      10 FORMAT (/1X'HOIST SUBROUTINE')
00018         WRITE (5,20)
00019      20 FORMAT (3(1H1/),1X'INITIALIZE HOISTURE [illegible]:'/)
00020         DO 50 I=1,4
00021         L = I
00022   C
00023   C     INITIALIZE IWARAY
00024   C
00025         CALL INIT (1,5)
00026         WRITE (5,40) (HEAD(J,I),J=1,3)
00027      40 FORMAT (3A5)
00028         CALL TTYIN (HSTFMT,HSTFNT,HSTNUM,1,5)
00029         CALL FILE (HSTFMT,HSTNUM,L,1,5)
00030         CALL FIX(L)
00031      50 CONTINUE
00032         CALL CHECK (HSTFMT,HSTFNT,HSTNUM,HS[illegible],4,TITLE)
00033         WRITE (4,60)
00034      60 FORMAT (/)
00035         CALL REPEAT(630,670,TITLE,CHOICE)
00036         CLOSE (UNIT=4,DEVICE='DSK',ACCESS='A[illegible]')
00037         RETURN 1
00038      70 RETURN 2
00039         END
```

```
      SUBROUTINE PLOTIN(*)
C
C     SUBROUTINE INPUTS 'SETPLT' PARAMETERS
C
      PARAMETER BLINE=28
      INTEGER PLTNUM(17),PLTSPN(2,BLINE)
      COMMON IBARAY(5,20),LINE(BLINE,15),LABSIZ(BLINE),FSHCUT(BLINE,15)
      DIMENSION PLTFMT(3,17),PLTFNT(17),IFMT(3)
      DATA PLTFMT/'# OF DAYS/PAGE ','YEAR ','JULIAN DATE ',
     1 'DAY OF WEEK ','YEAR ','JULIAN DATE ',
     2 'DAY OF WEEK ','XAXIS OR YAXIS ','LABELED NUMBER ',
     3 'CHANNEL NAME ','OUTPUT UNITS ','MINIMUM VALUE ',
     4 'INCREMENT ','MAXIMUM VALUE ','THRESHOLD ',
     5 'HYSTERESIS ','AXIS TYPE '/
      DATA PLTFNT /'I2','I4','I3','A3','I4','I3','A3','A5','I3',
     1 '2A4','2A4',5*'F5.0','A5'/
      DATA PLTSPN(1,1),PLTSPN(2,1),PLTSPN(1,2),PLTSPN(2,2),PLTSPN(1,3),
     1 PLTSPN(2,3),PLTSPN(1,4),PLTSPN(2,4) /1,1,2,4,5,7,8,8/
      DATA PLTNUM,TITLE,IXAXIS,IYAXIS,IXAXIS,IYAXIS,IES,IAVEC,IPVEC
     1  /9*1,2*2,6*1,'PLOT','XAXIS','YAXIS','XAXIS','YAXIS','YES',
     2 'AVEC','PVEC'/
      DO 10 I=1,BLINE
      PLTSPN(1,I) = 9
      PLTSPN(2,I) = 17
   10 CONTINUE
      KVEC = 0
      WRITE (5,20)
   20 FORMAT (3(1H*)/'INITIALIZE SETPLT PARAMETERS:'/)
C
C     INITIALIZE IBARAY
C
      CALL INIT(1,17)
C
C     GET TITLE FOR OUTPUT
C
      CALL PAGE(ITITLE)
C
C     WRITE KEYWORD AND PAGE TITLE
C
      OPEN (UNIT=4,DEVICE='DSK',ACCESS='APPEND')
      WRITE (4,30) ITITLE
   30 FORMAT (1X'PLOT PARAMETERS'/4,5A5)
      CLOSE (UNIT=4,DEVICE='DSK',ACCESS='APPEND')
C
C     GET NUMBER OF DAYS PER PLOT SET
C
      CALL TTYIN(PLTFMT,PLTFNT,PLTNUM,1,1)
      CALL FILE(PLTFMT,PLTNUM,1,1,1)
      WRITE (5,40)
   40 FORMAT (/'INITIAL PLOT DATE:'/)
C
C     GET YEAR, DATE, AND DAY FOR START OF EACH PLOT SET
C
      CALL TTYIN(PLTFMT,PLTFNT,PLTNUM,2,4)
      CALL FILE(PLTFMT,PLTNUM,2,2,4)
      WRITE (5,50)
```



```
   50 FORMAT (/'FINAL PLOT DATE:'/)
C
C     GET YEAR,DATE, AND DAY FOR END OF LAST PLOT SET
C
      CALL TTYIN(PLTFMT,PLTFNT,PLTNUM,5,7)
      CALL FILE(PLTFMT,PLTNUM,3,5,7)
      L = 4
C
C     ASK FOR X-AXIS PARAMETERS
C
   60 CALL TTYIN(PLTFMT,PLTFNT,PLTNUM,8,8)
C
C     WRITE ERROR MESSAGE IF X-AXIS IS NOT CHOSEN
C
      IF (IBARAY(1,8) .EQ. IXAXIS) GO TO 80
      WRITE (5,70)
   70 FORMAT (1X'XAXIS MUST BE ENTERED FIRST'/)
      CALL INIT(8,8)
      GO TO 60
   80 CALL FILE(PLTFMT,PLTNUM,L,8,8)
      PLTSPN(1,L) = 8
      PLTSPN(2,L) = 8
      L = L + 1
   90 CALL TTYIN(PLTFMT,PLTFNT,PLTNUM,9,17)
      CALL FILE(PLTFMT,PLTNUM,L,9,17)
      L = L + 1
      CALL INIT(8,17)
  100 CALL TTYIN(PLTFMT,PLTFNT,PLTNUM,8,8)
      IF (IBARAY(1,8) .EQ. IYAXIS) GO TO 130
      IF (IBARAY(1,8) .EQ. IXAXIS) GO TO 120
      WRITE (5,110) IBARAY(1,8)
  110 FORMAT (1X'',A5,'' IS NOT UNDERSTOOD.'/1X'PLEASE TRY AGAIN.')
      GO TO 100
  120 L = L - 1
      GO TO 90
  130 CALL FILE(PLTFMT,PLTNUM,L,8,8)
      PLTSPN(1,L) = 8
      PLTSPN(2,L) = 8
      L = L + 1
  140 CALL TTYIN(PLTFMT,PLTFNT,PLTNUM,9,17)
      CALL FILE(PLTFMT,PLTNUM,L,9,17)
      IAXIS = IBARAY(1,17)
      L = L + 1
      CALL INIT(8,17)
      IF ((IAXIS .EQ. IAVEC) .OR. (IAXIS .EQ. IPVEC)) KVEC = KVEC + 1
      IF (KVEC .EQ. 1) GO TO 140
      KVEC = 0
      WRITE (5,150)
  150 FORMAT (1X'SPECIFICATIONS FOR ANOTHER AXIS: ')
      READ (5,160) CHOICE
  160 FORMAT (A5)
      IF (CHOICE .EQ. YES) GO TO 170
      IF (CHOICE .EQ. XAXIS) GO TO 100
      IF (CHOICE .EQ. YAXIS) GO TO 140
      L = L - 1
      OPEN (UNIT=4,DEVICE='DSK',ACCESS='APPEND')
```

```
PLOTIN  RETINP.FOR      FORTRAN V.5A(621) /KI   16-JUL-81       16:26   PAGE 1-2

00113          CALL CHECK(PLTPRT,PLTFRT,PLTHUB,PLF[illegible],1,1,1,4)
00114          CLOSE(UNIT=4,DEVICE='DSK',ACCESS='APPEND')
00115          RETURN 1
00116      170 CALL TTYIN(PLTPRT,PLTFRT,PLTHUB,A,B)
00117          IF (INARAY(1,B) .EQ. IYAXIS) GO TO 190
00118          IF (INARAY(1,B) .EQ. IXAXIS) GO TO 80
00119          WRITE (5,110) INARAY(1,B)
00120          GO TO 170
00121      180 INARAY(1,B) = IXAXIS
00122          GO TO 80
00123          END
```

```
PAGE    RETINP.FOR      FORTRAN V.5A(621) /KI   16-JUL-81       16:26   PAGE 1

00001          SUBROUTINE PAGE(ITITLE)
00002  C
00003  C       SUBROUTINE READS IN THE PAGE HEADING [illegible] THE OUTPUT
00004  C
00005          PARAMETER NLINE=28
00006          INTEGER PAGNUM
00007          COMMON INARAY(5,20),LINE(NLINE,15),[illegible],FBHOUT(NLINE,15)
00008          DIMENSION PAGPRT(3,1),ITITLE(5)
00009          DATA PAGPRT,PAGFRT,PAGNUM,STAR/'OUTPUT [illegible] ','SAVE',5,'*'/
00010      10  CALL TTYIN(PAGPRT,PAGFRT,PAGNUM,1,1)
00011          CALL FILE(PAGFRT,PAGNUM,1,1,1)
00012          WRITE (5,20)
00013      20  FORMAT (X'VERIFY OUTPUT PAGE TITLE [illegible] TO CONTINUE,'.
00014         1 '* TO EDIT)'/)
00015          WRITE (5,30) (LINE(1,I),I=1,LENGTH(1))
00016      30  FORMAT (X,5A5,$)
00017          READ (5,40) CHAR
00018      40  FORMAT (A1)
00019          IF (CHAR .EQ. STAR) GO TO 10
00020          DO 50 I=1,5
00021      50  ITITLE(I) = LINE(1,I)
00022          CALL INIT(1,1)
00023          WRITE (5,60)
00024      60  FORMAT (//)
00025          RETURN
00026          END
```

```
FIX     RETINP.FOR      FORTRAN V.5A(621) /KI   16-JUL-81       16:26   PAGE 1

00001          SUBROUTINE FIX(NLINE)
00002  C
00003  C       SUBROUTINE CHECKS CHANNEL NUMBER AND [illegible] SWITCH AND SETS THEM TO ZERO
00004  C       IF A BLANK WAS PRESENT
00005  C
00006          PARAMETER NLINE=28
00007          COMMON INARAY(5,20),LINE(NLINE,15),[illegible],FBHOUT(NLINE,15)
00008          DATA IBLANK/'  '/
00009          IF (LINE(NLINE,1) .EQ. IBLANK) LINE([illegible],1) = 0
00010          IF (LINE(NLINE,9) .EQ. IBLANK) LINE([illegible],9) = 0
00011          RETURN
00012          END
```

```
ILEGAL  RETINP.FOR      FORTRAN V.5A(621) /KI   16-JUL-81       16:26   PAGE 1

00001          SUBROUTINE ILEGAL(*,WORD)
00002  C
00003  C       SUBROUTINE WRITES AN 'ILLEGAL CHOICE' [illegible]
00004  C
00005          WRITE (5,10) WORD
00006      10  FORMAT (1X,'"',A5,'" IS NOT ALLOWED.'/1X,'PLEASE TRY AGAIN OR',
00007         1 ' TYPE "EXIT" TO QUIT: '$)
00008          RETURN 1
00009          END
```

```
REPEAT  RETINP.FOR      FORTRAN V.5A(621) /KI   16-JUL-81       16:26   PAGE 1

00001          SUBROUTINE REPEAT (*,*,WORD,CHOICE)
00002  C
00003  C       SUBROUTINE ASKS WHETHER THE PREVIOUS [illegible] INPUT SHOULD BE REPEATED
00004  C
00005          DATA EXIT,YES,RNO,PLOT,SCRT
00006         1 /'EXIT','YES ','NO  ','PLOT','SCRT'/
00007          WRITE (5,10) WORD
00008      10  FORMAT (//X'REPEAT PARAMETER INPUT FOR SUBROUTINE "',A5,'"?: '$)
00009      20  READ (5,30) CHOICE
00010      30  FORMAT (A5)
00011          IF (CHOICE .EQ. YES) GO TO 40
00012          IF (CHOICE .EQ. RNO) RETURN
00013          IF (CHOICE .EQ. EXIT) RETURN 2
00014          IF (CHOICE .EQ. PLOT) RETURN 2
00015          IF (CHOICE .EQ. SCRT) RETURN 2
00016          CALL ILEGAL (*20,CHOICE)
00017      40  WRITE (5,50)
00018      50  FORMAT (//)
00019          RETURN 1
00020          END
```

```
TTYIN   METINP.FOR      FORTRAN V.5A(621) /KI   16-Dec-81       16:26   PAGE 1

00001          SUBROUTINE TTYIN (PROMPT,FORMIN,NUM,NFIRST,NLAST)
00002   C
00003   C      SUBROUTINE WRITES SUBSCRIPTED MESSAGES FROM ARRAY 'PROMPT' AND
00004   C      READS RESPONSES FROM THE TTY ACCORDING TO 'FORMIN'
00005   C
00006          PARAMETER NLINE=20
00007          COMMON IWARAY(5,20),LINE(NLINE,15),...(NLINE),FRMOUT(NLINE,15)
00008          DIMENSION PROMPT(3,20),FORMIN(20),NUM(20)
00009          DIMENSION STRING(5),FORM(3)
00010          DATA STAR,BLEFT,PLUS,RIGHT/'*    ','(','+','1X',')'/
00011          DATA DOLLAR,BLANK,IBLANK/'$',' ',    '/
00012   C
00013   C      SET POINTER, # OF CHARACTERS, AND # OF WORDS
00014   C
00015          IPOINT = NFIRST
00016       10 NUMBER = 5 * NUM(IPOINT)
00017   C
00018   C      SET UP OUTPUT FORMAT
00019   C
00020          FORM(1) = PLUS
00021          FORM(2) = FORMIN(IPOINT)
00022          FORM(3) = DOLLAR
00023          WRITE (5,20) (PROMPT(I,IPOINT),I=1,3),...,FORM(2),RIGHT
00024       20 FORMAT (1X,3A5,A1,A3,A1,'*',$)
00025   C
00026   C      WRITE PREVIOUS VALUE, IF ANY.  OTHERWISE, GO TO INPUT.
00027   C
00028          EMPTY = 1
00029          DO 25 I=1,NUM(IPOINT)
00030       25 IF (IWARAY(I,IPOINT) .NE. IBLANK) EMPTY = 0
00031          IF (EMPTY .EQ. 1) GO TO 40
00032          WRITE (5,FORM) (IWARAY(I,IPOINT),I=1,...(IPOINT))
00033          WRITE (5,30)
00034       30 FORMAT ('+? ')
00035   C
00036   C      SET UP FORMAT TO DECODE DATA
00037   C
00038       40 FORM(3) = RIGHT
00039          FORM(1) = BLEFT
00040   C
00041   C      INPUT RESPONSE
00042   C
00043          READ (5,50) (STRING(I),I=1,5)
00044       50 FORMAT (5A5)
00045   C
00046   C      TEST FIRST CHARACTER FOR '*'
00047   C
00048          DECODE (1,60,STRING) CHAR
00049       60 FORMAT (A1)
00050          IF (CHAR .NE. STAR) GO TO 70
00051   C
00052   C      CHARACTER IS A '*', THEREFORE BACK UP TO PREVIOUS INPUT & REPEAT
00053   C
00054          IPOINT = IPOINT - 1
00055          IF (IPOINT .LT. NFIRST) IPOINT = NFIRST
00056          GO TO 10
```

```
TTYIN   METINP.FOR      FORTRAN V.5A(621) /KI   16-Dec-81       16:26   PAGE 1-1

00057   C
00058   C      CHARACTER IS NOT '*', SO DECODE VALUE FOR CURRENT VARIABLE
00059   C
00060       70 DECODE (5,65,STRING) CHAR
00061       65 FORMAT (A5)
00062          IF (CHAR .EQ. BLANK) GO TO 75
00063          DECODE (NUMBER,FORM,STRING) (IWARAY(...),I=1,NUM(IPOINT))
00064       75 IF (IPOINT .GE. NLAST) GO TO 80
00065          IPOINT = IPOINT + 1
00066          GO TO 10
00067       80 WRITE (5,90)
00068       90 FORMAT (/)
00069          RETURN
00070          END
```

```
FILE    METINP.FOR      FORTRAN V.5A(621) /KI   16-Dec-81       16:26   PAGE 1

00001          SUBROUTINE FILE (FORMIN,NUM,NLINE,NFIRST,NLAST)
00002   C
00003   C      SUBROUTINE COMPOSES A LINE AND THE CORRESPONDING FORMAT FOR THE OUTPUT FILE
00004   C
00005          PARAMETER NLINE=20
00006          COMMON IWARAY(5,20),LINE(NLINE,15),...(NLINE),FRMOUT(FLINE,15)
00007          DIMENSION FORMIN(20),NUM(20)
00008          I = 1
00009          J = 1
00010          DO 20 IPOINT=NFIRST,NLAST
00011          DO 10 IWORD=1,NUM(IPOINT)
00012          LINE(NLINE,I) = IWARAY(IWORD,IPOINT)
00013          I = I + 1
00014       10 CONTINUE
00015          FRMOUT(NLINE,J) = FORMIN(IPOINT)
00016          J = J + 1
00017       20 CONTINUE
00018          LENGTH(NLINE) = I - 1
00019          RETURN
00020          END
```

```
CHECK   RETINP.FOR      FORTRAN V.5A(021) /AI   16-...-81        16:26   PAGE 1

00001         SUBROUTINE CHECK (PROMPT,FORMIN,NUM,...,...,NORD)
00002   C
00003   C     SUBROUTINE WRITES OUTPUT ON TTY LINE M ...S AND WRITES VERIFIED LINES TO THE DISK (FOR04.DAT)
00004   C
00005         PARAMETER NLINE=28
00006         COMMON INARAY(5,20),LINE(NLINE,15),...(NLINE),FRMOUT(NLINE,15)
00007         DIMENSION PROMPT(3,20),FORMIN(20),NUM(20)
00008         DIMENSION OUT(22),ISPAN(2,NLINE)
00009         DATA STAR,OUT(1),X,BIGNT,DOLLAR/'*    ','(14,',',1X,',')','$'/
00010         WRITE (5,5)
00011       5 FORMAT (1X,'VERIFY FILE (TYPE RETURN TO PROCEED, * TO EDIT)')
00012         WRITE (5,10) NORD
00013      10 FORMAT (1X,A5,' PARAMETERS')
00014         DO 40 NLINE=1,LAST
00015         LINNUM = NLINE
00016         M = LENGTH(NLINE)
00017         L = 1
00018         N = 2
00019      15 IF (FRMOUT(NLINE,L) .EQ. 0) GO TO 20
00020         OUT(N) = FRMOUT(NLINE,L)
00021         OUT(N+1) = X
00022         L = L + 1
00023         N = N + 2
00024         GO TO 15
00025      20 OUT(N) = DOLLAR
00026      25 WRITE (5,OUT) (LINE(NLINE,L),L=1,N)
00027         READ (5,30) CHAR
00028      30 FORMAT (A1)
00029         IF (CHAR .EQ. STAR) CALL EDIT (PROMPT,FORMIN,NUM,LINBUF,ISPAN,625)
00030         OUT(N) = BIGNT
00031         WRITE (4,OUT) (LINE(NLINE,L),L=1,N)
00032      40 CONTINUE
00033         RETURN
00034         END
```

```
EDIT    RETINP.FOR      FORTRAN V.5A(021) /AI   16-...-81        16:26   PAGE 1

00001         SUBROUTINE EDIT (PROMPT,FORMIN,NUM,...,...)
00002   C
00003   C     SUBROUTINE ALLOWS CORRECTION OF ERRORS ... THE PARAMETER FILE
00004   C
00005         PARAMETER NLINE=28
00006         COMMON INARAY(5,20),LINE(NLINE,15),...(NLINE),FRMOUT(NLINE,15)
00007         DIMENSION PROMPT(3,20),FORMIN(20),NUM(20),ISPAN(2,NLINE)
00008         WRITE (5,10) N
00009      10 FORMAT (1H0,'EDIT LINE # ',I2,':')
00010         J = ISPAN(1,N)
00011         K = ISPAN(2,N)
00012      20 CALL RESTOR (NUM,N,J,K)
00013         CALL TTYIN (PROMPT,FORMIN,NUM,J,K)
00014         CALL FILE (FORMIN,NUM,N,J,K)
00015         RETURN 1
00016         END
```

```
INIT    RETINP.FOR      FORTRAN V.5A(021) /AI   16-...-81        16:26   PAGE 1

00001         SUBROUTINE INIT (NFIRST,NLAST)
00002   C
00003   C     SUBROUTINE INITIALIZES INPUT ARRAY: ...(I,J) = '     '
00004   C
00005         PARAMETER NLINE=28
00006         COMMON INARAY(5,20),LINE(NLINE,15),...(NLINE),FRMOUT(NLINE,15)
00007         DATA IBLANK/'     '/
00008         DO 10 J=NFIRST,NLAST
00009         DO 10 I=1,5
00010         INARAY(I,J) = IBLANK
00011      10 CONTINUE
00012         RETURN
00013         END
```

```
RESTOR  RETINP.FOR      FORTRAN V.5A(021) /AI   16-...-81        16:26   PAGE 1

00001         SUBROUTINE RESTOR (NUM,NLINE,NFIRST,...)
00002   C
00003   C     SUBROUTINE RESTORES VALUES TO INARAY
00004   C
00005         PARAMETER NLINE=28
00006         COMMON INARAY(5,20),LINE(NLINE,15),...(NLINE),FRMOUT(NLINE,15)
00007         DIMENSION NUM(20)
00008         I = 1
00009         DO 10 IPOINT=NFIRST,NLAST
00010         DO 10 INORD=1,NUM(IPOINT)
00011         INARAY(INORD,IPOINT) = LINE(NLINE,I)
00012         I = I + 1
00013      10 CONTINUE
00014         RETURN
00015         END
```

Appendix E
METINP Terminal Session
Creation of a Parameter File

```
PARAMETERS FOR WHICH PROGRAM (SORT,CALC,PLOT)?: SORT

INITIALIZE SORT PARAMETERS:

OUTPUT TITLE    (54S)=MAYES 1981 CRUISE

VERIFY OUTPUT PAGE TITLE (RETURN TO CONTINUE,* TO EDIT)

MAYES 1981 CRUISE


# OF CHANNELS  (I3 )= 19


INITIAL DATE:

YEAR           (I4 )=1981
JULIAN DATE    (I3 )= 29
DAY OF WEEK    (A3 )=19/91/THU


FINAL DATE:

YEAR           (I4 )=1981
JULIAN DATE    (I3 )= 29
DAY OF WEEK    (A3 )=THU


CHANNEL VARIABLES:

CHANNEL NUMBER (I3 )= 40
CHANNEL NAME   (2A4)=OUT.TEMP
INPUT UNITS    (A2 )= C
OUTPUT UNITS   (2A4)= DEG C
OUTPUT FORMAT  (3A4)=2X,F6.1,2X,
PRINT SWITCH   (I1 )=1

CHANNEL NUMBER (I3 )= 27
CHANNEL NAME   (2A4)=PWR.SUP.
INPUT UNITS    (A2 )= V
OUTPUT UNITS   (2A4)= VOLTS
OUTPUT FORMAT  (3A4)=2X,F7.3,1X,
PRINT SWITCH   (I1 )=0

CHANNEL NUMBER (I3 )= 26
CHANNEL NAME   (2A4)=POS.PWR.
INPUT UNITS    (A2 )= V
OUTPUT UNITS   (2A4)= VOLTS
OUTPUT FORMAT  (3A4)=2X,F7.3,2/2/1X,
PRINT SWITCH   (I1 )=0

CHANNEL NUMBER (I3 )= 22
CHANNEL NAME   (2A4)=OUT.TEMP

INPUT UNITS    (A2 )= V
OUTPUT UNITS   (2A4)= DEG C
OUTPUT FORMAT  (3A4)=2X,F6.1,2X,
PRINT SWITCH   (I1 )=1

CHANNEL NUMBER (I3 )= 21
CHANNEL NAME   (2A4)=REL. / /HUM.
INPUT UNITS    (A2 )= V
OUTPUT UNITS   (2A4)=PER CENT
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,
PRINT SWITCH   (I1 )=0

CHANNEL NUMBER (I3 )= 20
CHANNEL NAME   (2A4)=PRESSURE
INPUT UNITS    (A2 )=MV
OUTPUT UNITS   (2A4)= TORR
OUTPUT FORMAT  (3A4)=2X,F6.1,2X,
PRINT SWITCH   (I1 )=0

CHANNEL NUMBER (I3 )= 15
CHANNEL NAME   (2A4)=SHP.HEAD
INPUT UNITS    (A2 )=MV
OUTPUT UNITS   (2A4)= DEGREES
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,
PRINT SWITCH   (I1 )=1

CHANNEL NUMBER (I3 )= 14
CHANNEL NAME   (2A4)=SHP.SPD.
INPUT UNITS    (A2 )=MV
OUTPUT UNITS   (2A4)= KNOTS
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,
PRINT SWITCH   (I1 )=1

CHANNEL NUMBER (I3 )= 13
CHANNEL NAME   (2A4)=WND.DIR.
INPUT UNITS    (A2 )= V
OUTPUT UNITS   (2A4)= DEGREES
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,
PRINT SWITCH   (I1 )=1

CHANNEL NUMBER (I3 )= 12
CHANNEL NAME   (2A4)=WND.SPD.
INPUT UNITS    (A2 )= V
OUTPUT UNITS   (2A4)= KNOTS
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,
PRINT SWITCH   (I1 )=1

VERIFY FILE (TYPE RETURN TO PROCEED, * TO EDIT)

SORT  PARAMETERS
 10
1981  29 THU
1981  29 THU
40 OUT.TEMP  C  DEG C     2X,F6.1,2X,   1
27 PWR.SUP.  V  VOLTS     2X,F7.3,1X,   0
26 POS.PWR.  V  VOLTS     2X,F7.3,1X,   0
22 OUT.TEMP  V  DEG C     2X,F6.1,2X,   1
21 REL.HUM.  V  PER CENT  2X,F6.0,2X,   0
20 PRESSURE MV  TORR      2X,F6.1,2X,   0
15 SHP.HEAD MV  DEGREES   2X,F6.0,2X,   1
14 SHP.SPD. MV  KNOTS     2X,F6.0,2X,   1
13 WND.DIR.  V  DEGREES   2X,F6.0,2X,   1
12 WND.SPD.  V  KNOTS     2X,F6.0,2X,   1
```

```
PARAMETERS FOR ANOTHER PROGRAM?: YES_


PARAMETERS FOR WHICH PROGRAM (SORT,CALC,PLOT)?: xxxx_
OUTPUT TITLE   (5A5)=_

VERIFY OUTPUT PAGE TITLE (RETURN TO CONTINUE,* TO EDIT)

-


PARAMETERS FOR WHICH FUNCTION (WIND,ASSIST)?: WIND_
```

```
INITIALIZE ABSOLUTE WIND VELOCITY PARAMETERS:

RELATIVE WIND SPEED
CHANNEL NUMBER (I3 )= 12_
CHANNEL NAME   (2A4)=REL.SPD._
OUTPUT UNITS   (2A4)= KNOTS_
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,_
PRINT SWITCH   (I1 )=0_

RELATIVE WIND DIRECTION
CHANNEL NUMBER (I3 )= 13_
CHANNEL NAME   (2A4)=REL.DIR._
OUTPUT UNITS   (2A4)= DEGREES_
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,_
PRINT SWITCH   (I1 )=0_

ABSOLUTE SHIP SPEED
CHANNEL NUMBER (I3 )= 14_
CHANNEL NAME   (2A4)=SHP.SPD._
OUTPUT UNITS   (2A4)= KNOTS_
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,_
PRINT SWITCH   (I1 )=0_

ABSOLUTE SHIP DIRECTION
CHANNEL NUMBER (I3 )= 15_
CHANNEL NAME   (2A4)=SHP.HEAD_
OUTPUT UNITS   (2A4)= DEGREES_
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,_
PRINT SWITCH   (I1 )=0_

ABSOLUTE WIND SPEED
CHANNEL NUMBER (I3 )=100_
CHANNEL NAME   (2A4)=ABS.SPD._
OUTPUT UNITS   (2A4)= KNOTS_
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,_
PRINT SWITCH   (I1 )=1_

ABSOLUTE WIND DIRECTION
CHANNEL NUMBER (I3 )=101_
CHANNEL NAME   (2A4)=ABS.DIR._
OUTPUT UNITS   (2A4)= DEGREES_
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,_
PRINT SWITCH   (I1 )=1_

VERIFY FILE (TYPE RETURN TO PROCEED, * TO EDIT)

WIND  PARAMETERS
 12 REL.SPD.  KNOTS   2X,F6.0,2X,  0 _
 13 REL.DIR.  DEGREES 2X,F6.0,2X,  0 _
 14 SHP.SPD.  KNOTS   2X,F6.0,2X,  0 _
 15 SHP.HEAD  DEGREES 2X,F6.0,2X,  0 _
100 ABS.SPD.  KNOTS   2X,F6.0,2X,  1 _
101 ABS.DIR.  DEGREES 2X,F6.0,2X,  1 _
```

```
REPEAT PARAMETER INPUT FOR SUBROUTINE "WIND "?: NO

PARAMETERS FOR ANOTHER FUNCTION?: NO(ST
```

```
INITIALIZE MOISTURE PARAMETERS:

AIR TEMPERATURE
CHANNEL NUMBER (I3 )= 40
CHANNEL NAME   (2A4)=OUT.TEMP
OUTPUT UNITS   (2A4)= DEG C
OUTPUT FORMAT  (3A4)=2X,F6.1,2X
PRINT SWITCH   (I1 )=0

AIR PRESSURE
CHANNEL NUMBER (I3 )= 20
CHANNEL NAME   (2A4)=PRESSURE
OUTPUT UNITS   (2A4)= TORR
OUTPUT FORMAT  (3A4)=2X,F6.1,2X
PRINT SWITCH   (I1 )=0

REL. HUMIDITY
CHANNEL NUMBER (I3 )= 21
CHANNEL NAME   (2A4)=REL.HUM.
OUTPUT UNITS   (2A4)=PER CENT
OUTPUT FORMAT  (3A4)=2X,F6.0,2X
PRINT SWITCH   (I1 )=0

H2O VAPOR CONC.
CHANNEL NUMBER (I3 )=102
CHANNEL NAME   (2A4)=H2O VAP
OUTPUT UNITS   (2A4)=  PPMV
OUTPUT FORMAT  (3A4)=2X,F6.0,2X
PRINT SWITCH   (I1 )=1

VERIFY FILE (TYPE RETURN TO PROCEED, * TO EDIT)

MOIST PARAMETERS
 40 OUT.TEMP  DEG C   2X,F6.1,2X,  0
 20 PRESSURE  TORR    2X,F6.1,2X,  0
 21 REL.HUM.  PER CENT 2X,F6.0,2X,  0
102 H2O VAP    PPMV   2X,F6.0,2X,  1

REPEAT PARAMETER INPUT FOR SUBROUTINE "MOIST"?: NO

AIR TEMPERATURE
CHANNEL NUMBER (I3 )= 22
CHANNEL NAME   (2A4)=OUT.TEMP
OUTPUT UNITS   (2A4)= DEG C
OUTPUT FORMAT  (3A4)=2X,F6.1,2X
PRINT SWITCH   (I1 )=0

AIR PRESSURE
CHANNEL NUMBER (I3 )= 20
CHANNEL NAME   (2A4)=PRESSURE
OUTPUT UNITS   (2A4)=  TORR
OUTPUT FORMAT  (3A4)=2X,F6.1,2X
PRINT SWITCH   (I1 )=1
```

```
REL. HUMIDITY
CHANNEL NUMBER  (I3 )=_21_
CHANNEL NAME    (2A4)=REL.HUM._
OUTPUT UNITS    (2A4)=PER CENT_
OUTPUT FORMAT   (3A4)=2X,F6.0,2A._
PRINT SWITCH    (I1 )=1_

H2C VAPOR CONC.
CHANNEL NUMBER  (I3 )=_
CHANNEL NAME    (2A4)=_
OUTPUT UNITS    (2A4)=_
OUTPUT FORMAT   (3A4)=_
PRINT SWITCH    (I1 )=1_

VERIFY FILE (TYPE RETURN TO PROCEED, * TO EDIT)

HCIST PARAMETERS
 22 OUT.TEMP  DEG C   2X,F6.1,2X,  0 _
 20 PRESSURE  TORR    2X,F6.1,2X,  1 _
 21 REL.HUM.  PER CENT 2X,F6.0,2X, 1 _
  0                                1 _

REPEAT PARAMETER INPUT FOR SUBROUTINE "HOIST"?: NO_

PARAMETERS FOR ANOTHER FUNCTION?: NO_
```

```
INITIALIZE HEIGHT PARAMETERS:

OUTPUT TITLE    (5A5)=_

VERIFY OUTPUT PAGE TITLE (RETURN TO CONTINUE,* TO EDIT)

# OF DAYS/PAGE  (I2 )=_1_

INITIAL PLOT DATE:

YEAR            (I4 )=1981_
JULIAN DATE     (I3 )=_29_
DAY OF WEEK     (A3 )=THU_

FINAL PLOT DATE:

YEAR            (I4 )=1981_
JULIAN DATE     (I3 )=_29_
DAY OF WEEK     (A3 )=THU_

XAXIS OR YAXIS  (A5 )=XAXIS_

XAXIS MUST BE ENTERED FIRST

XAXIS OR YAXIS  (A5 )=XAXIS_

CHANNEL NUMBER  (I3 )=W_
CHANNEL NAME    (2A4)=_
OUTPUT UNITS    (2A4)=_
MINIMUM VALUE   (F5.)=0._
INCREMENT       (F5.)=0._
MAXIMUM VALUE   (F5.)=0._
THRESHOLD       (F5.)=0._
HYSTERESIS      (F5.)=100._
AXIS TYPE       (A5 )=LIN_

XAXIS OR YAXIS  (A5 )=YAXIS_

CHANNEL NUMBER  (I3 )=300_
CHANNEL NAME    (2A4)=WND.VEL._
OUTPUT UNITS    (2A4)=_KNOTS_
MINIMUM VALUE   (F5.)=0._
INCREMENT       (F5.)=5._
MAXIMUM VALUE   (F5.)=40._
THRESHOLD       (F5.)=2._
HYSTERESIS      (F5.)=-1._
AXIS TYPE       (A5 )=MIXC_
```

```
CHANNEL NUMBER  (I3 )=101
CHANNEL NAME    (2A4)=ABS.DIR.
OUTPUT UNITS    (2A4)= DEGREES
MINIMUM VALUE   (F5.)=0.
INCREMENT       (F5.)=45.
MAXIMUM VALUE   (F5.)=360.
THRESHOLD       (F5.)=0.
HYSTERESIS      (F5.)=10.
AXIS TYPE       (A5 )=RTHT

SPECIFICATIONS FOR ANOTHER AXIS?: XAXIS
CHANNEL NUMBER  (I3 )=0
CHANNEL NAME    (2A4)=
OUTPUT UNITS    (2A4)=
MINIMUM VALUE   (F5.)=0.
INCREMENT       (F5.)=0.
MAXIMUM VALUE   (F5.)=0.
THRESHOLD       (F5.)=0.
HYSTERESIS      (F5.)=0.
AXIS TYPE       (A5 )=LIN

XAXIS OR YAXIS (A5 )=XAXIS

CHANNEL NUMBER  (I3 )=300
CHANNEL NAME    (2A4)=ABS.SPD.
OUTPUT UNITS    (2A4)=KNOTS
MINIMUM VALUE   (F5.)=0.
INCREMENT       (F5.)=5.
MAXIMUM VALUE   (F5.)=99.
THRESHOLD       (F5.)=0.
HYSTERESIS      (F5.)=-1.
AXIS TYPE       (A5 )=LIN

SPECIFICATIONS FOR ANOTHER AXIS?: YAXIS
CHANNEL NUMBER  (I3 )=101
CHANNEL NAME    (2A4)=ABS.DIR.
OUTPUT UNITS    ( 4)=
CHANNEL NAME       )=ABS,DIR.? ABS.DIR.
OUTPUT UNITS       = DEGREES
MINIMUM VALUE   F5.)=0.
INCREMENT       F5.)=45.
MAXIMUM VALUE   (
THRESHOLD       (
HYSTERESIS
AXIS TYPE       5

SPECIFICATIONS FOR ANOTHER AXIS?: NO
VERIFY FILE (TYPE RETURN TO PROCEED, * TO EDIT)

PLOT  PARAMETERS
 1
1981  29 THU
1981  29 THU
XAXIS
  0                    0.    0.    0.    0.   360.  ...
YAXIS
100 WND.VEC.  KNOTS      0.    5.   40.    2.    -1. ...
101 ABS.DIR.  DEGREES    0.   45.  360.    0.   360. ...
XAXIS
  0                    0.    0.    0.    0.     0. ...
YAXIS
100 ABS.SPD.  KNOTS      0.    5.   40.    0.    -1. ...
101 ABS.DIR.  DEGREES    0.   45.  360.    0.    -1. ...
```

Appendix F
METSRT Error Messages

Several references to the error detection capabilities of METSRT have previously been made. In this section, these capabilities will be discussed in more detail and some examples will be give.

Five different classes of errors may be distinguished, as follows:

1) Failure to find some piece of required information.
2) Errors occurring during data input.
3) Conditions which were detected and flagged by the data logger.
4) Inability to properly process some type of data.
5) Errors arising from the processing of a specific piece of data.

Errors of the first type are detected in the main program when an attempt to OPEN a file fails or when the keyword "SORT" cannot be found in the parameter file. In all cases, execution is immediately aborted, and the reason is printed out.

Subroutines SEARCH and DATAIN look for errors of the second type, which include such things as read errors, duplicate channel numbers, and incorrectly formatted data records. Each such problem is listed and, insofar as possible, analysis of the remaining data continues.

Once a record has been read and found to be valid, it is checked by FLAG to determine if any of the data logger warning flags were set. Broken or overloaded sensors are detected at this point and these data are ignored. If the data logger's preset upper or lower limits were exceeded, this will also be be detected but, in these cases, the data is not rejected. Finally, if an illegal character appears in one of the positions reserved for data logger flags, this fact will be reported.

In general, each type of data will require a special function subprogram to perform the appropriate calculations and conversion of units. MANIP uses a computed GO TO statement to select the correct function for each channel. In the event that the channel number lies outside the range of the GO TO, or if no function has been defined at the specified statement label, then an error message will be printed. The data will be left in its original (input) form.

During processing of a specific data record, various errors may occur. For the most part, it is left up to the individual function subprograms to make any tests which may be required. For example, the values of various power supply voltages are tested and messages printed if they lie outside the specified ranges.

One general requirement of all data, regardless of type, is that it must have the correct units (those expected by the corresponding subprogram). For each channel, the proper units are declared in the parameter file at input time. DIDDLE compares the actual units with the expected units and lists all

discrepancies.

Most of the error messages presently incorporated in METSRT are illustrated in the following three tables. Table F1 shows the complete error listing as it appears in FORØ6.DAT. In Table F2, those messages produced during data input are related to the errors in the data file which caused them. Table F3 similarly correlates processing time error messages with the erroneous data records.

In all of these examples, the parameter and data files were specifically constructed to exercise the maximum possible number of error trapping routines. Not illustrated are those (such as "NO DATA FILE FOUND") which are mutually exclusive.

Table F1
METSRT Error Messages

```
                                                    ..... MESSAGE DEMO.
                                                          #A*NING!


1979 307 2200 CHANNEL 18 : LOWER LIMIT EXCEEDED.
1979 307 2200 CHANNEL 22 : SENSOR OVERLOAD
1979 307 2200 CHANNEL 23 : UNEXPECTED CHARACTER : ....
1979 307 2200 CHANNEL 24 : SENSOR BROKEN
INCOMPREHENSIBLE DATA AT 2200 HOURS ON DAY 307,1979:
A BAD LINE
SYSTEM RESET AFTER 2200 HOURS ON DAY 307,1979. DATA SET(S) SKIPPED.
ACQUISITION RESUMED AT 2230 HOURS ON DAY 307,1979
ILLEGAL RECORD AT 2230 HOURS ON DAY 307,1979:
//////
1979 307 2230 CHANNEL 17 : UNEXPECTED CHARACTER : ..... = "
END OF DATA SET FOR 2230 HOURS ON DAY 307,1979 :   3 CHANNELS READ
ERROR DURING SEARCH: 1979 307 2230
10011979
1979 307 2300 CHANNEL 22 : SENSOR IN RANGE
1979 307 2300 CHANNEL 24 : SENSOR IS OK
1979 307 2300 CHANNEL 23 : UPPER LIMIT EXCEEDED.
DUPLICATE DATA FOUND FOR CHANNEL 24 AT 2300 HOURS ON DAY 307,1979:
A 24 -0.3100 V
ECF CUEING SEARCH: 1979 307 2300
UNSPECIFIED PROCESS FOR CHANNEL 16 (ITERATION =  1)
1979 307 2200 CHANNEL 18 : HUMIDITY OUT OF RANGE (  ....); CORRECTED TO   0.
1979 307 2200 CHANNEL 21 : HUMIDITY OUT OF RANGE ( 1....); CORRECTED TO 100.
1979 307 2200 CHANNEL 23 : UNEXPECTED UNITS (MV) FOUND
1979 307 2230 CHANNEL 23 : CORRECT UNITS ( V) FOUND
1979 307 2230 CHANNEL 25 : M.D. VOLTAGE OUT OF RANGE ( 5.0270 V)
1979 307 2230 CHANNEL 26 : POS. PWR. SUPPLY OUT OF RANGE ( 44.198 V)
1979 307 2300 CHANNEL 27 : NEG. PWR. SUPPLY OUT OF RANGE (  0.000 V)
ILLEGAL CHANNEL NUMBER 55 (ITERATION = 13)
```

Table F2
METSRT Input Errors Correlated with Error Messages

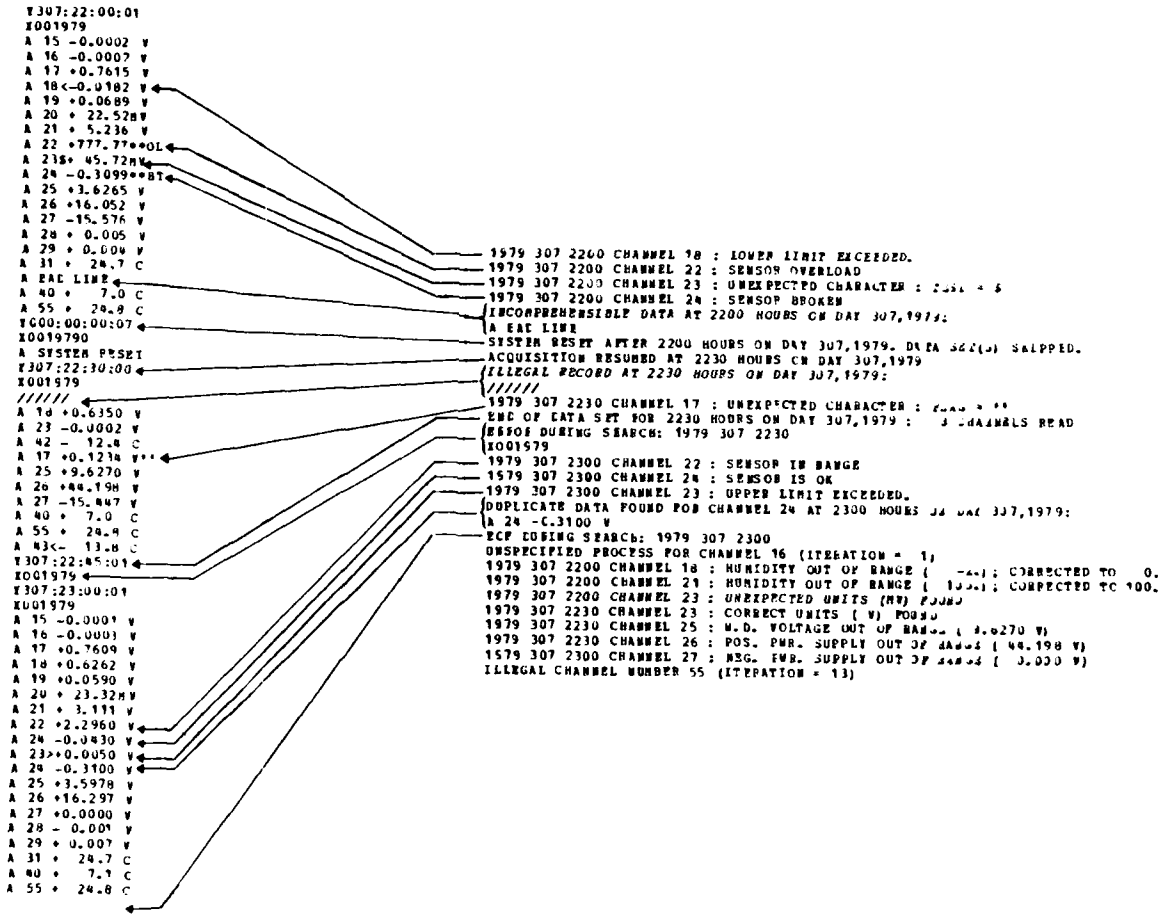
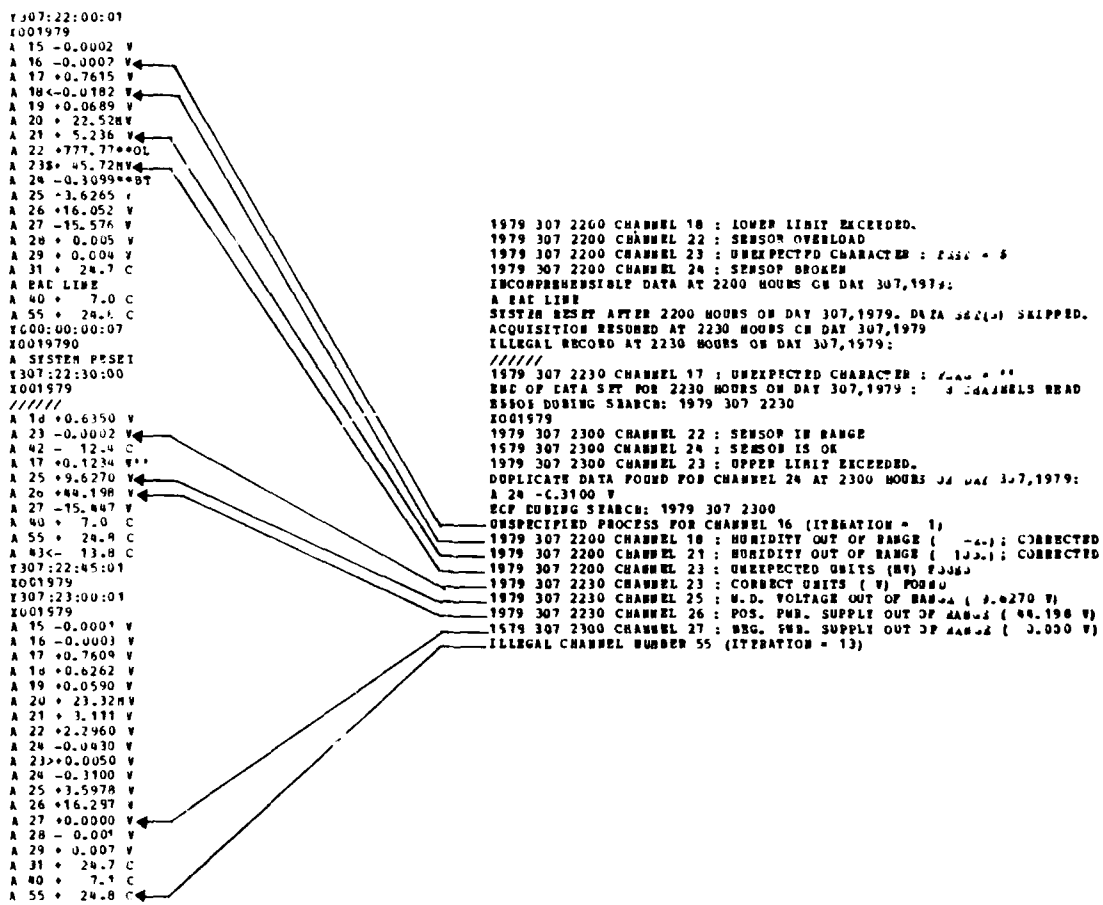


Table F3

METSRT Processing Time Errors and the Corresponding Warning Messages



Appendix G
Non-Standard FORTRAN

An attempt has been made to restrict the statements used in these programs to the set defined by 1966 ANSI standard FORTRAN. However, in several places non-standard statements have been used. Often this was done because the desired function was sufficiently complex that no simple alternative was available. In other cases, the non-standard statements were considered to involve relatively trivial functions which could easily be deleted by other users without detriment to the overall program function.

In this appendix we briefly discuss these non-standard features and suggest possible alternatives for some of them.

I. PROGRAM 'name'

This statement assigns a name to the main program just as FUNCTION or SUBROUTINE are used to designate subprograms. It may be omitted without affecting any program functions.

II. PARAMETER M=n

PARAMETER M = n assigns, at compile time, the value 'n' to the constant 'M'. In the METEOR package, PARAMETER statements are used to set MDATA (the maximum allowable number of data sets) and MCHAN (maximum possible number of data channels). These two constants are then used to dimension many of the arrays in both the main programs and in the subprograms. A PARAMETER statement must appear in each subprogram in which MDATA or MCHAN are to be used.

If the PARAMETER statements are omitted, then each occurrence of MDATA and MCHAN must be replaced by explicit values.

III. OPEN/CLOSE

These statements control the characteristics of the files used for input and output. The following arguments may be used with OPEN or CLOSE statements:

1)	UNIT	= n	Defines the logical unit number.
2)	DEVICE	= 'DSK'	Specifies that the device is a disk.
3)	ACCESS	= 'SEQOUT'	Initializes device for write.
		= 'SEQIN'	Sets device for read.
		= 'APPEND'	Sets device for write but does not initialize. New data will be added to the end of the existing file.
4)	DISPOSE	= 'DELETE'	Delete file after it is closed.
		= 'SAVE'	Save file after close. This is the default.
5)	FILE	= 'filename'	Allows new files to be named.

		The default name is FORØn.DAT, where 'n' is the logical unit number.
6)	ERR = s	Causes a branch to statement number 's' if an I/O error occurs.

In many systems, the functions of the OPEN and CLOSE statements may be performed by job control commands external to the program. However, the error recovery function may not be available in these cases.

METINP closes and reopens file FORØ4.DAT at several points. These statements could be eliminated and the file allowed to remain open continuously during program execution.

In METCLC, subroutine DATOUT closes FORØ7.DAT for input, reopens it for output, and rewrites the entire file (with modifications). The equivalent effect might be achieved by defining a new logical unit to receive this output [change the WRITE (7,f) statements to WRITE (8,f), for example], deleting the old file 7, and renaming the new file (file 8, in this example).

IV. STOP 'string'

This statement causes the message 'string' to be written to the default device (TTY for interactive jobs, LOG file for batch jobs) at the time that the STOP is encountered. These statements serve little purpose in batch jobs (in most cases the same message is available in FORØ6.DAT) but have proven to be convenient in debugging from a terminal. They may be replaced by standard STOPs.

V. RETURN n

This statement allows subroutines to return to any point in the calling program. Any subroutine that uses this feature must have one or more '&s' arguments (where 's' is a statement number) in the CALL and corresponding dummy arguments ,'*', in the SUBROUTINE statement. A RETURN n will then return to the statement number represented by the n^{th} asterisk (counting from the left).

A substitute for this function might involve setting the value of a variable within the subroutine and then using a computed GO TO in the calling program to branch to the desired statement number.

VI. END = s/ERR = s

The END = s feature is used as part of the READ statement to direct the program to statement 's' if an End of File (EOF) is read. The format of the statement is [READ (n,f,END = s) 'list']. Since this is only used to enable the program to print out an appropriate message before termination of the program it is not really necessary and may be omitted.

If it is desirable to retain this feature, the function could possibly be

simulated by placing some standard character in the last record of each file. The input may then be tested for the presence of this character and an EOF routine called if it is found. This procedure might be implemented by doing a READ with an A-format, then using a BACKSPACE and another READ (with a different format) if the special character was not found.

ERR = s is also used in a READ statement in essentially the same fashion as the END function. ERR operates as described in the discussion of OPEN and CLOSE statements. Since the error, if present, is detected by the operating system's I/O routines there is little that can be done to mimic this function within the FORTRAN program. However, it is possible that the job control language may provide commands by means of which an error recovery may be accomplished.

VII. SKIP RECORD n

The SKIP RECORD statement causes the next record on device 'n' to be skipped during input. It is equivalent to the construction:

```
      READ (n,f)
    f FORMAT (/)
```

VIII. ENCODE/DECODE

These two statements allow data to be reformatted within the computer. They both require arguments as follows:

```
ENCODE (n,f,'array') 'list'
DECODE (n,f,'array') 'list'
```

where 'n' is the number of characters to be transferred and 'f' is a format statement number.

ENCODE is somewhat like WRITE in that information in the variables specified by 'list' is transferred to a string under control of a FORMAT statement. However, instead of being written to an output device, the string is written into variable 'array'.

Conversely, DECODE reads 'n' characters contained in 'array', formats them as specified by FORMAT statement 'f', and stores the results in the variables given in 'list'.

ENCODE is used by METPLT to create character strings which are used as captions and axis labels. These strings could be explicitly defined in the program, read in from a file, or they may be omitted entirely without significantly altering the functions of these programs.

METINP reads a record once in an A-format, then uses DECODE to reformat the string as required. This could be accomplished by using a BACKSPACE followed by another READ.