UNITED STATES
DEPARTMENT OF INTERIOR
GEOLOGICAL SURVEY

TMPLOT -- TRANSVERSE MERCATOR PLOT PROGRAM FOR MULTICS

By

R. D. Koch and R. J. Miller
345 Middlefield Road
Menlo Park, California 94025

Open-File Report 80-1139

1980

CONTENTS

	Page
Abstract	1
Introduction	2
Purpose	2
Acknowledgments	3
Control parameters	3
Steps program takes during execution	4
Selected program features	4
Lat/lon skip	4
Proximity skip	5
Qualification-code skip	5
Proximity label skip	5
Map skip	7
Data input	7
Character input files	7
Number of numeric variables to read	8
Specifying whether data contains qualifiers	8
Format specification for character input	8
U.S. Geological Surveys Statpac input files	10
Program output	10
Plot output	10
Line printer output	11
Running the program on Multics	11
Conversion to other computer systems	15
Driver exec_com (tmplot.ec)	15
Retrieving data from the system clock	16
File attaching and opening, closing and detaching	16
Character-type variables	16
Subroutine-name and variable-name length	16
Encode and decode built-in subroutines	17
References	17
Appendix A. Format and description of control parameters in the control file	18
B. Formats for control file parameter records	23
C. Location of components of a Statpac data record	24
D. Plotter character set	25
E. Source code listings	26
tmplot.ec	26
tmplot	28
tmplotdriver	40
begin1	41
ckinumof	43
decimal	43
dms	44
dmsi	44
getlst, get, pack	45
juldat, date_prog	47
labskip	48
openfl, openf, closef	49

pcheck---------- 52
readcr, reader1---------- 53
readsp, readsp1---------- 55
sortm---------- 57
strpsort---------- 58
whichpltrof---------- 60

TABLES

Table 1. Qualification codes used in U.S.G.S. Statpac files------ 6
2. Data order and input formats for ASCII character input data records---------- 8

TMPLOT -- TRANVERSE MERCATOR PLOT PROGRAM FOR MULTICS

By R. D. Koch and R. J. Miller

ABSTRACT

TMPLOT is a FORTRAN program which generates plot maps from points with latitude and longitude coordinates. The program is set up to run on the U.S. Geological Survey's Honeywell computer under the Multics operating system. Maps are plotted using the Universal Transverse Mercator projection. User-specified symbols mark the location of data points. Symbol size and type can vary with the value of the plotted data. Points can be labelled with alpha-numeric or numeric labels. Data can be entered from ASCII character files or U.S. Geological Survey Statpac data files. The program is designed to handle unqualified data and data containing the standard codes used in the U.S. Geological Survey Statpac system to qualify goechemical data from U.S.G.S. analytical laboratories.

INTRODUCTION

TMPLOT is a FORTRAN program, with associated subroutines, which generates Universal Transverse Mercator (UTM) projection plot maps of data points having coordinates as (west) latitude and (north) longitude. The X-Y UTM projection map coordinates are computed relative to the center of each map so that there is no difficulty plotting maps which cross UTM zone boundaries. The basic formulae for conversion of geodetic coordinates to plot coordinates are discussed in Thomas (1952) and Plouff (1968).

As described here, TMPLOT is set up to operate on the U.S. Geological Survey's Honeywell Multics computer in Menlo Park, California. Plot files (segments) are written onto disk for transfer to tape. Plots can be made on Benson-Lehner or Houston pen plotters, or on a Versatec plotter. TMPLOT calls 6 system-maintained plot subroutines: plots, plot, letter, number, spot, and symbol.

Purpose

Scatter plots, showing the location of points on a map, are frequently employed to display data. TMPLOT was written to plot such maps with a UTM projection. It can process unqualified numeric data or data which includes the standard qualification codes used by U.S.G.S. analysts for geochemical data. Data locations are represented on the plots by user-specified symbols.

Map plots can be made in several modes:

1. plots of unlabelled symbols showing location of data;
2. plots of symbols with alpha-numeric labels up to 8 characters in length;
3. plots of symbols with one or two numeric labels (one or two data values associated with the point); and

4. "anomally plots" with 1 to 4 symbols of different sizes corresponding to the magnitude of a data value at each point. (These points can be labelled with one or two numeric data values.)

Acknowledgments

The original coding of the section for conversion of latitude and longitude coordinates to X-Y UTM plot coordinates was done by Donald Plouff, U.S. Geological Survey, Menlo Park, California. Subroutine dms was adapted from a routine by Roger Bowen, U.S. Geological Survey, Reston, Virgina and routines dmsi, getlst, and juldat were written by George Van Trump, Jr., U.S. Geological Survey, Denver, Colorado. George has been an invaluable source of advice and assistance with programming difficulties.

CONTROL PARAMETERS

Control parameters for the program are entered in an ASCII character, card-image file (segment); the Control File. Each batch of data processed by the program is referred to as a Data Set. One Control File is required for each Data Set processed. The first two card-images or records in the Control File contain parameters which describe the Data Set and give instructions about how to treat it. These records are called Data Set Parameter Records Number 1 and 2.

Following the Data Set Parameter Records are two records (Map Parameter Records Nos. 1 and 2) for each map to be plotted from that Data Set. The parameters entered on these records describe a map to be plotted and define a subset of the Data Set to appear on that map. The Data Set Parameters and Map Parameters are described in appendix A.

STEPS PROGRAM TAKES DURING EXECUTION

The program first reads both Map Parameter Records from the Control File. For character input data, the program also reads the number of numeric variables to be input and the input format, from the first two records of the data file. Then the data from a Data Set is read by the program and "unacceptable" data discarded (see Lat/Lon Skip, Proximity Skip, Qualification-Code Skip). The accepted data is sorted by latitude and longitude into strips running east-west. This is done to minimize movement of the plotter pen. (This sorting is normally omitted when the plots are for the Versatec plotter.) Parts of the data are then selected using the Map Parameters and maps plotted for the data as specified by each pair of Map Parameter Records in the Control File.

SELECTED PROGRAM FEATURES

Program features and options are discussed in the description of Data-Set and Map Parameters in appendix A. Additional explanation of selected features is given below.

Lat/Lon Skip

Included among the Data Set Parameters are geographic acceptance limits which define an area by latitude and longitude (see Data Set Parameter Record 1, variables SLAT thru WLM in appendix A). Only points within this area will be "accepted" for plotting. Other points in the Data Set will be ignored (not stored for plotting). If Data Set Parameter OP2 is other than blank or zero, a listing is made in the lineprinter output file of all data points lying outside the geographic acceptance area.

Proximity Skip

If Data Set Parameter SKMIN is other than blank or zero, the program will test every point as it is read from the Data Set against the two previously accepted data points. Points within SKMIN minutes of either previous point will be ignored (not stored for plotting). This option is effective only if the data as entered is sorted by location so that closely-spaced points will be likely to be read consecutively.

Qualification-Code Skip

Data in U.S. Geological Survey Statpac files may be qualified with letter codes designed to qualify the results from chemical analyses performed by the U.S. Geological Survey. If points are to be plotted using a numeric "Z-value" (if Data Set Parameter KS1 > 0), these standard codes are checked by the program and action taken as described in table 1. If alpha-numeric labels or no labels are specified (Data Set Parameter KS1 $\leq$ 0), qualification codes are not checked. Qualification-code checking is not performed for ASCII character input files because the only alphabetic information expected there is the row or sample ID.

Proximity Label Skip

For some plots where points are clustered tightly, the labels may heavily overprint and obscure each other and the point locations. If Data Set Parameter SKLAB is other than blank or zero, the program will compare each point with NPP (also a Data Set Parameter) previous points. Any points lying within SKLAB minutes of each other will be plotted but will not be labeled on the plot. The Proximity Label Skip test is performed after the points in a

Table 1.--Qualification codes used in U.S. Geological Survey Statpac data files

Code	Standard Meaning	Action taken by the program
No code	Normal, unqualified data.	Treats data normally.
B	No data.	Skip this point.
N	Nothing detected.	Set value = 0.0, plot the point.
L	Detected but value below limit of determinability.	Set value = 0.0, plot the point.
T	Trace.	Set value = 0.0, plot the point.
H	Interference.	Skip the point.
G	Value greater than upper limit of determinability.	Ignores the code and treats data as if unqualified.

Data Set have been sorted by location into horizontal strips for plotting. Since points are examined consecutively within the storage arrays, the check does not scan across plot-strip edges. (If the plot is for the Versatec plotter, this sorting of points into plot strips is done only if SKLAB > 0.)

Map Skip

If the program finds less than MPSK (a Map Parameter) points which would plot on a requested map, a message is printed and that map is not plotted.

DATA INPUT

Data may be entered from either an ASCII character file or a standard U.S. Geological Survey Statpac data-file (Van Trump and Miesch, 1977). The number of data points in a Data-Set is not limited; however, only the first 3000 points "accepted" will be stored for plotting (see Lat/Lon Skip, Proximity Skip, and Qualification-Code Skip).

Character Input Files

Input data for each point to be plotted must occur in the order shown in table 2. The input format is specified by the user. The first information read from each input data record is an alpha-numeric record identifier of up to 8 characters. This is followed by the latitude and then longitude, both as degrees, minutes, and seconds. The data list consists of numeric data values. These data values may occur in any order. Data-Set Parameters KS1 and KS2 are used to specify which of these values are desired for plotting.

Table 2.--Data order and input formats of ASCII character input data records

Data item	Number of computer words	Example format	Description
1. Sample ID	2	2a4	Up to 8-character alpha-numeric station ID.
2. Latitude	3	3f2.0	Degrees, minutes, seconds read as real (floating point) numbers.
3. Longitude	3	f3.0, 2f2.0	Degrees, minutes, seconds read as as real (floating point) numbers.
4. Data list	1 for each data item read.	6f5.0 or 4(a1, f7.1)	String of numeric data values in any order. Data-Set Parameters KS1 and KS2 are used to identify which elements in the data list are desired for plotting. (For example, KS1=3 if the third numeric value in each record is to be plotted.)

Number of numeric variables to read - When data is entered from an ASCII character file, the program needs to know how many numeric values to read from each input data record. This value (NV) is entered, right-justified, in the first 6 spaces of the first record in the data file. The value given must be between 0 and 100 inclusive and must be greater than or equal to both KS1 and KS2. When data qualifiers are read with the data, NV qualifier-value pairs are read from each input record.

Specifying whether data contains qualifiers - Data entered from an ASCII character file may contain one-letter qualification symbols immediately preceeding each data value (see table 1). To cause the program to read

qualifiers along with the data values, column 10 of the first record in the data file must contain "1". If the qualifiers are not to be read, (the default condition), column 10 of this record should be zero, blank, or "2".

Format specification for character input - A standard FORTRAN format specification must be given if the input data are in an ASCII character file. The format is entered as the second record in the data file (after the record containing the number of numeric data values to be read and before the first data record).

The format specification may be up to 80 characters long, all on one line. It must be enclosed in parenthesis and must be a legal FORTRAN format specification. It is important that the format specify the proper number of computer words for each input item (see table 2). For example, the sample ID need not be a full 8 characters long, but the specification must address 2 words. For a 3-character sample number, "a1,a2" and "a2,a1" are acceptable formats, but "a3" is not because it specifies only one computer word while the program is filling two words. Coordinates may actually be given as decimal degrees or as degrees and decimal minutes, but the format must include specifications for degrees, minutes, and seconds. If data qualifiers are to be read, each qualifier-data value pair is considered a single data item and format specifications must be supplied for both the character portion and the numeric portion of each pair. The format may over-specify the data items, (include specifications for more data items than are indicated by the number of variables to read, NV), but it must include specifications for at least that many.

U.S. Geological Statpac Input Files

Input can only be read from the first Data-Set within a Statpac file. Latitude and longitude are read as words 6 and 7 of each data record, not from data columns. Statpac files created from RASS II data files with program B860 (RASS II retrieval program) are always of this form, even if latitude and longitude values are also placed into columns of the data matrix. Appendix C shows the position of pertinant elements of a U.S. Geological Survey Statpac data record compatable with this program. Data-Set Parameters KS1 and KS2 are used to identify the columns containing the data values to be plotted.

PROGRAM OUTPUT

Plot Output

Plot output consists of maps at any scale in sizes up to 29" by 100" (Benson-Lehner), 34.5" by 100" (Houston), and 39" by 120" (Versatec) (N-S by E-W). Any number of plots can be made from each Data-Set by increasing the number of pairs of Map Parameter Records in the Control File. Data points are indicated by symbols as specified by Data Set Parameters SMBOL. Grid-tics are provided for registration at a spacing determined by Map Parameter INTV and map corners are labeled with latitude and longitude. The date and two titles (TITLE and TITLE2) are printed below the map. If input is from a Statpac file, the column ID of the column specified by Data Set Parameter KS1 is printed beside the lower right corner of the plot. A 10 by 20 inch rectangular scale check is automatically plotted as the first frame of each run and a bit-skip check is plotted for each map.

Alpha-numeric point labels are handled differently for the Versatec and the pen plotters. The labels are stored as 2 computer words of up to 4 characters each. For Versatec plots, the labels are printed above the point

symbol, one word after the other. For the pen plotters, the first word is centered above the point symbol and the other below it. This helps reduce overprinting in crowded areas because the minimum label size for the pen plotters is nearly twice that of the Versatec.

Numeric point labels are handled in the same way on all plotters. The variable specified by Data Set Parameter KS1 is plotted above the point symbol. This value is used to determine symbol size and type on "anomally plots" (see Data Set Parameters VALIM, SMBOL, and VALAB). The variable specified by Data Set Parameter KS2 is plotted below the point symbol.

Line Printer Output

Line printer output consists of headings listing the Data Set Parameters each data set and the Map Parameters for each plot. Totals and lists of data points meeting different criteria can also be made (see Data Set Parameters OP1 to OP5 in appendix A).

RUNNING THE PROGRAM ON MULTICS

On Multics, TMPLOT is run through an exec_com segment (file) which sets the necessary search rules so that all parts of the TMPLOT package and all appropriate plotter software modules will be found. The general preparation for running this program is given below. Descriptions of the Multics system commands used in the driver exec_com and in this section are given by Honeywell Information Systems (1977). To clarify the following discussion, items which the user types are either explained, or enclosed in single quotes if literal. Prompts from the program are underlined.

1. Create one Control File for each Data-Set to be run through the program. Formats and explanations of the parameters in the Control File are given in appendix A and the formats are shown on a chart in appendix B.
2. Create the input data file; either an ASCII character file, or a standard U.S. Geological Survey Statpac file with the desired data in the first Data-Set within that file (see input data).
3. Give the 'new_proc' command to clear out possible interferring elements in the current process history. This is usually unnecessary but is a good thing to try if you seem to be having trouble running a program on Multics. A new_proc command must be given to clear plot software reference names from Multics initiated_segment table if a program other than TMPLOT has been run using one plotter, and a different plotter will be used for the next plot.
4. If plots are to be made on the Versatec plotter, it may be necessary to create an additional parameter file. Several factors, such as the size of plot allowed on the Versatec, have default values which may be unacceptable for a particular plot. If so, these "initial parameters" to control the Versatec should be put into a file. See the Versatec Manual (Versatec, 1976) for details on changing these default parameters.
5. Establish a link to the driver exec_com file by typing:

   'link >udd>Amrap>rdklib>obj>tmplot.ec'

   This link will be perminant and the command need not be repeated when running the program from the same working directory.

6. Execute the program and respond to prompts as follows.

'ec tmplot'

Which plotter? 1)Benson-Lehner 2)Versatec 3)Houston: enter '1', '2', or '3' to continue, 'q' or 'quit' to stop.

If you entered '2', you will be prompted for an initial parameter file.

Initial Parameter File: enter the filename. If you wish to use the default initial parameters, enter 'none' or just a carriage return. Enter 'q' or 'quit' to stop.

CONTROL FILE: enter filename, or 'q' or just a carriage return to stop.

IS INPUT FILE: 1)CHARACTER 2)STATPAC: enter '1' or '2' to continue, enter 'q' or 'quit' to stop.

DATA FILE: enter name of data input file. Entry of 'q' or carriage return will return you to the prompt CONTROL FILE:.

OUTPUT ON TTY: enter 'y' to direct the line printer output to your terminal or 'n' to put it into a file on disk. (A file is recommended.) If the response is 'n', you will receive the following prompt:

PRINTER OUTPUT FILE: enter the name you wish this file to have. Entry of carriage return, 'q', or 'quit' will shift you back to the prompt DATA FILE:.

The program will now read and process the data, and write plot instructions for all maps specified in the Control File. It will then resume prompting with the CONTROL FILE: to allow you to enter another Control File and a data file (which could be the same one again to plot different data from it).

7. To exit the program, enter carriage return, 'q', or 'quit' to the prompt CONTROL FILE. If you have generated plot files for the Versatec plotter, a warning message will be printed at this point reminding you to wait patiently. The plot software is now rasterizing and compressing the vector plot instructions. Wait for the STOP message signifying the end of program execution.

8. Plot output will be contained in one or more files as shown below:

Plotter used	Output
Benson-Lehner	All plot frames in one file named "bl_plot".
Houston	All plot frames in one file named "hous_plot"
Versatec	Each plot frame in a separate file with names in the series "vplt00", "vplt01", "vplt02", vplt03", etc.

Programs are provided by the system to copy the plot files to magnetic tape and to signal the operator to run the tape on the appropriate plotter.

a) For the Benson-Lehner plotter, type the following:

'link >iml>bl_lib>plotter.ec'

'ec plotter bl_plot PAPER-SIZE PEN-SIZE ACCOUNT'

where: PAPER-SIZE is '10' or '30' (inches)

PEN-SIZE is '00', '0', '1', etc.

ACCOUNT is your 9-digit billing number.

b) For the Houston plotter, type the following:

'link >iml>v_plot>multi_houston.ec'

'ec multi_houston'

You will now be prompted for the required information.

c) For the Versatec plotter, type the following:

'link >iml>v_plot>gpt'

'gpt'

You will now be prompted for the required information.

CONVERSION TO OTHER COMPUTER SYSTEMS

The configuration of TMPLOT described here is set up specifically to operate on the Geological Survey's Honeywell Multics computer in Menlo Park, California. Portions of the TMPLOT system have been written around specific features of the offline plotter software available at that site and the Multics operating system. Most of the system-dependant features are isolated in certain subroutines and sections of the program and conversion to other systems with similar plotting capabilities is possible without a major program re-write. Versions of TMPLOT have been run on other systems including the Geological Survey's IBM 371/55 computer in Reston, Virginia.

The following discussion covers aspects of TMPLOT which may need to be handled differently for non-Multics implementation.

Driver exec_com (tmplot.ec)

This routine is a collection of Multics operating system commands to configure a proper program run environment on Multics. This assures that the program will "find" all necessary subroutines and parts of the appropriate plotter software package. Equivalent functions will be performed differently on other systems.

Retrieving Date From the System Clock

Subroutine juldat retrieves the current date from the Multics system calendar clock. Changes would be necessary to address the clock on other systems. If the call to juldat is omitted, TMPLOT will run normally and the date will be listed as "00/00/00".

File Attaching and Opening, Closing and Detaching

Attaching and opening of files is handled by subroutine openf and closing and detaching by subroutine closef. These routines use Multics FORTRAN "open" and "close" statements. Equivalent statements commonly exist in other interactive implementations of FORTRAN but sintax will probably vary. If calls to subroutines "begin1", "openf", and "closef" were eliminated, TMPLOT could be run without explicit file attachments and openings.

Character-type Variables

The ability of the Multics system to handle relatively long character strings results in common usage of long strings for naming file. Character-type variables are a non-ASCII-standard feature of Multics FORTRAN allowing easy manipulation of long character strings. On systems where use of long strings is not permitted, character-type variables can be converted to integer type.

Subroutine-name and Variable-name Length

ASCII-standard FORTRAN allows variable and subroutine-names to a maximum of 6 characters in length. Many implementations of FORTRAN, including Multics, allows these names to be somewhat longer. A small number of variables and several subroutines in the TMPLOT system have names in excess of

6 characters. These names might have to be changed to be compatible with some FORTRAN compilers.

Encode and Decode Built-in Subroutines

The encode and decode statements (used in subroutines reader and readsp) are non-standard enhancements to ANSI-standard FORTRAN. They are present in other implementations of FORTRAN but syntax may vary. The equivalent operation may be performed with the reread statement in IBM FORTRAN.

REFERENCES

Honeywell Information Systems, 1977, Multics prgrammers' manual: Commands and active functions: HIS AG-92.

Plouff, Donald, 1968, Determination of rectangular coordinates for map projections - modifications of basic formulas and applications to computer plotting: U.S. Geological Survey Professional Paper 600-C, p. C174-C176.

Thomas, P. D., 1952, Conformal projections in goedesy and cartography: U.S. Coast and Geodetic Survey Special Publication 251, 142 p.

Van Trump, George, Jr., and Miesch, A. T., 1977, The U.S. Geological Survey RASS-Statpac system for management and statistical reduction of geochemical data: Computers and Geosciences, v. 3, no. 3, p. 475-488.

Versatec Inc., 1976, Versaplot-07 Graphics programming manual: Document Number 50028-90001, 66p.

APPENDIX A

Format and description of control parameters in the Control File for program TMPLOT. Asterisk (*) indicate minimum input requirements for the program to run. All other entries are optional. Values designated with an "i" format must be right-justified in the specified columns. Leaving an entry blank has the same effect as entering 0 (zero) and will result in the default condition (if any).

Parameter	Column	Format	Description
			Data Set Parameter Record No. 1
*ZCAL	1- 7	f7.0	Reciprocal of largest map scale (smallest reciprocal) to be plotted.
*SLAT	9-10	f2.0	Southern latitude limit of Data Acceptance Area-degrees.
*SLM	11-12	f2.0	Southern latitude limit of Data Acceptance Area-degrees.
*TLAT	14-15	f2.0	Northern latitude limit of Data Acceptance Area-degrees.
*TLM	16-17	f2.0	Northern latitude limit of Data Accepatnace Area-minutes.
*ELON	19-21	f2.0	Eastern longitude limit of Data Acceptance Area-degrees.
*ELM	22-23	f2.0	Eastern longitude limit of Data Acceptance Area-minutes.
*WLON	25-27	f3.0	Western longitude limit of Data Acceptance Area-degrees.
*WLM	28-29	f2.0	Western longitude limit of Data Acceptance Area-minutes.
SKMIN	30-33	f4.2	=0, Don't perform Proximity Skip. >0, Distance (decimal minutes) between points Proximity Skip.

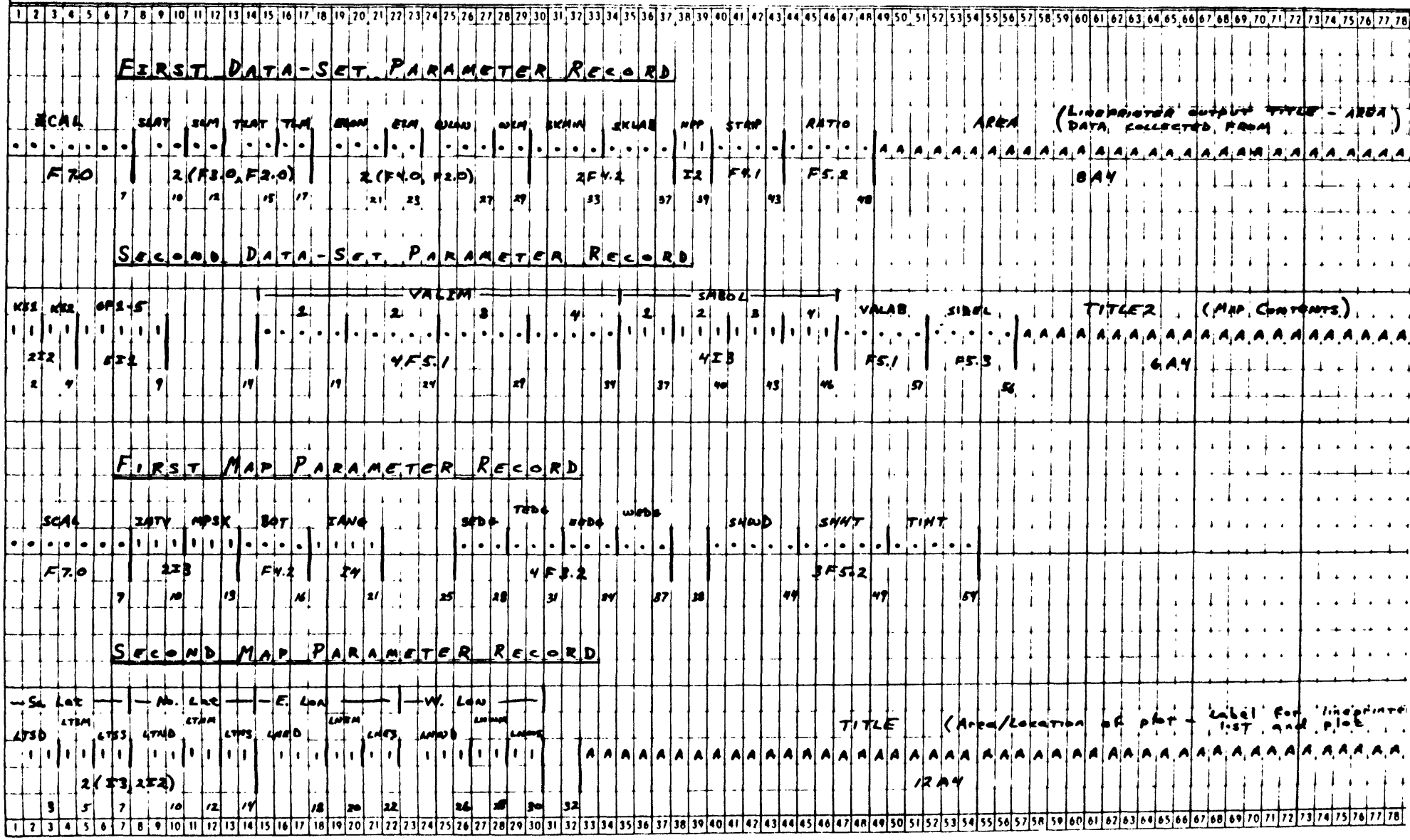
Parameter	Column	Format	Description
			Data Set Parameter Record No. 1--Continued
SKLAB	34-37	f4.2	=0, Plot all labels as specified by KS1 and KS2. (Don't perform Proximity Label Skip.) >0, Omit labels on points within SKLAB minutes of each other.
NPP	38-39	i2	Number of previous points to check in Proximity Label Skip (0 to 10). Default = 10.
STRIP	40-43	f4.1	Width (inches) of plot strips. Default = 2.2.
RATIO	44-48	f5.2	Ratio of 1 minute of map distance in latitude over 1 minutes of map distance in longitude. Used only if SKLAB>0. Default =1.738 (correct value for 55^{o} latitude).
AREA	49-80	8a4	Data-set location/description for line-printer output.
			Data Set Parameter Record No. 2
*KS1	1- 2	i2	Tell what label type to plot at labelled points. = -2 Label point with secondary row ID or "tag number" (Statpac input only). = -1 Don't plot the point labels, just plot the point symbols. = 0 Label point with primary row ID (station or sample ID). = 1-100 Label with corresponding (1st to 100th) numerical value read from each data record. Label is plotted above the point. This value is used to determine symbol type and size on "anomally plots". (See VALIM, SMBOL, and VALAB.)
KS2	3- 4	i2	= 1-100 tell which numeric variable (1st to 100th) read from each data record, to plot below the point. If KS1 $\leq$ 0, the value of KS2 is reset to equal KS1.
OP1	1	i1	= 0, list station number, coordinates, and tag number for all points accepted.

Parameter	Column	Format	Description
			Data Set Parameter Record No. 2--Continued
OP2	6	i1	= 0, list station number, coordinates, and tag number for all Lat/Lon Skips.
OP3	7	i1	= 0, list point labels, coordinates and tag numbers for accepted points after sorting into plot strips (only if sorting is done).
OP4	8	i1	= 0, list point labels and coordinates of points for each map plotted.
OP5	9	i1	= 0, list point labels and coordinates of points with skipped labels (only if SKMIN.NE.0). For OP? ≠ 0, don't do the above. If the input data-set is large, it is recommended that OP1, OP3, and OP4 be set = 1 unless those lists are actually needed.
VALIM	15-19	f5.1	Inclusive value of bottom of class interval 1. If the data value specified by KS1 is less than VALIM (2) the point is not plotted.
	20-24	f5.1	Inclusive value of bottom of class interval 2.
	24-29	f5.1	Inclusive value of bottom of class interval 3.
	30-34	f5.1	Inclusive value of bottom of class interval 4. Used only if KS1>0.
SMBOL	35-37	i3	Number specifying symbol to plot for C.I. 1.
	38-40	i3	Number specifying symbol to plot for C.I. 2.
	41-43	i3	Number specifying symbol to plot for C.I. 3.
	44-46	i3	Number specifying symbol to plot for C.I. 4. Used only if KS1>0. Default symbol for C.I. 1 = 66 (a "+") Default symbol for C.I. 2-4 = 69 (an octagon). See Appendix D for list of symbols. Plotted symbol sizes for CI 1 to 4 are 0.06, 0.12, 0.18, and 0.24 inches, respectively.
VALAB	47-51	f5.1	Lowest value of points to be labeled. Used only if KS1 >0. The variable specified by KS1 is the value the tested to determine the plot symbol's size, type, and whether it should be labelled.

Parameter	Column	Format	Description
			Data Set Parameter Record No. 2--Continued
SIZEL	52-56	f5.3	Height (inches) of characters in plotted point labels. (Defaults and minimum values: Benson-Lehner and Houston = 0.06, Versatec = 0.035.)
TITLE2	57-80	6a4	Map contents (for example: sample sites, Cu data). Printed below plot starting at center of sheet, and on line-printer listing.
			Map Parameter Record No. 1
*SCAL	1- 7	f7.0	Reciprocal of scale for the plot. (For example 63360, 250000 and so on.)
*INTV	8-10	i4	Interval (minutes) between grid-tics. Must divide evenly into map dimensions in both directions (N-S and E-W).
MPSK	11-13	i3	Minimum number of points required to plot a map. (Default = 1).
BOT	14-17	f3.1	Margin width (inches) at bottom of plot. (From lowest grid point to base of "plot area") (Default = 2.0).
SEDG	26-28	f3.2	Distance (decimal deg) to plot points outside bottom (south) map margin. (Default = 0.0).
TEDG	29-31	f3.2	Distance (decimal deg) to plot points outside top (north) map margin. (Default = 0.0).
EEDG	32-34	f3.2	Distance (decimal deg) to plot points outside right (east) map margin. (Default = 0.0).
WEDG	35-37	f3.2	Distance (decimal deg) to plot points outside left (west) map margin. (Default = 0.0).
SHWD	40-44	f5.2	E-W boundary line length (inches).
SHHT	45-49	f5.2	N-S boundary line length (inches).
			If both SHWD and SHHT > 0, program will draw a rectangular box arround the map.

Parameter	Column	Format	Description
			Map Parameter Record No. 1--Continued
TIHT	50-54	f5.2	Distance (inches) between base of characters in title line and bottom of 'plot area". Default = 0.03.
			Map Parameter Record No. 2
*LTSD	2- 3	i2	Latitude of map southern margin - Degrees.
*LTSM	4- 5	i2	Latitude of map southern margin - Minutes.
*LTSS	6- 7	i2	Latitude of map southern margin - Seconds.
*LTND	9-10	i2	Latitude of map northern margin - Degrees.
*LTNM	11-12	i2	Latitude of map northern margin - Minutes.
*LTNS	13-14	i2	Latitude of map northern margin - Seconds.
*LNED	16-18	i3	Longitude of map eastern margin - Degrees.
*LNEM	19-20	i2	Longitude of map eastern margin - Minutes.
*LNES	21-22	i2	Longitude of map eastern margin - Seconds.
*LNWD	24-26	i3	Longitude of map western margin - Degrees.
*LNWM	27-28	i2	Longitude of map western margin - Minutes.
*LNWS	29-30	i2	Longitude of map western margin - Seconds.
TITLE	33-80	12a4	Location title for map. Printed on line printer ouput and below plot starting at left edge.

Appendix B: Format Key for Control File Parameter Records



Appendix C

Location of components of Statpac data record

Word in Statpac data record	Contents
1	Sequential data-row number within a Statpac data set.
2 and 3	8-character alpha-numberic string used to label points if KS1 = 0 (the "primary row ID").
4 and 5	8-character alpha-numeric string used to label points if KS1 = 2 (the "secondary row ID"). NOTE: For Statpace files created with program B860 (RASS II retrieval program): if option 2 = 1-4. words 2 and 3 contain laboratory "tag number" words 4 and 5 contain field number. If option 2 = 5-8. The above positions are reversed.
6	Latitude in degrees, minutes and seconds.
7	Longitude in degrees, minutes and seconds.
8 to (n + 7)	Contain n number data values.
(n + 8) to ((n+9)/10) + (n+8))	Contain n qualification codes packed 10 codes per word.

Appendix D: Plotter character set

The numeric value to the left of each plot symbol is the code used to specify that symbol for plotting.

1	A	13	M	25	Y	37	7	49	/	61	∏
2	B	14	N	26	Z	38	8	50	?	62	±
3	C	15	O	27	%	39	9	51	!	63	↙
4	D	16	P	28	@	40	$	52	¡	64	←
5	E	17	Q	29	&	41	¢	53	¡¡	65	→
6	F	18	R	30	0	42	<	54	▫	66	┼
7	G	19	S	31	1	43	>	55	=	67	✳
8	H	20	T	32	2	44	(	56	▫9	68	#
9	I	21	U	33	3	45	)	57	▫▫	69	◯
10	J	22	V	34	4	46	¬	58	9	70	△
11	K	23	W	35	5	47	|	59	_	71	◇
12	L	24	X	36	6	48	—	60	⊖	72	□

```
&      tmplot.ec
&        add_search_rule.ec
&
&       exec_com driver for plot program tmplot via subroutine
&        tmplotdriver.
&
& ================================================================
&goto &ec_name
&
&label tmplot
&command_line off
&if [exists non_null_link value_seg] &then unlink value_seg
&else &if [exists segment value_seg] &then &goto DEL_VAL_SEG
&goto CREATE_VAL_SEG
&
&label DEL_VAL_SEG
safety_sw_off value_seg
delete value_seg
&
&label CREATE_VAL_SEG
&if [exists segment [pd]>value_seg] &then
&else create [pd]>value_seg
value$set_seg [pd]>value_seg
&
ec &ec_dir>add_search_rule  >udd>Amrap>RKoch>obj -after working_dir
ec &ec_dir>add_search_rule  >udd>Amrap>rdklib>obj -after working_dir
&
&label WHICH_PLOTTER
value$set wp [response "Which Plotter? 1)Benson-Lehner  2)Versatec  3)Houston: "]
&if [equal [value wp] "undefined!"] &then &quit
&if [equal [value wp] "q"         ] &then &quit
&if [equal [value wp] "quit"      ] &then &quit
&if [equal [value wp] "1"] &then &goto BPLOT
&if [equal [value wp] "2"] &then &goto VPLOT
&if [equal [value wp] "3"] &then &goto HPLOT
&else &goto WHICH_PLOTTER
&
& ----------------------------------------------------------------
&      Run the plot program for the Benson-Lehner pen plotter.
&
&label BPLOT
ec &ec_dir>add_search_rule  >iml>bl_lib -after >udd>Amrap>rdklib>obj
>sss>run tmplotb
&
dsr >iml>bl_lib
&quit
& ----------------------------------------------------------------
&      Run the plot program for the Versatec plotter.
&
&label VPLOT
&print To use default parameters, enter carriage-return only.
&label VPLOT2
value$set ipfile [response "Initial Parameter File: "]
&if [equal [value ipfile] "undefined!"] &then &goto VPLOT3
&if [equal [value ipfile] "none"      ] &then &goto VPLOT3
&if [equal [value ipfile] "q"         ] &then &quit
&if [equal [value ipfile] "quit"      ] &then &quit
&if [exists file [value ipfile]] &then &goto ATTACH_INIT_VAL
ioa_ "File Not Found:  "[value ipfile]
&goto VPLOT2
```

```
&
&label ATTACH_INIT_VAL
io attach init_vals vfile_ [value ipfile]
&
&label VPLOT3
ec &ec_dir>add_search_rule  >iml>v_plot -after >udd>Amrap>rdklib>obj
>sss>run tmplotv
&
&if [attached init_vals] &then io (detach destroy_iocb) init_vals
dsr >iml>v_plot
&quit
&
& ------------------------------------------------------------------
&     Run the program for the Houston plotter.
&
&label HPLOT
ec &ec_dir>add_search_rule  >iml>BLH -after >udd>Amrap>rdklib>obj
ec &ec_dir>add_search_rule  >iml>houston -after >iml>BLH
>sss>run tmploth
&
dsr >iml>BLH  >iml>houston
&quit
& ==================================================================
& ==================================================================
&label add_search_rule
&
&   Multics exec_com routine to add a specified directory in
&    the Multics storage system to the process search rules.
&    Routine first checks to see if the directory is already in the
&    search rules.  If so, it is removed, then added again in the
&    position specified (if any).  This augments the standard Multics
&    system add_search_rules command by allowing the user to move a
&    directory from one place to another in the search order.
&
&   Arguments:
&    &1 - relative or absolute pathname of directory to place in the
&         search rules.
&    &2,&3 - (optional), arguments acceptable to the add_search_rules
&            command to specify the desired position in the search order
& ==================================================================
&command_line off
&if [exists argument &4] &then &goto A_S_R_ENTRY_ERROR1
&if [not [exists argument &1]] &then &quit
&if [not [exists directory &1]] &then &goto A_S_R_BAD_DIR
&if [not [on command_error "" -bf discard_output asr &f1]] &then &quit
delete_search_rules &1
add_search_rules &f1
&quit
&
&label A_S_R_ENTRY_ERROR1
ioa_ " Too Many Arguments: &f1"
&quit
&
&label A_S_R_BAD_DIR
ioa_ "Directory Does Not Exist:  &1"
&quit
```

```
      SUBROUTINE TMPLOT (IP)
C
C---- VERSION 1  FOR HONEYWELL MULTICS.        OLD_FORTRAN   (06/24/80)
C
C---- PLOT & LABEL POINTS HAVING LAT & LON COORDS ON A TRANSVERSE
C----   MERCATOR PROJECTION PLOT USING BENSON-LEHNER, VERSATEC, OR
C----   HOUSTON PLOTTER. PLOT "ANOMALY" MAPS FOR NUMERIC DATA.
C---- READS INPUT DATA FROM ASCII FILE OR USGS STATPAC BINARY FILE.
C
C---- NOTES:
C---- 1) On Multics, tmplot and all subroutines having non-standard
C----  returns must be compiled with the old_fortran compiler because
C----  arrays dent, cba1, p, and q may hold alpha or real type data
C----  values and the new_fortran compiler won't allow that.
C
C---- 2) Subroutine juldat was written by George Van Trump, Jr.
C----  (USGS, Denver) to get the current date from the Multics system.
C
C---- PROGRAM REQUIRES THE FOLLOWING SYSTEM-MAINTAINED PLOT ROUTINES:
C----  PLOTS, PLOT, LETTER, NUMBER, SPOT, SYMBOL
C=====================================================================
      CHARACTER*32 IDATA
      CHARACTER*80 IFMT
      INTEGER  OP1,OP2,OP3,OP4,OP5,OP6,SMBOL(4),PLUS,XXXX
      COMMON /SORTBK/ XX(3000),INDX(3000),IND(3000),P(3000),Q(3000)
      COMMON /LABELBK/ DENT(3000),CBA1(3000),X(3000),Y(3000)
      COMMON /READBK/ STA(2),TAG(2),VAR1,VAR2,QCODE1,QCODE2,
     1                TDEG,TMIN,TSEC,GDEG,GMIN,GSEC
      DIMENSION  IDDS(2),L1(4),VID(2),ALAB(2),
     1  R(300), S(300), VALIM(4),AREA(8), TITLE(12), TITLE2(6)
      DATA  BLNK1,GEE,BLANK,PLUS,XXXX/' ','G','    ','+   ','X   '/
      DATA  BEE,HHHH,JD,JM,JY/'B','H',0,0,0/
      DATA  A,EQ,IOT,INC,INF,LP/6378206.,6.768658E-3,0,20,30,6/
      DEGR=3.141593/180.0
C=====================================================================
C
      CALL JULDAT (JY,JM,JD)
C
C---------------------------------------------------------------------
      IF (IP.LT.1 .OR. IP.GT.3) GO TO 504
      NDS = 0
      NMAP = 0
C---- DETERMINE INPUT AND OUTPUT FILES AND PERFORM FILE ATTACHING.
   20 IDTYPE = -1
      NV = 100
      CALL BEGIN1 (INC,INF,IOT,LP,15,1,IDTYPE,0,NDS,IDATA,$500,$504)
C
      IF (IDTYPE .NE. 1) GO TO 28
      NSR = 5000
C---- DETERMINE NO. NUMERIC VARIABLES & INPUT FORMAT.
      IFMT = '                '
      CALL READCR1 (IOT,INF,LP,NV,KQ,IFMT,$500)
C
C---- SET APPROPRIATE CONSTANTS FOR THE PLOTTER TO BE USED.
   28 CALL WHICHPLTROF (IOT,IP,XFUDGE,YFUDGE,XLMAX,YLMAX,CSMIN,$500)
C
C---- PLOT A CHECK FOR BIT SKIP & A 10 X 20 INCH SCALE CHECK.
      NDS = NDS + 1
      IF (NDS .NE. 1) GO TO 30
      IF (IP .EQ. 2) CALL PLOTS (0.0,0.0,0.0)
```

```
      CALL PCHECK (XFUDGE,YFUDGE)
      IF (IP .EQ. 2) CALL PLOT (0.0,0.0,+999)
C
C-----------------------------------------------------------------------
C---- READ DATA SET PARAMETER CARD #1
C
   30 READ (INC,31,END=500) ZCAL,SLAT,SLM,TLAT,TLM,ELON,ELM,WLON,WLM,
     1                      SKMIN,SKLAB,NPP,STRIP,RATIO,AREA
   31 FORMAT (F7.0,2(F3.0,F2.0),2(F4.0,F2.0),2F4.2,I2,F4.1,F5.2,8A4)
C-----------------------------------------------------------------------
C---- READ DATA-SET PARAMETER CARD NO 2.
C
      READ (INC,33,END=34) KS1,KS2,OP1,OP2,OP3,OP4,OP5,OP6,
     1                     VALIM,SMBOL,VALAB,SIZEL,TITLE2
   33 FORMAT (2I2,6I1,4X,4F5.1,4I3,F5.1,F5.3,6A4)
      GO TO 36
C-----------------------------------------------------------------------
   34 WRITE (IOT,35)
   35 FORMAT (/' UNEXPECTED END-OF-FILE -- CONTROL FILE '/)
      GO TO 500
C
C-----------------------------------------------------------------------
   36 IF (KS1.GT.NV .OR. KS2.GT.NV) GO TO 37
      IF (NV.LT.0 .OR. NV.GT.100) GO TO 37
      IF (KS1 .EQ. 0) KS2=0
      IF (KS1 .EQ. -1) KS2=-1
      IF (KS1 .EQ. -2) KS2=-2
      IF (KS1 .GE. -2) GO TO 39
C
   37 WRITE (IOT,38)
   38 FORMAT (/' ERROR: BAD SELECTED VARIABLE NO. OR NO. VARIABLES')
      GO TO 499
C
   39 SKDEG = SKMIN/60.
      IF (RATIO .LE. 0.0) RATIO=1.738
      IF (SIZEL .LT. CSMIN) SIZEL=CSMIN
      LSIZE = (SIZEL/0.06) + .005
      IF (NPP.LT.1 .OR. NPP.GT.10) NPP=10
      IF (STRIP.LE.0.0 .OR. STRIP.GT.5.0) STRIP=2.2
      IF (SMBOL(1).LT.1 .OR. SMBOL(1).GT.72) SMBOL(1)=66
      DO 40 I=2,4
   40 IF (SMBOL(I).LT.1 .OR. SMBOL(I).GT.72) SMBOL(I)=69
C
      WRITE (LP,505) JM,JD,JY,NDS,IDATA
      WRITE (LP,510) AREA,TITLE2,ZCAL,STRIP,SKMIN,SKLAB
      WRITE (LP,515) SLAT,SLM,TLAT,TLM,ELON,ELM,WLON,WLM,KS1,KS2,OP1,OP2
     1,OP3,OP4,OP5
      WRITE (LP,516)  VALIM,SMBOL,VALAB,SIZEL
      WRITE (LP,517) RATIO
C-----------------------------------------------------------------------
      SKLAB=SKLAB/60.
      SLAT=SLAT+SLM/60.
      TLAT=TLAT+TLM/60.
      ELON=ELON+ELM/60.
      WLON=WLON+WLM/60.
      PLMAX=0.0
      PLMIN=100.0
      N=0
      NE=0
      NR=0
```

```
      NBH=0
      NLLS=0
      PRPLAT=500.
      PRPLON=500.
      OLPLAT=500.
      OLPLON=500.
C-----------------------------------------------------------------------
      GO TO (41,43) IDTYPE
C---- FOR CHARACTER INPUT DATA, WRITE OUT THE NO. VAR. & FORMAT.
   41 WRITE (LP,518) NV,IFMT
      GO TO 45
C
C---- READ STATPAC DATA-SET HEADER RECORD.
C
   43 CALL READSP1 (IOT,INF,LP,KS1,KS2,NSR,NSC,VID,$500)
C
C=======================================================================
   45 DO 125 L=1,NSR
C---- READ DATA FOR NEXT STATION.
      GO TO (47,48) IDTYPE
C
   47 CALL READCR (IOT,INF,IFMT,NV,KS1,KS2,KQ,$130)
      GO TO 49
C
   48 CALL READSP (INF,IOT,LP,KS1,KS2,NSC,$500)
C
   49 NR = NR + 1
      PLAT=TDEG+TMIN/60.+TSEC/3600.
      PLON=GDEG+GMIN/60.0+GSEC/3600.
      IF (PLAT.LT.SLAT.OR.PLAT.GT.TLAT) GO TO 50
      IF (PLON.LT.ELON.OR.PLON.GT.WLON) GO TO 50
      IF (PLAT.LT.PLMIN) PLMIN=PLAT
      IF (PLAT.GT.PLMAX) PLMAX=PLAT
      GO TO 60
   50 NLLS = NLLS + 1
      IF (OP2.NE.0) GO TO 125
      WRITE (LP,55) STA,TDEG,TMIN,TSEC,GDEG,GMIN,GSEC,TAG
   55 FORMAT (11X,2A4,2(F6.0,2F3.0),5X,2A4,6X,'LAT OR LON SKIP')
      GO TO 125
C-----------------------------------------------------------------------
   60 IF (SKDEG.EQ.0.0) GO TO 75
C---- MAKE PROXIMITY CHECK.
      DPLAT=ABS(PRPLAT-PLAT)
      DPLON=(ABS(PRPLON-PLON))/RATIO
      DOLAT=ABS(OLPLAT-PLAT)
      DOLON = (ABS(OLPLON-PLON))/RATIO
      IF ((DPLAT.GT.SKDEG.OR.DPLON.GT.SKDEG).AND.(DOLAT.GT.SKDEG.OR.DOLO
     1N.GT.SKDEG)) GO TO 70
      WRITE (LP,65) STA,TDEG,TMIN,TSEC,GDEG,GMIN,GSEC,TAG
   65 FORMAT (11X,2A4,2(F6.0,2F3.0),5X,2A4,6X,'PROXIMITY SKIP')
      GO TO 125
   70 OLPLAT=PRPLAT
      OLPLON=PRPLON
      PRPLAT=PLAT
      PRPLON=PLON
C-----------------------------------------------------------------------
   75 NE = NE + 1
      N = N + 1
      IF (N-3000) 91,85,80
   80 N=N-1
```

```
      GO TO 125
   85 WRITE (LP,90)  STA,TAG
   90 FORMAT ('3000 STATIONS ACCEPTED.',2(6X,2A4),'  IS NO. 3000')
C
C---- LOAD PT. LABEL & COORDINATE ARRAYS -- STATION ELIGIBLE TO PLOT.
   91 DENT(N) = BLANK
      CBA1(N) = BLANK
      IF (KS1 .NE. -2) GO TO 92
      DENT(N) = TAG(1)
      CBA1(N) = TAG(2)
      GO TO 100
   92 IF (KS1) 98,93,94
   93 DENT(N) = STA(1)
      CBA1(N) = STA(2)
      GO TO 100
C---- FOR QCODE = B OR H, DON'T PLOT THE POINT.
   94 IF (QCODE1.EQ.BEE .OR. QCODE1.EQ.HHHH) GO TO 96
C---- ** FOR QCODE=N,L,T  SET VAL=0.0
      IF (QCODE1.NE.BLNK1 .AND. QCODE1.NE.GEE) VAR1=0.0
      IF (VAR1 .GE. VALIM(1)) GO TO 97
   96 N = N - 1
      NBH = NBH + 1
      GO TO 125
   97 DENT(N) = VAR1
   98 IF (KS2) 100,100,99
   99 IF (QCODE2.NE.BLNK1 .AND. QCODE2.NE.GEE) GO TO 100
      CBA1(N) = VAR2
  100 X(N)=PLON
      Y(N)=PLAT
      IF (OP1.GT.0) GO TO 125
      IF (KS1 .GT. 0) GO TO 105
C---- FOR PLOT OF STATION NUMBERS OR TAG NUMBERS.
      WRITE (LP,103) N,STA,TDEG,TMIN,TSEC,GDEG,GMIN,GSEC,TAG,
     1      DENT(N),CBA1(N)
  103 FORMAT (' ',I5,5X,2A4,2(F6.0,2F3.0),5X,2A4,6X,2A4)
      GO TO 125
C---- FOR A PLOT WITH DATA VALUE(S) WRITTEN AT EA PLOTTED POINT.
  105 WRITE (LP,108) N,STA,TDEG,TMIN,TSEC,GDEG,GMIN,GSEC,TAG,
     1      DENT(N),CBA1(N)
  108 FORMAT (' ',I5,5X,2A4,2(F6.0,2F3.0),5X,2A4,6X,F10.2,F15.2)
  125 CONTINUE
C-------------------------------------------------------------------
  130 WRITE (LP,135) NR,NBH,NLLS,NE,N,NDS
  135 FORMAT ('0'/'0',I6,' SAMPLES READ',30X,I6,' SAMPLES WITH B OR H',
     1  ' QUAL. CODE',11X,I6,' LATITUDE OR LONGITUDE SKIPS'/
     2  '0',I6,' SAMPLES ELIGIBLE TO BE PLOTTED', 12X,I6,' SAMPLES TO',
     3  ' BE PLOTTED FROM DATA-SET',I3 /)
C===================================================================
C---- SORT THE DATA BY LOCATION.  ---------------------------------
      IF (IP .NE. 2) GO TO 138
C---- FOR VERSATEC, SORT IFF PROXIMITY LABEL SKIP IS TO BE DONE OR
C----   STATION LABELS ARE NOT TO BE PLOTTED.
      IF (SKLAB.EQ.0.0 .OR. KS1.GT.0) GO TO 295
C
  138 DELTA=PLMAX-PLMIN
      YTOTP=60.0*1.85*39370.0*DELTA/ZCAL
      JDIFF=1.0+YTOTP/STRIP
      YMIN=PLMIN-0.001
      DELTA=DELTA/JDIFF
      JDIFF=JDIFF+1
```

```
      WRITE (LP,140) JDIFF,PLMIN,PLMAX
  140 FORMAT ('1POINTS TO BE SORTED INTO',I4,' STRIPS BTWN MOST S & N'
     1,' STATIONS AT LAT',F8.4,' AND',F8.4/)
C-----------------------------------------------------------------------
C---- SORT THE DATA INTO PLOT STRIPS. ----------------------------------
C
      CALL STRPSORT (DELTA,JDIFF,LN,LP,N,YMIN)
C
C-----------------------------------------------------------------------
      WRITE (LP,190) LN,JDIFF,N
  190 FORMAT ('0',I5,' SAMPLES SORTED INTO',I3,' STRIPS FROM',I6,'
     1ELIGIBLE STATIONS')
C-----------------------------------------------------------------------
      IF (OP3.GT.0) GO TO 270
C---- LIST ALL STATIONS & DATA SORTED BY GEOGRAPHIC LOCATION.
      WRITE (LP,225)  NDS
  225 FORMAT ( 'OPOINTS FROM DATA SET',I3,' ELIGIBLE TO BE PLOTTED.
     1    ' SORTED BY LAT & LON INTO PLOT STRIPS.'/)
      IF (KS1 .GE. 1) GO TO 255
      DO 230 J=1,N
  230 WRITE (LP,235) J,DENT(J),CBA1(J),Y(J),X(J)
  235 FORMAT (2X,I6,5X,2A4,F12.4,F13.4)
      GO TO 270
  255 DO 260 J=1,N
  260 WRITE (LP,265) J,Y(J),X(J),DENT(J),CBA1(J)
  265 FORMAT (I10,2F12.4,2X,2F13.2)
C-----------------------------------------------------------------------
  270 IF (SKLAB.EQ.0.0 .OR. KS1.GT.0) GO TO 295
C-----------------------------------------------------------------------
C---- MAKE PROXIMITY CHECK FOR LABEL SKIP  -----------------------------
C
      CALL LABSKIP (KS1,OP5,RATIO,SKLAB,N,NPP,LP)
C
C-----------------------------------------------------------------------
C---- READ FIRST MAP PARAMETER CARD
C
  295 READ (INC,300,END=499) SCAL,INTV,MPSK,BOT,IANG,SEDG,TEDG,EEDG,
     1  WEDG,SHWD,SHHT,TIHT
  300 FORMAT (F7.0,2I3,F4.2,I4,4X,4F3.2,2X,3F5.2)
C
      NMAP=NMAP+1
      IF (BOT.LE.0.3) BOT=2.0
      IF (MPSK .LE. 0) MPSK=1
      IF (INTV .GT. 0) GO TO 304
      WRITE (IOT,303)
      WRITE (LP,303)
  303 FORMAT (' ERROR:  VALUE OF INTV .LE. 0.    -- STOP')
      GO TO 499
C-----------------------------------------------------------------------
C---- READ SECOND MAP PARAMETER CARD
C
C
  304 READ (INC,305,END=34) LTSD,LTSM,LTSS,LTND,LTNM,LTNS,LNED,LNEM,
     1  LNES,LNWD,LNWM,LNWS,TITLE
  305 FORMAT (2(I3,2I2),2(I4,2I2),2X,12A4)
C
      NAP = 0
      IF (TIHT.EQ.0.0) TIHT=0.03
      DO 306 I=1,4
  306 L1(I) = 0
```

```
C---- PRINT OUTPUT PARAMETERS, TOTALS & DESCRIPT INFO FOR THIS MAP.
      WRITE (LP,520) NMAP,NDS,AREA,JM,JD,JY,TITLE,TITLE2,SCAL
      WRITE (LP,525) LTSD,LTSM,LTSS,LTND,LTNM,LTNS,LNED,LNEM,LNES,LNWD,L
     1NWM,LNWS,SEDG,TEDG,EEDG,WEDG
      WRITE (LP,530) INTV,SHWD,SHHT,BOT,TIHT,IANG,MPSK
C=====================================================================
      SCALE=100.0/(SCAL*2.54005)
      SCALE=0.9996*SCALE
C---- STORE MAP CORNER COORDS AS D,M,S FOR MAP LABELING.
      TSD = LTSD
      TSM = LTSM
      TND = LTND
      TNM = LTNM
      GED = LNED
      GEM = LNEM
      GWD = LNWD
      GWM = LNWM
C
      KTSM=(60*LTSD+LTSM)*60+LTSS
      KTNM=(60*LTND+LTNM)*60+LTNS
      KNEM=(60*LNED+LNEM)*60+LNES
      KNWM=(60*LNWD+LNWM)*60+LNWS
      KNORT=(KTNM-KTSM)/INTV
      KWEST=(KNWM-KNEM)/INTV
C---------------------------------------------------------------------
C---- DOES GRIDMARK INTRVL DIVIDE EVENLY INTO MAP DIM. FROM LTSD,LTND...
      ITEST=INTV*KNORT-KTNM+KTSM
      IF (ITEST.NE.0) GO TO 310
      ITEST=INTV*KWEST-KNWM+KNEM
      IF (ITEST.EQ.0) GO TO 320
  310 WRITE (LP,315)
      WRITE (IOT,315)
  315 FORMAT (' GRID DOES NOT DIVIDE EVENLY INTO MAP DIMENSION. STOP.'/)
      GO TO 295
C---------------------------------------------------------------------
C---- PRELIMINARY CALC. FOR GRID TIC & POINT X-Y COORDS. --------------
  320 KNORT=KNORT/60
      KWEST=KWEST/60
      JNORT=KNORT+1
      JWEST=KWEST+1
      FTSM=KTSM
      FTNM=KTNM
      FNEM=KNEM
      FNWM=KNWM
      KTSM=KTSM/60
      KTNM=KTNM/60
      KNEM=KNEM/60
      KNWM=KNWM/60
      DSOUT=FTSM/3600.0
      DEAST=FNEM/3600.0
      DNORT=FTNM/3600.0
      DWEST=FNWM/3600.0
      CEN=0.5*(DEAST+DWEST)
      CLAT=(FTSM+FTNM)/7200.0
      RCLAT=DEGR*CLAT
      CP=COS(RCLAT)
      SP=SIN(RCLAT)
      CP2=CP*CP
      SP2=1.0-CP2
      SCP=SP*CP
```

```
      SC5=0.5*SCP
      F1141=1141.7-9.6*CP2
      FM1=111699.3-CP2*F1141
      SC228=F1141*SCP
      GE=1.0+0.5*EQ*SP2*(1.0+0.75*EQ*SP2)
      GEF=EQ*SCP
      AGE=A*GE
      AGEF=A*GEF
C---- IS NO. OF GRID TICKS .LE. 300 (SIZE OF THEIR COORD ARRAYS)? ------
      NTIC = JNORT * JWEST
      IF (NTIC .LE. 300) GO TO 323
      WRITE (LP,322) NTIC
      WRITE (IOT,322) NTIC
  322 FORMAT (//I6,' GRID TICKS COMPUTED .GT. STORAGE.   INCREASE INTV')
      GO TO 500
C=====================================================================
C---- COMPUTE COORDS OF GRID TICS.  ----------------------------------
  323 L = 0
      ASC5 = A * SC5
      ASCALE = A * SCALE
      FTIK = 60.0 * (INTV/3600.0)
      DO 325 J=1,JNORT
      FJ=J-1
      PLAT=DSOUT+FTIK*FJ
      DP=PLAT-CLAT
      DPR=DEGR*DP
      G=GE+DPR*GEF
      XF=ASCALE*G*(CP-DPR*SP)
      YF=DP*(FM1+DPR*SC228)
      ASCP=G*ASC5
C---- COMPUTE COORDS OF GRID TICS IN ONE PLOT ROW.  ------------------
      DO 325 K=1,JWEST
      FK=K-1
      L=L+1
      PLON=DEAST+FTIK*FK
      DLB=DEGR*(CEN-PLON)
      R(L)=DLB*XF*XFUDGE
      S(L)=SCALE*(YF+DLB*DLB*ASCP)*YFUDGE
  325 CONTINUE
C---------------------------------------------------------------------
C---- FIND SMALLEST X & Y COORDS OF GRID TICS ON THIS MAP.  -----------
      SMX=R(JWEST)
      SMY=S(1)
      DO 335 J=2,JWEST
      IF (S(J)-SMY) 330,335,335
  330 SMY=S(J)
  335 CONTINUE
      DO 345 J=JWEST,NTIC,KNORT
      IF (R(J)-SMX) 340,345,345
  340 SMX=R(J)
  345 CONTINUE
C---------------------------------------------------------------------
C---- SHIFT GRID TICS NORTH & WEST BY SMAX & WMAX.
      SMAX=BOT-SMY
      WMAX=-SMX
      DO 350 J=1,NTIC
      R(J)=R(J)+WMAX
  350 S(J)=S(J)+SMAX
      KEAST = NTIC - KWEST
      YKE=S(KEAST)
```

```
      X1=R(1)
      Y1=S(1)
      XKE=R(KEAST)
      XN = R(NTIC)
      YN = S(NTIC)
      XME=R(JWEST)
      YME=S(JWEST)
C---- WILL PLOT GRID FIT ON THE PLOT PAPER? ---------------------------
      IF (YN.LT.YLMAX .AND. YKE.LT.YLMAX) GO TO 360
      WRITE (LP,355) YN,YLMAX,Y1,XN,X1,YKE
  355 FORMAT (' ',F6.2,'-IN. NORTH BORDER EXCEEDS',f6.2,' IN.',4F7.2//)
      GO TO 295
  360 IF (XME.LT.XLMAX .AND. XN.LT.XLMAX) GO TO 370
      WRITE (LP,365) XLMAX,XME,YME,XN,YN
  365 FORMAT (' ','EAST BORDER EXCEEDS',F7.2,' INCHES',4F7.2//)
      GO TO 295
C
  370 TMW=ABS(R(JWEST)-R(1))
      HALFW2=0.5*TMW
      SHWB=HALFW2-SHWD/2.0
      SHEB=HALFW2+SHWD/2.0
C=======================================================================
C---- DETERMINE WHICH PTS. TO PLOT ON THIS MAP & COMPUTE X-Y COORDS. ---
      DSOUTT=DSOUT-SEDG
      DNORTT=DNORT+TEDG
      DEASTT=DEAST-EEDG
      DWESTT=DWEST+WEDG
      MSTA=0
      DO 375 J=1,N
      PLAT=Y(J)
      PLON=X(J)
      IF (PLAT.LT.DSOUTT.OR.PLAT.GT.DNORTT.OR.PLON.LT.DEASTT.OR.PLON.GT.
     1DWESTT) GO TO 375
      MSTA=MSTA+1
C---- COMPUTE X & Y MAP COORDINATES & SHIFT THEM V & W BY SMAX & WMAX.
      INDX(MSTA)=J
      DP=PLAT-CLAT
      DPR=DEGR*DP
      DLB=DEGR*(CEN-PLON)
      G=AGE+DPR*AGEF
      GDLB=G*DLB
      XX(MSTA)=(SCALE*GDLB*(CP-DPR*SP)*XFUDGE)+WMAX
      Q(MSTA)=(SCALE*(DP*(FM1+DPR*SC228)+DLB*GDLB*SC5)*YFUDGE)+SMAX
  375 CONTINUE
C=======================================================================
      IF (OP4.GT.0) GO TO 395
C---- LIST POINTS FOR THIS MAP SORTED BY LOCATION.
      WRITE (LP,380)
  380 FORMAT ('0ELIGIBLE STATIONS FOR THIS MAP SORTED BY LOCATION'/)
      IF (MSTA .EQ. 0) GO TO 395
      IF (KS1 .GT. 0)  GO TO 391
      DO 384 J=1,MSTA
      K=INDX(J)
      XL = X(K)
      YL = Y(K)
      CALL DMS (YL,LAT1,LAT2,LAT3)
      CALL DMS (XL,LON1,LON2,LON3)
  384 WRITE (LP,385) J,DENT(K),CBA1(K),LAT1,LAT2,LAT3,LON1,LON2,LON3,K
  385 FORMAT (' ',I4,3X,2A4,2(I6,2A2),I9)
      GO TO 395
```

```
  391 DO 394 J=1,MSTA
      K = INDX(J)
      XL = X(K)
      YL = Y(K)
      CALL DMS (YL,LAT1,LAT2,LAT3)
      CALL DMS (XL,LON1,LON2,LON3)
      WRITE (LP,393) J,IDENT(K),CBA1(K),LAT1,LAT2,LAT3,LON1,LON2,LON3,K
  393 FORMAT (' ',I4,3X,2F12.2,2(I6,2A2),I9)
  394 CONTINUE
C=====================================================================
  395 WRITE (LP,400) MSTA
  400 FORMAT ('0',I7,' POINTS FOUND TO BE PLOTTED ON THIS MAP.'/)
C---- ARE THERE ENOUGH ACCEPTABLE POINTS TO PLOT THIS MAP?
      IF (MSTA.GT.MPSK) GO TO 410
      WRITE (LP,405) MPSK
  405 FORMAT ('0# STATIONS THIS PLOT <',I5,' THIS MAP NOT PLOTTED.')
      GO TO 295
C---- PLOT MAP TITLE-LINE INFO (MAP AREA,CONTENTS,DATE)  ---------------
  410 RJM=FLOAT(JM)
      RJD=FLOAT(JD)
      RJY=FLOAT(JY)
      CALL LETTER(48,2,0,XME,TIHT,TITLE)
      CALL SPOT(HALFW2,TIHT+0.06,0.12,PLUS,0.0)
      CALL LETTER (24,2,0,HALFW2+.2,TIHT,TITLE2)
C---- PLOT MAP SYMBOLS & ASSOC. INCLUSIVE LOWER CLASS INTERVAL LIMIT ---
      RCP = HALFW2 + ((X1-HALFW2)/2.0)
      IF (RCP .LT. 3.12) RCP=3.12
      XEX = RCP - 1.35
      XX2 = RCP + 0.3
      TIHT1 = TIHT + .2
      TIHT2 = TIHT + .6
      CALL SPOT (XEX,TIHT2+.06,0.06,SMBOL(1),0.0)
      CALL SYMBOL (XEX+.12,TIHT2,0.12,' - ',0.0,3)
      CALL NUMBER (XEX+.48,TIHT2,0.12,VALIM(1),0.0,2)
      IF (VALIM(2) .EQ. 0.0) GO TO 411
      CALL SPOT (XEX,TIHT1+.06,0.12,SMBOL(2),0.0)
      CALL SYMBOL (XEX+.12,TIHT1,0.12,' - ',0.0,3)
      CALL NUMBER (XEX+.48,TIHT1,0.12,VALIM(2),0.0,2)
  411 IF (VALIM(3) .EQ. 0.0) GO TO 412
      CALL SPOT (XX2,TIHT2+.06,0.18,SMBOL(3),0.0)
      CALL SYMBOL (XX2+.12,TIHT2,0.12,' - ',0.0,3)
      CALL NUMBER (XX2+.48,TIHT2,0.12,VALIM(3),0.0,2)
  412 IF (VALIM(4) .EQ. 0.0) GO TO 413
      CALL SPOT (XX2,TIHT1+.06,0.24,SMBOL(4),0.0)
      CALL SYMBOL (XX2+.12,TIHT1,0.12,' - ',0.0,3)
      CALL NUMBER (XX2+.48,TIHT1,0.12,VALIM(4),0.0,2)
C---- PLOT THE PROGRAM RUN DATE. ---------------------------------------
  413 CALL NUMBER (X1-1.00,TIHT,0.12,RJM,0.0,-1)
      CALL SYMBOL (X1-0.76,TIHT,0.12,'/  /',0.0,4)
      CALL NUMBER (X1-0.64,TIHT,0.12,RJD,0.0,-1)
      CALL NUMBER (X1-0.24,TIHT,0.12,RJY,0.0,-1)
      IF (SHWD.EQ.0.0.OR.SHHT.EQ.0.0) GO TO 415
C---- DRAW RECTANGULAR BOUNDARY LINE ARROUND THE MAP  ------------------
      CALL PLOT(SHEB,0.02,3)
      CALL PLOT(SHEB,SHHT,2)
      CALL PLOT(SHWB,SHHT,2)
      CALL PLOT(SHWB,0.02,2)
      CALL PLOT(SHEB,0.02,2)
C---- PLOT MAP GRIDMARKS & LABEL MAP CORNERS W/ LAT & LON  -------------
  415 JWEST1 = JWEST + 1
```

```
      DO 420 L=1,JWEST
      K=JWEST1-L
  420 CALL SPOT(R(K),S(K),0.24,PLUS,0.0)
      CALL NUMBER(X1+0.16,Y1-0.07,0.12,TSD,0.0,-1)
      CALL NUMBER(X1+0.49,Y1-0.07,0.12,TSM,0.0,-1)
      LIN=-1
      DO 445 L=1,KNORT-1
      LIN=-LIN
      IF (LIN) 435,425,425
C EAST TO WEST
  425 JB=L*JWEST+1
      KE=JB+KWEST
      DO 430 K=JB,KE
  430 CALL SPOT(R(K),S(K),0.24,PLUS,0.0)
      GO TO 445
  435 JB=(L+1)*JWEST+1
      DO 440 M=1,JWEST
      K=JB-M
  440 CALL SPOT(R(K),S(K),0.24,PLUS,0.0)
  445 CONTINUE
      IF (LIN) 450,460,460
  450 CALL SPOT(XKE,YKE,0.24,PLUS,0.0)
      CALL NUMBER(XKE+0.16,YKE-0.07,0.12,TND,0.0,-1)
      CALL NUMBER(XKE+0.49,YKE-0.07,0.12,TNM,0.0,-1)
      CALL NUMBER(XKE-0.36,YKE+0.13,0.12,GED,0.0,-1)
      CALL NUMBER(XKE+0.12,YKE+0.13,0.12,GEM,0.0,-1)
      DO 455 K=KEAST+1,NTIC
  455 CALL SPOT(R(K),S(K),0.24,PLUS,0.0)
      CALL NUMBER(XN-0.36,YN+0.13,0.12,GWD,0.0,-1)
      CALL NUMBER(XN+0.12,YN+0.13,0.12,GWM,0.0,-1)
      GO TO 470
  460 CALL SPOT(XN,YN,0.24,PLUS,0.0)
      CALL NUMBER(XN-0.36,YN+0.13,0.12,GWD,0.0,-1)
      CALL NUMBER(XN+0.12,YN+0.13,0.12,GWM,0.0,-1)
      DO 465 L=1,JWEST-1
      K = NTIC - L
  465 CALL SPOT(R(K),S(K),0.24,PLUS,0.0)
      CALL NUMBER(XKE+0.16,YKE-0.07,0.12,TND,0.0,-1)
      CALL NUMBER(XKE+0.49,YKE-0.07,0.12,TNM,0.0,-1)
      CALL NUMBER(XKE-0.36,YKE+0.13,0.12,GED,0.0,-1)
      CALL NUMBER(XKE+0.12,YKE+0.13,0.12,GEM,0.0,-1)
  470 CONTINUE
C---- END OF MAP LABELING    ------------------------------------------
C---- PLOT AND LABEL STATIONS ON THE MAP.
      DO 490 JJ=1,MSTA
      J=INDX(JJ)
      YJ=Q(JJ)
      XJ=XX(JJ)
C---- CHECK THAT THIS POINT FALLS ON A 30 X 110 IN. PAGE.
      IF (XJ.LE.-10.0.OR.XJ.GE.100.0) GO TO 490
      IF (YJ.LE.0.1.OR.YJ.GE.29.9) GO TO 490
C---- SET SIZE OF SYMBOL TO BE PLOTTED.    ----------------------------
      SIZE = .06
      ISMBOL = SMBOL(1)
      IF (KS1 .LE. 0) GO TO 473
C---- DETERMINE WHICH SYMBOL & SIZE OF SYMBOL TO BE PLOTTED.-----------
      IF (VALIM(2) .EQ. 0.0) GO TO 471
      IF (DENT(JJ) .LT. VALIM(2)) GO TO 473
      SIZE = 0.12
      ISMBOL = SMBOL(2)
```

```
  471 IF (VALIM(3) .EQ. 0.0) GO TO 472
      IF (DENT(JJ) .LT. VALIM(3)) GO TO 473
      ISMBOL = SMBOL(3)
      SIZE = 0.18
  472 IF (VALIM(4) .EQ. 0.0) GO TO 473
      IF (DENT(JJ) .LT. VALIM(4)) GO TO 473
      ISMBOL = SMBOL(4)
      SIZE = 0.24
C-------------------------------------------------------------------
C---- PLOT THE POINT.
  473 CALL SPOT (XJ,YJ,SIZE,ISMBOL,0.0)
      NAP = NAP + 1
      M = ((SIZE/0.06)+0.0005)
      L1(M) = L1(M) + 1
      IF (KS1 .LE. 0) GO TO 475
      IF (DENT(JJ) .GE. VALAB) GO TO 475
      GO TO 490
C---- LABEL THE POINT ---------------------------------------------------
  475 SIZE2 = SIZE/2.0
      VOSU = (SIZE2 + .01)
      VOSD = (SIZE2 + .01 + (LSIZE*0.06))
      IF (KS1 .GT. 0) GO TO 485
      GO TO (477,478,477) IP
  477 CALL LETTER (4,LSIZE,IANG,XJ-0.10,YJ-VOSD,CBA1(J))
      CALL LETTER (4,LSIZE,IANG,XJ-0.10,YJ+VOSU,DENT(J))
      GO TO 490
  478 ALAB(1) = DENT(J)
      ALAB(2) = CBA1(J)
      CALL SYMBOL (XJ-0.10,YJ+VOSU,SIZEL,ALAB,IANG,8)
      GO TO 490
  485 CALL DECIMAL (DENT(J),NDEC)
      CALL NUMBER (XJ-0.15,YJ+VOSU,SIZEL,DENT(J),IANG,NDEC)
      IF (KS2 .LE. 0) GO TO 490
      CALL DECIMAL (CBA1(J),NDEC)
      CALL NUMBER (XJ-0.15,YJ-VOSD,SIZEL,CBA1(J),IANG,NDEC)
  490 CONTINUE
      WRITE (LP,495) NAP
  495 FORMAT (//I8,' STATIONS PLOTTED ON THIS MAP'//
     1// ' SAMPLES PLOTTED WITHIN EACH CLASS INTERVAL'                 // )
      WRITE (LP,496) (I,VALIM(I),L1(I),I=1,4)
  496 FORMAT (' C.I.',I3,'     LOWER LIMIT',F7.2,'     NO. SAMPLES',I6/)
C---- COMPLETE THE CHECK FOR 'BIT-LOSS'.
      CALL SPOT(HALFW2,TIHT+0.06,0.06,XXXX,0.0)
C---- LABEL PLOT WITH STATPAC COLUMN ID.  -------------------------------
      IF (KS1.LE.0 .OR. IDTYPE.NE.2) GO TO 497
      CALL SYMBOL (TMW+2.0,BOT+2.0,0.12,'VARIABLE PLOTTED:  ',90.,19)
      CALL SYMBOL (TMW+2.0,BOT+4.3,0.12,VID,90.,8)
C---- ADVANCE PEN FOR A NEW PLOT FRAME.  --------------------------------
  497 CALL PLOT(TMW+12.,0.0,-3)
      IF (IP .EQ. 2) CALL PLOT (0.0,0.0,+999)
      GO TO 295
  499 CALL CLOSEF (INC)
      CALL CLOSEF (INF)
      GO TO 20
  500 IF (IP .EQ. 2) WRITE (IOT,501)
  501 FORMAT ('BE PATIENT! HARD-WORKING VERSATEC IS COMPRESSING YOUR',
     1         ' PLOT FILES.')
      IF (NDS .LE. 0) GO TO 502
      IF (IP .NE. 2) CALL PLOT (0.0,0.0,999)
      IF (IP .EQ. 2) CALL PLOT (0.0,0.0,-999)
```

```
  502 IF (LP .EQ. IOT) GO TO 504
      CALL CLOSEF (LP)
  504 RETURN
C-----------------------------------------------------------------------
  505 FORMAT ('1PROGRAM TMPLOT',11X,'* UNIVERSAL TRANSVERSE MERCATOR PLO
     1T *',12X,'VERSION 1    (06/24/80)',18X,'DATE:',I3,'/',I2,'/',I2//
     2  ' DATA SET NO.',I3,9X,' DATA FILE:  ',A32///)
  510 FORMAT (' ',8('-'),' MAP LOCATION ',8('-'),5X,'----- MAP CONTENTS
     1-----',5X,'MAX SCALE',5X,'SORT STRIP',6X,2('PROXIMITY',5X)//' ',78X
     2,'WIDTH,INCHES    POINT SKIP    LABEL SKIP'/' ',8A4,3X,6A4,5X,'
     31/',F8.0,F11.1,F16.2,' MIN',F10.2,' MIN'/)
  515 FORMAT ('0',4X,'---  GEOGRAPHIC ACCEPTANCE LIMITS  ----'/' ',6X,'L
     1ATITUDE RANGE',6X,'LONGITUDE RANGE',6X,'VARIABLES',4X,10('-'),'  O
     2PTIONS  ',10('-'),/' ',2(4X,'MINIMUM   MAXIMUM'),5X,'KS1   KS2
     3OP1  OP2  OP3  OP4  OP5  '//' ',1X,2(F7.0,F3.0),1X,2(F7.0,F3.0),2X
     4,2I6,1X,5I5//)
  516 FORMAT (' ',38X,'PLOT CLASS INTERVALS',9X,'LABEL',5X,'LABEL'/
     1 ' ',37X,'1',6X,'2',6X,'3',6X,'4',8X,'VALUE',5X,'HEIGHT'/
     2 ' ',4X,'LOWER CLASS INTERVAL LIMITS:',2X,4(F7.2)/
     3 ' ',4X,'PLOT SYMBOLS:',17X,4I7,F11.2,F10.3//)
  517 FORMAT (' RATIO OF LATITUDE TO LONGITUDE MAP DISTANCE: ',F7.2/)
  518 FORMAT ('0NUMBER OF NUMERIC VALUES TO READ PER RECORD:',I8/
     1        '0INPUT FORMAT:    ',A80//)
  520 FORMAT ('1MAP NO.',I3,10X,'DATA SET NO.',I3,9X,7A4,A2,37X,'MAP DAT
     1E:',I3,2('/',I2)///' ',16('-'),'  MAP LOCATION  ',16('-'),10X,'---
     2-- MAP CONTENTS -----',16X,'MAP SCALE'//' ',12A4,10X,6A4,16X,'1/',
     3F8.0///)
  525 FORMAT (' ',6X,'LATITUDE RANGE',12X,'LONGITUDE RANGE',10X,'DISTANC
     1E POINTS ARE PLOTTED INTO MAP MARGINS'/'      ',2('MINIMUM   MAXIMU
     2M',10X),'  SOUTH',6X,'NORTH',6X,'EAST',7X,'WEST'/5X,2(I3,2I2,I6,2I
     32,10X),4(F6.2,5X),'DEGREES'///)
  530 FORMAT (' ',5X,'GRIDMARK',9X,'SPECIFIED MAP',14X,'BOTTOM MAP',8X,'
     1TITLE LINE',8X,'PT LABEL',8X,'MIN # POINTS'/' ',5X,'INTERVAL',9X,'
     2MARGIN SIZE (WD/HT)',8X,'MARGIN',12X,'HEIGHT',12X,'ANGLE',11X,'TO
     3PLOT MAP'/' ',I8,'MINUTES',6X,F5.2,' X',F6.2,'INCHES',F11.1,' INCH
     4ES',F12.2,' INCHES',I11,' DEG',I13//)
C
      END
```

```
      subroutine tmplotdriver
c
c---- version 1 for Honeywell Multics.   new_fortran    (12/18/79)
c
c---- Driver for plotrn program tmplot.
c----  Provides a call to the program with an argument indicating which
c----  plotter will be used.
c
c---- Output Argument:
c----    ip     = 1, Benson-Lehner Plotter
c----           = 2, Versatec Plotter
c----           = 3, Houston Plotter
c=====================================================================
      return
      entry tmplotb
      ip = 1
      go to 400
c
      entry tmplotv
      ip = 2
      go to 400
c
      entry tmplotn
      ip = 3
c---------------------------------------------------------------------
  400 call tmplot (ip)
      return
      end
```

```
      subroutine begin1(inc,ind,iot,lp,iout,iccf,indf,of2,nds,idata,***)
c
c---- version 1 for Honeywell Multics.     old_fortran     (05/09/80)
c
c---- determines attach & open input & output files.
c
c---- begin calls 3 subroutines:
c       openf   - opens files and attaches switchnames.
c       closef  - closes files and detaches switchnames.
c       ckinumof- accept and evaluate an integer numeric response
c
c---- Input Arguments:
c       inc    - unit number for control card file.
c       ind    - unit number for input data file.  (will reset,see below)
c       iot    - unit number for I/O to tty.
c       lp     - unit number for lineprinter output file.
c       iout   - unit number for 2nd output file.
c
c      note:  unit nos. 21 & 32 must be reserved for ind.
c
c----   Control of file opening
c
c       iccf   = 0  don't open a control file.
c              = 1  open control file for stream input (ASCII input).
c       indf   < 0  prompt for now to open the input file.
c              = 0  don't open an input data file.
c              = 1  open input file for stream input (ASCII input).
c              = 2  open input file for sequential input (binary input).
c       of2    = 0  don't open 2nd output file.
c              = 1  open 2nd output file for stream I/O (ASCII).
c              = 2  open 2nd output file for sequential I/O (binary).
c       nds    = 0  open line-printer output file.
c              > 0  don't open printer file.
c
c---- Output Arguments:
c      lp     = iot  printer output to user terminal.
c             = lp   printer output to disk file.
c      indf   = 1  Input file is ascii (character type) file.
c             = 2  Input file is binary file.
c      ind    = unit number of input data file.
c             = 21 for binary file.
c             = 32 for a character file.
c
c     ioa_$nnl  is a Multics operating system routine to write text
c               to TTY with no following carriage return.
c=====================================================================
      character*4 list
      character*32 icard,idata,olist,ofile2
      integer acci,acco,accb,accbo,blank,of2
      data acci,acco,accb,accbo/'si  ','sio ','sqi ','sqio'/
      data blank,iyes,iyesl,list/' ','Y','y','tty'/
c---------------------------------------------------------------------
   20 if (iccf .le. 0) go to 35
      call ioa_$nnl ('1/ CONTROL FILE : ')
      read (iot,30) icard
   30 format (a32)
      if (icard .eq. '          ') go to 600
      if (icard .eq. 'q'   ) go to 600
      if (icard .eq. 'quit') go to 600
   34 call openf (inc,icard,acci)
```

```
c
   35 if (indf .eq. 0) go to 40
      if (indf .ge. 0) go to 38
   36 ind = 21
      call ioa_$nnl (' IS INPUT FILE:  1)CHARACTER  2)STATPAC : ')
      call ckinumof (iot,idtype,$650,$650,$650)
      if (idtype.ne.1 .and. idtype.ne.2) go to 36
      if (idtype .eq. 1) ind=32
      indf = idtype
c
   38 call ioa_$nnl (' DATA FILE : ')
      read (iot,30) idata
      if (idata .eq. '          ') go to 650
      if (idata .eq. 'q'    ) go to 650
      if (idata .eq. 'quit') go to 650
   39 if (indf.eq.1) call openf (ind,idata,acci)
      if (indf.eq.2) call openf (ind,idata,accb)
c
   40 if (nds .gt. 0) go to 500
      call ioa_$nnl (' OUTPUT ON TTY? (Y/N): ')
      read (iot,42) ians
   42 format (a1)
      if (ians .ne. 'y') go to 70
   46 lp = iot
      olist = list
      go to 110
c
   70 call ioa_$nnl (' PRINTER OUTPUT FILE : ')
      read (iot,30) olist
      if (olist .eq. '          ') go to 700
      if (olist .eq. 'q'    ) go to 700
      if (olist .eq. 'quit') go to 700
      if (olist .eq. idata) go to 70
  100 if (lp .eq. 0) lp = 6
      call openf (lp,olist,acco)
c
  110 if (of2 .le. 0) go to 500
  120 call ioa_$nnl (' OUTPUT FILE : ')
      read (iot,30) ofile2
      if (ofile2 .eq. '    ') go to 750
      if (ofile2 .eq. 'q'    ) go to 750
      if (ofile2 .eq. 'quit') go to 750
      if (ofile2 .eq. idata) go to 120
  130 if (of2.eq.1) call openf (iout,ofile2,acco)
      if (of2.eq.2) call openf (iout,ofile2,accbo)
c
  500 return
  600 if (nds .eq. 0) go to 610
      return 1
  610 return 2
  650 if (iccf.gt.0) call closef (inc)
      go to 20
  700 if (indf.gt.0) call closef (ind)
      go to 35
  750 call closef (lp)
      go to 40
      end
```

```
      subroutine ckinumof (iot,i,*,*,*)
c
c---- version "of" for Honeywell Multics.    old_fortran  (06/05/79)
c
c---- Checks the numeric value contained in the input character
c----   string ans.
c---- Identical to ckinum.fortran but programs compiled with Multics
c----  old_fortran cannot accept non-standard returns from routines
c----  compiled in new_fortran.
c
c---- Input Arguments:
c----   iot    = unit no. for IO to tty.
c---- Output Argument:
c----   i      = integer (fixed pt) value contained in ans.
c---- Returns:
c----   normal = returns the integer (fixed pt) value from ans in i.
c----   *1     = ans = blank
c----   *2     = ans = 'q','quit','Q','QUIT','stop','STOP','exit','EXIT'
c----   *3     = not used
c=====================================================================
      character*35 ans
c
      read (iot,21) ans
   21 format (a35)
c
      if (ans .eq. '           ') return 1
      if (ans.eq.'q' .or. ans.eq.'quit') return 2
      if (ans.eq.'Q' .or. ans.eq.'QUIT') return 2
      if (ans.eq.'stop' .or. ans.eq.'STOP') return 2
      if (ans.eq.'exit' .or. ans.eq.'EXIT') return 2
c
      decode (ans,31) i
   31 format (v)
c
      return
      end



      subroutine decimal  (x,ndec)
c
c---- version 1 for Honeywell Multics.   new_fortran          (12/03/78)
c
c---- Determine the number of non-zero decimal digits to the right
c----   of the decimal point contained in the input value  x.
c---- Return an integer  ndec  whose value depends on the number
c----   of decimal places in the value of x.
c
c---- The output is in a form suitable for use as the input argument
c----   NDEC to CALCOMP subroutine NUMBER to control the number of
c----   decimal places of the output format.
c
c---- Input Arguments:
c----   x    = a real (floating point) number
c
c---- Output Arguments:
c----   ndec = -1    - x has no decimal part
c----        = 1-8   - x has from 1 to 8 decimal places
c=====================================================================
      ndec = -1
      y = abs(x)
c
      do 200 i=1,8
      if (amod(y,1.0) .gt. 0.00001) go to 30
      go to 500
   30 ndec = i
      y = 10.0 * y
  200 continue
  500 return
      end
```

```
      subroutine dms (dec,deg,min,sec)
c
c---- Written by Roger W. Bowen, USGS, Reston, Virginia
c
c---- version 1 for Honeywell Multics.     new_fortran      (09/25/79)
c
c     dms(dec,deg,min,sec) accepts the floating point number dec and
c     converts it to degrees(deg), minutes(min), and seconds(sec) which
c     are all of type integer and designed to be printed out respective-
c     ly under i3, a2, and a2 format specifications.
c=====================================================================
      integer pict(60),deg,sec,min,dir,dc(4)
      data pict/
     1'00','01','02','03','04','05','06','07','08','09','10',
     2'11','12','13','14','15','16','17','18','19','20','21',
     3'22','23','24','25','26','27','28','29','30','31','32',
     4'33','34','35','36','37','38','39','40','41','42','43',
     5'44','45','46','47','48','49','50','51','52','53','54',
     6'55','56','57','58','59'/
      data dc/'S','E','N','W'/
      deg=dec
      r=60.*abs(dec-deg)
      min=r
      r=60.*(r-min)
      sec=r
      if (r-sec.ge..5) sec=sec+1
      if (sec.lt.60) go to 100
      sec=0
      min=min+1
      if (min.lt.60) go to 100
      min=0
      deg=deg+sign(1.,dec)
  100 min=pict(min+1)
      sec=pict(sec+1)
      if (deg.lt.0) deg=-deg
      return
      end
```

```
      subroutine dmsi (icoord,ideg,imin,isec)
c
c---- Written by George Van Trump Jr., USGS, Denver, Colo.
c
c---- dmsi accepts icoord (lat or lon) as an integer values where:
c----   deg = digits 5,6 (&7)      (Digits numbered from R to L)
c----   min = digits 3 & 4
c----   sec = digits 1 & 2
c---- dmsi returns the coordinate as integer values for deg, min & sec.
c=====================================================================
      ideg = icoord/10000
      imin = mod(icoord,10000)/100
      isec = mod(icoord,100)
      return
      end
```

```
      subroutine getlst (unit,ir,id,loc,x,ia,m,*)
c
c---- Written by George Van Trump Jr.,  USGS,  Denver,  Colo.
c
c---- Read one data row of a USGS Statpac data-set.
c---- Arguments:
c----  unit   - input unit no.
c----  ir     - sequence number of the row.
c----  id     - 4-word alpha/numeric rowid as 2 parts.
c----           words 1&2- Primary rowid
c----                 3&4- Secondary rowid
c----  loc    - lat & lon as single integers of DMS
c----  x      - array of data values.
c----  ia     - array of 1-char alpha qual. codes.
c----  m      - no. Statpac data col. in this data-set.
c----  *1     - Unexpected EOF.
c=====================================================================
      integer unit
      dimension id(4),loc(2),x(m),ia(m),iy(20)
      n=(m+9)/10
      call get (unit,ir,id,loc,x,m,iy,n,$20)
      call pack (1,ia,m,iy,n)
      return
   20 return 1
      end
      subroutine get (unit,ir,id,loc,x,m,iy,n,*)
c
c---- Written by George Van Trump,  Jr.,  USGS,  Denver,  Colo.
c=====================================================================
      integer unit
      dimension istr(7),x(m),iy(n),id(4),loc(2),idr(4),locr(2)
      equivalence (istr(1),i),(istr(2),idr(1)),(istr(6),locr(1))
      it=iabs(unit)
      read (it,end=20) istr,x,iy
      ir=i
      do 10 i=1,4
   10 id(i)=idr(i)
      loc(1)=locr(1)
      loc(2)=locr(2)
      return
   20 if (unit.gt.0) print 30,it
   30 format (" ERROR ... An End-Of-File was encountered while reading",
     1 " Unit",i3)
      return 1
      end
      subroutine pack (icode,ia,no,iy,n)
c
c---- Written by Geroge Van Trump Jr.,  USGS,  Denver,  Colo.
c
c     This subroutine packs and unpacks 10 qualifying codes into 1
c     word,  using as many words as necessary to pack "no" codes.
c        icode = 0 packs
c              = 1 unpacks
c        ia    = array of unpacked codes ("no" words of 1 code per word)
c        no    = number of codes contained in array "ia"
c        iy    = array of packed codes ("n" words of 10 codes per word)
c        n     = number of words in array "iy" containing packed codes
c=====================================================================
      dimension ic(8),ia(200),iy(20),icl(7)
      data ic/" ","B","L","N","G","T","H","*"/
```

```
      data icl/" ","b","l","n","g","t","h"/
      n=(no+9)/10
      n1=10*n
      if (icode.ne.0) go to 100
c
c ... pack codes.
c
      do 5 i=1,no
      do 3 j=2,7
      if (ia(i).ne.icl(j)) go to 3
      ia(i)=ic(j)
      go to 5
    3 continue
    5 continue
      do 10 i=1,n
   10 iy(i)=0
      i=0
      do 50 j=1,n1
      if (mod(j-1,10).eq.0) i=i+1
      if (j.gt.no) go to 45
      do 40 k=1,7
      kx=k
      if (ia(j).eq.ic(k)) go to 50
   40 continue
   45 kx=8
   50 iy(i)=8*iy(i)+kx-1
      return
c
c ... unpacks codes.
c
  100 k=n1+1
      i=n+1
      do 110 j=1,n1
      k=k-1
      if (mod(j-1,10).eq.0) i=i-1
      if (k.gt.no) go to 110
      ll=mod(iy(i),8)
      ia(k)=ic(ll+1)
  110 iy(i)=iy(i)/8
      return
      end
```

```
      subroutine juldat (iy,im,id)
c
c---- Written by Geroge Van Trump Jr., USGS, Denver, Colo.
c
c---- version 1 for Honeywell Multics.      new_fortran
c
c---- Get the current date from the Multics system clock.
c=====================================================================
      external date_prog (descriptors)
      character*6 date
      call date_prog (date)
      decode (date,10) iy,im,id
   10 format (3i2)
      return
      end
```

```
date_prog:proc (new_date);

/*    Written by Geroge Van Trump Jr., USGS, Denver, Colo.

   PL/1 routine to get the current date from the Multics system clock. */
/* ================================================================== */
dcl new_date char(6);
dcl date builtin;
new_date=date;
return;
end;
```

```
      SUBROUTINE LABSKIP (KS1,OP5,RATIO,SKLAB,N,NPP,LP)
C
C---- VERSION 1 FOR HONEYWELL MULTICS.   NEW_FORTRAN       (01/11/78)
C
C---- TEST THE DISTANCE BETWEEN EACH GROUP OF NPP POINTS IN THE LIST
C----   AND BLANK THE LABEL ARRAYS DENT AND CBA1 IF THE POINTS LIE
C----   CLOSER THAN SKLAB.
C----   COMPARISON MADE IN TERMS OF LONGITUDE MAP DISTANCE.
C
C---- SELECTED VARIABLES:
C----  KS1     .LE.0, LABELS FOR THE POINTS ARE ALPHA-NUMERIC
C----          .GT.0, LABELS FOR THE POINTS ARE NUMERIC
C----  OP5     = 0,  LIST ANY LABELS BLANKED OUT
C----          .NE.0, DON'T LIST THE BLANKED OUT LABELS
C----  RATIO   = RATIO OF MAP DISTANCES OF 1 DEG LAT TO 1 DEG LONG
C----  SKLAB   = MINIMUM ACCEPTABLE SPACING (DECIMAL DEGREES)
C----  N       = NUMBER OF DATA POINTS
C----  NPP     = NUMBER OF PREVIOUS POINTS TO CHECK
C----  LP      = UNIT NUMBER FOR OUTPUT TO LINEPRINTER FILE
C=====================================================================
      INTEGER  F280(8), F280A(8), F280N(8), OP5
      DIMENSION PRVLAT(20),PRVLON(20)
      COMMON  /LABELBK/ DENT(3000),CBA1(3000),X(3000),Y(3000)
      DATA F280A/'(1H ',',I5,',',2(3X',',2A4',',2F1',',2.5,',',4X))',',   '/
      DATA F280N/'(1H ',',I5,',',2(3X',',2F1',',2.2,',',2F12',',.5,4',',X))'/
      DATA  BLANK/' '/
C---------------------------------------------------------------------
      DO 269 II=1,NPP
      PRVLAT(II) = 500.
  269 PRVLON(II) = 500.
      NL=0
      IF (OP5 .NE. 0) GO TO 276
      IF (KS1 .LE. 0) GO TO 272
      DO 271 II=1,8
  271 F280(II) = F280N(II)
      GO TO 274
  272 DO 273 II=1,8
  273 F280(II) = F280A(II)
  274 WRITE (LP,275)  SKLAB
  275 FORMAT ('1  LABELS OMITTED IN PROXIMITY LABEL-SKIP TEST.',15X,'SKL
     1AB =',F6.4,' DEGREES'/)
  276 NPT = NPP + 1
      DO 290 J= NPT,N
C---- NOTE: FIRST NPP PTS. IN THE DATA ARRAYS NOT CHECKED.
      DO 278 II=1,NPP
      JMX = (NPP+1) - II
      JMY = J - JMX
      DIFLAT = ABS(PRVLAT(JMX)-Y(J)) / RATIO
      DIFLON = ABS(PRVLON(JMX)-X(J))
      IF (DIFLAT.GE.SKLAB .OR. DIFLON.GE.SKLAB) GO TO 278
      NL = NL + 1
      IF (OP5.EQ.0) WRITE (LP,F280) NL,DENT(JMY),CBA1(JMY),Y(JMY),
     1   X(JMY),DENT(J),CBA1(J),Y(J),X(J)
      DENT(JMY) = BLANK
      CBA1(JMY) = BLANK
      DENT(J)   = BLANK
      CBA1(J)   = BLANK
  278 CONTINUE
      NPP1 = NPP - 1
      DO 288 II=1,NPP1
      JMX = 6 - II
      JMY = JMX + 1
      PRVLAT(JMX) = PRVLAT(JMY)
  288 PRVLON(JMX) = PRVLON(JMY)
      PRVLAT(1) = Y(J)
      PRVLON(1) = X(J)
  290 CONTINUE
      RETURN
      END
```

```
      subroutine openfl (unit,filnaml,filtyp)
c
c---- version "l" for Honeywell Multics.  new_fortran      (04/07/80)
c
c---- Program to do file attaching and opening for FORTRAN Programs
c----   using the Multics vfile_ I/O module.
c---- This routine provides continuity between use of the Multics
c----  command-level I/O commands (thru calls to subroutine iox_),
c----  and the Multics FORTRAN open & close statements.  It allows
c----  the calling program to specify file opening type by the
c----  same char strings used in calls to iox_.  The actual attaching
c----  opening, & closing are done with the FORTRAN language open & close
c----  statements so that both the FORTRAN file table, and the system's
c----  file (_iocb) table will be updated.
c
c---- Assumptions:  Several simplifying assumptions are made.
c----  File types used will be similar to what Multics FORTRAN would
c----   use for default openings: "stream" for formatted ("character")
c----   I/O, "sequential" for unformatted ("binary") I/O.
c----   keyed_sequential & direct openings are only made for "unformatted"
c----   and sequential  (sqi,sqo,sqio,squ) are sequential, not direct.
c----  Unit no. 6,7,8 when opened for stream or sequential output
c----   will have carriage control; otherwise cc  will be off.
c----  "binary stream" openings will not be made.
c
c----   input pathnames contained in character-type variable .
c----   entry point openf:
c----     filename argument must be character*32 in main prog.
c----   entry point openfl:
c----     filename argument must be character*80 in main prog.
c
c---- control parameters:
c      1. iunit: integer value of the FORTRAN unit to be opened.
c                example: 10  (same as "file10")
c      2. filnam: an alpha string containing the filename (ex: "seds.dat")
c
c      3. nt    :  an integer value corresponding to one of the following
c                  file types.
c         filtyp:  a 4-char abbrev for one of the following file types.
c              1  "si"  - stream_input               formatted I
c              2  "so"  - stream_output              formatted O
c              3  "sio" - stream_input_output        formatted I/O
c              4  "sqi" - sequential_input           unformatted I
c              5  "sqo" - sequential_output          unformatted O
c              6  "sqio"- sequential_input_output    unformatted I/O
c              7  "squ" - sequential_update          unformatted I/O
c              8  "ksqi"- keyed_sequential_input
c              9  "ksqo"- keyed_sequential_output
c             10  "ksqu"- keyed_sequential_update
c             11  "di"  - direct_input
c             12  "do"  - direct_output
c             13  "du"  - direct_update
c
c=====================================================================
      character*4  filtyp
      character*8  io
      character*12 acc,fform
      character*32 filnam
      character*80 filnaml
      logical  bs,cc
```

```
      integer  unit
      data  iot/0/
c-----------------------------------------------------------------------
      namel = 2
      go to 35
c
      entry openf (unit,filnam,filtyp)
      namel = 1
   35 call iofiltyp (filtyp,nt,$520)
      bs = .false.
      cc = .false.
      if (nt.ne.2 .and. nt.ne.3 .and. nt.ne.5 .and. nt.ne.6) go to 40
      if (unit.ge.6 .and. unit.le.8) cc=.true.
c
   40 if (unit .le. 0) go to 510
      if (nt.lt.1 .or. nt.gt.13) go to 520
c
      fform = "unformatted"
      if (nt .le. 3) fform="formatted"
c
      acc = "sequential"
      if (nt .gt. 7) acc="direct"
c
      io = "inout"
      if (nt.ne.1 .and. nt.ne.4 .and. nt.ne.8 .and. nt.ne.11) go to 75
      io = "in"
      cc = .false.
      go to 100
   75 if (nt.eq.2 .or. nt.eq.5 .or. nt.eq.9 .or. nt.eq.12) io="out"
c
  100 go to (110,120) namel
  110 open (unit,file=filnam,form=fform,access=acc,mode=io,
     1      binary stream=bs,carriage=cc,err=550)
      return
  120 open (unit,file=filnaml,form=fform,access=acc,mode=io,
     1      binary stream=bs,carriage=cc,err=550)
c
      return
c-----------------------------------------------------------------------
      entry closef  (unit)
c
      close (unit,err=600)
c
      return
c-----------------------------------------------------------------------
  510 write (iot,511) unit
  511 format ('ERROR: Bad Unit Number: ',i10)
      go to 590
  520 write (iot,521) nt
  521 format ('ERROR: Bad Value For I/O-Type: ',i10)
      go to 590
  550 go to (551,552) namel
  551 write (iot,553) unit,filnam
  552 write (iot,554) unit,filnaml
  553 format ('ERROR In Opening Unit',i3,'.  File:  ',a32)
  554 format ('ERROR In Opening Unit',i3,'.  File:  ',a80)
      go to 590
  590 close (unit,err=900)
c-----------------------------------------------------------------------
  600 write (iot,601) unit
```

```
  601 format ('ERROR Closing Unit:',i3)
  900 return
      end
      subroutine iofiltyp (filtyp,iotype,*)
c
c---- version 1 for Honeywell Multics.       new_fortran     (04/18/80)
c
c---- Accept a 4-char. abbrev specifying a file I/O type.
c---- Return a corresponding integer value.
c=====================================================================
      character*4 filtyp,types(13)
      data  iot/0/
      data types/'si  ','so  ','sio ','soi ','sqo ','sqio','squ ',
     1           'ksqi','ksqo','ksqu','di  ','do  ','du  '/
c---------------------------------------------------------------------
      do 50 i=1,13
      if (filtyp .ne.types(i)) go to 50
      iotype = i
      go to 100
   50 continue
      iotype = 0
      write (iot,61) filtyp
   61 format ('ERROR: Bad I/O Specified:  ',a4)
      return 1
  100 return
      end
```

```
      subroutine pcheck  (xfact,yfact)
c
c---- version 1 for Honeywell Multics.     new_fortran   (03/21/79)
c
c---- plot a 10 X 20 inch scale check on an off line plotter.
c---- Variables:
c----  xfact & yfact are x & y coordinate fudge-factors to correct for
c----  inaccurate scale adjustment of the plotter.
c
c---- Program uses the following system-maintained plot routines:
c----   plot, spot, number, symbol
c=====================================================================
      external symbol (descriptors),  spot (descriptors)
      if (xfact .eq. 0.0) xfact=1.0
      if (yfact .eq. 0.0) yfact=1.0
c
c---- Begin check for bit skips.
      call spot (2.0,2.0,0.24,'+',0.0)
c---- Begin plotting the scale check.
      x10 = 10.0 * xfact
      y10 = 10.0 * yfact
      y20 = 20.0 * yfact
      call plot (0.1,0.0,3)
      call plot (0.3,0.0,2)
      call plot (0.0,y10,2)
      call plot (0.2,y10,2)
      call plot (0.3,y10,2)
      call plot (0.0,y20,2)
      call plot (0.2,y20,2)
      call symbol (0.25,18.90,0.06,'X',90.,1)
      call number (0.25,19.00,0.06,yfact,90.,6)
      call plot (0.0,0.0,3)
      call plot (x10,0.0,2)
      call plot (x10,0.1,2)
      call symbol (9.90,0.25,0.06,'X',0,1)
      call number (9.0,0.25,0.06,xfact,0.0,6)
c---- Complete plotting the check for skipped bits.
      call spot (2.0,2.0,0.18,'X',0.0)
c---- Shift paper for the next plot frame.
      call plot (13.0,0.0,-3)
      return
      end
```

```
      subroutine readcr (iot,inf,ifmt,nv,ks1,ks2,kq,*)
c
c---- version 1 for honeywell multics.        old_fortran  (12/17/79)
c
c---- subroutine called by plot programs (versions of tmplot)
c---- to read one record of ascii character input data.
c
c---- Selected Variables:
c----  iot    - I/O unit number for tty.
c----  inf    - Input unit to use to read the data.
c----  ifmt   - Format to use to read the data.
c----  nv     - Number of numeric (float pt) variables to read.
c----  ks1    - Which variable to load into var1.
c----  ks2    - Which variable to load into var2.
c----  kq     = 1, read alternating qual-code & variable
c----         = 2, data contains no qual-codes.
c----  *1     - Unexpected EOF.
c=====================================================================
      character*1  form(80)
      character*80 ifmt
      dimension itemp(1),var(100),iq(100)
      common /readbk/ sta(2),tag(2),var1,var2,qcode1,qcode2,
     1                tdeg,tmin,tsec,gdeg,gmin,gsec
c---------------------------------------------------------------------
      var1 = 0.0
      var2 = 0.0
      qcode1 = '    '
      qcode2 = '    '
      tag(1) = '    '
      tag(2) = '    '
      if (kq .ne. 1) kq=2
c
      if (nv .gt. 0) go to 40
c---- Read data containing only record id, lat and lon.
      read (inf,ifmt,end=310) sta,tdeg,tmin,tsec,gdeg,gmin,gsec
      go to 200
c---------------------------------------------------------------------
c---- Read data including numeric variables.
   40 go to (45,41) kq
c---- Data contain no qualifiers.
   41 read (inf,ifmt,end=310) sta,tdeg,tmin,tsec,gdeg,gmin,gsec,
     1                        (var(j),j=1,nv)
      if (ks1 .le. 0) go to 200
      var1 = var(ks1)
c
      if (ks2 .le. 0) go to 200
      var2 = var(ks2)
      go to 200
c---------------------------------------------------------------------
c---- Data include qualification codes.
   45 read (inf,ifmt,end=310) sta,tdeg,tmin,tsec,gdeg,gmin,gsec,
     1                        (iq(j),var(j),j=1,nv)
      if (ks1 .le. 0) go to 200
      var1 = var(ks1)
      itemp(1) = iq(ks1)
      decode (itemp,71) qcode1
c
      if (ks2 .le. 0) go to 200
      var2 = var(ks2)
      itemp(1) = iq(ks2)
```

```
      decode (item3,71) qcode2
   71 format (a1)
c
  200 return
  310 return 1
c-----------------------------------------------------------------------
      entry readcr1 (iot,inc,lp,nv,kq,ifmt,*)
c
c---- If variable ifmt is blank, read a standard FORTRAN format
c----  specification from an ascii character file and check the format
c----  to see that it is enclosed in parrens.
c
c---- Selected Variables:
c----  inc  - input unit for file containing the format specification.
c----  nv   - number of numeric data values to read per data record.
c----  lp   - unit number for output to lineprinter file
c=======================================================================
      read (inc,415,end=520) nv,kq
  415 format (i6,i4)
      if (nv.lt.0 .or. nv.gt.100) go to 525
      if (kq .eq. 0) kq=2
      if (kq.lt.1 .or. kq.gt.2) go to 528
c
  422 read (inc,423,end=520) ifmt
  423 format (a80)
      decode (ifmt,424) form
  424 format (80a1)
      do 426 i=1,77
      if (form(i) .eq. ' ') go to 426
      if (form(i) .eq. '(') go to 428
      go to 430
  426 continue
      go to 430
c
  428 do 429 i=1,75
      j = 81 - i
      if (form(j) .eq. ' ') go to 429
      if (form(j) .eq. ')') go to 443
      go to 430
  429 continue
c
  430 write (iot,431) ifmt
  431 format (' ERROR In Input Format. Missing Parren.'/a80)
      go to 520
c
c
  443 return
  520 return 1
  525 write (iot,526) nv
  526 format ('Error: Bad Value For No. Of Variables:',i8)
      return 1
  528 write (iot,529) kq
  529 format ('Error: Bad Value KQ (select reading of qual codes):'i8)
      return 1
      end
```

```
      subroutine readsp (inf,iot,lp,ks1,ks2,nsc,*)
c
c---- version 1 for Honeywell Multics.     old_fortran   (12/11/77)
c
c---- Read one data record (=row =sample) of a USGS STATPAC data file.
c---- Return primary & secondary row-ids, lat & lon (deg,min,sec as
c----    real no.), and 2 numeric values with corresponding qualification
c----    codes (as specified by ks1 and ks2).
c
c---- Note:
c---- readsp assumes that the Statpac file has sample no. selected
c----  as the primary row-id, usually with a lab no. as the
c----  secondary row-id.
c
c---- readsp requires these routines from >udd>Cmptappl>gvtlib>obj
c----    getlst, get, oack
c----  These routines (written by George Van Trump, Jr., USGS) read
c----   and unpack one data record of a USGS Statpac data file.
c
c---- Selected Variables:
c---- inf      - unit number for data input
c---- iot      - unit number for I/O to TTY
c---- lp       - unit number for output to lineprinter file
c---- ks1,ks2 - tell which data values to return
c---- nsc      - number of data columns in the Statpac input file
c---- sta      - 8-character alpha-numeric primary row-id
c---- tag      - 8-character alpha-numeric secondary row-id
c---- var1,var2 - return the data values requested by ks1,ks2
c---- *1       - non-standard return for unexpected end of file
c=====================================================================
      common /readsk/ sta(2),tag(2),var1,var2,qcode1,qcode2,
     1                tdeg,tmin,tsec,gdeg,gmin,gsec
      dimension id(4),idds(2),cid(100,2),vid(2),loc(2),v(100),iq(100)
c---------------------------------------------------------------------
      call getlst (inf,ir,id,loc,v,iq,nsc,*650)
      decode (id(1),50) sta(1)
      decode (id(2),50) sta(2)
      decode (id(3),50) tag(1)
      decode (id(4),50) tag(2)
   50 format (a4)
c
      if (ks1 .ge. 1) var1=v(ks1)
      if (ks2 .ge. 1) var2=v(ks2)
      if (ks1 .ge. 1) decode (iq(ks1),60) qcode1
      if (ks2 .ge. 1) decode (iq(ks2),60) qcode2
   60 format (a1)
c
c---- convert lat & lon from integer to deg,min,sec
      call dmsi (loc(1),la1,la2,la3)
      call dmsi (loc(2),lo1,lo2,lo3)
      tdeg = la1
      tmin = la2
      tsec = la3
      gdeg = lo1
      gmin = lo2
      gsec = lo3
      return
c
  650 write (lp,652)
      write (iot,652)
```

```
  652 format (///' unexpected EOF while reading input STATPAC file ')
      return 1
c-----------------------------------------------------------------------
      entry readsp1 (iot,inf,lp,ks1,ks2,nsr,nsc,vid,*)
c
c---- read statpac data-set header record.
c
c---- Selected Variables:
c----  idds  - 3-character alpha-numeric Statpac Data-Set ID
c----  nsr   - number of data rows in this Statpac Data-Set
c----  cid   - array of 8-character data column ID's
c=======================================================================
      read (inf,end=650) idds,nsr,nsc,((cid(i,k),k=1,2),i=1,nsc)
c
      write (lp,135) idds,nsr,nsc
      write (iot,136) idds,nsr,nsc
  135 format (' DATA-SET:   ',2a4,i7,' ROWS.',i5,' COLUMNS.'/)
  136 format (' Data-Set:  ',2a4,i7,' Rows.',i5,' Columns.')
      if (ks1.le.nsc .and. ks2.le.nsc) go to 142
      write (lp,141)  nsc
      write (iot,141) nsc
  141 format (///' COL NO OF SELECTED VAR GT NO STATPAC INPUT COLS=',i5)
      return 1
c
  142 if (ks1 .le. 0) return
      vid(1) = cid(ks1,1)
      vid(2) = cid(ks1,2)
      write (lp,143)  vid
      write (iot,144) vid
  143 format (' COLUMN ID OF SELECTED STATPAC VARIABLE: ',2a4///)
  144 format (' Column ID of Selected Statpac Variable: ',2a4)
      return
      end
```

```
      subroutine sortm (jn,kasc)
c
c---- version 1 for Honeywell Multics.   new_fortran       (02/07/78)
c
c---- order points by longitude (x coordinate) within a plot strip.
c
c---- Selected Variables:
c----  jn   - no. pts. within current group to be sorted
c----  kasc - .ge.0, sort in ascending order of xx
c----         .lt.0, sort in descending order of xx
c----  indx - locations of points in original data arrays
c----  r    - temporary storage
c----  xx   - x coordinates (long in dec. deg) of pts to sort
c=====================================================================
      common /sortbk/ xx(3000),indx(3000),ind(3000),p(3000),q(3000)
      dimension index(3000), r(3000), jdex(3000)
c---------------------------------------------------------------------
      n1=jn+1
      do 5 j=1,jn
      index(j)=j
    5 r(j)=xx(j)
c
      do 25 j=2,jn
      jm=j-1
      jp=j+1
      xj=xx(j)
c
      do 20 k=1,jm
      if (xj-r(k)) 10,10,20
   10 jk=j-k
      do 15 l=1,jk
      lr=jp-l
      lr1=lr-1
      index(lr)=index(lr1)
   15 r(lr)=r(lr1)
      index(k)=j
      r(k)=xj
      go to 25
   20 continue
   25 continue
      do 30 j=1,jn
      id=index(j)
   30 jdex(j)=indx(id)
c
c---- Order index values for this strip (L to R, or R to L)
      if (kasc) 45,35,35
   35 do 40 j=1,jn
   40 indx(j)=jdex(j)
      go to 55
   45 do 50 j=1,jn
      l=n1-j
   50 indx(j)=jdex(l)
   55 return
      end
```

```
      subroutine strpsort (delta,jdiff,ln,lp,n,ymin)
c
c---- version 1 for Honeywell Multics.   old_fortran       (02/07/78)
c
c---- Sort data for program tmplot by latitude (x) into horizontal plot
c----   strips.  (each strip runs the width of the plot)
c---- Load array ind containing array index locations of data in arrays
c----    dent,cha1,x,y in the order the points are to be plotted;
c----    then re-order the data according to the values in ind.
c
c---- requires subroutine:   sortm
c
c---- Arguments:
c----  delta - width of plot strips (decimal degrees)
c----  jdiff - number of plot strips this map is divided into
c----  ln    - total no. pts. loaded into plot strips so far.
c----  n     - total number of data points to be plotted.
c----  ymin  - latitude of bottom of current strip (decimal degrees)
c---- Selected Variables:
c----  cha1  - data value for the points
c----  dent  - data value for the points
c----  indx  - position w/in the origional data arrays of pts in
c----          current strip.
c----  jj    - main array index for last point assigned to this strip
c----  jn    - no. points loaded into current strip
c----  kasc  - flag tells to plot strip R to L, or L to R
c----  njf   - no. points left to be placed in a strip
c----  p & q - temporary storage
c----  x & y - lat & lon coordinates of data pts (decimal degrees)
c----  qmin  - lat of top of current strip (decimal degrees)
c=====================================================================
      common /sortbx/ xx(3000),indx(3000),ind(3000),p(3000),q(3000)
      common /labelbx/ dent(3000),cha1(3000),x(3000),y(3000)
c---------------------------------------------------------------------
      ln = 0
      kasc = -1
      n1 = n + 1
      do 185 l=1,jdiff
      jmin=ymin+delta
      jn=0
c---- find all pts w/in this strip & load xx & indx for sorting by long.
c----   w/in the strip.
      do 155 j=1,n
      yj=y(j)
      if (yj-ymin) 155,150,145
  145 if (yj-qmin) 150,155,155
  150 jn=jn+1
      indx(jn)=j
      xx(jn)=x(j)
      jj=j
      ln=ln+1
  155 continue
c
      write (lp,160) l,ymin,qmin,jn
  160 format (' ','STRIP',i5,10x,'FROM',f7.2,3x,'to',f7.2,3x,'CONTAINS',
     1i5,3x,'STATIONS')
c
      ymin=qmin
      njf=n1-ln+jn
      if (jn) 185,185,165
```

```
  165 if (jn-1) 185,170,175
  170 ind(njf-1)=jj
      go to 185
  175 kasc=-kasc
c
c---- Sort the data within this plot strip by longitude  --------------
      call sortm (jn,kasc)
c
      do 180 j=1,jn
      jl=njf-j
  180 ind(jl)=indx(j)
  185 continue
c-----------------------------------------------------------------------
c---- Reorder arrays x,y,dent,cbal -sorted by lat & lon into plot strips
      do 195 j=1,n
      id = ind(j)
      p(j) = x(id)
  195 q(j) = y(id)
      do 205 j=1,n
      x(j) = p(j)
  205 y(j) = q(j)
      do 215 j=1,n
      id=ind(j)
      q(j)=cbal(id)
  215 p(j)=dent(id)
      do 220 j=1,n
      cbal(j)=q(j)
  220 dent(j)=p(j)
c-----------------------------------------------------------------------
      return
      end
```

```
      subroutine unicholtrof (iot,ip,xf,yf,xlmax,ylmax,csmin,*)
c
c---- version 1 for Honeywell Multics.     old_fortran   (12/15/79)
c
c---- Subroutine for plot programs to set appropriate values for
c----   constants for the specified plotter.
c
c---- Arguments:
c----  iot      = unit no. for tty I/O.
c----  ip       = 1, Benson-Lehner Plotter
c----           = 2, Versatec Plotter
c----           = 3, Houston Plotter
c----  xf       = Fudge factor to correct x axis length.
c----  yf       = Fudge factor to correct y axis length.
c----  xlmax    = max possible plot dimension - x direction. (inches)
c----  ylmax    = max possible plot dimension - y direction. (inches)
c----  csmin    = min. permissible character height (inches).
c----  *        = Bad input.  ip not 1,2, or 3.
c=====================================================================
      if (ip.lt.1 .or. ip.gt.3) return 1
c
      go to (100,150,200) ip
c
c---- Benson-Lehner
  100 xf = 1.00
      yf = 1.00
      xlmax = 100.0
      ylmax = 28.8
      csmin = 0.06
      return
c
c---- Versatec
  150 xf = 1.0020300
      yf = 1.0000135
      xlmax = 120.0
      ylmax = 39.0
      csmin = 0.035
      return
c
c---- Houston
  200 xf = 1.00
      yf = 1.00
      xlmax = 120.0
      ylmax = 34.5
      csmin = 0.06
      return
      end
```