NASA TM-85648

NASA Technical Memorandum 85648

NASA-TM-85648 19850022696

COMPUTER PROGRAM FOR PLOTTING AND FAIRING WIND-TUNNEL DATA

Harry L. Morgan, Jr.

July 1983

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

SUMMARY

This report contains a detailed description of the Langley computer program PLOTWD which plots and fairs experimental wind-tunnel data. The program was written for use primarily on the Langley CDC computer and CALCOMP plotters. The fundamental operating features of the program are that the input data are read and written to a random-access file for use during program execution, that the data for a selected run can be sorted and edited to delete duplicate points, and that the data can be plotted and faired using tension splines, least-squares polynomial, or least-squares cubic-spline curves. The most noteworthy feature of the program is the simplicity of the user-supplied input requirements. Several subroutines are also included that can be used to draw grid lines, zero lines, axis scale values and labels, and legends. A detailed description of the program operational features and each subprogram are presented. The general application of the program is also discussed together with the input and output for two typical plot types. A listing of the program code, user-guide, and output description are presented in appendices. The program has been in use at Langley for several years and has proven to be both easy to use and versatile.

INTRODUCTION

Although the primary function of the existing NASA Langley wind-tunnel complex is the same as that of its earlier predecessor NACA, the capabilities of the existing facilities are far superior to those of its predecessor with regard to the relative speeds obtainable, Reynolds number range, and the types and quality of data that can be taken. The advent of modern strain-gage balances, pressure and acoustic transducers, flow visualization systems, engine simulators, attitude measuring devices, and tunnel control and data acquisition systems have resulted in a tremendous increase in the amount of data generated during a typical test program. The recent "explosion" in mini-computer technology has further enhanced the ability to acquire even more data. Without the availability of both on- and off-line high-speed digital and analog computers, the test engineers' ability to acquire, reduce, and analyze this large volume of data would be totally inadequate.

Following a typical wind-tunnel test program, the test engineer is faced with a large amount of data that must be plotted and faired for analysis and report presentation. During the earlier NACA period, these data were plotted and faired manually which required the services of a rather large supporting staff. This manual operation was slow, very tedious and monotonous, and often very inaccurate. However, since the introduction of high-speed computers with peripheral mechanical and optical plotting devices, manual plotting is no longer necessary or desirable. The Langley computing complex currently consists of some of the advanced CDC computers and CALCOMP plotting devices available. The speed, quality, and accuracy of the plots generated by the CALCOMP devices are generally far superior to those generated manually. The available CALCOMP computer software package is extremely versatile and relatively easy to use.

The purpose of this report is to describe the features and application of a computer program written specifically to plot and fair wind-tunnel generated data. This program has the internal designation "PLOTWD" and was written strictly for use on the Langley CDC computer and CALCOMP plotting systems. The program can, however, be adapted to other computer and plotting systems provided the computer has random-access file capability and the plotters have comparable system software routines. This computer program has been used extensively by personnel in the Langley 4- by 7-Meter and 7- by 10-Foot High-Speed wind tunnels for several years and has proven to be very reliable, flexible, and easy to use. In fact, one of the most noteworthy features of this program is the simplicity of the input requirements.

This report contains a discussion of the overall program capabilities and operational structure. A detailed description is included of each subprogram contained in the program. A listing of the computer code is presented in appendix A. Only the subprograms that were written specifically for this program are listed in the appendix. The external subroutines called that are part of the Langley CALCOMP software package are not listed. A description of the user-supplied input requirements and a description of a sample output are presented in appendixes B and C, respectively.

SYMBOLS

$a_0, a_1, a_2, \ldots, a_m$	polynomial coefficients
c_i, d_i, e_i, f_i	coefficients of spline equation
h_i	increment in spline independent variable such as $(x - x_i)$, $(y - y_i)$, or $(t - t_i)$
M	order of polynomial
N	number of points
P(x)	polynomial equation
S	sum-of-squares of differences
SP	cubic-spline smoothing parameter
W	weighting factor
t	independent variable
x	x-axis dependent variable
x"	d^2x/dt^2
y	y-axis dependent variable
y"	d^2y/dt^2
$\bar{\sigma}$	spline tension factor
δ	increment

Subscripts:

o	origin
p	plotting sheet
sf	scale factor
wt	wind-tunnel data

PROGRAM OPERATIONAL FEATURES

The computer program PLOTWD was originally formulated to plot and fair sequentially obtained wind-tunnel data on the large 33-inch CALCOMP plotter paper with either 10 or 20 grid lines per inch. The plots generated could be used either as working figures or, after proper border masking and labeling, as final report figures. The later change in the Langley editorial figure presentation standards due to the enhancement of the CALCOMP software package allows for the acceptance of final report figures that are almost totally generated with computer graphics. As a result, additional subroutines were added to the original program version that can be used to draw grid lines, axis scale values and labels, and a legend containing the run numbers and corresponding plot symbols. These plots can be drawn on the plain white CALCOMP paper and, after additional legend information has been added, are suitable report quality figures.

The data on a figure for a particular run are identified by one of 22 available symbols as illustrated in Table I. Each symbol can be drawn with three basic sizes: (1) small (0.100 in.), (2) medium (0.132 in.), and (3) large (0.168 in.). The data are faired using the standard curve fairing technique which utilizes tension-splines to compute a maximum of 100 enhancement points between each pair of input data points. All enhancement points are plotted as a continuous solid line which intersects but does not pass through the symbol surrounding each input data point. During a typical wind-tunnel test, the engineer will often request that repeat data points be taken to either establish anomalies in performance trends or to provide a routine check value. Fitting a spline curve through the input data with the repeat points included will almost always produce curves with unrealistic oscillations, especially between original and repeat data points. To overcome this problem, a subroutine has been included in the plot program which first sorts the data so that the data points are in monotonic increasing order and then deletes all but one data point in each cluster of duplicate data points. The deleted points are not used as input to the spline curve fit subroutines, but they are plotted as individual data points with the appropriate symbol.

Additional subroutines are available in the plot program that can compute and plot either least-squares polynomial or least-squares cubic-spline curve fits to the input data. These least-squares curve fits are plotted with a continuous dashed line corresponding to the particular symbol as illustrated in Table I. If the user desires, a group of subroutines are available to draw grid lines, axis scale values and labels, and legends. The axis labels that can be drawn consist of a set of nine standard aerodynamic coefficients as illustrated in Table II. A convenient place has been designated in the subroutine that draws labels where the user can add the coding needed to draw other axis labels. The character sets available in the CALCOMP software package are very versatile and the user should be able to draw almost any imaginable label.

The user-supplied input requirements were formulated to be simple and at the same time provide as much flexibility as possible. This flexibility consists of the ability to plot multiple figures on a single sheet, to plot almost any combination of data variables, and to plot almost any combination of runs on a sheet. The input requirements for the basic type of plot without drawn grid lines, axis scale values and labels, and legends are divided into two parts. The first part, designated as the plotting setup information, identifies the plotting sheet boundaries, the symbol size and starting number, curve fairing and editing options, the data array location of the independent test variable, and the combinations of test variables to be plotted with corresponding axis origin locations and scale factors. The second part, designated as the plotting sheet information, is a set of namelists, each containing a sheet identification number and a list of the runs to be plotted on the sheets.

The wind-tunnel data to be plotted must be written on a file designated as TAPE1 and attached as part of the job execution procedure. The data must be written on TAPE1 with the following unformatted write statement:

```
WRITE(1) IRUN, ITEST, IPT, (DATA(I), I = 1, IMAX)
```

where IRUN is the run number, ITEST is the test number, IPT is the data point number, DATA is an array containing the wind-tunnel data, and IMAX is the maximum number of elements in the DATA array. This input data requirment should be able to accommodate almost any type of wind-tunnel data that is taken sequentially. After reading the setup information during program execution, a subroutine is called that reads the data on TAPE1 and reloads it on the random-access file TAPE2 with an equivalent run number indexing. Thereafter, anytime a particular run is to be plotted, the corresponding data for that run is read from the random-access file and loaded into a plotting array. Retrieving the data in this manner simply means that the file read of the data for a particular run begins immediately at the first data point rather than each time having to read the data file from the beginning to locate the first point. This random-access feature greatly reduces the total computing costs by reducing the execution time and the number of I/O operations.

DESCRIPTION OF COMPUTER PROGRAM

The wind-tunnel data plotting program PLOTWD consists of a main program, 18 subroutines, and 2 function subprograms. A computer listing of the program coding is presented in Appendix A. A description of the input requirements for the program is presented in Appendix B and a description of the output for a sample input case listed in Table III is presented in Appendix C. The primary function and execution sequence for the main program and each subprogram are discussed in this section. A supplemental list and corresponding description of the input variables and internal parameters for each routine is included.

Program PLOT

The function of program PLOT is to control the overall execution of the plotting process. Listed at the beginning of the program are a group of comment statements that describe the program input requirements. A group of dimension and common statements appear next consisting of the arrays containing the run numbers and plotting variables, axis origin locations, and axis scale and label information. Several global program constants are then defined and calls made to subroutines PSEUDO and LEROY to initialize the plot vector file SAVPLT for subsequent post-process plotting on a variety of plotters at Langley. A call is then made to subroutine SETUP which reads the input plotting setup data. This is followed by a call to subroutine DATADK which loads the input wind-tunnel data that is sequentially written on the input file TAPE1 into the random-access working file TAPE2. This subroutine is called only once during the program execution; therefore, if the plotting setup information is changed during the program execution, the test number may not change.

The data array locations of the independent and dependent variables to be plotted are then loaded into the work array NLOC. A check is then made to determine if any parts of the plotting setup have common abscissas or ordinates and, if so, the corresponding internal XPLT and YPLT array elements are set equal to 1.0. These internal arrays are checked during the axis labeling and zero-line drawing execution step to prevent duplication. The next execution step is the read of the input namelist SHEET which contains the sheet number and corresponding run numbers to be plotted. The test, sheet, and run numbers are then drawn at the bottom of the plotting sheet and the sheet frame drawn. Arrowheads are then drawn along the bottom and left-hand edge of the sheet frame to indicate the origins. The corresponding axis scale and label are then drawn adjacent to each arrowhead. Zero lines which run from the arrowhead to the opposite frame border are then drawn if the parameter LZERO is equal to zero.

Execution then proceeds to plot the desired wind-tunnel data array variables for each run. The first step is an identification of the plotting symbol to use. The 22 symbols available are presented in Table I in the standard order. The number of the symbol to use is the same as the input order of the run number in the SHEET namelist. If the user desires to skip a particular symbol, simply input a run number of zero at the appropriate place in the run number list in the SHEET namelist. If the user desires to set up a symbol order different from the standard order, this can easily be accomplished by redefining the sequential values in the internal array LSYM.

The next step is to read the particular wind-tunnel data corresponding to the variable to be plotted from the random-access file TAPE2 and to load these data into the internal array VAR. The next step is to convert the wind-tunnel data from cofficient form to the equivalent x_p and y_p locations on the plotting sheet using equations

$$x_p = x_{wt}/x_{sf} - x_o \tag{1}$$

$$y_p = y_{wt}/y_{sf} - y_o, \tag{2}$$

and to load the converted values into arrays X and Y, respectively. At the same time, the values for the independent data variable are loaded into array T and a check is made of the x_p and y_p data to determine if any points are outside the sheet frame. If any points fall outside the sheet frame, a message will be printed stating the run and plot number and the number of points outside the frame. The next step is to sort and edit the x_p and y_p data if the user specified a nonzero value for the input parameter IEDIT. The final step is to call subroutine CURPLT which plots and fairs the x_p and

y_p data using tension splines. After plotting all of the specified variable combinations for a particular run, the above sequence of steps is repeated for the next run.

After plotting all the specified runs for a particular sheet, the plotter is advanced to the start location for the next sheet, the next SHEET namelist read, and the entire sequence of scaling and plotting steps are repeated. The next execution step in the program is to print a summary of the sort and edit information. The final execution step is to draw a statement indicating that all plotting has been completed and to call subroutine CALPLT to close the plot vector file SAVPLT. The following is a list and description of the parameters used in this program:

LSYM	array containing integer values corresponding to the symbol order
RUN	array containing run numbers to be plotted
NPT	array containing the directory or index information on the random-access file TAPE2 (dimensional size equivalent to number of data points on input wind-tunnel data file TAPE1)
XTAPE, YTAPE	arrays containing wind-tunnel data array indices of x_{wt} and y_{wt} variables
XOFFSET, YOFFSET	arrays containing origin x_o and y_o locations on plotting sheet, in.
XSCALE, YSCALE	arrays containing axis scale factors x_{sf} and y_{sf}, change in coefficient per inch
XLABEL, YLABEL	arrays containing x- and y-axis labels
XPLT, YPLT	arrays containing a value of 1.0 if the particular x- or y-axis is repeated
X,Y	arrays containing scale x_p and y_p values, in.
T	array containing value of independent variable

VAR	two-dimensional array containing y_{wt} data in one level and x_{wt} data in the next level
NLOC	array containing consecutive DATA array indices of y_{wt} and x_{wt} variables
NO	sheet number
NEWCASE	if = 1, new setup deck follows the current SHEET namelist if = 0, new SHEET namelist follows
JREAD	tape number of file containing input setup and namelist data
NRNMAX	maximum number of allowable RUNS per sheet
NPLMAX	maximum number of allowable plots per sheet
NDMAX	maximum number of allowable data points per run
NPMAX	maximum number of total data points on TAPE1
IPRT	if = 0, print sort-and-edit summary data if = 1, do not print sort-and-edit summary data
LZERO	if = 0, draw zero lines if = 1, do not draw zero lines
NORG	if = 0, draw arrowheads at axis origins if = 1, do not draw arrowhead at axis origins
IDISK	if = 0, random-access file TAPE2 has not been loaded if = 1, TAPE2 has been loaded
ITEST	test number
SHEETW, SHEETH	plotting sheet width and height, in.
SPACE	space between plotting sheets, in.
ISYM	starting symbol number
ISIZE	symbol size
IOP	data fairing option
TENSION	spline tension factor σ

IEDIT	data sort-and-edit option
TOLR	editing tolerance of independent variable
IW	DATA array index of independent variable
NPLOT	number of desired plots per sheet
IERR	if = 1 after call to subroutine SETUP, indicates that last case has been read or error occurred if = 1 after call to subroutine DATADK indicates that error occurred during either a read or write of TAPE2
NVAR	number of variables read from TAPE2 for each data point (2*NPLOT+1)
NP	number of data points to be plotted for a specified run

Subroutine SETUP

The function of subroutine SETUP is to read and print the plotting setup information as described in Appendix B. A sample of the print formats are presented as page 1 output in Table IV. As various parameters are read from the input file, checks are performed to insure that the parameter is within the program limitations. If the particular parameter cannot be safely redefined within the program limitations, an error message is printed and program execution terminated. All the parameters in the subroutine argument list are defined in the description of program PLOT.

Subroutine DATADK

The primary function of subroutine DATADK is to read data from or write data to the random-access file TAPE2. If the ICODE parameter equals 0, the wind-tunnel data on the input file TAPE1 is read and then written on the random-access file using the mass storage write statement WRITMS. Each data point is read from TAPE1 using the following read statement:

```
READ(1) IRUN, ITEST, IPT, (DATA(I), I = 1, IMAX)
```

The information in the data array is then shifted forward three array locations and the values of IRUN, ITEST, and IPOINT loaded into the first three locations. The information in the resultant DATA array is then written on the random-access file TAPE2 and the access location loaded into the array NPT. This access location will be needed during all subsequent reads of this file; therefore, the user should not add any additional coding that would destroy the contents of this array. If the data being written on TAPE2 corresponds to that for the first point of each run, the run number and the array index in the NPT array are written on a local file TAPE12 for use during the later reads of TAPE2. The default value for the parameter IMAX is 300. If the input data tape contains either more or less variables for each data point, the value of IMAX must be redefined and the dimension size of the DATA array redefined to a value equal to IMAX + 3. After transferring the data on TAPE1 to TAPE2, a summary of the run numbers available and the total number of data points transferred is printed as illustrated in Table IV as page 2 output.

If the ICODE parameter equals 1, all the wind-tunnel data points for a specified run number IRUN are read from the random-access file TAPE2 and the specified variables defined by the user-supplied input values for XTAPE, YTAPE, and IW are then loaded into the two-dimensional array VAR. The random-access index location for the first point of the specified run is determined by reading the run and NPT array index information written previously on TAPE12. During the read of each data point on TAPE2, the data are loaded into the array DATA. The subsequent transfer of the information in the DATA array to the VAR array is shifted back three array locations to prevent loading the values for IRUN, ITEST, and IPT into the VAR array. The test number read from TAPE2 is, however, checked against the test number specified by the user in the setup information to ensure that the correct test data were input on TAPE1.

Several types of errors can occur which may cause the program execution to either terminate or proceed to the next specified run number. The types of errors that may occur include (1) there are no data on TAPE1, (2) the total number of input data points exceeds the allowable maximum, (3) the input and specified test numbers are different, (4) the specified run number is not on TAPE2, (5) number of data points for a specified run number exceeds the allowable maximum, (6) the number of data points on TAPE2 for a specified run cannot be determined from the information written on TAPE12, and (7) the specified XTAPE or YTAPE value exceeds the allowable maximum IMAX. If any of these errors occur, an appropriate error message will be printed. Most of the important parameters used in this subroutine are defined in the description of program PLOT. The following additional parameters are used in this subroutine:

DATA	array containing x_{wt} and y_{wt} data read from either TAPE1 or TAPE2 (must be dimensioned by IMAX+3)
IMAX	maximum number of elements in data array
IRUN	specified run number
IPT	input data point number
ICODE	if = 0, read data on TAPE1 and write to random-access file TAPE2 if = 1, read data on TAPE2 for a specified run and load into the VAR array
OPENMS	system subroutine to open and initialize random-access file
READMS	system subroutine to read information from random-access file
WRITMS	system subroutine to write information on random-access file

Subroutine EDIT

The function of subroutine EDIT is to sort and edit a set of x_p and y_p data to be plotted for a specified run. If the user desires to fair the wind-tunnel data with the tension spline method, the set of data to be faired must be arranged so that the corresponding values for the independent variable, t, are monotonically increasing, single-valued, and reasonably spaced to prevent large oscillations in the faired curve. The first time this subroutine is called with a specified run number, the results of the sort and edit procedure are written on the output file TAPE7 and the run number on output file TAPE11 for future printing during the call to subroutine PRTEDT. Therefore, the first execution step in this subroutine is a check of the information on TAPE11 to determine whether or not summary data have previously been written on TAPE7. If so, the parameter NSTORE is set equal to a value of 1.

The next execution step is to sort the input values of the independent variable, t, so that the final set of values are monotonically increasing. The sorting is accomplished in an iterative manner with the following steps:

(1) Load remaining unsorted values of the independent variable and their corresponding indices into temporary arrays TEDT and IEDT, respectively.

(2) Determine the minimum TEDT value and its corresponding index and load index value into array ISAVE.

(3) Delete the minimum TEDT value from the TEDT array.

(4) Repeat steps 1, 2, and 3 until all input values of the independent variable have been processed.

(5) Load sorted independent variable values and corresponding y_p values into temporary arrays using index information in the ISAVE array.

If the user specified on edit tolerance TOLR that is greater than zero, the sorted data will be edited next to delete all but one data point in each cluster of repeat points. Repeat points are defined as any two adjacent points whose difference in the absolute values of the independent variable is less than a specified tolerance. The edit procedure consists simply of connecting the two data points adjacent to a particular cluster of repeat points with a straight line and then determining the point in the cluster closest to the line. The closest point will be saved and the remaining points in the cluster will be deleted. These deleted points are, however, plotted immediately on the plotting sheet with the appropriate symbol.

Although this editing procedure is simple, four problems may be encountered which will prevent the sorted data from being properly edited. If any of these particular problems occur, the error flag IERR will be set equal to the problem number, all data points input for the particular run plotted with the appropriate symbol, and control returned to the main program PLOT. Program execution will then proceed to plot and fair the next specified run number. The first error which may occur is that all the values for the sorted independent variable are within the specified tolerance TOLR. This error usually occurs because the user specified a tolerance which is too large. The remaining errors occur due to the presence of several adjacent clusters of repeat points which prevents formation of the straight line between adjacent data points. The second flagged error indicates that all data points except the last are clustered; the third flagged error, that all data points prior to the particular point being edited are clustered; and the fourth flagged error, that all data points except the first are clustered. If any of the latter three flagged errors are printed in the output, the user should carefully examine the data and decide which data points to delete.

Following the editing procedure, a summary of the sort and edit information is written on TAPE7 provided the value of the NSTORE parameter is zero. The last execution step is to reload the independent variable t, x_p, and y_p input arrays with the sorted and edited values. The following additional parameters are used in this subroutine:

KEDT	number of run numbers written on TAPE11
TIN	array containing input values of independent variable t
TEDT	array containing sorted and edited values of independent variable
TSAVE, YSAVE	temporary arrays containing values of t and y_p
IEDT, ITEMP, ISAVE	temporary arrays containing indices of sorted or edited t values
TMIN	minimum value of t
NSTORE	if = 0, summary of sort and edit information for particular run is to be written on TAPE7
NSYM	plot symbol number for deleted points

Subroutine PRTEDT

The function of subroutine PRTEDT is to read the sort and edit information generated during the calls to subroutine EDIT. This information is read from TAPE7 and consists of the input, sorted, edited, and deleted values of the independent variable t for all of the runs requested during the job execution. A sample of the output is illustrated as page 4 output in Table IV. As previously noted, the sort information is the same regardless of the number of times the particular run is plotted. However, the edit information is only applicable to the first plot of the run. All of the important parameters used in this subroutine are defined in the description of subroutine EDIT.

Subroutine ZEROLN

The function of subroutine ZEROLN is to draw heavy lines on the plot perpendicular to the axis at each XOFFSET and YOFFSET location to indicate origins or zero lines for the variables to be plotted. Each heavy line is generated by simply drawing a straight line perpendicular to the sheet border at the specified origin location and then drawing two additional lines parallel to the original line and offset to either side of it. The following subroutine arguments and parameters are used:

XO, YO	starting x_p and y_p plot locations of the zero line, in.
DIST	length of zero line, in.
LINE	if = 1, zero line parallel to x-axis
	if = 2, zero line parallel to y-axis
D	offset distance of additional lines drawn parallel to zero line (width of zero line = 2 x D + plotting pen width), in.

Subroutine CURPLT

The function of subroutine CURPLT is to plot and fair a tension spline curve through an input set of x_p and y_p values. The subroutine contains a plot option to plot the symbols only (IOP = 0) and a plot option to plot the symbols and draw a faired curve between them (IOP = 1). For both plot options, the symbols are plotted by calls to subroutine PNTPLT which has been attached as part of the CALCOMP software package. If the curve fairing option has been chosen, the input data points x_p and y_p are faired with a tension spline that has a tension factor equal to the value of the input parameter TENSION. The coordinates of the faired curve are determined by fitting a tension spline through both the x_p and y_p as a function of the independent variable t and then interpolating the two curves to obtain the resultant y_p as a function of x_p faired curve which is then drawn on the plot. If the x_p is a scaler function of t (XTAPE value equals IW), the double interpolation process is not necessary. The input value of the parameter IVAR must be equal to 0 if x_p and t are equal and equal to 1 if they are not equal. The first execution step is a check to ensure that the independent variable values are monotonically increasing. The next step is a call to subroutine CUBSPL which determines the value of the second derivative x_p'' and y_p'' of the tension spline at the input points. The final execution step is to interpolate a specified number of spline coordinates between each pair of input points and to draw the curve so that it does not cross the borders of the symbol surrounding each data point. This capability represents a unique feature of this program.

The following discussion outlines the procedure used to determine the intersection points of the spline curve with the border of each end point symbol and the number of coordinates to interpolate between the pair of points. The straight line distance between the pair of points is computed first. If the length of the line is less than the nominal width of the symbol, the spline curve cannot be drawn between the pair of symbols and execution advances to the next pair of points. If the length of the line is greater than the nominal width of the symbol, the interpolation increment of the independent variable, t, between the pair of points is determined using the formula

used to determine if enough room between symbols to generate fairing.

$$\delta = \frac{t_{i+1} - t_i}{(NH-1)\ \ I(t)} \qquad (3)$$

where NH = 101 and I(t) is the integer portion of $(t_{i+1} - t_i)$. Spline coordinates are then computed at consecutive δ increments between the pair of points.

The next step is to determine the intersection points between the spline curve and the borders of the symbols drawn at each point. This step is accomplished utilizing an iterative technique that is centered around the capabilities of subroutine SYMBOL which determines the coordinates of the border of a specified symbol as a function of the angular position of a radial line drawn through the symbol center. The iteration cycle begins at a point on the spline curve from the symbol center corresponding to a value of t equal to an incremental value, H, based on the symbol size. The coordinates of the spline curve are interpolated using the function subprogram FUNC and then the straight line length and angular location computed from the center of the symbol. The angular value is then input to subroutine SYMBOL which determines the corresonding coordinates on the symbol border. The straight line length from the symbol center to the border is then computed. If the spline straight line length is less than that from the center to the border, the value of t is incremented by H and the above steps are repeated. If the spline straight line length is greater than that from the center to the border, the increment H is halved and added to the previous t value and the above steps repeated. This procedure is repeated until the difference between the spline and symbol-border straight line lengths is within a tolerance equal to the value of the parameter EPS.

After determining the intersection coordinates of the spline curve and the symbol borders for a pair of points, the final step is to plot the spline curve between the intersection coordinates. The LINE subroutine, which is part of the CALCOMP software package, is used to plot this curve. This subroutine draws a series of connecting straight lines between an input set of coordinates. By plotting a relatively large number of closely spaced coordinates, the smoothness of the curve between each pair of input points is retained. The following additional subroutine arguments and parameters are used in this subroutine:

IVAR	if = 0, independent variable t is plotted on x-axis
	if = 1, a dependent variable is plotted on x-axis
IOP	if = 0, plot symbols only
	if = 1, plot symbols and spline curve

MX, MY	arrays containing second derivative values x_p'' and y_p'' of tension spline curve fit
DS1, DS2	arrays containing interpolated x_p and y_p spline curve coordinates between a pair of input points
DUMX, DUMY	arrays containing interpolated x_p and y_p spline coordinates between the borders of the symbols surrounding a pair of input points
NH	number of interpolated spline coordinates between a pair of input points
EPS	tolerance used to determine the intersection point between each symbol border and the spline curve
H	initial increment in t value

Function ATANF

The purpose of function subprogram ATANF is to compute the angular position of a specified coordinate point in degrees with the quadrants defined from 0 to 360 degrees. This function is used with subroutine CURPLT to determine the radial angle of a straight line connecting a specified point on the spline curve and the center of the nearest symbol.

Subroutine SYMBOL

The function of subroutine SYMBOL is to compute the coordinates of the intersection between the border of a specified symbol and a radial line drawn from the center of the symbol. This requires that the border of each symbol presented in Table I be defined as a function of the angular equivalent of border coordinates with the origin of the axis system at the symbol center. The shapes of 22 symbols are defined in this subroutine. The first 10 symbols are the NASA standard open symbols: (1) circle, (2) square, (3) diamond, (4) triangle, (5) right triangle, (6) quadrant, (7) dog house, (8) fan, (9) long diamond, and (10) house. The next 10 symbols are the standard centered (center of symbol indicated by +) symbols corresponding to the standard open symbols. The 21st symbol is a period and the 22nd symbol is a plus sign. The symbol border for the last two symbols is an imaginary circle with a radius 10 percent larger than the symbol height. The shapes of the 10 basic open and centered symbols are made up of various combinations of circles, squares, and triangles. Most of the symbols are symmetric about the local y-axis except the right triangle and the quadrant. The shape of each symbol is determined by computing the straight line or circular arc coordinates of the various segments that make up the symbol. The equations and angular positions of the segments that make up each of the 10 basic symbols are relatively easy to derive and, therefore, are not presented in this report. The following arguments and parameters are used in this subroutine:

NO	symbol number
IS	symbol size (1 for small, 2 for medium, and 3 for large)
X,Y	x and y coordinates of intersection of symbol border and radial line drawn from the center of the symbol
T	angular position of radial line, deg.
SCALE	relative height of three basic symbol sizes, in.

Subroutine CUBSPL

The function of subroutine CUBSPL is to fit a cubic or tension spline through an input set of x and y values. The cubic spline represents a special case of the tension spline where the tension is zero. The input x and y data are used to compute a matrix of simultaneous equations in the form of

$$e_i y''_{i-1} + d_i y''_i + f_i y''_{i+1} = c_i \tag{4}$$

for i = 2, 3, . . ., N-1 where for the cubic spline

$$e_i = \frac{h_{i-1}}{6}, \quad d_i = \frac{h_{i-1} + h_i}{3}, \text{ and } f_i = \frac{h_i}{6} \tag{5}$$

and for the tension spline

$$c_i = \frac{1}{\sigma^2} \left[\frac{1}{h_{i-1}} - \frac{\sigma}{\sinh(\sigma h_{i-1})} \right] ,$$

$$d_i = \frac{1}{\sigma^2} \left[\frac{\sigma \cosh(\sigma h_{i-1})}{\sinh(\sigma h_{i-1})} - \frac{1}{h_{i-1}} + \frac{\sigma \cosh(\sigma h_i)}{\sinh(\sigma h_i)} - \frac{1}{h_i} \right] , \tag{6}$$

and $$f_i = \frac{1}{\sigma^2} \left[\frac{1}{h_i} - \frac{\sigma}{\sinh(\sigma h_i)} \right]$$

For both cubic and tension spline,

$$c_i = \left(\frac{y_{i+1} - y_i}{h_i} \right) - \left(\frac{y_i - y_{i-1}}{h_{i-1}} \right) ,$$

$$h_i = x_{i+1} - x_i , \tag{7}$$

and $h_{i-1} = x_i - x_{i-1}$.

For the tension spline, the tension parameter σ is the average tension per length of the input x values and determined using the equation

$$\sigma = \frac{\bar{\sigma}\,(N-1)}{(x_N - x_1)} \tag{8}$$

where $\bar{\sigma}$ is the user-supplied value of the input parameter TENSION and N is the number of input values. A complete derivation and description of cubic and tension splines are presented in reference 1.

The second derivatives are the unknowns in the system of equations generated by equation (4). The matrix of terms is tridiagonal with two less unknowns than equations; therefore, the second derivative at the end points must be specified. In this subroutine, the second derivatives at the end points are computed by fitting a second-order polynomial of the form

$$y = a_1 x^2 + a_2 x + a_3 \tag{9}$$

to each end point and its two adjacent input points and then differentiating to determine the second derivative which is

$$y'' = 2a_1 \quad . \tag{10}$$

The resultant tridiagonal matrix can be solved using the Crout reduction method which is described in detail in reference 2. Using this method, the solution becomes a simple back substitution

$$y''_N = c_N \quad \text{for } i = N$$

$$\text{and } y''_i = \bar{c}_i - \bar{f}_i \, y''_{i+1} \quad \text{for } i = N-1, N-2, \ldots \quad 1 \tag{11}$$

$$\text{where } \bar{d}_i = d_i - e_i \, \bar{f}_{i-1} \, ,$$

$$\bar{f}_i = f_i / \bar{d}_i \quad , \tag{12}$$

and

$$\bar{c}_i = \frac{c_i - e_i \bar{c}_{i-1}}{\bar{d}_i}$$

The following is a description of the parameters in the argument list for this subroutine:

X,Y	input arrays containing table of x- and y-values
N	number of input coordinates
YPP	output array containing y" values
TENSION	input tension factor $\bar{\sigma}$
TENS	output dimensionalized tension factor σ
A	internal work array used during back-substitution process

Function FUNC

The function of subprogram FUNC is to compute the y-value of a spline curve at a given x-value in a specified interval $x_i < x > x_{i+1}$.

The interpolation equation for the cubic spline is

$$y(x) = y''_i \left[\frac{(x_{i+1} - x)^3}{6h_i} - \frac{(x_{i+1} - x)\, h_i}{6}\right] +$$

$$y''_{i+1} \left[\frac{(x - x_i)^3}{6h_i} - \frac{(x - x_i)\, h_i}{6}\right] + \left[\frac{y_i\,(x_{i+1} - x) + y_{i+1}\,(x - x_i)}{h_i}\right] \quad (13)$$

and for the tension spline is

$$y(x) = \frac{y''i}{\sigma^2} \left[\frac{\sinh\left[\sigma(x_{i+1} - x)\right]}{\sinh\,(\sigma h_i)} - \frac{(x_{i+1} - x)}{h_i}\right] +$$

$$\frac{y''_{i+1}}{\sigma^2} \left[\frac{\sinh\left[\sigma(x - x_i)\right]}{\sinh\,(\sigma h_i)} - \frac{(x - x_i)}{h_i}\right] + \left[\frac{y_i(x_{i+1} - x) + y_{i+1}(x - x_i)}{h_i}\right] \quad (14)$$

where $h_i = (x - x_i)$.

Subroutine DASHLN

The function of subroutine DASHLN is to draw a dashed line spline curve through an input set of x_p and y_p values. The spline curve fit procedure for this subroutine is very similar to that described for subroutine CURPLT with the major difference being that symbols are not drawn around each symbol with this subroutine. Therefore, the complex procedure to determine the intersection between the spline curve and the symbol border is not included in this subroutine. The dashed lines that can be drawn consist of various combinations of long and short dashes as illustrated in table I.

The subroutine CUBSPL is called initially to determine the y''_p values of a tension or cubic spline fit of the input data. A specified number of equally spaced points are then interpolated between the input end points and the total length of the curve computed using a simple triangular summation method. The basic length of the long and short dash combination and the space between combinations are then scaled so that the final plotted dashed line will start and end with a complete long and short dash combination. The scaled dashed line is then drawn using linear interpolation of the spline curve to determine the end points x_p and y_p values for each consecutive long or short dash. The following additional subroutine arguments and parameters are used in this subroutine:

NSYM	symbol or dashed line number
IOP	if = 0, plot symbols only
	if = 1, plot dashed line only
NL	number of long dashes in each combination
NS	number of short dashes in each combination
XI, YI	arrays containing interpolated x_p and y_p values of spline curve
S	array containing interpolated length of spline curve
NP	number of interpolated spline curve values
SL	length of long dash, in.
SS	length of short dash, in.
SP	length of space between dashes, in.
NLT	total number of long and short dash combinations to be plotted
DTN/DT	scaling factor for each dash combination

Subroutine LINEAR

The function of subroutine LINEAR is to determine the x and y values at a specified t value using linear (straight line) interpolation. The first execution step is to determine the interval containing the specified t value. The final step is to compute the corresponding x and y values using the following linear interpolation equations:

$$x = x_{i-1} + \delta (x_i - x_{i-1})$$
$$y = y_{i-1} + \delta (y_i - y_{i-1}) \qquad (15)$$

where $\delta = (t - t_{i-1})/(t_i - t_{i-1})$.

Subroutine LSQPLT

The function of subroutine LSQPLT is to draw a least-squares polynomial or least-squares cubic spline curve through an input set of x and y values. For each set of input data, the input points are plotted with the specified symbol and the least-squares curve plotted with the corresponding dashed line. The curve-fitting procedure used in this subroutine is similar to that used in subroutine CURPLT. A separate least-squares curve is determined for each set of input x and y values as a function of the independent variable t and a y(x) curve then obtained by interpolation. If the user specifies that a least-squares polynomial curve be drawn through the input data, subroutine LSQ is called which determines the coefficients of the specified-order polynomial. If the user specifies that a least-squares cubic-spline curve be drawn, subroutine CSDS is called which determines the coefficients of the third-order piecewise polynomials that constitute the spline curve.

Following the call to the selected least-squares subroutine, the x and y values at each input t value are computed and sum-of-the-squares of the differences between the input and least-squares values are computed. If the user specifies a nonzero value for the parameter IPRINT, a summary of the least-squares curve fit process will be printed as illustrated in table V and VI. The final execution step is a call to subroutine DASHLN which draws the least-squares compute x and y values with the specified dashed line. The following additional arguments and parameters are used in this subroutine:

ILSQ	if = 0, fit input data with least-squares polynomial
	if = 1, fit input data with least-squares cubic spline
NPOL	order of polynomial if ILSQ = 0
DF	standard deviation for least-squares cubic spline if ILSQ = 1
COEF	two-dimensional array containing coefficients of piecewise third-order polynomials of cubic spline
CX,CY	arrays containing coefficients of least-squares polynomial
XNEW,YNEW	arrays containing new x and y values after curve fit
WT	weighting factor used in least-squares polynomial curve fit
IPRINT	if = 0, do not print summary data
	if = 1, print summary data
ERRX,ERRY	sum-of-squares of the differences between input and least-squares x and y data

Subroutine LSQ

The function of subroutine LSQ is to determine the coefficients of a polynomial that best fits an input set of weighted x and y data. The polynomial is in the form

$$p(x_i) = a_o + a_1 x_i + a_2 x_i^2 + \dots + a_M x_i^M \qquad (16)$$

where M is the order of the polynomial. The sum-of-the-squares differences between the weighted input and polynomial fit values are

$$S = \sum_{i=1}^{N} (P(x_i) - w_i y_i)^2 \qquad (17)$$

The least or minimum error is

$$\frac{\partial S}{\partial a_j} = 2 \sum_{i=1}^{N} \left[P(x_i) - w_i y_i \right] x_i^{\,j} = 0 \tag{18}$$

where j = 0, 1, 2 . . ., M. This reduces to a set of simultaneous equations in the form

$$a_o \sum x_i^{\,j} + a_1 \sum x_i^{\,j+1} \; . \; . \; . + a_M \sum x_i^{\,j+M} = \sum w_i y_i x_i^{\,j} \tag{19}$$

which can be solved for the polynomial coefficients a_j using a simplified solution technique that takes advantage of the symmetric properties of the matrix of terms generated by the left-hand side of equation (19). The following arguments and parameters are used in this subroutine:

X,Y	arrays containing input x_i and y_i data
W	array containing input weighting factor w_i
NP	number of input points N
N	order of polynomial M
C	array containing values of the coefficients of the least-squares polynomial
A	two-dimensional work array

Subroutine CSDS

The function of subroutine CSDS is to fit a least-squares cubic-spline through an input set of x and y data. The method used in this subroutine defines a continuous cubic-spline function in the form of the third-order polynomial

$$P(x) = a_o + a_1 h_i + a_2 h_i^{\,2} + a_3 h_i^{\,3} \tag{20}$$

for i = 1, 2, 3, . . ., N - 1 where $h_i = (x - x_i)$ and N is the number of input points. The coefficients a_o, a_1, a_2, and a_3 are computed in a least-squares manner so that

$$\sum_{i=1}^{N} \left[\frac{P(x_i) - y_i}{\delta y_i}\right] \leq SP \tag{21}$$

and

$$\int_{x_1}^{x_N} \left[\frac{d^2P(x)}{dx^2}\right] dx \text{ is a minimum} \tag{22}$$

where the smoothing parameter SP is in the interval $N - \sqrt{2N} \leq SP \leq N + \sqrt{2N}$ and δy_i is specified allowable standard deviation in the error in y_i.

A detailed discussion of the least-squares cubic-spline method is presented in reference 3 and, therefore, will not be included in this report. This subroutine is also a part of the standard math-library subprogram package available on the Langley CDC computer system and is identified by the same name and argument list. A complete description of the input and output parameters in the argument list are presented at the beginning of the listing of the subroutine in appendix A.

Subroutine AXISLB

The primary function of subroutine AXISLB is to draw scale values on the x or y axis. The scale value is drawn in one-inch increments starting at the origin and continuing for a specified length. Each scale value drawn is centered adjacent to the inch mark for the y-axis or centered below the inch mark for the x-axis. The following is a description of the subroutine arguments:

IAXIS	if = 1, draw x-axis scale values
	if = 2, draw y-axis scale values
XORG, YORG	x_p and y_p values of axis origin, in.
ORG	scale value at origin
SCALE	increment in scale value per inch

HT	height of scale values, in.
NDIG	number of significant figures to draw on the right side of the decimal point (NDIG = -1 will drop decimal point)

Subroutine COEFSY

The function of subroutine COEFSY is to draw a specified standard aerodynamic label as illustrated in table II. The first seven labels are the standard designations for the lift, drag, pitching-moment, rolling-moment, yawing-moment, and side-force coefficients and lift-drag ratio. The last two labels are the standard designations for the angle of attack and angle of sideslip. The width-height ratio for each label is also listed in table II and is useful to the user when determining the x_p and y_p values needed to position the label. If the user desires to add coding to draw other labels, the additional coding can be inserted following the statement 3 CONTINUE. The following is a description of the subroutine arguments:

XO	x_p location of left-hand edge of main character in label, in.
YO	y_p location of centerline of main character in label, in.
HT	height of main character, in.
ISYM	identifying number of label

Subroutine LAM

The function of subroutine LAM is to draw the script "ℓ" character used as the subscript for the rolling-moment coefficient label generated by subroutine COEFSY. This special character is not available as part of the standard character series contained in the CALCOMP plotting subprogram package.

Subroutine GRIDLN

The function of subroutine GRIDLN is to draw a series of horizontal and vertical grid lines within a specified frame size. This process is accomplished using simple straight line plotting pen movements. The following is a description of the subroutine arguments:

XO,YO	x_p and y_p values for the lower left-hand corner of the frame, in.
XL	width of frame, in.
YH	height of frame, in.
NDIV	number of grid lines per inch

Subroutine RUNKEY

The function of subroutine RUNKEY is to draw a legend containing a list of run numbers for the data plotted and a corresponding list of symbols. The run numbers are spaced so that a corresponding list of descriptors can be typed on a sheet of gum-back paper using the IBM Executive model typewriter and then cut and attached to the legend adjacent to the list of run numbers. This particular typewriter is widely used at Langley to type figure titles and legends. After drawing the symbols and run numbers the words "Symbol" and "Run" are drawn and underlined above the corresponding lists. The user often desires to scale a series of plots to either a smaller or larger size using the magnification factors available as part of the postprocessor PLOT control card. Using these magnification factors would also cause a corresponding change in the legend which would prohibit the use of the IBM Executive Model typewriter to type the list of descriptors. To overcome this problem, a magnification factor, XM, has been included in this subroutine that adjusts the letter and number heights and spacings to compensate for the postprocessor scaling. The following subroutine arguments and parameters are used in this subroutine:

XO, YO	The x_p and y_p values of the lower left-hand corner of the legend, in.

RUN	array containing input run numbers
LSYM	array containing input symbol order
ISIZE	symbol size
HT	letter and number character height, in.
NRNMAX	maximum allowable number of runs per sheet
XM	magnification factor for character and spacings

APPLICATION OF COMPUTER PROGRAM

The basic program variables have been dimensioned for a maximum of 10 runs per sheet, 10 plots per sheet, 50 data points per run, 5000 data points per test, and 300 wind-tunnel test parameters per data point. The following procedure outlines the changes necessary to either increase or decrease these basic program variables:

(1) To change the maximum number of allowable runs per sheet, change the dimension of the variable RUN and the value of the parameter NRMAX in the program PLOT.

(2) To change the maximum number of plots per sheet, change the dimension of the variables XPLT, YPLT, YTAPE, YOFFSET, YSCALE, YLABEL, XTAPE, XOFFSET, XSCALE, XLABEL, VAR, and NLOC and the value of the parameter NPLMAX in program PLOT.

(3) To change the maximum number of points per run, change the dimension of the variables VAR, ARRAY1, ARRAY2 and the value of the parameter NDMAX in program PLOT; change the dimension of the variables TIN, TEDT, TSAVE, YSAVE, IEDT, ITEMP, and ISAVE in subroutine EDIT; change the dimension of the variables TIN, ISAVE, TSAVE, IEDT, TEDT, IUN, and TUN in subroutine PRTEDT; change the dimension of the variables MX, MY, and A in subroutines CURPLT and DASHLN; and change the dimension of the variables XNEW, YNEW, COEF, and SDV and the value of the parameter NMAX in subroutine LSQPLT.

(4) To change the maximum number of allowable data points per test, change the dimension of the variable NPT and the value of the parameter NPMAX in program PLOT.

(5) To change the maximum number of allowable wind-tunnel test parameters per data point, change the dimension of the variable DATA and the value of the parameter IMAX in subroutine DATADK.

The best way to demonstrate the application of the program is to present and discuss the output for several sample cases. Two sample cases will be presented (1) a single plot with three variables per sheet and (2) four plots with a single variable per sheet. The input data for these two cases are presented in table III. Both cases contain one sheet namelist with identical sheet and run numbers. The tabulated output for the first case is presented in table IV and the plotted output, in figure 1. If the user desires to plot dashed lines instead of symbols with connecting solid lines, the call to subroutine CURPLT in program PLOT can be replaced with a call to subroutine DASHLN which will produce the plot presented in figure 2. In a like manner, if the user desires to least-squares curve fit the input data, the call to CURPLT can be replaced with a call to subroutine LSQPLT which will produce the plot presented in figure 3(a) for the least-squares polynomial option and in figure 3(b) for the least-squares cubic-spline option. The programming changes required to obtain these optional plots are very simple and require very little programming knowledge.

Some programming knowledge is required if the user desires to generate a report quality plot on blank (no grid) paper by drawing the appropriate grid lines, axis scale values and labels, and the legends. The input requirements for the subroutines that generate the various parts of the plot are relatively simple and, after a short period of time, the average user should become very proficient in their use. The computer code required to generate report quality plots for the two sample cases presented in table III are presented in table VII and VIII, respectively, and the corresponding plots in figures 4 and 5. Each set of code was inserted in the PLOT program following card number 233 (see appendix A) and the original code from cards number 237 to 269 deleted. Careful examination of the code for both cases will reveal that the same general procedure was used to generate a particular plot. This procedure is outlined as follows:

(1) position plotting pen and draw grid lines (call subroutine GRIDLN)

(2) draw all zero lines (call subroutine ZEROLN)

(3) draw scale values on x-axis (call subroutine AXISLB)

(4) position and draw label for x-axis (call subroutine COEFSY)

(5) repeat steps 3 and 4 for y-axis

(6) position and draw legend (call subroutine RUNKEY)

The two types of plots presented represent only a small example of the wide variety of the types of plots that can be generated using the PLOT program as the baseline. Most engineers that test in the same facility generally prefer to present the experimental data taken during a typical test program in a well-established standard format. The computer code to generate these standard report quality plots can be written by a programmer with more experience and simply inserted in the PLOT program code by the user with either the EDIT, XEDIT, UPDATE, or MODIFY editing postprocessor commands.

CONCLUDING REMARKS

The wind-tunnel data plotting program PLOTWD described in this report has been used successfully at Langley for several years with very few reported execution problems. The required format for the input data is somewhat restrictive, but the required user input is very simple and the variety of plots generated numerous. With relatively minor code changes, the original version of the program can be changed to handle either larger or smaller amounts of test data. The program code presented in this report was written especially for use on the Langley CDC computer system. The program can be adapted to other computer systems provided the system selected has a random-access file capability and a similar CALCOMP software package. A copy of the basic source code for this program can be obtained using the following control statements:

```
GET, PLOTWD/UN = 621288N
UPDATE,F,P = PLOTWD, S = SOURCE, I = 0, L = 0.
```

REFERENCES

1. Cline, A. K.: Scalar- and Planar-Valued Curve Fitting Using Splines Under Tension. Communications of the ACM, Vol. 17, No. 14, 1984, pp. 218-220.
2. Hildebrand, Frances B.: Methods of Applied Mathematics. Second Edition, Prentice-Hall, Inc., 1965.
3. Reinsch, Christian H.: Smoothing by Spline Functions. Numerische Mathematik, Vol. 10, No. 3, 1967, pp. 177-183.

APPENDIX A

COMPUTER LISTING OF WIND-TUNNEL DATA PLOTTING PROGRAM PLOTWD

This appendix contains a computer listing of the wind-tunnel data plotting program PLOTWD which consists of a main program, eighteen subroutines, and two function subprograms.

```
      PROGRAM PLOT(INPUT,OUTPUT,TAPE5=INPUT,TAPE1,TAPE2,TAPE7=1001,
     1TAPE11=1001,TAPE12=1001)
C
C     PROGRAM TO PLOT AND FAIR WIND TUNNEL DATA
C
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983
C
C ********************************************************************
C *                                                                  *
C *              -- DEFINITION OF SETUP DECK CARDS --                *
C *                                                                  *
C * CARD NO.             DESCRIPTION                          FORMAT *
C *                                                                  *
C *    1      TEST - TEST NUMBER                             (F10.0) *
C *                                                                  *
C *    2      SHEETW - WIDTH OF PLOTTING SHEET (INCHES)     (3F10.0) *
C *           SHEETH - HEIGHT OF PLOTTING SHEET (INCHES)             *
C *           SPACE - SPACE BETWEEN PLOTTING SHEETS (INCHES)         *
C *                                                                  *
C *    3      ISYM - STARTING SYMBOL NUMBER                 (2F10.0) *
C *           ISIZE - SYMBOL SIZE (1 = SMALL  2 = MEDIUM  3 = LARGE) *
C *                                                                  *
C *    4      IOP - PLOTTING OPTION  IOP=0 PLOT DATA ONLY   (2F10.0) *
C *                                  IOP=1 PLOT AND FAIR DATA        *
C *           TENSION - SPLINE TENSION FACTOR (FOR IOP=1 ONLY)       *
C *                                                                  *
C *    5      IEDIT - EDIT OPTION  IEDIT=0 NO EDITING       (2F10.0) *
C *                                IEDIT=1 EDIT WITH GIVEN TOLERANCE *
C *           TOLR - EDIT TOLERANCE FOR INDEPENDENT VARIABLE         *
C *                                                                  *
C *    6      IW - DATA ARRAY LOCATION OF INDEPENDENT VARIABLE (F10.0) *
C *                                                                  *
C *    7      NPLOT - NUMBER OF PLOTS PER SHEET              (F10.0) *
C *                                                                  *
C *    8      YTAPE - ARRAY LOCATION OF Y VARIABLE    (2(3F10.0,A10)) *
C *           YOFFSET - ORIGIN OFFSET OF Y VARIABLE (INCHES)         *
C *           YSCALE - SCALE VALUE PER INCH FOR Y VARIABLE           *
C *           YLABEL - LABEL FOR Y VARIABLE                          *
C *           XTAPE - ARRAY LOCATION OF X VARIABLE                   *
C *           XOFFSET - ORIGIN OFFSET OF X VARIABLE (INCHES)         *
```

CARD NO.

```
C  *           XSCALE - SCALE VALUE PER INCH FOR X VARIABLE            * PT  41
C  *           XLABEL - LABEL FOR X VARIABLE                           * PT  42
C  *                NOTE - CARD 8 IS REPEATED NPLOT TIMES              * PT  43
C  *                                                                   * PT  44
C  *          -- DEFINITION OF SHEET NAMELIST PARAMETERS --            * PT  45
C  *                                                                   * PT  46
C  *      $SHEET  - NAMELIST LABEL                                     * PT  47
C  *      NO      - SHEET NUMBER                                       * PT  48
C  *      RUN     - RUN NUMBERS TO BE PLOTTED ON SHEET (MAXIMUM OF 10) * PT  49
C  *      NEWCASE - IF NEW SETUP DECK FOLLOWS SET NEWCASE=1            * PT  50
C  *      $END    - END OF NAMELIST                                    * PT  51
C  *                                                                   * PT  52
C  ********************************************************************* PT  53
C                                                                        PT  54
      INTEGER RUN                                                        PT  55
C                                                                        PT  56
C         DIMENSION ARRAY LSYM WHICH CONTAINS ORDER OF PLOTTING SYMBOLS  PT  57
C                                                                        PT  58
      DIMENSION LSYM(22)                                                 PT  59
C                                                                        PT  60
C         DIMENSION RUN(NRNMAX) WHERE NRNMAX IS THE MAXIMUM NUMBER OF    PT  61
C         ALLOWABLE RUNS PER SHEET                                       PT  62
C                                                                        PT  63
      DIMENSION RUN(10)                                                  PT  64
C                                                                        PT  65
C         DIMENSION NPT(NPMAX) WHERE NPMAX IS THE MAXIMUM NUMBER OF DATA PT  66
C         POINTS FOR TOTAL TEST                                          PT  67
C           NOTE -- DO NOT DESTROY THIS ARRAY DURING PROGRAM EXECUTION   PT  68
C                                                                        PT  69
      DIMENSION NPT(5000)                                                PT  70
C                                                                        PT  71
C         DIMENSION EACH OF THE FOLLOWING ARRAYS BY NPLMAX WHICH IS THE  PT  72
C         MAXIMUM NUMBER OF ALLOWABLE PLOTS PER SHEET                    PT  73
C                                                                        PT  74
      DIMENSION XPLT(10), YPLT(10), YTAPE(10), YOFFSET(10), YSCALE(10),  PT  75
     1YLABEL(10), XTAPE(10), XOFFSET(10), XSCALE(10), XLABEL(10)         PT  76
C                                                                        PT  77
C         DIMENSION EACH OF THE FOLLOWING ARRAYS BY NDMAX WHICH IS THE   PT  78
C         MAXIMUM NUMBER OF ALLOWABLE POINTS FOR A GIVEN RUN             PT  79
C                                                                        PT  80
```

CARD NO.

```
      DIMENSION X(50), Y(50), T(50)
C
C        DIMENSION VAR(NDMAX,NVAR) AND NLOC(NVAR) WHERE NVAR=2*NPLMAX+1
C
      DIMENSION VAR(50,21), NLOC(21)
C
C        DIMENSION ARRAY1(7*NDMAX.GE.500) AND ARRAY2(3*NDMAX)
C
      COMMON /WORK/ ARRAY1(500)
      COMMON /PLT/ ARRAY2(150)
C
C        SHEET NAMELIST PARAMETERS
C
      NAMELIST /SHEET/ NO,RUN,NEWCASE
C
C        DEFINE INPUT TAPE NUMBER
C
      DATA JREAD/5/
C
C        DEFINE ORDER OF PLOTTING SYMBOLS
C
      DATA LSYM/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22
     1/
C
C     DEFINE MAXIMUM NUMBER OF ALLOWABLE RUNS PER SHEET
C
      DATA NRNMAX/10/
C
C        DEFINE MAXIMUM NUMBER OF ALLOWABLE PLOTS PER SHEET
C
      DATA NPLMAX/10/
C
C        DEFINE MAXIMUM NUMBER OF ALLOWABLE DATA POINTS FOR A GIVEN RUN
C
      DATA NDMAX/50/
C
C     DEFINE MAXIMUM NUMBER OF ALLOWABLE TOTAL DATA POINTS
C
      NPMAX=5000
C
```

CARD NO.

```
      C     DEFINE PROGRAM OPTIONS
      C
      C        IF IPRT=0, PRINT EDITED DATA
      C        IF IPRT=1, DO NOT PRINT EDITED DATA
            IPRT=0
      C        LZERO=0, DRAW ZERO LINES
      C        LZERO=1, DO NOT DRAW ZERO LINES
            LZERO=1
      C        NORG=0, DRAW ARROWHEAD AT ORIGIN OF PLOTTING VARIABLES
      C        NORG=1, DO NOT DRAW ARROWHEAD AT ORIGIN OF PLOTTING VARIABLES
            NORG=0
      C
      C     SET PROGRAM CONSTANTS
      C
            IDISK=0
      C
      C     INITIALIZE PLOTTING DEVICE
      C
            CALL PSEUDO
            CALL LEROY
      C
      C     READ SETUP DECK
      C
      1     CALL SETUP (ITEST,SHEETW,SHEETH,SPACE,ISYM,ISIZE,IOP,TENSION,IEDIT
           1,TCLR,IW,NPLOT,YTAPE,YOFFSET,YSCALE,YLABEL,XTAPE,XOFFSET,XSCALE,XL
           2ABEL,JREAD,NPLMAX,IERR)
            IF (IERR.NE.0) GO TO 23
      C
            REWIND 7
            REWIND 11
            KEDT=0
      C
      C     LOAD DATA DISK
      C
            IF (IDISK) 3,2,3
      2     CALL DATADK (0,ITEST,IRUN,VAR,NVAR,NLOC,NP,NPT,NPMAX,NDMAX,IERR)
            IF (IERR.NE.0) GO TO 23
      C
      3     IDISK=1
            PRINT 24
```

```
      C
      C       LOAD PLOTTING VARIABLES
      C
              DO 4 I=1,NPLOT
              J2=2*I
              J1=J2-1
              NLOC(J1)=IFIX(YTAPE(I)+0.0001)
            4 NLOC(J2)=IFIX(XTAPE(I)+0.0001)
              NVAR=2*NPLOT+1
              NLOC(NVAR)=IW
      C
      C       CHECK FOR DUPLICATE PLOTTING VARIABLES
      C
              DO 9 I=1,NPLOT
              YPLT(I)=XPLT(I)=0.
              IF (I.EQ.1) GO TO 9
              J1=I-1
              DO 8 J=1,J1
              IF (XTAPE(J)-XTAPE(I)) 6,5,6
            5 IF (XOFFSET(J).EQ.XOFFSET(I).AND.XSCALE(J).EQ.XSCALE(I)) XPLT(I)=1
             1.0
            6 IF (YTAPE(J)-YTAPE(I)) 8,7,8
            7 IF (YOFFSET(J).EQ.YOFFSET(I).AND.YSCALE(J).EQ.YSCALE(I)) YPLT(I)=1
             1.0
            8 CONTINUE
            9 CONTINUE
      C
      C       READ SHEET NUMBER AND RUNS TO BE PLOTTED
      C
           10 DO 11 I=1,NRNMAX
           11 RUN(I)=0
              NEWCASE=NO=0
      C
              READ (JREAD,SHEET)
              IF (EOF(JREAD)) 23,12
           12 CONTINUE
      C
              PRINT 25, NO,RUN
      C
      C       PERFORM GRID CHECK
```

CARD NO.

```
      C
            CALL CALPLT (0.,0.,-3)
            CALL GRIDCK
      C
      C     POSITION PLOTTING PEN AT LOWER LEFT HAND CORNER OF PLOTTING SHEET
      C
            CALL CALPLT (2.,2.,-3)
      C
      C     LABEL TEST, SHEET, AND RUN NUMBERS
      C
            SZ=0.15
            D=-1.9
      C        LABEL TEST NUMBER
            CALL NOTATE (D,-2.0,SZ,5HTEST ,0.,5)
            D=30.*SZ/7.+D
            TP=FLOAT(ITEST)
            CALL NUMBER (D,-2.0,SZ,TP,0.,-1)
            D=36.*SZ/7.+D
      C        LABEL SHEET NUMBER
            CALL NOTATE (D,-2.0,SZ,7H,SHEET ,0.,7)
            D=42.*SZ/7.+D
            TP=FLOAT(NO)
            CALL NUMBER (D,-2.0,SZ,TP,0.,-1)
            D=36.*SZ/7.+D
      C        LABEL RUN NUMBERS
            CALL NOTATE (D,-2.0,SZ,6H,RUNS ,0.,6)
            D=42.*SZ/7.+D
            DO 13 I=1,NRNMAX
            TP=FLOAT(RUN(I))
            CALL NUMBER (D,-2.0,SZ,TP,0.,-1)
       13   D=42.*SZ/7.+D
            IX=IFIX(D)
            DMAX=FLOAT(IX)+SPACE+2.0
      C
      C     DRAW PLOTTING SHEET FRAME
      C
            CALL CALPLT (0.,0.,3)
            CALL CALPLT (SHEETW,0.,2)
            CALL CALPLT (SHEETW,SHEETH,2)
            CALL CALPLT (0.,SHEETH,2)
```

CARD NO.

```
            CALL CALPLT (0.,0.,2)
      C
      C     LABEL ORIGIN, VARIABLE DESCRIPTION, AND SCALE
      C
            DO 15 I=1,NPLOT
            IF (YPLT(I).EQ.1.) GO TO 14
            D=YOFFSET(I)
            IF (NORG.EQ.0) CALL PARROW (-0.3,D,0.,D,-2.0,.3)
            CALL NOTATE (-1.8,D,0.15,YLABEL(I),0.,10)
            D=YOFFSET(I)-0.2
            CALL NUMBER (-1.8,D,0.15,YSCALE(I),0.,4)
   14       IF (XPLT(I).EQ.1.) GO TO 15
            D=XOFFSET(I)
            IF (NORG.EQ.0) CALL PARROW (D,-0.3,D,0.,-2.0,.3)
            D=XOFFSET(I)-0.65
            CALL NOTATE (D,-0.6,0.15,XLABEL(I),0.,10)
            CALL NUMBER (D,-0.8,0.15,XSCALE(I),0.,4)
   15       CONTINUE
      C
      C     DRAW ZERO LINES
      C
            IF (LZERO.NE.0) GO TO 18
            DO 17 I=1,NPLOT
            IF (YPLT(I).EQ.1.) GO TO 16
            CALL ZEROLN (0.,YOFFSET(I),SHEETW,1)
   16       IF (XPLT(I).EQ.1.) GO TO 17
            CALL ZEROLN (XOFFSET(I),0.,SHEETH,2)
   17       CONTINUE
   18       CONTINUE
      C
      C     READ DATA FROM DATA DISK AND PLOT
      C
            KSYM=ISYM-1
      C
            DO 21 IR=1,NRNMAX
      C        DEFINE RUN NUMBER
            IRUN=RUN(IR)
      C        DEFINE PLOTTING SYMBOL
            KSYM=KSYM+1
            IF (KSYM.GT.22) KSYM=ISYM
```

CARD NO.

```
                IF (KSYM.LT.1) KSYM=1                                       PT 281
                NSYM=LSYM(KSYM)                                             PT 282
          C                                                                 PT 283
                IF (IRUN.EQ.0) GO TO 21                                     PT 284
          C        READ DATA FROM DATA DISK                                 PT 285
                CALL DATADK (1,ITEST,IRUN,VAR,NVAR,NLOC,NP,NPT,NPMAX,NDMAX,IERR)  PT 286
                IF (IERR.NE.0) GO TO 21                                     PT 287
          C        LOAD AND PLOT DATA                                       PT 288
                DO 20 I=1,NPLOT                                             PT 289
                J2=2*I                                                      PT 290
                J1=J2-1                                                     PT 291
                IX=NLOC(J2)                                                 PT 292
          C             LOAD PLOTTING ARRAYS                                PT 293
                N=0                                                         PT 294
                DO 19 J=1,NP                                                PT 295
                XP=VAR(J,J2)/XSCALE(I)+XOFFSET(I)                           PT 296
                YP=VAR(J,J1)/YSCALE(I)+YOFFSET(I)                           PT 297
                TP=VAR(J,NVAR)                                              PT 298
          C             CHECK TO SEE IF DATA IS WITHIN PLOTTING FRAME       PT 299
                IF (XP.LT.0.0.OR.XP.GT.SHEETW) GO TO 19                     PT 300
                IF (YP.LT.0.0.OR.YP.GT.SHEETH) GO TO 19                     PT 301
                N=N+1                                                       PT 302
                X(N)=XP                                                     PT 303
                Y(N)=YP                                                     PT 304
                T(N)=TP                                                     PT 305
          19    CONTINUE                                                    PT 306
          C             PRINT MESSAGE IF SOME POINTS NOT WITHIN PLOTTING FRAME    PT 307
                JS=NP-N                                                     PT 308
                IF (N.NE.NP) PRINT 26, IRUN,JS,I                            PT 309
          C             EDIT DATA                                           PT 310
                IERR=0                                                      PT 311
                IF (IEDIT.NE.0) CALL EDIT (T,X,Y,N,NSYM,ISIZE,IRUN,TOLP,KEDT,IERR) PT 312
                IF (IERR.NE.0) GO TO 20                                     PT 313
          C             PLOT VARIABLES                                      PT 314
                JS=0                                                        PT 315
                IF (IX.NE.IW) JS=1                                          PT 316
                CALL CURPLT (T,X,Y,N,JS,NSYM,ISIZE,IOP,IRUN,TENSION)        PT 317
          C             DO NEXT VARIABLE                                    PT 318
          20    CONTINUE                                                    PT 319
          C        DO NEXT RUN                                              PT 320
```

CARD NO.

```
   21    CONTINUE
   C        ADVANCE TO ORIGIN OF NEXT FRAME
         D=SHEETW+SPACE+2.0
         IF (D.LT.DMAX) D=DMAX
         CALL NFRAME (D,0.)
   C
   C     READ NEXT SHEET NAMELIST UNLESS NEW SETUP DECK FOLLOWS
   C
         IF (NEWCASE.NE.0) GO TO 22
         GO TO 10
   C
   C     PRINT EDITED DATA
   C
   22    IF (IEDIT.NE.0.AND.IPRT.EQ.0) CALL PRTEDT
         PRINT 27
         GO TO 1
   23    IF (IEDIT.NE.0.AND.IPRT.EQ.0) CALL PRTEDT
   C
   C     FINALIZE PLOTTING DEVICE
   C
         CALL NOTATE (2.,2.,0.5,15HEND OF PLOTTING,90.,15)
         CALL NFRAME (4.,0.)
         CALL CALPLT (0.,0.,999)
   C
   C     PRINT CLOSING MESSAGE
   C
         PRINT 28
         STOP
   C
   24    FORMAT (1H1,3X,12HDATA PLOTTED)
   25    FORMAT (/4X,10HSHEET NO =,I5,5X,6HRUNS =,9(I7,1H,),I7/(30X,9(I7,1H
        1,),I7))
   26    FORMAT (/4X,9HFOR RUN =,I7,3H , ,I4,55H POINTS WERE NOT WITHIN PLO
        1TTING SHEET FRAME FOR PLOT =,I3)
   27    FORMAT (/4X,22HNEW CASE SHOULD FOLLOW)
   28    FORMAT (/4X,22HALL PLOTTING COMPLETED)
         END
```

CARD NO.

```
      SUBROUTINE SETUP (ITEST,SHEETW,SHEETH,SPACE,ISYM,TSIZE,IOP,TENSION ST   1
     1,IEDIT,TOLR,IW,NPLOT,YTAPE,YOFFSET,YSCALE,YLABEL,XTAPE,XOFFSET,XSC ST   2
     2ALE,XLABEL,JREAD,NPLMAX,IERR)                                      ST   3
C                                                                        ST   4
C     ROUTINE TO READ PLOTTING SETUP DECK                                ST   5
C                                                                        ST   6
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983    ST   7
C                                                                        ST   8
C     PARAMETER DEFINITION                                               ST   9
C        ITEST - TEST NUMBER                                             ST  10
C        SHEETW - WIDTH OF PLOTTING SHEET (IN)                           ST  11
C        SHEETH - HEIGHT OF PLOTTING SHEET (IN)                          ST  12
C        SPACE - SPACE BETWEEN PLOTTING SHEETS (IN)                      ST  13
C        ISYM - STARTING SYMBOL NUMBER                                   ST  14
C        IOP - PLOTTING OPTION - IOP=0, PLOT SYMBOLS ONLY                ST  15
C                                IOP=1, PLOT AND FAIR DATA               ST  16
C        TENSION - SPLINE TENSION FACTOR                                 ST  17
C        IEDIT - EDIT OPTION - IEDIT=0, DO NOT SORT AND EDIT DATA        ST  18
C                              IEDIT=1, SORT AND EDIT DATA               ST  19
C        TOLR - EDIT TOLERANCE OF INDEPENDENT VARIABLE                   ST  20
C        IW - DATA ARRAY LOCATION OF INDEPENDENT VARIABLE                ST  21
C        NPLOT - NUMBER OF PLOTS PER SHEET                               ST  22
C        YTAPE - DATA ARRAY LOCATION OF Y VARIABLE                       ST  23
C        YOFFSET - ORIGIN OF Y VARIABLE (IN)                             ST  24
C        YSCALE - VALUE PER INCH FOR Y VARIABLE                          ST  25
C        YLABEL - LABEL FOR Y VARIABLE                                   ST  26
C        XTAPE - DATA ARRAY LOCATION OF X VARIABLE                       ST  27
C        XOFFSET - ORIGIN OF X VARIABLE (IN)                             ST  28
C        XSCALE - VALUE PER INCH FOR X VARIABLE                          ST  29
C        XLABEL - LABEL FOR X VARIABLE                                   ST  30
C        JREAD - TAPE NUMBER OF INPUT FILE                               ST  31
C        IERR - ERROR INDICATOR - IERR=0, NO ERRORS OCCURRED             ST  32
C                                 IERR=1, ERRORS OCCURRED                ST  33
C                                                                        ST  34
C        XTAPE,XOFFSET,XSCALE,XLABEL,YTAPE,YOFFSET,YSCALE, AND YLABEL    ST  35
C        MUST BE PROPERLY DIMENSIONED IN CALLING PROGRAM                 ST  36
      DIMENSION XTAPE(1), XOFFSET(1), XSCALE(1), XLABEL(1), YTAPE(1), YO ST  37
     1FFSET(1), YSCALE(1), YLABEL(1)                                     ST  38
C        DIMENSION AND DEFINE SYMBOL SIZE ARRAY AND NMAX                 ST  39
      DIMENSION IS(3)                                                    ST  40
```

CARD NO.

```
                DATA IS/6HSMALL ,6HMEDIUM,6HLARGE /
          C
          C     INITIALIZE ERROR INDICATOR
          C
                IERR=0
          C
          C     READ AND PRINT TEST NUMBER
          C
                READ (JREAD,6) D1
                IF (EOF(JREAD)) 4,1
          1     ITEST=IFIX(D1+0.0001)
                PRINT 7, ITEST
          C
          C     READ AND PRINT SHEETW, SHEETH, AND SPACE
          C
                READ (JREAD,6) SHEETW,SHEETH,SPACE
                IF (SHEETW.LT.11.0) SHEETW=11.0
                PRINT 8, SHEETW,SHEETH,SPACE
          C
          C     READ AND PRINT ISYM AND ISIZE
          C
                READ (JREAD,6) D1,D2
                ISYM=IFIX(D1+0.0001)
                ISIZE=IFIX(D2+0.0001)
                IF (ISYM.LT.1.OR.ISYM.GT.22) ISYM=1
                IF (ISIZE.LT.1.OR.ISIZE.GT.3) ISIZE=2
                PRINT 9, ISYM,IS(ISIZE)
          C
          C     READ AND PRINT IOP AND TENSION
          C
                READ (JREAD,6) D1,TENSION
                IOP=IFIX(D1+0.0001)
                IF (IOP.LT.0.OR.IOP.GT.1) IOP=0
                IF (IOP.EQ.0) PRINT 10
                IF (IOP.EQ.1) PRINT 11, TENSION
          C
          C     READ AND PRINT IEDIT AND TOLR
          C
                READ (JREAD,6) D1,TOLR
                IEDIT=IFIX(D1+0.0001)
```

CARD NO.

```
      IF (IEDIT.LT.0.OR.IEDIT.GT.1) IEDIT=0
      IF (IEDIT.EQ.1) PRINT 12, TOLR
C
C     READ AND PRINT IW
C
      READ (JREAD,6) D1
      IW=IFIX(D1+0.0001)
      PRINT 13,IW
      IF (IW.LE.0) GO TO 4
C
C     READ AND PRINT NPLOT
C
      READ (JREAD,6) D1
      NPLOT=IFIX(D1+0.0001)
      PRINT 14, NPLOT
      IF (NPLOT.LE.0.OR.NPLOT.GT.NPLMAX) GO TO 3
C
C     READ AND PRINT YTAPE,YOFFSET,YSCALE,YLABEL,XTAPE,XOFFSET,XSCALE,
C     AND XLABEL
C
      DO 2 I=1,NPLOT
      READ (JREAD,15) YTAPE(I),YOFFSET(I),YSCALE(I),YLABEL(I),XTAPE(I),X
     1OFFSET(I),XSCALE(I),XLABEL(I)
      IX=IFIX(XTAPE(I)+0.0001)
      IY=IFIX(YTAPE(I)+0.0001)
      PRINT 16, I,IY,YOFFSET(I),YSCALE(I),YLABEL(I),IX,XOFFSET(I),XSCALE
     1(I),XLABEL(I)
    2 CONTINUE
      GO TO 5
C
C     PRINT ERROR MESSAGES THAT MAY HAVE OCCURRED
C
    3 PRINT 17, NPLOT
      IERR=1
      GO TO 5
    4 IERR=1
C
C     RETURN TO CALLING PROGRAM
C
    5 RETURN
```

LISTING OF DECK: SETUP

CARD NO.

```
                                                                                  ST 121
      C                                                                           ST 122
      6     FORMAT (4F10.4)                                                       ST 123
      7     FORMAT (1H1,3X,30HPLOTTING SETUP DECK FOR TEST =,I5)                  ST 124
      8     FORMAT (4X,13HSHEET WIDTH =,F8.2,5X,14HSHEET HEIGHT =,F8.2,5X,22HS    ST 125
           1PACE BETWEEN SHEETS =,F8.2)                                           ST 126
      9     FORMAT (4X,24HSTARTING SYMBOL NUMBER =,I3,5X,15HSYMBOL SIZE IS ,A6    ST 127
           1)                                                                     ST 128
      10    FORMAT (4X,25HDATA WILL BE PLOTTED ONLY)                              ST 129
      11    FORMAT (4X,59HDATA WILL BE FAIRED AND PLOTTED USING SPLINE WITH TE    ST 130
           1NSION =,F6.2)                                                         ST 131
      12    FORMAT (4X,44HDATA WILL BE EDITED WITH AN EDIT TOLERANCE =,F8.3)      ST 132
      13    FORMAT (4X,39HTAPE LOCATION OF INDEPENDENT VARIABLE =,I4)             ST 133
      14    FORMAT (4X,27HNUMBER OF PLOTS PER SHEET =,I4/7X,8HPLOT NO.,5X,5HYT    ST 134
           1APE,5X,7HYOFFSET,6X,6HYSCALE,6X,6HYLABEL,9X,5HXTAPE,5X,7HXOFFSET,6    ST 135
           2X,6HXSCALE,6X,6HXLABEL)                                               ST 136
      15    FORMAT (3F10.4,A10,3F10.4,A10)                                        ST 137
      16    FORMAT (8X,I3,1X,2(8X,I4,5X,F7.2,2X,F12.4,2X,A10))                    ST 138
      17    FORMAT (/4X,40HERROR OCCURRED --  IN SETUP DECK NPLOT =,I4)           ST 139-
            END
```

CARD NO.

```
      SUBROUTINE DATADK (ICODE,ITEST,IRUN,VAR,NVAR,NLOC,NP,NPT,NPMAX,NDM
     1AX,IERR)
C
C     ROUTINE TO READ FROM OR WRITE TO A DATA DISK
C
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983
C
C     PARAMETER DEFINITION
C
C        ICODE - READ OR WRITE OPTION
C        ITEST - TEST NUMBER
C                ICODE=0, WRITE IN DATA DISK
C                ICODE=1, READ DATA FROM DATA DISK
C        IRUN - RUN NUMBER TO BE READ FROM DATA DISK
C        VAR(I,J) - TWO DIMENSIONAL ARRAY CONTAINING DATA READ FROM
C                   DATA DISK. (I=1,....,NP AND J=1,....,NVAR)
C        NVAR - NUMBER OF VARIABLES READ FROM DISK FOR EACH POINT OF
C               DATA
C        NLOC(J) - ARRAY CONTAINING DATA ARRAY LOCATIONS OF DESIRED
C                  VARIABLES. (J=1,....,NVAR)
C        NP - NUMBER OF DATA POINTS READ FROM DISK FOR DESIRED RUN
C        NPT - ARRAY CONTAINING COUNT NUMBER OF DATA POINTS STORED ON
C              THE DATA DISK
C        NPMAX - MAXIMUM NUMBER OF DATA POINTS
C        NDMAX - MAXIMUM NUMBER OF ALLOWABLE DATA POINTS FOR ANY GIVEN
C                RUN
C        IERR - ERROR INDICATOR
C               IERR=0, NO ERRORS OCCURRED
C               IERR=1, ERROR OCCURRED DURING READ OR WRITE
C               IERR=2, IRUN NOT ON DATA DISK
C
C        NOTE -- VAR, NLOC, AND NPT MUST BE PROPERLY DIMENSIONED IN THE
C                CALLING PROGRAM
C
      DIMENSION VAR(NDMAX,1), NLOC(1), NPT(1)
C
C        WORK ARRAY
C
      COMMON /PLT/ NPRT(10)
C
```

CARD NO.

```
C        DIMENSION DATA(IMAX+3) WHERE IMAX IS THE MAXIMUM NUMBER OF DATA DK  41
C        VARIABLES STORED ON THE INPUT DATA TAPE                         DK  42
C                                                                        DK  43
      COMMON /WORK/ DATA(303)                                            DK  44
C                                                                        DK  45
C        DEFINE ITAPE (INPUT DATA TAPE NUMBER) AND NDISK (DATA DISK      DK  46
C        NUMBER)                                                         DK  47
C                                                                        DK  48
      DATA ITAPE/1/,NDISK/2/                                             DK  49
C        DEFINE IMAX                                                     DK  50
      DATA IMAX/300/                                                     DK  51
C                                                                        DK  52
C     COMPUTE ROUTINE CONSTANTS                                          DK  53
C                                                                        DK  54
      IERR=0                                                             DK  55
      I3=IMAX+3                                                          DK  56
      REWIND 12                                                          DK  57
C                                                                        DK  58
C     SELECT READ OR WRITE PROCEDURE                                     DK  59
C                                                                        DK  60
      IF (ICODE) 1,1,16                                                  DK  61
C                                                                        DK  62
C     WRITE DATA ON DATA DISK                                            DK  63
C                                                                        DK  64
C        INITIALIZE NPT                                                  DK  65
    1 DO 2 I=1,NPMAX                                                     DK  66
    2 NPT(I)=0                                                           DK  67
C        OPEN DATA DISK                                                  DK  68
      CALL OPENMS (NDISK,NPT,NPMAX,0)                                    DK  69
C        COMPUTE NEEDED CONSTANTS                                        DK  70
      I1=IMAX+1                                                          DK  71
      K=0                                                                DK  72
      IRUNP=0                                                            DK  73
      IC=0                                                               DK  74
C        REWIND INPUT TAPE                                               DK  75
      REWIND ITAPE                                                       DK  76
C        READ DATA FROM INPUT TAPE                                       DK  77
    3 READ (ITAPE) IRUN,ITESTC,IPT,(DATA(I),I=1,IMAX)                    DK  78
      IF (EOF(ITAPE)) 10,4                                               DK  79
    4 CONTINUE                                                           DK  80
```

CARD NO.

```
      IF (IRUN.GT.99999) GO TO 10
      IF (ITESTC.NE.ITEST) GO TO 15
C
C        NOTE -- POINTS OR RUNS CAN BE VOIDED HERE
C
C        LOAD DATA ARRAY
      DO 5 I=1,IMAX
      J=I1-I
      J3=J+3
      DATA (J3)=DATA(J)
5     CONTINUE
      DATA (1)=FLOAT(IRUN)
      DATA (2)=FLOAT(ITESTC)
      DATA (3)=FLOAT(IPT)
      K=K+1
C        CHECK TO SEE IF POINT COUNT EXCEEDS ALLOWABLE MAXIMUM
      IF (K-NPMAX) 6,14,14
C        WRITE ON TAPE 12 RUN NUMBER AND COUNT NUMBER OF FIRST DATA
C        POINT OF RUN
6     IF (IRUNP) 7,8,7
7     IF (IRUN-IRUNP) 8,9,8
8     IC=IC+1
      WRITE (12) IRUN,K
      IRUNP=IRUN
C        WRITE DATA ON DATA DISK
9     CALL WRITMS (NDISK,DATA,I3,K,0,0)
C        READ NEXT DATA POINT ON INPUT TAPE
      GO TO 3
C        WRITE ON TAPE 12 RUN NUMBER AND COUNT NUMBER OF LAST DATA
C        POINT + 1 OF RUN
10    K=K+1
      WRITE (12) IRUNP,K
      END FILE 12
C
C     PRINT SUMMARY OF RUNS STORED ON DATA DISK
C
      PRINT 29, ITESTC
      IF (IC.EQ.0) GO TO 13
      REWIND 12
      K=0
```

CARD NO.

```
      DO 12 I=1,IC
      READ (12) IRUN,JS
      K=K+1
      NPRT(K)=IRUN
      IF (K-10) 12,11,11
   11 PRINT 30, (NPRT(J),J=1,10)
      K=0
   12 CONTINUE
      IF (K.GT.0.AND.K.LT.10) PRINT 30, (NPRT(J),J=1,K)
      READ (12) IRUN,K
      K=K-1
      PRINT 31, K
      REWIND 12
      GO TO 28
C
C     PRINT ERROR THAT MAY HAVE OCCURRED DURING WRITE TO DISK
C
   13 PRINT 32
      IERR=1
      GO TO 28
   14 PRINT 33, NPMAX
      IERR=1
      GO TO 28
   15 PRINT 34, ITEST,ITESTC
      IERR=1
      GO TO 28
C
C     READ DATA FROM DATA DISK
C
C        READ TAPE 12 TO FIND POINT COUNT OF FIRST DATA POINT FOR
C        DESIRED RUN
   16 READ (12) J,K
      IF (EOF(12)) 23,17
   17 IF (J-IRUN) 16,18,16
C        READ TAPE 12 TO FIND POINT COUNT OF FIRST DATA POINT FOR
C        NEXT RUN ON DATA DISK
   18 READ (12) J,JS
      IF (EOF(12)) 24,19
C        COMPUTE NUMBER OF DATA POINTS ON DATA DISK FOR DESIRED RUN
   19 NP=JS-K
```

CARD NO.

```
      IF (NP.GT.NDMAX) GO TO 26
C        READ DATA FROM DATA DISK
      DO 22 I=1,NP
      CALL READMS (NDISK,DATA,I3,K)
C        CHECK TEST NUMBER
      ITESTC=IFIX(DATA(2)+0.0001)
      IF (ITESTC.NE.ITEST) GO TO 25
C        STORE DATA NEEDED FOR PLOTTING
      DO 21 J=1,NVAR
      J3=NLOC(J)+3
      IF (J3-I3) 20,20,27
   20 VAR(I,J)=DATA(J3)
   21 CONTINUE
   22 K=K+1
      GO TO 28
C
C     PRINT ERROR THAT MAY HAVE OCCURRED DURING READ FROM DATA DISK
C
   23 PRINT 35, IRUN
      IERR=2
      GO TO 28
   24 PRINT 36, IRUN
      IERR=1
      GO TO 28
   25 PRINT 37, ITEST,ITESTC
      IERR=1
      GO TO 28
   26 PRINT 38, IRUN,NDMAX
      IERR=1
      GO TO 28
   27 PRINT 39, JS,IRUN,IMAX
      IERR=1
C
C     RETURN TO CALLING PROGRAM
C
   28 RETURN
C
   29 FORMAT (1H1,5X,71HTHE FOLLOWING LIST OF RUNS ARE AVAILABLE ON THE
     1DATA DISK FOR TEST NO. ,I5/)
   30 FORMAT (5X,10I10)
```

CARD NO.

```
      31  FORMAT (/5X,29HTOTAL NUMBER OF DATA POINTS =,I7)                  DK 201
      32  FORMAT (/5X,67HERROR OCCURRED DURING WRITE TO DATA DISK ---  NO DA DK 202
         1TA ON INPUT TAPE)                                                 DK 203
      33  FORMAT (/5X,92HERROR OCCURRED DURING WRITE TO DATA DISK ---  POINT DK 204
         1 COUNT HAS EXCEEDED ALLOWABLE MAXIMUM OF ,I7)                     DK 205
      34  FORMAT (/5X,71HERROR OCCURRED DURING WRITE TO DATA DISK ---  REQUE DK 206
         1STED TEST NUMBER IS ,I5,34H AND TEST NUMBER ON INPUT TAPE IS ,I5)  DK 207
      35  FORMAT (/5X,58HERROR OCCURRED DURING READ FROM DATA DISK ---  RUN  DK 208
         1NUMBER ,I7,15H IS NOT ON DISK)                                    DK 209
      36  FORMAT (/5X,91HERROR OCCURRED DURING READ FROM DATA DISK ---  SEAR DK 210
         1CH FOR STARTING POINT FOR RUN FOLLOWING ,I7,14H WAS NOT FOUND)    DK 211
      37  FORMAT (/5X,72HERROR OCCURRED DURING READ FROM DATA DISK ---  REQU DK 212
         1ESTED TEST NUMBER IS ,I5,28H AND TEST NUMBER ON DISK IS ,I5)      DK 213
      38  FORMAT (/5X,47HERROR OCCURRED DURING READ FROM DATA DISK ---  /5X, DK 214
         125HNUMBER OF POINTS FOR RUN ,I7,30H EXCEEDS ALLOWABLE MAXIMUM OF , DK 215
         2I4)                                                               DK 216
      39  FORMAT (/5X,47HERROR OCCURRED DURING READ FROM DATA DISK ---  /5X, DK 217
         118HVARIABLE LOCATION ,I4,23H REQUESTED FOR RUN NO. ,I7,30H EXCEEDS DK 218
         2 ALLOWABLE MAXIMUM OF ,I4)                                        DK 219
          END                                                               DK 220-
```

CARD NO.

```
          SUBROUTINE EDIT (T,X,Y,N,NSYM,ISIZE,IRUN,TOLR,KEDT,IERR)           ED   1
    C                                                                        ED   2
    C                                                                        ED   3
    C     ROUTINE TO SORT AND EDIT DATA                                      ED   4
    C                                                                        ED   5
    C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983    ED   6
    C                                                                        ED   7
    C     PARAMETER DEFINITION                                               ED   8
    C        T - ARRAY CONTAINING INDEPENDENT VARIABLE                       ED   9
    C        X - ARRAY CONTAINING X VARIABLE                                 ED  10
    C        Y - ARRAY CONTAINING Y VARIABLE                                 ED  11
    C        N - NUMBER OF POINTS TO BE SORTED AND EDITED                    ED  12
    C        NSYM - SYMBOL NUMBER TO BE USED TO PLOT UNEDITED POINTS         ED  13
    C        ISIZE - SYMBOL SIZE                                             ED  14
    C        IRUN - RUN NUMBER                                               ED  15
    C        TOLR - EDIT TOLERANCE OF INDEPENDENT VARIABLE                   ED  16
    C        KEDT - NUMBER OF RUNS ALREADY EDITED AND STORED FOR FUTURE      ED  17
    C               PRINT OUT                                                ED  18
    C        IERR - EDIT ERROR INDICATOR                                     ED  19
    C               IERR=0, NO ERRORS OCCURRED                               ED  20
    C               IERR=1, DURING EDIT OF FIRST SORT POINT ALL GREATER      ED  21
    C                       POINTS ARE WITHIN EDIT TOLERANCE                 ED  22
    C               IERR=2, DURING EDIT OF LAST SORT POINT ALL LESSER        ED  23
    C                       POINTS ARE WITHIN EDIT TOLERANCE                 ED  24
    C               IERR=3, DURING EDIT OF INTERMEDIATE SORT POINT ALL       ED  25
    C                       LESSER POINTS ARE WITHIN EDIT TOLERANCE          ED  26
    C               IERR=4, DURING EDIT OF INTERMEDIATE SORT POINT ALL       ED  27
    C                       GREATER POINTS ARE WITHIN EDIT TOLERANCE         ED  28
    C                                                                        ED  29
    C        T, X, AND Y MUST BE PROPERLY DIMENSIONED IN CALLING PROGRAM     ED  30
          DIMENSION T(1), X(1), Y(1)                                         ED  31
    C        WORK ARRAYS                                                     ED  32
          COMMON /WORK/ TIN(50),TEDT(50),TSAVE(50),YSAVE(50),IEDT(50),ITEMP( ED  33
         150),ISAVE(50)                                                      ED  34
    C                                                                        ED  35
    C     CHECK TO SEE IF THIS RUN HAS ALREADY BEEN EDITED AND STORED FOR    ED  36
    C     FUTURE PRINT OUT                                                   ED  37
    C                                                                        ED  38
          IERR=0                                                             ED  39
          IF (N.LT.1) RETURN                                                 ED  40
```

CARD NO.

```
        NSTORE=1
        REWIND 11
        IF (KEDT) 3,3,1
      1 READ (11) NRUN
        IF (EOF(11)) 3,2
      2 IF (IRUN-NRUN) 1,4,1
      3 KEDT=KEDT+1
        NSTORE=0
        IF (KEDT.GT.1) BACKSPACE 11
        WRITE (11) IRUN
        END FILE 11
      4 IN=N
      C
      C SET INITIAL VALUES OF TIN, TEDT, IEDT, AND ITEMP
      C
        DO 5 I=1,N
        TIN(I)=T(I)
        TEDT(I)=0.0
        IEDT(I)=0
      5 ITEMP(I)=I
      C
      C SORT DATA
      C
        K1=N
        DO 12 I=1,N
        IF (I.EQ.1) GO TO 7
        K=0
        DO 6 JJ=1,K1
        IF (JJ.EQ.JSAVE) GO TO 6
        K=K+1
        TEDT(K)=T(JJ)
        IEDT(K)=ITEMP(JJ)
      6 CONTINUE
        TMIN=TEDT(1)
        K1=K1-1
        GO TO 8
      7 TMIN=T(1)
      8 JSAVE=1
        DO 11 J=1,K1
        IF (I.EQ.1) TEDT(J)=T(J)
```

CARD NO.

```
              IF (I.EQ.1) IEDT(J)=ITEMP(J)
              IF (TEDT(J).LE.TMIN) GO TO 9
              GO TO 10
        9     JSAVE=J
              TMIN=TEDT(J)
       10     T(J)=TEDT(J)
              ITEMP(J)=IEDT(J)
       11     CONTINUE
              ISAVE(I)=IEDT(JSAVE)
       12     TSAVE(I)=TMIN
              DO 13 I=1,IN
              K1=ISAVE(I)
       13     YSAVE(I)=Y(K1)
        C
        C     EDIT DATA
        C
              N1=N-1
              IF (N1.LT.1.OR.TOLR.LE.0.0) GO TO 31
              K1=0
              DO 15 I=1,N1
              ITEMP(I)=0
              DIFF=ABS(TSAVE(I+1)-TSAVE(I))
              IF (DIFF.LE.TOLR) GO TO 14
              GO TO 15
       14     ITEMP(I)=I
              K1=1
       15     CONTINUE
              IF (K1.EQ.0) GO TO 31
              N=1
              I=1
       16     K1=I
              J=0
              IF (I.GT.N1) GO TO 27
              IF (ITEMP(I).NE.0) GO TO 17
              IEDT(N)=ISAVE(I)
              N=N+1
              I=I+1
              IEDT(N)=ISAVE(I)
              GO TO 16
       17     I=I+1
```

CARD NO.

```
                J=J+1                                              ED 121
                IF (I.EQ.IN) GO TO 21                              ED 122
                IF (ITEMP(I).EQ.0) GO TO 18                        ED 123
                GO TO 17                                           ED 124
          18    K2=K1+J                                            ED 125
                IF (K1.EQ.1) GO TO 19                              ED 126
                I1=K1-1                                            ED 127
                I2=K2+1                                            ED 128
                GO TO 23                                           ED 129
          19    I1=K2                                              ED 130
          20    I1=I1+1                                            ED 131
                I2=I1+1                                            ED 132
                IF (I2.GT.IN) IERR=1                               ED 133
                IF (I2.GT.IN) GO TO 29                             ED 134
                DIFF=ABS(TSAVE(I2)-TSAVE(I1))                      ED 135
                IF (DIFF.LE.TOLR) GO TO 20                         ED 136
                GO TO 23                                           ED 137
          21    I2=K1                                              ED 138
          22    I2=I2-1                                            ED 139
                I1=I2-1                                            ED 140
                IF (I1.LT.1) IERR=2                                ED 141
                IF (I1.LT.1) GO TO 29                              ED 142
                DIFF=ABS(TSAVE(I2)-TSAVE(I1))                      ED 143
                IF (DIFF.LE.TOLR) GO TO 22                         ED 144
                K2=K1+J                                            ED 145
          23    IF (I1.LT.1) IERR=3                                ED 146
                IF (I1.LT.1) GO TO 29                              ED 147
                IF (I2.GT.IN) IERR=4                               ED 148
                IF (I2.GT.IN) GO TO 29                             ED 149
                IEDT(N)=ISAVE(K1)                                  ED 150
                T(1)=YSAVE(I1)*TSAVE(I2)-YSAVE(I2)*TSAVE(I1)       ED 151
                T(2)=YSAVE(I2)-YSAVE(I1)                           ED 152
                T(3)=TSAVE(I2)-TSAVE(I1)                           ED 153
                TMIN=ABS(YSAVE(K1)-(T(1)+T(2)*TSAVE(K1))/T(3))     ED 154
                K1=K1+1                                            ED 155
                DO 25 K=K1,K2                                      ED 156
                DIFF=ABS(YSAVE(K)-(T(1)+T(2)*TSAVE(K))/T(3))       ED 157
                IF (DIFF.LE.TMIN) GO TO 24                         ED 158
                GO TO 25                                           ED 159
          24    TMIN=DIFF                                          ED 160
```

CARD NO.

```
          IEDT(N)=ISAVE(K)
   25     CONTINUE
          IF (K2.EQ.IN) GO TO 27
          IF (I.EQ.N1) GO TO 26
          N=N+1
          I=I+1
          GO TO 16
   26     N=N+1
          IEDT(N)=ISAVE(IN)
   27     DO 28 K=1,N
          K1=IEDT(K)
   28     TEDT(K)=TIN(K1)
          GO TO 33
   29     IF (NSTORE.EQ.0) PRINT 40, IRUN,TOLR,IERR
          DO 30 I=1,IN
          K1=ISAVE(I)
          CALL PNTPLT (X(K1),Y(K1),NSYM,ISIZE)
          IF (NSTORE.EQ.0) PRINT 41, I,K1,TSAVE(I),X(K1),Y(K1)
   30     CONTINUE
          RETURN
   31     DO 32 I=1,IN
          K1=ISAVE(I)
          TEDT(I)=TIN(K1)
   32     IEDT(I)=K1
C
C         WRITE EDITED DATA ON TAPE 7 FOR USE IN SUBROUTINE PRTEDT
C
   33     IF (NSTORE.EQ.1) GO TO 34
          WRITE (7) IRUN,IN,N,TOLR
          WRITE (7) (TIN(I),I=1,IN)
          WRITE (7) (ISAVE(I),TSAVE(I),I=1,IN)
          WRITE (7) (IEDT(I),TEDT(I),I=1,N)
C
C         REORDER T, X, AND Y FOR PLOTTING
C
   34     DO 35 I=1,IN
   35     T(I)=TEDT(I)
          DO 36 I=1,IN
          TSAVE(I)=X(I)
   36     TEDT(I)=Y(I)
```

```
      DO 37 I=1,IN
      K=IEDT(I)
      X(I)=TSAVE(K)
   37 Y(I)=TEDT(K)
C
C     PLOT DATA POINTS NOT EDITED
C
      IF (N.EQ.IN) RETURN
      DO 39 I=1,IN
      K1=0
      DO 38 J=1,N
      IF (IEDT(J).EQ.I) K1=1
   38 CONTINUE
      IF (K1.EQ.0) CALL PNTPLT (TSAVE(I),TEDT(I),NSYM,ISIZE)
      IF ((K1.EQ.0).AND.(NSTORE.EQ.0)) WRITE (7) I,TIN(I)
   39 CONTINUE
C
C
C     RETURN TO CALLING PROGRAM
C
      RETURN
C
   40 FORMAT (//2X,19HATTEMPT TO EDIT RUN,I7,1X,29HFAILED FOR AN EDIT TO
     1LERANCE=,F8.3,10X,9HERROR NO.,I3//4X,1HI,4X,2HIS,6X,1HT,9X,1HX,9X,
     21HY/)
   41 FORMAT (2I5,3F10.4)
      END
```

CARD NO.

```
      SUBROUTINE PRTEDT
C
C     ROUTINE TO PRINT EDITED DATA STORED ON TAPE 7
C
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983
C
C        WORK ARRAYS
      COMMON /WORK/ TIN(50),ISAVE(50),TSAVE(50),IEDT(50),TEDT(50),IUN(50
     1),TUN(50)
C
      END FILE 7
      REWIND 7
      PRINT 7
    1 READ (7) IRUN,IN,N,TOLR
      IF (EOF(7)) 6,2
    2 NR=IN-N
      READ (7) (TIN(I),I=1,IN)
      READ (7) (ISAVE(I),TSAVE(I),I=1,IN)
      READ (7) (IEDT(I),TEDT(I),I=1,N)
      IF (NR.LE.0) GO TO 4
      DO 3 I=1,NR
    3 READ (7) IUN(I),TUN(I)
    4 PRINT 8, IRUN,TOLR
      DO 5 I=1,IN
      IF ((I.LE.N).AND.(I.LE.NR)) PRINT 9, I,TIN(I),ISAVE(I),TSAVE(I),IE
     1DT(I),TEDT(I),IUN(I),TUN(I)
      IF ((I.LE.N).AND.(I.GT.NR)) PRINT 10, I,TIN(I),ISAVE(I),TSAVE(I),I
     1EDT(I),TEDT(I)
      IF ((I.GT.N).AND.(I.LE.NR)) PRINT 11, I,TIN(I),ISAVE(I),TSAVE(I),I
     1UN(I),TUN(I)
      IF ((I.GT.N).AND.(I.GT.NR)) PRINT 12, I,TIN(I),ISAVE(I),TSAVE(I)
    5 CONTINUE
      GO TO 1
    6 RETURN
C
    7 FORMAT (1H1,10X,35HTHE FOLLOWING RUNS HAVE BEEN EDITED)
    8 FORMAT (//1X,9HRUN NO. =,I7,10X,11HTOLERANCE =,F8.3//9X,5HINPUT,19
     1X,6HSORTED,19X,6HEDITED,20X,7HDELETED//4(1X,9HPOINT NO.,3X,7HT-VAL
     2UE,5X)/)
    9 FORMAT (4(5X,I3,2X,F10.4,5X))
```

```
CARD NO.

   41        10     FORMAT (3(5X,I3,2X,F10.4,5X))                              PR  41
             11     FORMAT (2(5X,I3,2X,F10.4,5X),30X,I3,2X,F10.4)             PR  42
             12     FORMAT (2(5X,I3,2X,F10.4,5X))                              PR  43
                    END                                                        PR  44-
```

CARD NO.

```
      SUBROUTINE ZEROLN (XO,YO,DIST,LINE)
C
C     ROUTINE TO DRAW ZERO LINES
C
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983
C
C     PARAMETER DEFINITION
C        XO,YO - STARTING LOCATION OF ZERO LINE
C        DIST - LENGTH OF ZERO LINE
C        LINE - AXIS OPTION
C               LINE=1, ZERO LINE PARALLEL TO X-AXIS
C               LINE=2, ZERO LINE PARALLEL TO Y-AXIS
C
C        NOTE - THICKNESS OF ZERO LINE = 2.*D
      D=0.014
      CALL CALPLT (XO,YO,3)
      IF (LINE.EQ.2) GO TO 1
      X=XO+DIST
      CALL CALPLT (X,YO,2)
      Y=YO+D
      CALL CALPLT (X,Y,2)
      CALL CALPLT (XO,Y,2)
      Y=YO-D
      CALL CALPLT (XO,Y,2)
      CALL CALPLT (X,Y,2)
      CALL CALPLT (X,YO,2)
      GO TO 2
    1 Y=YO+DIST
      CALL CALPLT (XO,Y,2)
      X=XO+D
      CALL CALPLT (X,Y,2)
      CALL CALPLT (X,YO,2)
      X=XO-D
      CALL CALPLT (X,YO,2)
      CALL CALPLT (X,Y,2)
      CALL CALPLT (XO,Y,2)
    2 CALL CALPLT (XO,YO,3)
      RETURN
      END
```

CARD NO.

```
      SUBROUTINE CURPLT (T,X,Y,N,IVAR,NSYM,ISIZE,IOP,IRUN,TENSION)
C
C     ROUTINE TO PLOT AND FAIR DATA
C
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983
C
C     PARAMETER DEFINITION
C        T - ARRAY CONTAINING INDEPENDENT VARIABLE
C        X - ARRAY CONTAINING X VARIABLE
C        Y - ARRAY CONTAINING Y VARIABLE
C        N - NUMBER OF POINTS TO BE PLOTTED
C        IVAR - VARIABLE CODE  IVAR=0 IF X AND T ARE THE SAME
C                              IVAR=1 IF X AND T ARE DIFFERENT
C        NSYM - SYMBOL NUMBER
C        ISIZE - SYMBOL SIZE
C        IOP - PLOTTING OPTION  IOP=0 PLOT SYMBOLS ONLY
C                               IOP=1 PLOT CURVE AND SYMBOLS
C        IRUN - RUN NUMBER
C        TENSION - SPLINE TENSION
C
C        T, X, AND Y MUST BE PROPERLY DIMENSIONED IN CALLING ROUTINE
      DIMENSION T(1), X(1), Y(1), H(3)
C        WORK ARRAYS
      COMMON /WORK/ DS1(105),DS2(105),DUMX(105),DUMY(105)
      COMMON /PLT/ MX(50),MY(50),A(50)
C
      REAL MX,MY
C        ROUTINE CONSTANTS
      DATA NH/101/,XNT/100./,EPS/0.0001/,NTA/100/,H/.02,.03,.04/
C
C     SELECT PLOTTING OPTION
C
      IF (IOP.EQ.0.OR.N.LT.3) GO TO 1
      GO TO 3
C
C     PLOT SYMBOLS ONLY
C
1     DO 2 I=1,N
2     CALL PNTPLT (X(I),Y(I),NSYM,ISIZE)
      RETURN
```

CARD NO.

```
      C
      C     PLOT CURVE AND SYMBOL
      C
      C        CHECK TO SEE IF X OR T IS STRICTLY INCREASING
      3     DO 5 I=2,N
            IF (IVAR.EQ.1) GO TO 4
            IF (X(I).LT.X(I-1)) GO TO 6
            GO TO 5
      4     IF (T(I).LT.T(I-1)) GO TO 6
      5     CONTINUE
            GO TO 7
      C        PRINT ERROR MESSAGE IF X OR T NOT STRICTLY INCREASING
      6     IF (IVAR.EQ.0) PRINT 26, IRUN
            IF (IVAR.EQ.0) PRINT 27, (X(I),Y(I),I=1,N)
            IF (IVAR.EQ.1) PRINT 28, IRUN
            IF (IVAR.EQ.1) PRINT 29, (T(I),X(I),Y(I),I=1,N)
      C        PLOT SYMBOLS ONLY IF X OR T NOT STRICTLY INCREASING
            GO TO 1
      C        FIT SPLINE CURVE THROUGH DATA POINTS
      7     IF (IVAR.EQ.0) CALL CUBSPL (X,Y,N,MX,TENSION,TENS,A)
            IF (IVAR.EQ.1) CALL CUBSPL (T,X,N,MY,TENSION,TENS,A)
            IF (IVAR.EQ.1) CALL CUBSPL (T,Y,N,MY,TENSION,TENS,A)
      C        PLOT FIRST POINT
            CALL PNTPLT (X(1),Y(1),NSYM,ISIZE)
      C        PLOT AND FAIR REMAINING POINTS
            NN=N-1
            DO 25 I=1,NN
      C        COMPUTE STRAIGHT LINE DISTANCE BETWEEN TWO POINTS. IF DISTANCE
      C        LESS THAN SYMBOL DIAMETER, PLOT POINTS ONLY.
            X1=X(I+1)-X(I)
            Y1=Y(I+1)-Y(I)
            DS=SQRT(X1*X1+Y1*Y1)
            T1=ATANF(Y1,X1)
            CALL SYMBS (NSYM,ISIZE,XS1,YS1,T1)
            DSS1=SQRT(XS1*XS1+YS1*YS1)
            X1=-X1
            Y1=-Y1
            T1=ATANF(Y1,X1)
            CALL SYMBS (NSYM,ISIZE,XS2,YS2,T1)
            DSS2=SQRT(XS2*XS2+YS2*YS2)
```

CARD NO.

```
      IF ((DSS1+DSS2).GE.DS) GO TO 24
C        COMPUTE DISTANCE ALONG CURVE AS A FUNCTION OF X OR T BETWEEN
C        POINT I AND I+1
      IF (IVAR.EQ.0) NT=IFIX(XNT*(X(I+1)-X(I)))+1
      IF (IVAR.EQ.1) NT=IFIX(XNT*(T(I+1)-T(I)))+1
      IF (NT.LT.3) NT=3
      IF (NT.GT.NH) NT=NH
      IF (IVAR.EQ.0) DELTA=(X(I+1)-X(I))/FLOAT(NT-1)
      IF (IVAR.EQ.1) DELTA=(T(I+1)-T(I))/FLOAT(NT-1)
      DUMX(1)=X(I)
      DUMY(1)=Y(I)
      DS1(1)=0.0
      IF (IVAR.EQ.0) DA=X(I)
      IF (IVAR.EQ.1) DA=T(I)
C
      DO 9 J=2,NT
      DA=DA+DELTA
      IF (IVAR.EQ.1) GO TO 8
      DUMX(J)=DA
      DUMY(J)=FUNC(DA,X(I),X(I+1),Y(I),Y(I+1),MX(I),MX(I+1),TENS)
      GO TO 9
    8 DUMY(J)=FUNC(DA,T(I),T(I+1),Y(I),Y(I+1),MY(I),MY(I+1),TENS)
      DUMX(J)=FUNC(DA,T(I),T(I+1),X(I),X(I+1),MX(I),MX(I+1),TENS)
    9 DS1(J)=SQRT((DUMX(J)-X(I))**2+(DUMY(J)-Y(I))**2)
C
      DO 10 J=1,NT
      K=NT+1-J
   10 DS2(J)=SQRT((DUMX(K)-X(I+1))**2+(DUMY(K)-Y(I+1))**2)
C        FIND X AND Y LOCATION WHERE SYMBOL AND CURVE INTERSECT
      DELTA=H(ISIZE)
      IF (IVAR.EQ.0) DA=X(I)
      IF (IVAR.EQ.1) DA=T(I)
C
      DO 14 J=2,NTA
      DA=DA+DELTA
      IF (IVAR.EQ.1) GO TO 11
      IF (DA.GE.X(I+1)) GO TO 15
      X1=DA-X(I)
      Y1=FUNC(DA,X(I),X(I+1),Y(I),Y(I+1),MX(I),MX(I+1),TENS)-Y(I)
      GO TO 12
```

CARD NO.

```
      11     IF (DA.GE.T(I+1)) GO TO 15
             Y1=FUNC(DA,T(I),T(I+1),Y(I),Y(I+1),MY(I),MY(I+1),TENS)-Y(I)
             X1=FUNC(DA,T(I),T(I+1),X(I),X(I+1),MX(I),MX(I+1),TENS)-X(I)
      12     DS=SQRT(X1*X1+Y1*Y1)
             T1=ATANF(Y1,X1)
             CALL SYMBS (NSYM,ISIZE,XS1,YS1,T1)
             DSS1=SQRT(XS1*XS1+YS1*YS1)
             IF (ABS(DS-DSS1).LE.EPS) GO TO 15
             IF (DS.GT.DSS1) GO TO 13
             GO TO 14
      13     DA=DA-DELTA
             DELTA=DELTA/2.
      14     CONTINUE
      C
      15     XS1=XS1+X(I)
             YS1=YS1+Y(I)
             DELTA=H(ISIZE)
             IF (IVAR.EQ.0) DA=X(I+1)
             IF (IVAR.EQ.1) DA=T(I+1)
      C
             DO 19 J=2,NTA
             DA=DA-DELTA
             IF (IVAR.EQ.1) GO TO 16
             IF (DA.LE.X(I)) GO TO 20
             X1=DA-X(I+1)
             Y1=FUNC(DA,X(I),X(I+1),Y(I),Y(I+1),MX(I),MX(I+1),TENS)-Y(I+1)
             GO TO 17
      16     IF (DA.LE.T(I)) GO TO 20
             Y1=FUNC(DA,T(I),T(I+1),Y(I),Y(I+1),MY(I),MY(I+1),TENS)-Y(I+1)
             X1=FUNC(DA,T(I),T(I+1),X(I),X(I+1),MX(I),MX(I+1),TENS)-X(I+1)
      17     DS=SQRT(X1*X1+Y1*Y1)
             T1=ATANF(Y1,X1)
             CALL SYMBS (NSYM,ISIZE,XS2,YS2,T1)
             DSS2=SQRT(XS2*XS2+YS2*YS2)
             IF (ABS(DS-DSS2).LE.EPS) GO TO 20
             IF (DS.GT.DSS2) GO TO 18
             GO TO 19
      18     DA=DA+DELTA
             DELTA=DELTA/2.
      19     CONTINUE
```

CARD NO.

```
      C
   20 XS2=XS2+X(I+1)
      YS2=YS2+Y(I+1)
      NP=1
      NDS1=0
      C
      DO 21 J=2,NT
      IF ((DS1(J).LT.DSS1).AND.(NDS1.EQ.0)) GO TO 21
      IF ((DS1(J).GE.DSS1).AND.(NDS1.EQ.0)) NDS1=1
      NP=NP+1
      DUMX(NP)=DUMX(J)
      DUMY(NP)=DUMY(J)
   21 CONTINUE
      C    PLOT CURVE BETWEEN POINTS
      DUMX(1)=XS1
      DUMY(1)=YS1
      DO 22 J=1,NT
      IF (DS2(J).LE.DSS2) GO TO 22
      GO TO 23
   22 NP=NP-1
   23 NP=NP+1
      DUMX(NP)=XS2
      DUMY(NP)=YS2
      DUMX(NP+1)=DUMY(NP+1)=0.0
      DUMX(NP+2)=DUMY(NP+2)=1.0
      CALL LINE (DUMX,DUMY,NP,1,0,0,0)
      C    PLOT SYMBOL AT POINT I+1
   24 CALL PNTPLT (X(I+1),Y(I+1),NSYM,ISIZE)
      C    ADVANCE TO NEXT POINT
   25 CONTINUE
      C
      C    RETURN TO CALLING PROGRAM
      C
      RETURN
      C
   26 FORMAT (1H1//5X,36HX IS NOT STRICTLY INCREASING FOR RUN,I7/16X,1HX
     1,13X,1HY/)
   27 FORMAT (5X,2F15.4)
   28 FORMAT (1H1//5X,36HT IS NOT STRICTLY INCREASING FOR RUN,I7/16X,1HT
     1,13X,1HX,13X,1HY/)
```

CARD NO.

```
 201        29    FORMAT (5X,3F15.4)                                    CU 201
                  END                                                   CU 202-
```

CARD NO.

```
              FUNCTION ATANF (DY,DX)                                           AF   1
        C                                                                      AF   2
        C     ROUTINE TO COMPUTE ARCTANGENT FOR ANGLES FROM 0 TO 360 DEGREES   AF   3
        C                                                                      AF   4
        C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983  AF   5
        C                                                                      AF   6
              IF (DX.EQ.0.0) GO TO 1                                           AF   7
              ATANF=ABS(DY/DX)                                                 AF   8
              ATANF=ATAN(ATANF)*57.2957795131                                  AF   9
              IF ((DX.GT.0.0).AND.(DY.LT.0.0)) ATANF=360.-ATANF                AF  10
              IF ((DX.LT.0.0).AND.(DY.GT.0.0)) ATANF=180.-ATANF                AF  11
              IF ((DX.LT.0.0).AND.(DY.LT.0.0)) ATANF=180.+ATANF                AF  12
              RETURN                                                           AF  13
        1     IF (DY.LT.0.0) ATANF=270.                                        AF  14
              IF (DY.GE.0.0) ATANF=90.                                         AF  15
              RETURN                                                           AF  16
              END                                                              AF  17-
```

CARD NO.

```
      SUBROUTINE SYMBS (NO,IS,X,Y,T)                                    SY   1
C                                                                       SY   2
C     ROUTINE TO COMPUTE THE X AND Y INTERCEPTS OF A RADIAL LINE DRAWN  SY   3
C     FROM THE CENTER OF A GIVEN SYMBOL WITH THE SIDE OF THAT SYMBOL    SY   4
C                                                                       SY   5
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983   SY   6
C                                                                       SY   7
C     PARAMETER DEFINITION                                              SY   8
C        NO  -  SYMBOL NUMBER                                           SY   9
C        IS  -  SYMBOL SIZE   1-SMALL   2-MEDIUM   3-LARGE              SY  10
C        X AND Y  -  INTERCEPT OF SYMBOL AND RADIAL DRAWN FROM SYMBOL   SY  11
C                  CENTER AT ANGLE T.                                   SY  12
C        T  -  ANGLE OF RADIAL DRAWN FROM SYMBOL CENTER                 SY  13
C                                                                       SY  14
      DIMENSION SCALE(3)                                                SY  15
      DATA RAD/57.2957795131/,PI/3.141592654/                           SY  16
      DATA DA/1.4142135624/,R/1.7320508076/,SCALE/.13,.16,.19/          SY  17
      DATA T1/213.6900675260/,T2/326.3099324740/,T3/116.5650511771/,T4/3 SY  18
     133.4349488229/,T7/11.3099324740/,T8/168.6900675260/,T9/218.6598082 SY  19
     2541/,S1/321.3401917459/,S2/185.7105931375/,S3/354.2894068625/,S4/1 SY  20
     31.3099324740/,S5/168.6900675260/,S6/218.6598082541/,S7/321.3401917 SY  21
     4459/                                                              SY  22
C                                                                       SY  23
      IF ((NO.EQ.1).OR.(NO.EQ.11)) GO TO 1                              SY  24
      IF ((NO.EQ.2).OR.(NO.EQ.12)) GO TO 2                              SY  25
      IF ((NO.EQ.3).OR.(NO.EQ.13)) GO TO 8                              SY  26
      IF ((NO.EQ.4).OR.(NO.EQ.14)) GO TO 13                             SY  27
      IF ((NO.EQ.5).OR.(NO.EQ.15)) GO TO 17                             SY  28
      IF ((NO.EQ.6).OR.(NO.EQ.16)) GO TO 21                             SY  29
      IF ((NO.EQ.7).OR.(NO.EQ.17)) GO TO 26                             SY  30
      IF ((NO.EQ.8).OR.(NO.EQ.18)) GO TO 31                             SY  31
      IF ((NO.EQ.9).OR.(NO.EQ.19)) GO TO 35                             SY  32
      IF ((NO.EQ.10).OR.(NO.EQ.20)) GO TO 40                            SY  33
      IF ((NO.EQ.21).OR.(NO.EQ.22)) GO TO 45                            SY  34
      IF (NO.GT.22) RETURN                                              SY  35
C                                                                       SY  36
C        SYMBOL NUMBER 1 OR 11                                          SY  37
C                                                                       SY  38
    1 X=SCALE(IS)*.5525*COS(T/RAD)                                      SY  39
      Y=SCALE(IS)*.5525*SIN(T/RAD)                                      SY  40
```

CARD NO.

```
      RETURN
C
C        SYMBOL NUMBER 2 OR 12
C
    2 IF ((T.GE.0.).AND.(T.LT.45.)) GO TO 3
      IF ((T.GE.45.0).AND.(T.LT.135.)) GO TO 4
      IF ((T.GE.135.).AND.(T.LT.225.)) GO TO 5
      IF ((T.GE.225.).AND.(T.LT.315.)) GO TO 6
      IF ((T.GE.315.).AND.(T.LE.360.)) GO TO 7
    3 X=SCALE(IS)/2.
      Y=X*TAN(T/RAD)
      RETURN
    4 Y=SCALE(IS)/2.
      IF (T.EQ.90.) X=0.0
      IF (T.NE.90.) X=Y/TAN(T/RAD)
      RETURN
    5 X=-SCALE(IS)/2.
      Y=X*TAN(T/RAD)
      RETURN
    6 Y=-SCALE(IS)/2.
      IF (T.EQ.270.) X=0.0
      IF (T.NE.270.) X=Y/TAN(T/RAD)
      RETURN
    7 X=SCALE(IS)/2.
      Y=X*TAN(T/RAD)
      RETURN
C
C        SYMBOL NUMBER 3 OR 13
C
    8 IF ((T.GE.0.).AND.(T.LT.90.)) GO TO 9
      IF ((T.GE.90.).AND.(T.LT.180.)) GO TO 10
      IF ((T.GE.180.).AND.(T.LT.270.)) GO TO 11
      IF ((T.GE.270.).AND.(T.LE.360.)) GO TO 12
    9 X=SCALE(IS)*DA/2./(TAN(T/RAD)+1.)
      Y=-X+SCALE(IS)*DA/2.
      RETURN
   10 IF (T.EQ.90.) X=0.0
      IF (T.NE.90.) X=SCALE(IS)*DA/2./(TAN(T/RAD)-1.)
      Y=X+SCALE(IS)*DA/2.
      RETURN
```

```
 11   X=-SCALE(IS)*DA/2./(TAN(T/RAD)+1.)
      Y=-X-SCALE(IS)*DA/2.
      RETURN
 12   IF (T.EQ.270.) X=0.0
      IF (T.NE.270.) X=-SCALE(IS)*DA/2./(TAN(T/RAD)-1.)
      Y=X-SCALE(IS)*DA/2.
      RETURN
C
C        SYMBOL NUMBER 4 OR 14
C
 13   IF ((T.GE.0.).AND.(T.LT.90.)) GO TO 14
      IF ((T.GE.90.).AND.(T.LT.T1)) GO TO 15
      IF ((T.GE.T1).AND.(T.LT.T2)) GO TO 16
 14   X=(2./3.)*SCALE(IS)*1.105/(TAN(T/RAD)+2.)
      Y=-2.*X+2.*SCALE(IS)*1.105/3.
      RETURN
 15   IF (T.EQ.90.) X=0.0
      IF (T.NE.90.) X=(2./3.)*SCALE(IS)*1.105/(TAN(T/RAD)-2.)
      Y=2.*X+2.*SCALE(IS)*1.105/3.
      RETURN
 16   Y=-SCALE(IS)*1.105/3.
      IF (T.NE.270.) X=0.0
      IF (T.NE.270.) X=Y/TAN(T/RAD)
      RETURN
C
C        SYMBOL NUMBER 5 OR 15
C
 17   IF ((T.GE.0.).AND.(T.LT.T3)) GO TO 18
      IF ((T.GE.T3).AND.(T.LT.225.)) GO TO 19
      IF ((T.GE.225.).AND.(T.LT.T4)) GO TO 20
 18   IF (T.EQ.90.) X=0.0
      IF (T.NE.90.) X=SCALE(IS)*1.22222/3./(TAN(T/RAD)+1.)
      Y=-X+SCALE(IS)*1.22222/3.
      RETURN
 19   X=-SCALE(IS)*1.22222/3.
      Y=X*TAN(T/RAD)
      RETURN
 20   Y=-SCALE(IS)*1.22222/3.
      IF (T.EQ.270.) X=0.0
      IF (T.NE.270.) X=Y/TAN(T/RAD)
```

CARD NO.

```
          RETURN
    C
    C        SYMBOL NUMBER 6 OR 16
    C
    21    A=4.*SCALE(IS)*1.22222/(3.*PI)
          B=SCALE(IS)*1.22222-A
          T5=ATAN(A/B)*RAD
          T6=360.-T5
          T5=90.+T5
          IF ((T.GE.0.).AND.(T.LT.T5)) GO TO 22
          IF ((T.GE.T5).AND.(T.LT.225.)) GO TO 24
          IF ((T.GE.225.).AND.(T.LT.T6)) GO TO 25
    22    IF (T.EQ.90.) GO TO 23
          BB=-2.*A*(1.+TAN(T/RAD))
          AA=TAN(T/RAD)**2+1.
          CC=2.*A*A-(SCALE(IS)*1.22222)**2
          X=SQRT(BB*BB-4.*AA*CC)/(2.*AA)
          IF ((T.GE.0.).AND.(T.LT.90.)) X=X+BB/(2.*AA)
          IF ((T.GE.90.).AND.(T.LT.180.)) X=-X+BB/(2.*AA)
          IF ((T.GE.270.).AND.(T.LE.360.)) X=X+BB/(2.*AA)
          Y=X*TAN(T/RAD)
          RETURN
    23    X=0.0
          Y=-A+SQRT((SCALE(IS)*1.22222)**2-A*A)
          RETURN
    24    X=-A
          Y=X*TAN(T/RAD)
          RETURN
    25    Y=-A
          IF (T.EQ.270.) X=0.0
          IF (T.NE.270.) X=Y/TAN(T/RAD)
          RETURN
    C
    C        SYMBOL NUMBER 7 OR 17
    C
    26    IF ((T.GE.0.0).AND.(T.LT.T7)) GO TO 27
          IF ((T.GE.T7).AND.(T.LT.T8)) GO TO 28
          IF ((T.GE.T8).AND.(T.LT.T9)) GO TO 29
          IF ((T.GE.T9).AND.(T.LT.S1)) GO TO 30
    27    X=SCALE(IS)/2.
```

CARD NO.

```
                Y=X*TAN(T/RAD)                                                  SY 161
                RETURN                                                          SY 162
          28    X=SCALE(IS)*.1*SIN(T/RAD)*COS(T/RAD)+SCALE(IS)*COS(T/RAD)*SQRT(.25 SY 163
               1-(.1*COS(T/RAD))**2)                                            SY 164
                Y=X*TAN(T/RAD)                                                  SY 165
                RETURN                                                          SY 166
          29    X=-SCALE(IS)/2.                                                 SY 167
                Y=X*TAN(T/RAD)                                                  SY 168
                RETURN                                                          SY 169
          30    Y=-SCALE(IS)*.4                                                 SY 170
                IF (T.EQ.270.) X=0.0                                            SY 171
                IF (T.NE.270.) X=Y/TAN(T/RAD)                                   SY 172
                RETURN                                                          SY 173
          C                                                                     SY 174
          C        SYMBOL NUMBER 8 OR 18                                        SY 175
          C                                                                     SY 176
          31    IF ((T.GE.0.0).AND.(T.LT.S2)) GO TO 32                          SY 177
                IF ((T.GE.S2).AND.(T.LE.270.)) GO TO 33                         SY 178
                IF ((T.GT.270.).AND.(T.LE.S3)) GO TO 34                         SY 179
          32    X=-.05*SCALE(IS)*SIN(T/RAD)*COS(T/RAD)+SCALE(IS)*COS(T/RAD)*SQRT(. SY 180
               125-(.05*COS(T/RAD))**2)                                         SY 181
                Y=X*TAN(T/RAD)                                                  SY 182
                RETURN                                                          SY 183
          33    IF (T.EQ.270.) X=0.0                                            SY 184
                IF (T.EQ.270.) Y=-SCALE(IS)*.55                                 SY 185
                IF (T.NE.270.) X=-SCALE(IS)*.55/(TAN(T/RAD)+1.)                 SY 186
                IF (T.NE.270.) Y=X*TAN(T/RAD)                                   SY 187
                RETURN                                                          SY 188
          34    X=-SCALE(IS)*.55/(TAN(T/RAD)-1.)                                SY 189
                Y=X*TAN(T/RAD)                                                  SY 190
                RETURN                                                          SY 191
          C                                                                     SY 192
          C        SYMBOL NUMBER 9 OR 19                                        SY 193
          C                                                                     SY 194
          35    IF ((T.GE.0.0).AND.(T.LT.90.)) GO TO 36                         SY 195
                IF ((T.GE.90.).AND.(T.LT.180.)) GO TO 37                        SY 196
                IF ((T.GE.180.).AND.(T.LT.270.)) GO TO 38                       SY 197
                IF ((T.GE.270.).AND.(T.LE.360.)) GO TO 39                       SY 198
          36    X=SCALE(IS)*R/2./(TAN(T/RAD)+R)                                 SY 199
                Y=X*TAN(T/RAD)                                                  SY 200
```

CARD NO.

```
            RETURN                                               SY 201
   37       IF (T.EQ.90.) X=0.0                                  SY 202
            IF (T.NE.90.) X=SCALE(IS)*R/2./(TAN(T/RAD)-R)        SY 203
            IF (T.EQ.90.) Y=SCALE(IS)*R/2.                       SY 204
            IF (T.NE.90.) Y=X*TAN(T/RAD)                         SY 205
            RETURN                                               SY 206
   38       X=-SCALE(IS)*R/2./(TAN(T/RAD)+R)                     SY 207
            Y=X*TAN(T/RAD)                                       SY 208
            RETURN                                               SY 209
   39       IF (T.EQ.270.) X=0.0                                 SY 210
            IF (T.EQ.270.) Y=-SCALE(IS)*R/2.                     SY 211
            IF (T.NE.270.) X=-SCALE(IS)*R/2./(TAN(T/RAD)-R)      SY 212
            IF (T.NE.270.) Y=X*TAN(T/RAD)                        SY 213
            RETURN                                               SY 214
   C                                                             SY 215
   C           SYMBOL NUMBER 10 OR 20                            SY 216
   C                                                             SY 217
   40       IF ((T.GE.0.0).AND.(T.LT.S4)) GO TO 41               SY 218
            IF ((T.GE.S4).AND.(T.LT.S5)) GO TO 42                SY 219
            IF ((T.GE.S5).AND.(T.LT.S6)) GO TO 43                SY 220
            IF ((T.GE.S6).AND.(T.LT.S7)) GO TO 44                SY 221
   41       X=SCALE(IS)/2.                                       SY 222
            Y=X*TAN(T/RAD)                                       SY 223
            RETURN                                               SY 224
   42       IF (T.LT.90.) X=SCALE(IS)*.6/(TAN(T/RAD)+1.)         SY 225
            IF (T.EQ.90.) X=0.0                                  SY 226
            IF (T.GT.90.) X=SCALE(IS)*.6/(TAN(T/RAD)-1.)         SY 227
            IF (T.EQ.90.) Y=SCALE(IS)*.6                         SY 228
            IF (T.NE.90.) Y=X*TAN(T/RAD)                         SY 229
            RETURN                                               SY 230
   43       X=-SCALE(IS)/2.                                      SY 231
            Y=X*TAN(T/RAD)                                       SY 232
            RETURN                                               SY 233
   44       Y=-SCALE(IS)*.4                                      SY 234
            IF (T.EQ.270.) X=0.0                                 SY 235
            IF (T.NE.270.) X=Y/TAN(T/RAD)                        SY 236
            RETURN                                               SY 237
   C                                                             SY 238
   C           SYMBOL NUMBER 21 OR 22                            SY 239
   C                                                             SY 240
```

CARD NO.

```
241          45    X=.1*SCALE(IS)*COS(T/RAD)                                SY 241
                   Y=.1*SCALE(IS)*SIN(T/RAD)                                SY 242
                   RETURN                                                   SY 243
                   END                                                      SY 244-
```

CARD NO.

```
      SUBROUTINE CUBSPL (X,Y,N,YPP,TENSION,TENS,A)
C
C     THIS SUBROUTINE FITS A CUBIC SPLINE TO A SET OF Y VS X INPUT
C     POINTS
C
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983
C
C        IN CALLING PROGRAM DIMENSION X, Y, YPP, AND A BY N
C
      DIMENSION X(N), Y(N), YPP(N), A(N)
C
C     COMPUTE TENSION PARAMETER
C
      TENS=TENSION*FLOAT(N-1)/(X(N)-X(1))
      IF (TENS.LT.0.0) TENS=0.0
C
C     COMPUTE SECOND DERIVATIVE AT END POINTS BY FITTING
C     Y=A*X**2+B*X+C TO THE LAST THREE POINTS AND SOLVE FOR A.
C     SECOND DERIVATIVE AT END POINT IS THEN EQUAL TO 2.*A
C
      H1=X(2)-X(3)
      H2=X(3)-X(1)
      H3=X(1)-X(2)
      YPP(1)=2.*(Y(1)*H1+Y(2)*H2+Y(3)*H3)/(H1*X(1)**2+H2*X(2)**2+H3*X(3)
     1**2)
      H1=X(N-1)-X(N)
      H2=X(N)-X(N-2)
      H3=X(N-2)-X(N-1)
      YPP(N)=2.*(Y(N-2)*H1+Y(N-1)*H2+Y(N)*H3)/(H1*X(N-2)**2+H2*X(N-1)**2
     1+H3*X(N)**2)
C
C     PERFORM FORWARD ELIMINATION
C
      N1=N-1
      H1=X(2)-X(1)
      IF (TENS.NE.0.0) GO TO 1
      CON=6.
      E=H1
      GO TO 2
    1 CON=TENS*TENS
```

CARD NO.

```
            H3=TENS/SINH(TENS*H1)                              CB  41
            CTH=H3*COSH(TENS*H1)                               CB  42
            E=1./H1-H3                                         CB  43
            DP=CTH-1./H1                                       CB  44
      2     A(1)=0.0                                           CB  45
            DO 5 I=2,N1                                        CB  46
            H2=X(I+1)-X(I)                                     CB  47
            C=CON*((Y(I+1)-Y(I))/H2-(Y(I)-Y(I-1))/H1)          CB  48
            IF (TENS.NE.0.0) GO TO 3                           CB  49
            F=H2                                               CB  50
            D=H1*(2.-A(I-1))+2.*H2                             CB  51
            GO TO 4                                            CB  52
      3     H3=TENS/SINH(TENS*H2)                              CB  53
            CTH=H3*COSH(TENS*H2)                               CB  54
            F=1./H2-H3                                         CB  55
            DN=CTH-1./H2                                       CB  56
            D=DN+DP-E*A(I-1)                                   CB  57
            DP=DN                                              CB  58
      4     A(I)=F/D                                           CB  59
            YPP(I)=(C-E*YPP(I-1))/D                            CB  60
            E=F                                                CB  61
      5     H1=H2                                              CB  62
      C                                                        CB  63
      C     PERFORM BACK SUBSTITUTION                          CB  64
      C                                                        CB  65
            J=N                                                CB  66
            DO 6 I=2,N1                                        CB  67
            J=J-1                                              CB  68
      6     YPP(J)=YPP(J)-A(J)*YPP(J+1)                        CB  69
      C                                                        CB  70
      C     RETURN TO CALLING PROGRAM                          CB  71
      C                                                        CB  72
            RETURN                                             CB  73
            END                                                CB  74-
```

CARD NO.

```
      FUNCTION FUNC (X,X1,X2,Y1,Y2,YPP1,YPP2,TENS)
C
C     ROUTINE TO COMPUTE Y-VALUE AT A GIVEN X-VALUE ALONG SPLINE CURVE
C
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983
C
      DX1=X2-X
      DX2=X-X1
      DH=X2-X1
      IF (TENS.EQ.0.0) GO TO 1
      F1=YPP1/(TENS*TENS)
      F2=YPP2/(TENS*TENS)
      FUNC=(F1*SINH(TENS*DX1)+F2*SINH(TENS*DX2))/SINH(TENS*DH)+((Y1-F1)*
     1DX1+(Y2-F2)*DX2)/DH
      RETURN
    1 FUNC=YPP1*DX1**3/(6.*DH)+YPP2*DX2**3/(6.*DH)+(Y1-YPP1*DH*DH/6.)*DX
     11/DH+(Y2-YPP2*DH*DH/6.)*DX2/DH
      RETURN
      END
```

CARD NO.

```
      SUBROUTINE DASHLN (T,X,Y,N,NSYM,IOP,ISIZE,IRUN,TENSION)           DH   1
C                                                                       DH   2
C     ROUTINE TO FAIR DATA WITH DASH LINES                              DH   3
C                                                                       DH   4
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983   DH   5
C                                                                       DH   6
C     PARAMETER DEFINITION                                              DH   7
C        T - ARRAY CONTAINING INDEPENDENT VARIABLE                      DH   8
C        X - ARRAY CONTAINING X VARIABLE                                DH   9
C        Y - ARRAY CONTAINING Y VARIABLE                                DH  10
C        N - NUMBER OF POINTS TO BE PLOTTED                             DH  11
C        NSYM - LINE OR SYMBOL NUMBER                                   DH  12
C        IOP - PLOTTING OPTION  IOP=0 PLOT SYMBOLS ONLY                 DH  13
C                               IOP=1 PLOT DASH LINE                    DH  14
C        ISIZE - SYMBOL SIZE                                            DH  15
C        IRUN - RUN NUMBER                                              DH  16
C        TENSION - SPLINE TENSION                                       DH  17
C                                                                       DH  18
C        T, X, AND Y MUST BE PROPERLY DIMENSIONED IN CALLING ROUTINE    DH  19
      DIMENSION T(1), X(1), Y(1), NL(22), NS(22)                        DH  20
C        WORK ARRAYS                                                    DH  21
      COMMON /WORK/ S(166),XI(166),YI(166)                              DH  22
      COMMON /PLT/ MX(50),MY(50),A(50)                                  DH  23
C                                                                       DH  24
      REAL MX,MY                                                        DH  25
C        ROUTINE CONSTANTS                                              DH  26
C        NL = NUMBER OF LONG DASHES AND NS = NUMBER OF SHORT DASHES     DH  27
      DATA NL/1,1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,5/              DH  28
      DATA NS/0,1,1,2,3,1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4,5/              DH  29
C                                                                       DH  30
C     ROUTINE CONSTANTS                                                 DH  31
C                                                                       DH  32
      NP=166                                                            DH  33
      EPS=.02                                                           DH  34
      SL=.3                                                             DH  35
      IF (NSYM.EQ.2) SL=.2                                              DH  36
      SS=.1                                                             DH  37
      SP=.1                                                             DH  38
C                                                                       DH  39
C     SELECT PLOTTING OPTION                                            DH  40
```

CARD NO.

```
      C
            IF (IOP.EQ.0.OR.N.LT.3) GO TO 1
            GO TO 3
      C
      C     PLOT SYMBOLS ONLY
      C
      1     DO 2 I=1,N
      2     CALL PNTPLT (X(I),Y(I),NSYM,ISIZE)
            RETURN
      C
      C     PLOT DASH LINE
      C
      C        CHECK TO SEE IF T IS STRICTLY INCREASING
      3     DO 4 I=2,N
            IF (T(I).LT.T(I-1)) GO TO 5
      4     CONTINUE
            GO TO 6
      C        PRINT ERROR MESSAGE IF T IS NOT STRICTLY INCREASING
      5     PRINT 15, IRUN
            PRINT 16, (T(I),X(I),Y(I),I=1,N)
      C        IF T IS NOT STRICTLY INCREASING, PLOT SYMBOLS ONLY
            GO TO 1
      C        FIT SPLINE CURVE THROUGH DATA POINTS
      6     CALL CUBSPL (T,X,N,MX,TENSION,TENS,A)
            CALL CUBSPL (T,Y,N,MY,TENSION,TENS,A)
      C        COMPUTE NP EQUALLY SPACED POINTS ALONG CURVE
            IF (NSYM.EQ.1) NP=NP-2
            DT=(T(N)-T(1))/FLOAT(NP-1)
            TI=T(1)
            S(1)=0.
            M=2
            DO 9 I=1,NP
            IF (TI.LT.T(1)) TI=T(1)
            IF (TI.GT.T(N)) TI=T(N)
            DO 7 J=M,N
            K=J-1
            IF (TI.GE.T(K).AND.TI.LE.T(J)) GO TO 8
      7     CONTINUE
      8     M=K
            IF (M.LE.1) M=2
```

CARD NO.

```
      XI(I)=FUNC(TI,T(K),T(K+1),X(K),X(K+1),MX(K),MX(K+1),TENS)         DH  81
      YI(I)=FUNC(TI,T(K),T(K+1),Y(K),Y(K+1),MY(K),MY(K+1),TENS)         DH  82
      IF (I.EQ.1.OR.NSYM.EQ.1) GO TO 9                                  DH  83
      S(I)=S(I-1)+SQRT((XI(I)-XI(I-1))**2+(YI(I)-YI(I-1))**2)           DH  84
    9 TI=TI+DT                                                          DH  85
      ST=S(NP)                                                          DH  86
C        IF NSYM=1, PLOT CONTINUOUS LINE                                DH  87
      IF (NSYM.EQ.1) GO TO 13                                           DH  88
C        COMPUTE LENGTH OF LONG AND SHORT DASH AND SPACE BETWEEN EACH   DH  89
      NLG=NL(NSYM)                                                      DH  90
      NSG=NS(NSYM)                                                      DH  91
      DT=FLOAT(NLG)*SL+FLOAT(NSG)*SS+FLOAT(NLG+NSG)*SP                  DH  92
      NLT=IFIX((ST+SP)/DT)                                              DH  93
      IF (NLT.LE.0) NLT=1                                               DH  94
      DTN=(ST+SP)/FLOAT(NLT)                                            DH  95
      TI=DTN/DT                                                         DH  96
      SL=SL*TI                                                          DH  97
      SS=SS*TI                                                          DH  98
      SP=SP*TI                                                          DH  99
C        POSITION PEN AT START OF LINE                                  DH 100
      SI=0.                                                             DH 101
      CALL CALPLT (XI(1),YI(1),3)                                       DH 102
      JSTART=1                                                          DH 103
C        PLOT NL LONG DASHES                                            DH 104
   10 DO 11 I=1,NLG                                                     DH 105
      SI=SI+SL                                                          DH 106
      IF (SI+EPS.GE.ST) SI=ST                                           DH 107
      CALL LINEAR (SI,XX,YY,NP,S,XI,YI,JSTART)                          DH 108
      CALL CALPLT (XX,YY,2)                                             DH 109
      SI=SI+SP                                                          DH 110
      IF (SI.GE.ST) GO TO 14                                            DH 111
      CALL LINEAR (SI,XX,YY,NP,S,XI,YI,JSTART)                          DH 112
      CALL CALPLT (XX,YY,3)                                             DH 113
   11 CONTINUE                                                          DH 114
C        PLOT NS SHORT DASHES                                           DH 115
      IF (NSG.EQ.0) GO TO 10                                            DH 116
      DO 12 I=1,NSG                                                     DH 117
      SI=SI+SS                                                          DH 118
      IF (SI+EPS.GE.ST) SI=ST                                           DH 119
      CALL LINEAR (SI,XX,YY,NP,S,XI,YI,JSTART)                          DH 120
```

CARD NO.

```
            CALL CALPLT (XX,YY,2)                                            DH 121
            SI=SI+SP                                                         DH 122
            IF (SI.GE.ST) GO TO 14                                           DH 123
            CALL LINEAR (SI,XX,YY,NP,S,XI,YI,JSTART)                         DH 124
            CALL CALPLT (XX,YY,3)                                            DH 125
      12    CONTINUE                                                         DH 126
      C        CONTINUE PLOTTING DASH LINE                                   DH 127
            GO TO 10                                                         DH 128
      C        PLOT CONTINUOUS LINE                                          DH 129
      13    XI(NP+1)=0.0                                                     DH 130
            YI(NP+1)=0.0                                                     DH 131
            XI(NP+2)=YI(NP+2)=1.0                                            DH 132
            CALL LINE (XI,YI,NP,1,0,0,0.)                                    DH 133
      C                                                                      DH 134
      C     RETURN TO START OF CURVE                                         DH 135
      C                                                                      DH 136
      14    CALL CALPLT (XI(1),YI(1),3)                                      DH 137
      C                                                                      DH 138
      C     RETURN TO CALLING PROGRAM                                        DH 139
      C                                                                      DH 140
            RETURN                                                           DH 141
      C                                                                      DH 142
      15    FORMAT (1H1//5X,36HT IS NOT STRICTLY INCREASING FOR RUN,I7/16X,1HT DH 143
           1,13X,1HX,13X,1HY/)                                               DH 144
      16    FORMAT (5X,3F15.4)                                               DH 145
            END                                                              DH 146-
```

CARD NO.

```
      SUBROUTINE LINEAR (TINT,XINT,YINT,N,T,X,Y,JSTART)
C
C     LINEAR INTERPOLATION METHOD
C
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983
C
      DIMENSION T(1), X(1), Y(1)
      J=1
      IF (TINT.LE.T(1)) GO TO 3
      J=N
      IF (TINT.GE.T(N)) GO TO 3
      JEND=JSTART
      DO 1 I=JSTART,N
      J=I
      IF (T(J)-TINT) 1,3,2
1     JEND=J
2     DT=(TINT-T(J-1))/(T(J)-T(J-1))
      XINT=X(J-1)+(X(J)-X(J-1))*DT
      YINT=Y(J-1)+(Y(J)-Y(J-1))*DT
      JSTART=JEND
      RETURN
3     YINT=Y(J)
      XINT=X(J)
      JSTART=J
      RETURN
      END
```

CARD NO.

```
      SUBROUTINE LSQPLT (T,X,Y,N,NSYM,ISIZE,IOP,IRUN,TENSION,ILSQ,NPOL,D
     1F)
C
C     ROUTINE TO FAIR DATA WITH LEAST SQUARES CURVE FIT
C
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983
C
C     PARAMETER DEFINITION
C        T - ARRAY CONTAINING INDEPENDENT VARIABLE
C        X - ARRAY CONTAINING X VARIABLE
C        Y - ARRAY CONTAINING Y VARIABLE
C        N - NUMBER OF POINTS TO BE PLOTTED
C        NSYM - LINE AND SYMBOL NUMBER
C        ISIZE - SYMBOL SIZE
C        IOP - PLOTTING OPTION  IOP=0 PLOT SYMBOLS ONLY
C                               IOP=1 PLOT SYMBOLS AND CURVE
C        IRUN - RUN NUMBER
C        TENSION - SPLINE TENSION
C        ILSQ - LEAST SQUARES CURVE FIT OPTION
C              ILSQ=0 LEAST SQUARES POLYNOMIAL CURVE FIT
C              ILSQ=1 LEAST SQUARES CUBIC SPLINE FIT
C        NPOL - ORDER OF LEAST SQUARES POLYNOMIAL CURVE FIT
C        DF - STANDARD DEVIATION FOR LEAST-SQUARES CUBIC-SPLINE FIT
C     DIMENSION T, X, AND Y IN CALLING PROGRAM
      DIMENSION T(1), X(1), Y(1)
C     WORK ARRAYS
      DIMENSION XNEW(50), YNEW(50)
      DIMENSION COEF(50,4)
      COMMON /WORK/ WK(360),SDV(50),CX(11),CY(11)
C
C     ROUTINE CONSTANTS
C
      NMAX=50
      WT=1.0
      IPRINT=0
C
C     SELECT PLOTTING OPTION
C
      IF (IOP.EQ.0.OR.N.LT.3) GO TO 1
      GO TO 3
```

CARD NO.

```
      C
      C     PLOT SYMBOLS ONLY
      C
      1     DO 2 I=1,N
      2     CALL PNTPLT (X(I),Y(I),NSYM,ISIZE)
            RETURN
      C
      C     CHECK TO SEE IF T IS STRICTLY INCREASING
      C
      3     DO 4 I=2,N
            IF (T(I).LT.T(I-1)) GO TO 5
      4     CONTINUE
            GO TO 6
      5     PRINT 20, IRUN
            PRINT 21, (T(I),X(I),Y(I),I=1,N)
            GO TO 1
      C
      C     PLOT DATA POINTS WITH SYMBOLS
      C
      6     DO 7 I=1,N
            SDV(I)=WT
            IF (ILSQ.EQ.1) SDV(I)=DF
            CALL PNTPLT (X(I),Y(I),NSYM,ISIZE)
      7     CONTINUE
      C
      C     COMPUTE LEAST SQUARES CURVE
      C
            IF (ILSQ.GT.0) GO TO 10
      C        COMPUTE COEFFICIENTS OF LEAST SQUARES POLYNOMIAL
            MPOL=NPOL
            IF (MPOL.GT.N-1) MPOL=N-1
            IF (MPOL.LE.0) MPOL=1
            IF (MPOL.GT.10) MPOL=10
            M1=MPOL+1
            CALL LSQ (T,X,SDV,N,MPOL,CX)
            CALL LSQ (T,Y,SDV,N,MPOL,CY)
      C        COMPUTE NEW X AND Y VALUES AND ERRCR SUMMATIONS
            ERRX=ERRY=0.0
            DO 9 I=1,N
            XNEW(I)=CX(1)
```

```
              YNEW(I)=CY(1)
              DELTA=1.0
              DO 8 J=2,M1
              DELTA=DELTA*T(I)
              XNEW(I)=XNEW(I)+DELTA*CX(J)
    8         YNEW(I)=YNEW(I)+DELTA*CY(J)
              ERRX=ERRX+(X(I)-XNEW(I))**2
              ERRY=ERRY+(Y(I)-YNEW(I))**2
    9         CONTINUE
              GO TO 17
    C            COMPUTE COEFFICIENTS OF LEAST SQUARES CUBIC SPLINE
    C                  FIT X
    10        K=-1
              SM=FLOAT(N)
              CALL CSDS (NMAX,N,T,X,SDV,SM,K,COEF,WK,IERR)
              IF (IERR.NE.0) GO TO 19
    C                  COMPUTE NEW X AND ERROR SUMMATION
              ERRX=0.
              DO 13 I=1,N
              IF (I.EQ.N) GO TO 11
              XNEW(I)=COEF(I,1)
              GO TO 12
    11        K=I-1
              DELTA=T(I)-T(K)
              XNEW(I)=((COEF(K,4)*DELTA+COEF(K,3))*DELTA+COEF(K,2))*DELTA+COEF(K
             1,1)
    12        ERRX=ERRX+(X(I)-XNEW(I))**2
    13        CONTINUE
    C                  FIT Y
              K=-1
              SM=FLOAT(N)
              CALL CSDS (NMAX,N,T,Y,SDV,SM,K,COEF,WK,IERR)
              IF (IERR.NE.0) GO TO 19
    C                  COMPUTE NEW Y AND ERROR SUMMATION
              ERRY=0.
              DO 16 I=1,N
              IF (I.EQ.N) GO TO 14
              YNEW(I)=COEF(I,1)
              GO TO 15
    14        K=I-1
```

CARD NO.

```
            DELTA=T(I)-T(K)
            YNEW(I)=((COEF(K,4)*DELTA+COEF(K,3))*DELTA+COEF(K,2))*DELTA+COEF(K
           1,1)
   15       ERRY=ERRY+(Y(I)-YNEW(I))**2
   16       CONTINUE
C
C           PLOT DASHED LINE THRU NEW X AND Y VALUES
C
   17       TEN=TENSION
            IF (ILSQ.EQ.1) TEN=0.0
            CALL DASHLN (T,XNEW,YNEW,N,NSYM,IOP,ISIZE,IRUN,TEN)
C
C           PRINT RESULTS OF LEAST SQUARES CURVE FIT
C
            IF (IPRINT.EQ.0) RETURN
            IF (ILSQ.EQ.0) PRINT 22, IRUN,NSYM,MPOL
            IF (ILSQ.EQ.1) PRINT 23, IRUN,NSYM,DF
            DO 18 I=1,N
            DELTA=X(I)-XNEW(I)
            SM=Y(I)-YNEW(I)
            PRINT 24, I,T(I),X(I),XNEW(I),DELTA,Y(I),YNEW(I),SM
   18       CONTINUE
            PRINT 25, ERRY,ERRY
            RETURN
C
C           PRINT ERROR MESSAGE
C
   19       PRINT 26, IERR,IRUN,NSYM
            GO TO 1
C
   20       FORMAT (1H1//5X,36HT IS NOT STRICTLY INCREASING FOR RUN,I7/16X,1HT
           1,13X,1HX,13X,1HY/)
   21       FORMAT (5X,3F15.4)
   22       FORMAT (//5X,55HRESULTS OF LEAST SQUARES POLYNOMIAL CURVE FIT FOR
           1RUN =,I4,3X,6HNSYM =,I2,3X,6HNPOL =,I3//4X,1HI,6X,1HT,9X,1HX,9X,4H
           2XNEW,4X,6HX-XNEW,6X,1HY,9X,4HYNEW,4X,6HY-YNEW)
   23       FORMAT (//5X,57HRESULTS OF LEAST SQUARES CUBIC SPLINE CURVE FIT FO
           1R RUN =,I4,3X,6HNSYM =,I2,3X,4HDF =,F10.5//4X,1HI,6X,1HT,9X,1HX,9X
           2,4HXNEW,4X,6HX-XNEW,6X,1HY,9X,4HYNEW,4X,6HY-YNEW)
   24       FORMAT (I5,7F10.4)
```

CARD NO.

```
     25   FORMAT (/10X,12HERROR IN X =,E15.6,10X,12HERROR IN Y =,E15.6)        LS 161
     26   FORMAT (/5X,13HERROR NUMBER ,I3,42H OCCURRED IN CSDS CALL IN LSQPL LS 162
         1T FOR RUN =,I4,11H AND NSYM =,I3)                                    LS 163
          END                                                                  LS 164-
```

CARD NO.

```
      SUBROUTINE LSQ (X,Y,W,NP,N,C)
C
C     LEAST SQUARES WEIGHTED POLYNOMIAL CURVE FIT ROUTINE
C
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983
C
C     PARAMETER DEFINITION
C        X - ARRAY CONTAINING INDEPENDENT VARIABLE
C        Y - ARRAY CONTAINING DEPENDENT VARIABLE
C        W - ARRAY CONTAINING WEIGHTING VALUES
C        NP - NUMBER OF POINTS
C        N - ORDER OF POLYNOMIAL
C        C - COEFFICIENTS OF POLYNOMIAL
C
C     NEW Y VALUE = C(1)+C(2)*X+C(3)*X**2+...+C(N+1)*X**N
C
C        DIMENSION X(NP), Y(NP), W(NP), AND C(N+1) IN CALLING PROGRAM
      DIMENSION X(1), Y(1), W(1), C(1)
C        DIMENSION A(N+1,N+2)   N=10 FOR THIS VERSION
      COMMON /WORK/ A(11,12)
C
C     COMPUTE LEAST SQUARES MATRIX
C
      N1=N+1
      N2=N+2
      DO 1 I=1,N1
      DO 1 J=1,N2
    1 A(I,J)=0.
      DO 3 K=1,NP
      T1=1.
      DO 3 J=1,N1
      T2=T1
      DO 2 I=1,N1
      A(J,I)=A(J,I)+T2*W(K)
    2 T2=T2*X(K)
      A(J,N2)=A(J,N2)-Y(K)*T1*W(K)
    3 T1=T1*X(K)
C
C     SOLVE FOR COEFFICIENTS OF LEAST SQUARES POLYNOMIAL
C
```

CARD NO.

```
            DO 4 K=1,N                              LQ  41
            DO 4 J=K,N                              LQ  42
            T1=A(J+1,K)/A(K,K)                      LQ  43
            DO 4 I=K,N2                             LQ  44
      4     A(J+1,I)=A(J+1,I)-A(K,I)*T1             LQ  45
            C(N1)=-A(N1,N2)/A(N1,N1)                LQ  46
            DO 5 I=2,N1                             LQ  47
            K=N2-I                                  LQ  48
            C(K)=-A(K,N2)/A(K,K)                    LQ  49
            L=K+1                                   LQ  50
            DO 5 J=L,N1                             LQ  51
      5     C(K)=C(K)-C(J)*A(K,J)/A(K,K)            LQ  52
            RETURN                                  LQ  53
            END                                     LQ  54-
```

CARD NO.

```
      SUBROUTINE CSDS (MAX,IX,X,F,DF,S,IPT,COEF,WK,IERR)                CS   1
C************************************************************************ CS   2
C*                                                                     * CS   3
C*    PURPOSE:                                                         * CS   4
C*                 SUBROUTINE CSDS FITS A SMOOTH CUBIC SPLINE TO A     * CS   5
C*                 UNIVARIATE FUNCTION.  DATA MAY BE UNEQUALLY SPACED. * CS   6
C*                                                                     * CS   7
C*       USE:                                                          * CS   8
C*                 CALL CSDS(MAX,IX,X,F,DF,S,IPT,COEF,WK,IERR)         * CS   9
C*                                                                     * CS  10
C*         MAX     INPUT INTEGER SPECIFYING THE MAXIMUM NUMBER OF DATA * CS  11
C*                 POINTS FOR THE INDEPENDENT VARIABLE.                * CS  12
C*                                                                     * CS  13
C*          IX     INPUT INTEGER SPECIFYING THE ACTUAL NUMBER OF DATA  * CS  14
C*                 POINTS FOR THE INDEPENDENT VARIABLE.  IX∂MAX.       * CS  15
C*                                                                     * CS  16
C*           X     ONE-DIMENSIONAL INPUT ARRAY DIMENSIONED AT LEAST    * CS  17
C*                 IX IN THE CALLING PROGRAM.  UPON ENTRY TO CSDS,     * CS  18
C*                 X(I) MUST CONTAIN THE VALUE OF THE INDEPENDENT      * CS  19
C*                 VARIABLE AT POINT I.                                * CS  20
C*                                                                     * CS  21
C*           F     ONE-DIMENSIONAL INPUT ARRAY DIMENSIONED AT LEAST    * CS  22
C*                 IX IN THE CALLING PROGRAM.  UPON ENTRY TO CSDS,     * CS  23
C*                 F(I) MUST CONTAIN THE VALUE OF THE FUNCTION AT      * CS  24
C*                 POINT X(I).                                         * CS  25
C*                                                                     * CS  26
C*          DF     ONE-DIMENSIONAL INPUT ARRAY DIMENSIONED AT LEAST    * CS  27
C*                 IX IN THE CALLING PROGRAM.  UPON ENTRY TO CSDS,     * CS  28
C*                 DF(I) MUST CONTAIN AN ESTIMATE OF THE STANDARD      * CS  29
C*                 DEVIATION OF F(I).                                  * CS  30
C*                                                                     * CS  31
C*           S     A NON-NEGATIVE INPUT PARAMETER WHICH CONTROLS THE   * CS  32
C*                 EXTENT OF SMOOTHING.  S SHOULD BE IN THE RANGE      * CS  33
C*                 (IX-(2*IX)**.5)<S<(IX+(2*IX)**.5).                  * CS  34
C*                                                                     * CS  35
C*         IPT     INPUT INITIALIZATION PARAMETER.  THE USER MUST      * CS  36
C*                 SPECIFY IPT=-1 WHENEVER A NEW X ARRAY IS            * CS  37
C*                 INPUT.  THE ROUTINE WILL ALSO CHECK TO INSURE THAT  * CS  38
C*                 THE X ARRAY IS IN STRICTLY INCREASING ORDER.        * CS  39
C*                                                                     * CS  40
```

CARD NO.

```
C*        COEF     A TWO-DIMENSIONAL OUTPUT ARRAY DIMENSIONED (MAX,4)  * CS  41
C*                 IN THE CALLING PROGRAM.  UPON RETURN, COEF(I,J)     * CS  42
C*                 CONTAINS THE J-TH COEFFICIENT OF THE SPLINE FOR     * CS  43
C*                 THE INTERVAL BEGINNING AT POINT X(I).  THE          * CS  44
C*                 FUNCTIONAL VALUE OF THE SPLINE AT ABSCISSA X1,      * CS  45
C*                 WHERE X(I)<X1<X(I+1), IS GIVEN BY:                  * CS  46
C*                    F(X1)=((COEF(I,4)*H+COEF(I,3))*H+COEF(I,2))*H    * CS  47
C*                    +COEF(I,1)                                       * CS  48
C*                 WHERE H=X1-X(I)                                     * CS  49
C*                                                                     * CS  50
C*          WK     A ONE-DIMENSIONAL WORK AREA ARRAY DIMENSIONED AT    * CS  51
C*                 LEAST (7*IX+9) IN THE CALLING PROGRAM.              * CS  52
C*                                                                     * CS  53
C*        IERR     OUTPUT ERROR PARAMETER:                             * CS  54
C*                 =0   NORMAL RETURN.  NO ERROR DETECTED.             * CS  55
C*                 =J   THE J-TH ELEMENT OF THE X ARRAY IS NOT IN      * CS  56
C*                      STRICTLY INCREASING ORDER.                     * CS  57
C*                 =-1  THERE ARE LESS THAN FOUR VALUES IN THE X ARRAY.* CS  58
C*                                                                     * CS  59
C*                 UPON RETURN FROM CSDS, THIS PARAMETER SHOULD BE     * CS  60
C*                 TESTED IN THE CALLING PROGRAM.                      * CS  61
C*                                                                     * CS  62
C*                                                                     * CS  63
C*                                                                     * CS  64
C*    REQUIRED ROUTINES                    -NONE                       * CS  65
C*                                                                     * CS  66
C*    SOURCE                               IMSL ROUTINE ICSSMU MODIFIED BY * CS  67
C*                                         COMPUTER SCIENCES CORPORATION  * CS  68
C*                                                                     * CS  69
C*    LANGUAGE                             -FORTRAN                    * CS  70
C*                                                                     * CS  71
C*    DATE RELEASED                        SEPTEMBER 5, 1973           * CS  72
C*                                                                     * CS  73
C*    LATEST REVISION                      MARCH 1975                  * CS  74
C********************************************************************** CS  75
C                                                                        CS  76
C                                                                        CS  77
      DIMENSION X(1), F(1), DF(1), COEF(MAX,4), WK(1)                    CS  78
C                                                                        CS  79
C                                 SET UP WORKING AREAS                   CS  80
```

CARD NO.

```
C
      IERR=0
      IF (IPT.NE.-1) GO TO 4
      IPT=0
      IF (IX.LT.4) GO TO 1
      GO TO 2
    1 IERR=-1
      RETURN
    2 IX1=IX-1
      DO 3 I=1,IX1
      IF (X(I+1)-X(I).GT.0) GO TO 3
      IERR=I+1
      RETURN
    3 CONTINUE
      NP1=IX+1
      IB1=NP1
      IB2=IB1+NP1
      IB3=IB2+NP1+1
      IB4=IB3+NP1
      IB5=IB4+NP1
      IB6=IB5+NP1+1
      WK(1)=0.
      WK(2)=0.
      WK(IB2)=0.
      WK(IB3)=0.
      IJK2=IB2+NP1
      WK(IJK2)=0.
      IJK5=IB5+1
      WK(IJK5)=0.
      IJK5=IB5+2
      WK(IJK5)=0.
      WK(IB6)=0.
      IJK5=IB5+NP1
      WK(IJK5)=0.
    4 CONTINUE
      P=0.
      H=X(2)-X(1)
      F2=-S
      FF=(F(2)-F(1))/H
      IF (IX.LT.3) GO TO 10
```

```
            DO 5 I=3,IX
            G=H
            H=X(I)-X(I-1)
            E=FF
            FF=(F(I)-F(I-1))/H
            COFF(I-1,1)=FF-E
            IJK3=IB3+I
            WK(IJK3)=(G+H)*.66666666666667
            IJK4=IB4+I
            WK(IJK4)=H/3.
            IJK2=IB2+I
            WK(IJK2)=DF(I-2)/G
            WK(I)=DF(I)/H
            IJK1=IB1+I
            WK(IJK1)=-DF(I-1)/G-DF(I-1)/H
    5       CONTINUE
            DO 6 I=3,IX
            IJK1=IB1+I
            IJK2=IB2+I
            COEF(I-1,2)=WK(I)*WK(I)+WK(IJK1)*WK(IJK1)+WK(IJK2)*WK(IJK2)
            COEF(I-1,3)=WK(I)*WK(IJK1+1)+WK(IJK1)*WK(IJK2+1)
            COEF(I-1,4)=WK(I)*WK(IJK2+2)
    6       CONTINUE
    C
    C                               NEXT ITERATION
    C
    7       IF (IX.LT.3) GO TO 10
            DO 8 I=3,IX
            IJK1=IB1+I-1
            IJKO=I-1
            WK(IJK1)=FF*WK(IJKO)
            IJK2=IB2+I-2
            IJKO=I-2
            WK(IJK2)=G*WK(IJKO)
            IJKO=I
            IJK3=IB3+I
            WK(IJKO)=1./(P*COFF(I-1,2)+WK(IJK3)-FF*WK(IJK1)-G*WK(IJK2))
            IJK5=IB5+I
            IJKN=IJK5-1
            IJKO=IJKN-1
```

CARD NO.

```
            WK(IJK5)=COEF(I-1,1)-WK(IJK1)*WK(IJKN)-WK(IJK2)*WK(IJK0)
            IJK4=IB4+I
            FF=P*COEF(I-1,3)+WK(IJK4)-H*WK(IJK1)
            G=H
            H=COEF(I-1,4)*P
      8     CONTINUE
            DO 9 I=3,IX
            J=IX-I+3
            IJK5=IB5+J
            IJK6=IJK5+1
            IJK7=IJK6+1
            IJK1=IB1+J
            IJK2=IB2+J
            WK(IJK5)=WK(J)*WK(IJK5)-WK(IJK1)*WK(IJK6)-WK(IJK2)*WK(IJK7)
      9     CONTINUE
      10    E=0
            H=0
      C
      C                                      COMPUTE U AND ACCUMULATE E
      C
            DO 11 I=2,IX
            G=H
            IJK5=IB5+I
            H=(WK(IJK5+1)-WK(IJK5))/(X(I)-X(I-1))
            IJK6=IB6+I
            WK(IJK6)=(H-G)*DF(I-1)*DF(I-1)
            E=E+WK(IJK6)*(H-G)
      11    CONTINUE
            G=-H*DF(IX)*DF(IX)
            IJK6=IB6+NP1
            WK(IJK6)=G
            E=E-G*H
            G=F2
            F2=E*P*P
            IF (F2.GE.S.OR.F2.LF.G) GO TO 14
            FF=0.
            IJK6=IB6+2
            H=(WK(IJK6+1)-WK(IJK6))/(X(2)-X(1))
            IF (IX.LT.3) GO TO 13
            DO 12 I=3,IX
```

CARD NO.

```
            G=H
            IJK6=IB6+I
            H=(WK(IJK6+1)-WK(IJK6))/(X(I)-X(I-1))
            IJK1=IB1+I-1
            IJK2=IB2+I-2
            G=H-G-WK(IJK1)*WK(I-1)-WK(IJK2)*WK(I-2)
            FF=FF+G*WK(I)*G
            WK(I)=G
   12       CONTINUE
   13       H=E-P*FF
            IF (H.LE.0) GO TO 14
C
C                                  UPDATE THE LAGRANGE MULTIPLIER P
C                                     FOR THE NEXT ITERATION
C
            P=P+(S-F2)/((SQRT(S/E)+P)*H)
            GO TO 7
C
C                                  IF E LESS THAN OR EQUAL TO S,
C                                  COMPUTE THE COEFFICIENTS AND RETURN.
C
   14       DO 15 I=2,NP1
            IJK6=IB6+I
            COEF(I-1,1)=F(I-1)-P*WK(IJK6)
            IJK5=IB5+I
            COEF(I-1,3)=WK(IJK5)
   15       CONTINUE
            DO 16 I=2,IX
            H=X(I)-X(I-1)
            COEF(I-1,4)=(COEF(I,3)-COEF(I-1,3))/(3.*H)
            COEF(I-1,2)=(COEF(I,1)-COEF(I-1,1))/H-(H*COEF(I-1,4)+COEF(I-1,3))*
           1H
   16       CONTINUE
            RETURN
            END
```

CARD NO.

```
              SUBROUTINE AXISLB (IAXIS,XORG,YORG,ORG,SCALE,DIST,HT,NDIG)
        C
        C     ROUTINE TO DRAW SCALE VALUES ON AXIS
        C
        C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983
        C
        C     PARAMETER DEFINITION
        C        IAXIS = 1 - DRAW X-AXIS SCALES
        C        IAXIS = 2 - DRAW Y-AXIS SCALES
        C        XORG,YORG - X,Y COORDINATES OF BEGINNING OF AXIS
        C        ORG - SCALE VALUE AT BEGINNING OF AXIS
        C        SCALE - SCALE VALUE PER INCH
        C        DIST - LENGTH OF AXIS
        C        HT - HEIGHT OF VALUES DRAWN ON AXIS
        C        NDIG - NUMBER OF SIGNIFICANT FIGURES TO RIGHT OF DECIMAL POINT
        C             NDIG=-1 WILL DROP DECIMAL POINT
        C
        C     COMPUTE ROUTINE CONSTANTS
              WADJ=0.0
              IF (HT.LE.0.1) WADJ=0.01
              CONS=6.*HT/7.+WADJ
              CONS2=2.*HT/7.
              ADJ=0.4*HT
              IF (ADJ.LT.0.1) ADJ=0.1
              NL=IFIX(DIST+0.0001)+1
              XO=XORG
              YO=YORG
              VALUE=ORG
        C     POSITION PEN AT BEGINNING OF AXIS
              CALL CALPLT (XORG,YORG,3)
        C     DRAW AXIS VALUE AT EACH INCH MARK ALONG AXIS
              DO 8 I=1,NL
              VALUE=SCALE*FLOAT(I-1)+ORG
        C        COMPUTE NUMBER OF SIGNIFICANT FIGURES IN VALUE
              NDG=NDIG
              V=ABS(VALUE)
              IF (V-1.0E-5) 1,1,2
        1     NDG=-1
              NSIG=1
              V=0.0
```

CARD NO.

```
           VALUE=0.0
           GO TO 5
      2    NSIG=NDG+1
      3    S=V+SIGN(1.E-7,V)
           N=IFIX(S)
           IF (N) 5,5,4
      4    NSIG=NSIG+1
           V=0.10*V
           GO TO 3
      5    IF (VALUE.LT.0.0) NSIG=NSIG+1
           S=CONS*FLOAT(NSIG)-CONS2
      C       POSITION AND DRAW AXIS VALUE
           IF (IAXIS.EQ.2) GO TO 6
           XT=XO-0.5*S
           YT=YO-HT-ADJ
           XO=XO+1.
           GO TO 7
      6    XT=XO-S-ADJ
           YT=YO-0.5*HT
           YO=YO+1.
      7    CALL NUMBER (XT,YT,HT,VALUE,0.,NDG)
      C       ADVANCE TO NEXT INCH MARK
      8    CONTINUE
      C    RETURN TO BEGINNING OF AXIS
           CALL CALPLT (XORG,YORG,3)
      C    RETURN TO CALLING PROGRAM
           RETURN
           END
```

CARD NO.

```
      SUBROUTINE COEFSY (XO,YO,HT,ISYM)                                   CY   1
C                                                                         CY   2
C     ROUTINE TO DRAW AERODYNAMIC COEFFICIENTS                            CY   3
C                                                                         CY   4
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983     CY   5
C                                                                         CY   6
C     PARAMETER DEFINITION                                                CY   7
C        XO - X COORDINATE OF LEFT-HAND EDGE OF SYMBOL                    CY   8
C        YO - Y COORDINATE OF CENTERLINE OF MAIN LETTER OF SYMBOL         CY   9
C        HT - SYMBOL HEIGHT                                               CY  10
C        ISYM = 1 - LIFT                       WIDTH = 2.0*HT             CY  11
C        ISYM = 2 - DRAG                       WIDTH = 2.1*HT             CY  12
C        ISYM = 3 - PITCHING MOMENT            WIDTH = 2.2*HT             CY  13
C        ISYM = 4 - ROLLING MOMENT             WIDTH = 1.7*HT             CY  14
C        ISYM = 5 - YAWING MOMENT              WIDTH = 1.9*HT             CY  15
C        ISYM = 6 - SIDE FORCE                 WIDTH = 2.1*HT             CY  16
C        ISYM = 7 - LIFT/DRAG                  WIDTH = 2.6*HT             CY  17
C        ISYM = 8 - ALPHA,DEG                  WIDTH = 3.5*HT             CY  18
C        ISYM = 9 - BETA,DEG                   WIDTH = 3.4*HT             CY  19
C                                                                         CY  20
C     INITIALIZE CHARACTER SETS                                           CY  21
      CALL CHARST1                                                        CY  22
      CALL CHARST7                                                        CY  23
      CALL CHARST2                                                        CY  24
C     POSITION PEN AT START OF SYMBOL                                     CY  25
      CALL CALPLT (XO,YO,3)                                               CY  26
      Y1=YO-0.5*HT                                                        CY  27
      IF (ISYM.EQ.7) GO TO 1                                              CY  28
      IF (ISYM.EQ.8.OR.ISYM.EQ.9) GO TO 2                                 CY  29
      IF (ISYM.GT.9) GO TO 3                                              CY  30
C     DRAW LETTER C                                                       CY  31
      CALL CHARACT (XO,Y1,HT,2H)C,0.,2,0.0)                               CY  32
      CALL CHARWH (WD,X1,Y1,HT,2H)C,2,0.0)                                CY  33
      X1=XO+1.15*WD                                                       CY  34
      Y1=YO-HT                                                            CY  35
C     DRAW SUBSCRIPT OF C                                                 CY  36
      IF (ISYM.EQ.1) CALL CHARACT (X1,Y1,HT,2H)L,0.,2,0.0)                CY  37
      IF (ISYM.EQ.2) CALL CHARACT (X1,Y1,HT,2H)D,0.,2,0.0)                CY  38
      IF (ISYM.EQ.3) CALL CHARACT (X1,Y1,HT,2H(M,0.,2,0.0)                CY  39
      IF (ISYM.EQ.4) CALL LAM (X1,Y1,HT)                                  CY  40
```

CARD NO.

```
              IF (ISYM.EQ.5) CALL CHARACT (X1,Y1,HT,2H(N,0.,2,0.0)         CY  41
              IF (ISYM.EQ.6) CALL CHARACT (X1,Y1,HT,2H)Y,0.,2,0.0)         CY  42
              GO TO 4                                                      CY  43
      C       DRAW L/D                                                     CY  44
      1       CALL CHARACT (X0,Y1,HT,4H)L/D,0.,4,0.25)                     CY  45
              GO TO 4                                                      CY  46
      C       DRAW ALPHA,DEG OR BETA,DEG                                   CY  47
      2       IF (ISYM.EQ.8) CALL CHARACT (X0,Y1,HT,11H$7(A$2(,DEG,0.,11,0.20)  CY  48
              IF (ISYM.EQ.9) CALL CHARACT (X0,Y1,HT,11H$7(B$2(,DEG,0.,11,0.20)  CY  49
              GO TO 4                                                      CY  50
      C                                                                    CY  51
      C       ADD ADDITIONAL SYMBOLS HERE AS DESIRED                       CY  52
      3       CONTINUE                                                     CY  53
      C                                                                    CY  54
      C                                                                    CY  55
      C       RETURN PEN TO START OF SYMBOL                                CY  56
      4       CALL CALPLT (X0,Y0,3)                                        CY  57
              CALL CHNGSET (1)                                             CY  58
      C       RETURN TO CALLING PROGRAM                                    CY  59
              RETURN                                                       CY  60
              END                                                          CY  61-
```

CARD NO.

```
      SUBROUTINE LAM (XO,YO,HT)                                         LM   1
C                                                                       LM   2
C     ROUTINE TO DRAW ROLLING MOMENT SUBSCRIPT                          LM   3
C                                                                       LM   4
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983   LM   5
C                                                                       LM   6
C     PARAMETER DEFINITION                                              LM   7
C        XO,YO - X,Y COORDINATES OF LOWER LEFT-HAND CORNER OF SYMBOL    LM   8
C        HT - HEIGHT OF SYMBOL                                          LM   9
C                                                                       LM  10
      DIMENSION XM(17), YM(17)                                          LM  11
      DATA XM/3.,13.,0.,0.,4.,10.,13.,14.,13.,12.2,10.,4.,1.,1.,14.4,3.6 LM  12
     1,3./                                                              LM  13
      DATA YM/43.,43.,7.,2.,0.,0.,2.,5.,5.,2.6,1.,1.,2.6,7.,44.,44.,43./ LM  14
      HTH=HT/44.                                                        LM  15
      CALL CALPLT (XO,YO,3)                                             LM  16
      DO 1 I=1,17                                                       LM  17
      K=2                                                               LM  18
      IF (I.EQ.1) K=3                                                   LM  19
      X=XO+(XM(I)+2.)*HTH                                               LM  20
      Y=YO+YM(I)*HTH                                                    LM  21
    1 CALL CALPLT (X,Y,K)                                               LM  22
      CALL CALPLT (XO,YO,3)                                             LM  23
      RETURN                                                            LM  24
      END                                                               LM  25-
```

CARD NO.

```
      SUBROUTINE GRIDLN (XO,YO,XL,YH,NDIV)
C
C     ROUTINE TO DRAW AREA OF GRID LINES
C
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983
C
C     PARAMETER DEFINITION
C        XO,YO - X,Y COORDINATES OF LOWER LEFT-HAND CORNER OF GRID AREA
C        XL - LENGTH OF GRID AREA
C        YH - HEIGHT OF GRID AREA
C        NDIV - NUMBER OF GRID LINES PER INCH
C
C     COMPUTE NUMBER OF GRID LINES
      XD=FLOAT(NDIV)
      IF (XD.LE.0.0) XD=1.
      NGX=IFIX(XD*XL+0.0001)+1
      NGY=IFIX(XD*YH+0.0001)+1
C     DRAW GRID LINES PARALLEL TO Y-AXIS
      CALL CALPLT (XO,YO,3)
      D=0.
      IF (NGX.GT.1) D=XL/FLOAT(NGX-1)
      X1=XO
      Y1=YO
      Y2=YO+YH
      L=0
      DO 3 I=1,NGX
      L=L+1
      X2=X1+D
      IF (L-1) 1,1,2
    1 CALL CALPLT (X1,Y2,2)
      IF (I.EQ.NGX) GO TO 3
      CALL CALPLT (X2,Y2,3)
      GO TO 3
    2 CALL CALPLT (X1,Y1,2)
      IF (I.EQ.NGX) GO TO 3
      CALL CALPLT (X2,Y1,3)
      L=0
    3 X1=X2
C     DRAW GRID LINES PARALLEL TO X-AXIS
      CALL CALPLT (XO,YO,3)
```

CARD NO.

```
      D=0.
      IF (NGY.GT.1) D=YH/FLOAT(NGY-1)
      Y1=YO
      X1=XO
      X2=XO+XL
      L=0
      DO 6 I=1,NGY
      L=L+1
      Y2=Y1+D
      IF (L-1) 4,4,5
    4 CALL CALPLT (X2,Y1,2)
      IF (I.EQ.NGY) GO TO 6
      CALL CALPLT (X2,Y2,3)
      GO TO 6
    5 CALL CALPLT (X1,Y1,2)
      IF (I.EQ.NGY) GO TO 6
      CALL CALPLT (X1,Y2,3)
      L=0
    6 Y1=Y2
C     FRAME GRID AREA
      X1=XO+XL
      Y1=YO+YH
      CALL CALPLT (XO,YO,3)
      CALL CALPLT (X1,YO,2)
      CALL CALPLT (X1,Y1,2)
      CALL CALPLT (XO,Y1,2)
      CALL CALPLT (XO,YO,2)
      CALL CALPLT (XO,YO,3)
C     RETURN TO CALLING PROGRAM
      RETURN
      END
```

CARD NO.

```
      SUBROUTINE RUNKEY (XO,YO,RUN,LSYM,ISIZE,HT,NRNMAX)
C
C     ROUTINE TO DRAW TITLE BLOCK CONTAINING SYMBOLS AND RUN NUMBERS
C
C        CODED BY -- HARRY L. MORGAN        NASA/LARC/TAD/AAB    1983
C
C     PARAMETER DEFINITION
C        XO,YO - X,Y COORDINATES OF LOWER LEFT-HAND CORNER OF KEY
C        RUN - RUN NUMBERS
C        LSYM - SYMBOL ORDER
C        ISIZE - SYMBOL SIZE
C        HT - SYMBOL HEIGHT
C        NRNMAX - MAXIMUM NUMBER OF ALLOWABLE RUNS PER SHEET
C
      DIMENSION RUN(1), LSYM(1)
C        WORK ARRAYS
      COMMON /PLT/ KRUN(10),NRUN(10),LRUN(22),JSYM(22)
C
      INTEGER RUN
C     INITIALIZE ROUTINE CONSTANTS
      DATA XM/1.0/,XMARG/0.1/,XNSP/2.5/,SPC/0.2/
C     DEFINE LETTERING SIZE
      SZ=HT*XM
C     INITIALIZE CHARACTER SET
      CALL CHARST2
C
C     DETERMINE THE NUMBER OF NON-ZERO RUNS AND CORRESPONDING SYMBOLS
C
      NR=0
      DO 2 I=1,NRNMAX
      IF (RUN(I)) 1,2,1
    1 NR=NR+1
      LRUN(NR)=RUN(I)
      JSYM(NR)=LSYM(I)
    2 CONTINUE
      IF (NR.EQ.0) RETURN
C
C     COMPUTE WIDTH OF WORDS SYMBOL AND 99999
C
      CALL CHARWH (SYMW,HA,HBS,SZ,8H)S(YMBOL,8,SPC)
```

CARD NO.

```
      CALL CHARWH (WD,HA,HB,SZ,5H99999,5,SPC)
      XS=XO+0.5*SYMW+XMARG
      XR=XO+2.*SYMW+0.5*WD+XMARG
      YSR=YO+XMARG
C
C     DRAW SYMBOL AND RUN NUMBER
C
      DO 6 L=1,NR
      I=NR+1-L
      ENCODE (10,7,MRUN) LRUN(I)
      DECODE (10,8,MRUN) KRUN
C        FIND NUMBER OF DIGITS IN RUN NUMBER
      DO 3 J=1,10
    3 NRUN(J)=1H
      KDIG=0
      DO 4 J=1,10
      IF (KRUN(J).EQ.1H ) GO TO 4
      KDIG=KDIG+1
C        SHIFT RUN NUMBER FROM RIGHT TO LEFT JUSTIFIED
      NRUN(KDIG)=KRUN(J)
    4 CONTINUE
C        REFORMAT RUN NUMBER
      ENCODE (10,8,MRUN) NRUN
      DECODE (10,9,MRUN) IRUN
C        DRAW SYMBOL
      Y2=YSR+0.5*SZ
      CALL PNTPLT (XS,Y2,JSYM(I),ISIZE)
C        DRAW RUN NUMBER
      IF (KDIG.EQ.0) GO TO 5
      CALL CHARWH (WD,HA,HB,SZ,IRUN,KDIG,SPC)
      XX=XR-0.5*WD
      CALL CHARACT (XX,YSR,SZ,IRUN,0.,KDIG,SPC)
    5 YSR=YSR+XNSP*SZ
    6 CONTINUE
C
C     DRAW AND UNDERLINE WORD SYMBOL AND RUN
C
      CALL CHARWH (WD,HA,HB,SZ,5H)R(UN,5,SPC)
      IF (HB.LT.HBS) HBS=HB
      XX=XS-0.5*(SYMW+XMARG)
```

CARD NO.

```
      CALL CALPLT (XX,YSR,3)                                            KY  81
      XX=XS+0.5*(SYMW+XMARG)                                            KY  82
      CALL CALPLT (XX,YSR,2)                                            KY  83
      XX=XR-0.5*(WD+XMARG)                                              KY  84
      CALL CALPLT (XX,YSR,3)                                            KY  85
      XX=XR+0.5*(WD+XMARG)                                              KY  86
      CALL CALPLT (XX,YSR,2)                                            KY  87
      YSR=YSR-HBS+0.5*XMARG                                             KY  88
      XX=XS-0.5*SYMW                                                    KY  89
      CALL CALPLT (XX,YSR,3)                                            KY  90
      CALL CHARACT (XX,YSR,SZ,8H)S(YMBOL,0.,8,SPC)                      KY  91
      XX=XR-0.5*WD                                                      KY  92
      CALL CHARACT (XX,YSR,SZ,5H)R(UN,0.,5,SPC)                         KY  93
C                                                                       KY  94
C     RETURN TO TITLE BLOCK ORIGIN                                      KY  95
C                                                                       KY  96
      CALL CALPLT (XO,YO,3)                                             KY  97
      CALL CHARST1                                                      KY  98
C                                                                       KY  99
      RETURN                                                            KY 100
C                                                                       KY 101
7     FORMAT (I10)                                                      KY 102
8     FORMAT (10A1)                                                     KY 103
9     FORMAT (A10)                                                      KY 104
      END                                                               KY 105-
```

APPENDIX B

DESCRIPTION OF INPUT DATA FOR WIND-TUNNEL DATA

PLOTTING PROGRAM PLOTWD

This appendix contains a description of the input requirements for the wind-tunnel data plotting program PLOTWD. The input data is divided into two basic parts: (1) the plotting setup information and (2) the plotting sheet information. The setup information defines the data variables, axis scales and positions, and the type of data fairing desired. All setup variables are input as floating point quantities with a format of F10.0, except the YLABEL and XLABEL variables which are axis labels and are input with a format of A10.

Plotting Setup

CARD	VARIABLE	VALUE	DESCRIPTION
1	TEST	-	Test number
2	SHEETW	-	Total width of plotting sheet, in.
	SHEETH	-	Total height of plotting sheet, in.
	SPACE	-	Space between plotting sheets, in.
3	ISYM	-	Starting symbol number (See table I.)
	ISIZE	1	Small symbol size
		2	Medium symbol size
		3	Large symbol size
4	IOP	0	Plot data points only
		1	Plot data points and fair with tension spline
	TENSION	-	Tension factor, $\bar{\sigma}$
5	IEDIT	0	Do not sort or edit data
		1	Sort and edit data
	TOLR	-	Edit tolerance of independent test variable (TOLR=0 for sort only)
6	IW	-	Data array location of independent test variable

CARD	VARIABLE	VALUE	DESCRIPTION
7	NPLOT	-	Number of plots per sheet
8	YTAPE	-	Data array location of y-axis variable
	YOFFSET	-	Sheet height location of y-axis origin, in.
	YSCALE	-	Scale value per inch for y-axis
	YLABEL	-	Label for y-axis
	XTAPE	-	Data array location of x-axis variable
	XOFFSET	-	Sheet width location of x-axis origin, in.
	XSCALE	-	Scale value per inch for x-axis
	XLABEL	-	Label for x-axis

Card 8 is input NPLOT times and NPLOT is limited to a value of 10 for each setup.

Plotting Sheet Namelist

NAMELIST VARIABLE	VALUE	DESCRIPTION
$SHEET	-	Namelist label
NO	-	Sheet number
RUN	-	Run numbers to be plotted on sheet
NEWCASE	0 or default	A new SHEET namelist follows
	1	A new setup deck follows
$END	-	End of namelist

Note that no part of the namelist may be in column 1 and that a maximum of 10 run numbers may be input per sheet.

APPENDIX C

DESCRIPTION OF OUTPUT FOR WIND-TUNNEL DATA PLOTTING PROGRAM PLOTWD

This appendix contains a description of the typical output for the wind-tunnel data plotting program PLOTWD. A sample four-page output is presented in table IV and was generated from the upper set of sample input data presented in table III. Page 1 of the output is a printout of the input setup data as described in appendix B. This page should be carefully checked after each program execution to insure that the desired plotting variables and scales have been properly input. Page 2 of the output is a list of the run numbers contained in the data copied from the input file TAPE1 to the random-access file TAPE2. The total number of data points copied to the random-access file is printed following the list of run numbers. The quantity NPMAX and the dimensioned size of the array NPT in the main program PLOT should be equal to or greater than the total number of data points copied to file TAPE2.

Page 3 of the output is a list of the sheet and corresponding run numbers plotted. A run number with the value zero indicates that no data were plotted with the symbol corresponding to the order of the run number in the list. If the user-specified run number is not available on the random-access file, a message will be printed stating that the specified run could not be found. In addition, if during the plotting of the scaled data any data points fall outside the sheet boundaries, a message will be printed stating the number of data points outside the boundaries. These outside data points will not be plotted and, therefore, may require that the user redefine either the axis scale factors or the sheet height and width.

If the user selects the sort and edit option, page 4 will be output which contains a summary of the sort and edit information for each run called during the program execution. The first set of values listed in this summary are the input values of the independent variable t for the particular run. The sorted t values listed next represent the results of the sorting procedure which simply reorders the input values in a monotonically increasing order. The sorted t values for a particular run will be the same regardless of the number of times the run is plotted. The edited and deleted t values listed last represent the results of the editing procedure which deletes all but one of the multiple data points within the specified tolerance TOLR. The data points deleted are selected based on the input y values for the particular part of the figure and, therefore, may be different for each part. The edited and deleted t values listed are only for the first part of the figure. The deleted points are not used during the spline curve fairing process, but nevertheless, they are plotted with the appropriate symbol.

If the user replaces the call to subroutine CURPLT in the main program PLOT with a call to subroutine LSQPLT, the sample output summary information presented in table V will be listed for the least-squares polynomial curve fit option and in table VI for the least-squares cubic-spline curve fit option. The parameter IPRINT in subroutine LSQPLT must also be set equal to 1 for the summary information to be listed. This information will appear in the output immediately following the print of the particular sheet and corresponding run numbers on page 3 of the basic output. The summary information will be listed each time the subroutine is called; therefore, the user is cautioned that a large amount of printed output can easily be generated even if a moderate number of plots and runs are specified.

The summary information presented in tables V and VI consists of a tabulated listing of the input t, x, and y values, the new x and y values generated by the least-squares curve fit, and the differences between input and new x and y values. Immediately following the tabulated listing, the sum-of-the-squares of the differences between the input and new x and y values are listed as an error-in-x and -y, respectively. If the independent variable t and the x-axis variable are the same as it is for the sample output case, the error-in-x will be very small. If the least-square polynomial curve fit option is chosen, the error-in-x and -y will be a function of the degree of the polynomial. If the least-square cubic-spline option is chosen, the error-in-x and -y will always be equal to the product of the number of input points times the square of the standard deviation DF.

TABLE I.- LIST OF STANDARD SYMBOL AND DASHED LINE COMBINATIONS

ISYM	SYMBOL	DASHED LINE
1	○	———————————————
2	□	- - - - - - - - - - - - - -
3	◇	— - — - — - — - — - — -
4	△	— - - - — - - - — - - -
5	◺	— - - - — - - - — - - -
6	◗	— — - — — - — — -
7	⌓	— — - - — — - -
8	⬡	— — - - - — — - - -
9	◊	— — - - - - — — - - - -
10	⌂	— — — - — — — -
11	⊕	— — — - - — — — - -
12	⊞	—— —— —— - - -
13	◇+	—— —— —— - - - -
14	△+	—— —— —— —— -
15	◺+	—— —— —— —— - -
16	◗+	—— —— —— —— - - -
17	⌓+	— — — — - - - -
18	⬡+	— — — — — -
19	◊+	— — — — — - -
20	⌂+	— — — — — - - -
21	•	— — — — — - - - -
22	+	— — — — — - - - - -

TABLE II. - LIST OF STANDARD AERODYNAMIC SYMBOLS DRAWN BY SUBROUTINE COEFSY

ISYM	COEFFICIENT	SYMBOL	WIDTH/HEIGHT
1	Lift	C_L	2.0
2	Drag	C_D	2.1
3	Pitching Moment	C_m	2.2
4	Rolling Moment	C_l	1.7
5	Yawing Moment	C_n	1.9
6	Side Force	C_Y	2.1
7	Lift-Drag Ratio	L/D	2.6
8	Angle of Attack	α,deg	3.5
9	Angle of Sideslip	β,deg	3.4

TABLE III.- LISTING OF SAMPLE INPUT CASES

CASE 1 - Single Plot with Three Variables

```
496.
 20.        23.        10.
  1.         3.
  1.         0.
  1.         .5
  1.
  3.
  6.         2.         .4        CL         1.         4.         2.       ALPHA
  7.        11.         .2        CD         1.         4.         2.       ALPHA
  8.        20.         .4        CM         1.         4.         2.       ALPHA
 $SHEET NO=206,RUN=39,59$
```

CASE 2 - Four Plots with One Variable

```
496.
 12.        12.        10.
  1.         2.
  1.         0.
  1.         .5
  1.
  4.
  8.         2.         .4        CM         2.         1.         .8          CL
  9.         1.         4.        L/D        1.         8.        10.       ALPHA
  2.         8.         .8        CL         1.         1.        10.       ALPHA
  3.         7.         .2        CD         2.         8.         .8          CL
$SHEET NO=206,RUN=39,59$
```

TABLE IV.- SAMPLE OUTPUT FOR CASE 1 INPUT

PAGE 1 OUTPUT

```
PLOTTING SETUP DECK FOR TEST =  496
SHEET WIDTH =   20.00     SHEET HEIGHT =   23.00     SPACE BETWEEN SHEETS =   10.00
STARTING SYMBOL NUMBER =  1     SYMBOL SIZE IS LARGE
DATA WILL BE FAIRED AND PLOTTED USING SPLINE WITH TENSION =  0.00
DATA WILL BE EDITED WITH AN EDIT TOLERANCE =    .500
TAPE LOCATION OF INDEPENDENT VARIABLE =   1
NUMBER OF PLOTS PER SHEET =   3
```

PLOT NO.	YTAPE	YOFFSET	YSCALE	YLABEL	XTAPE	XOFFSET	XSCALE	XLABEL
1	6	2.00	.4000	CL	1	4.00	2.0000	ALPHA
2	7	11.00	.2000	CD	1	4.00	2.0000	ALPHA
3	8	20.00	.4000	CM	1	4.00	2.0000	ALPHA

PAGE 2 OUTPUT

THE FOLLOWING LIST OF RUNS ARE AVAILABLE ON THE DATA DISK FOR TEST NO.   496

2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21
22	23	24	25	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42
43	44	45	46	47	48	49	50	51	52
53	54	55	56	57	58	59	60	61	62
63	64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81	82
83	84	85	86	87	88	89	90	91	92
93	94	95	96	97	98	99	100	101	102
103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122
123	124	125	126	127	128	129	130	131	132
133	134	135	136	137	138	139	140	141	142
143	144	145	146	147	148	149	150	151	152
153	154	155	156	157	158	159	160	161	162
163	164	165	166	167	168	169	170	171	172
173	174	175	176	177	178	179	180	181	182
183	184	185	186	187	188	189	190	191	192
193	194	195	196	197	198	199	200	201	202
203	204	205	206	207	208	209	210	211	212
213	214	215	216	217	218	219	220	221	222
223	224	225	226	227	228	229	230	231	232
233	234	235	236	237	238	239	240	241	242
243	244	245	246	247	248	249	250	251	252
253	254								

TOTAL NUMBER OF DATA POINTS =   4901

PAGE 3 OUTPUT

DATA PLOTTED

SHEET NO = 206   RUNS =   39,   59,   0,   0,   0,   0,   0,   0,   0,   0

TABLE IV. - CONCLUDED

PAGE 4 OUTPUT

THE FOLLOWING RUNS HAVE BEEN EDITED

RUN NO. = 39 TOLERANCE = .500

INPUT		SORTED		EDITED		DELETED	
POINT NO.	T-VALUE	POINT NO.	T-VALUE	POINT NO.	T-VALUE	POINT NO.	T-VALUE
1	-6.1700	1	-6.1700	1	-6.1700	4	.3500
2	-4.1800	2	-4.1800	2	-4.1800		
3	-1.8800	3	-1.8800	3	-1.8800		
4	.3500	20	.1700	20	.1700		
5	2.0100	4	.3500	5	2.0100		
6	4.2800	5	2.0100	6	4.2800		
7	6.3700	6	4.2800	7	6.3700		
8	8.4500	7	6.3700	8	8.4500		
9	10.3200	8	8.4500	9	10.3200		
10	12.4100	9	10.3200	10	12.4100		
11	14.4000	10	12.4100	11	14.4000		
12	16.8000	11	14.4000	12	16.8000		
13	18.6700	12	16.8000	13	18.6700		
14	20.8600	13	18.6700	14	20.8600		
15	22.7500	14	20.8600	15	22.7500		
16	24.7800	15	22.7500	16	24.7800		
17	26.6700	16	24.7800	17	26.6700		
18	28.7500	17	26.6700	18	28.7500		
19	30.7000	18	28.7500	19	30.7000		
20	.1700	19	30.7000				

RUN NO. = 59 TOLERANCE = .500

INPUT		SORTED		EDITED		DELETED	
POINT NO.	T-VALUE	POINT NO.	T-VALUE	POINT NO.	T-VALUE	POINT NO.	T-VALUE
1	-6.0200	1	-6.0200	1	-6.0200	4	.2800
2	-4.4000	2	-4.4000	2	-4.4000		
3	-2.2200	3	-2.2200	3	-2.2200		
4	.2800	4	.2800	20	.2900		
5	2.0000	20	.2900	5	2.0000		
6	4.8700	5	2.0000	6	4.8700		
7	6.3000	6	4.8700	7	6.3000		
8	8.2000	7	6.3000	8	8.2000		
9	10.5800	8	8.2000	9	10.5800		
10	12.6400	9	10.5800	10	12.6400		
11	14.9700	10	12.6400	11	14.9700		
12	16.6100	11	14.9700	12	16.6100		
13	18.4600	12	16.6100	13	18.4600		
14	20.8900	13	18.4600	14	20.8900		
15	22.8300	14	20.8900	15	22.8300		
16	25.3700	15	22.8300	16	25.3700		
17	26.9800	16	25.3700	17	26.9800		
18	28.9200	17	26.9800	18	28.9200		
19	31.3400	18	28.9200	19	31.3400		
20	.2900	19	31.3400				

ALL PLOTTING COMPLETED

TABLE V. - SAMPLE OF LEAST-SQUARES POLYNOMIAL CURVE FIT

DATA PLOTTED

SHEET NO = 206 RUNS = 39, 59, 0, 0, 0, 0, 0, 0,

RESULTS OF LEAST SQUARES POLYNOMIAL CURVE FIT FOR RUN = 39 NSYM = 1 NPOL = 3

I	T	X	XNEW	X-XNEW	Y	YNEW	Y-YNEW
1	-6.1700	.9150	.9150	-.0000	1.1329	1.0588	.0741
2	-4.1800	1.9100	1.9100	.0000	1.4455	1.4068	.0387
3	-1.8800	3.0600	3.0600	.0000	1.7651	1.8479	-.0828
4	.1700	4.0850	4.0850	.0000	2.2144	2.2678	-.0534
5	2.0100	5.0050	5.0050	.0000	2.6249	2.6597	-.0348
6	4.2800	6.1400	6.1400	.0000	3.1164	3.1541	-.0377
7	6.3700	7.1850	7.1850	.0000	3.5735	3.6117	-.0382
8	8.4500	8.2250	8.2250	.0000	4.0115	4.0610	-.0495
9	10.3200	9.1600	9.1600	.0000	4.4649	4.4533	.0116
10	12.4100	10.2050	10.2050	0.0000	4.9459	4.8711	.0747
11	14.4000	11.2000	11.2000	-.0000	5.4040	5.2417	.1623
12	16.8000	12.4000	12.4000	-.0000	5.8305	5.6430	.1874
13	18.6700	13.3350	13.3350	-.0000	5.9419	5.9138	.0281
14	20.8600	14.4300	14.4300	-.0000	6.0995	6.1757	-.0763
15	22.7500	15.3750	15.3750	-.0000	6.2170	6.3472	-.1301
16	24.7800	16.3900	16.3900	-.0000	6.3325	6.4677	-.1352
17	26.6700	17.3350	17.3350	.0000	6.4900	6.5143	-.0243
18	28.7500	18.3750	18.3750	.0000	6.3650	6.4852	-.1202
19	30.7000	19.3500	19.3500	.0000	6.5800	6.3744	.2056

ERROR IN X = .154875E-22 ERROR IN Y = .189547E+00

TABLE VI. - SAMPLE OF LEAST-SQUARES CUBIC-SPLINE CURVE FIT

DATA PLOTTED

SHEET NO = 206 RUNS = 39, 59, 0, 0, 0, 0, 0, 0,

RESULTS OF LEAST SQUARES CUBIC SPLINE CURVE FIT FOR RUN = 39 NSYM = 1 DF = .10000

I	T	X	XNEW	X-XNEW	Y	YNEW	Y-YNEW
1	-6.1700	.9150	.9150	0.0000	1.1329	1.0207	.1122
2	-4.1800	1.9100	1.9100	.0000	1.4455	1.4164	.0291
3	-1.8800	3.0600	3.0600	0.0000	1.7651	1.8793	-.1142
4	.1700	4.0850	4.0850	0.0000	2.2144	2.3021	-.0877
5	2.0100	5.0050	5.0050	.0000	2.6249	2.6911	-.0662
6	4.2800	6.1400	6.1400	.0000	3.1164	3.1807	-.0643
7	6.3700	7.1850	7.1850	.0000	3.5735	3.6354	-.0619
8	8.4500	8.2250	8.2250	0.0000	4.0115	4.0837	-.0722
9	10.3200	9.1600	9.1600	0.0000	4.4649	4.4744	-.0095
10	12.4100	10.2050	10.2050	0.0000	4.9459	4.8855	.0604
11	14.4000	11.2000	11.2000	.0000	5.4040	5.2396	.1644
12	16.8000	12.4000	12.4000	.0000	5.8305	5.6055	.2250
13	18.6700	13.3350	13.3350	.0000	5.9419	5.8400	.1018
14	20.8600	14.4300	14.4300	.0000	6.0995	6.0624	.0371
15	22.7500	15.3750	15.3750	.0000	6.2170	6.2162	.0009
16	24.7800	16.3900	16.3900	.0000	6.3325	6.3511	-.0186
17	26.6700	17.3350	17.3350	0.0000	6.4900	6.4569	.0331
18	28.7500	18.3750	18.3750	.0000	6.3650	6.5606	-.1956
19	30.7000	19.3500	19.3500	.0000	6.5800	6.6538	-.0738

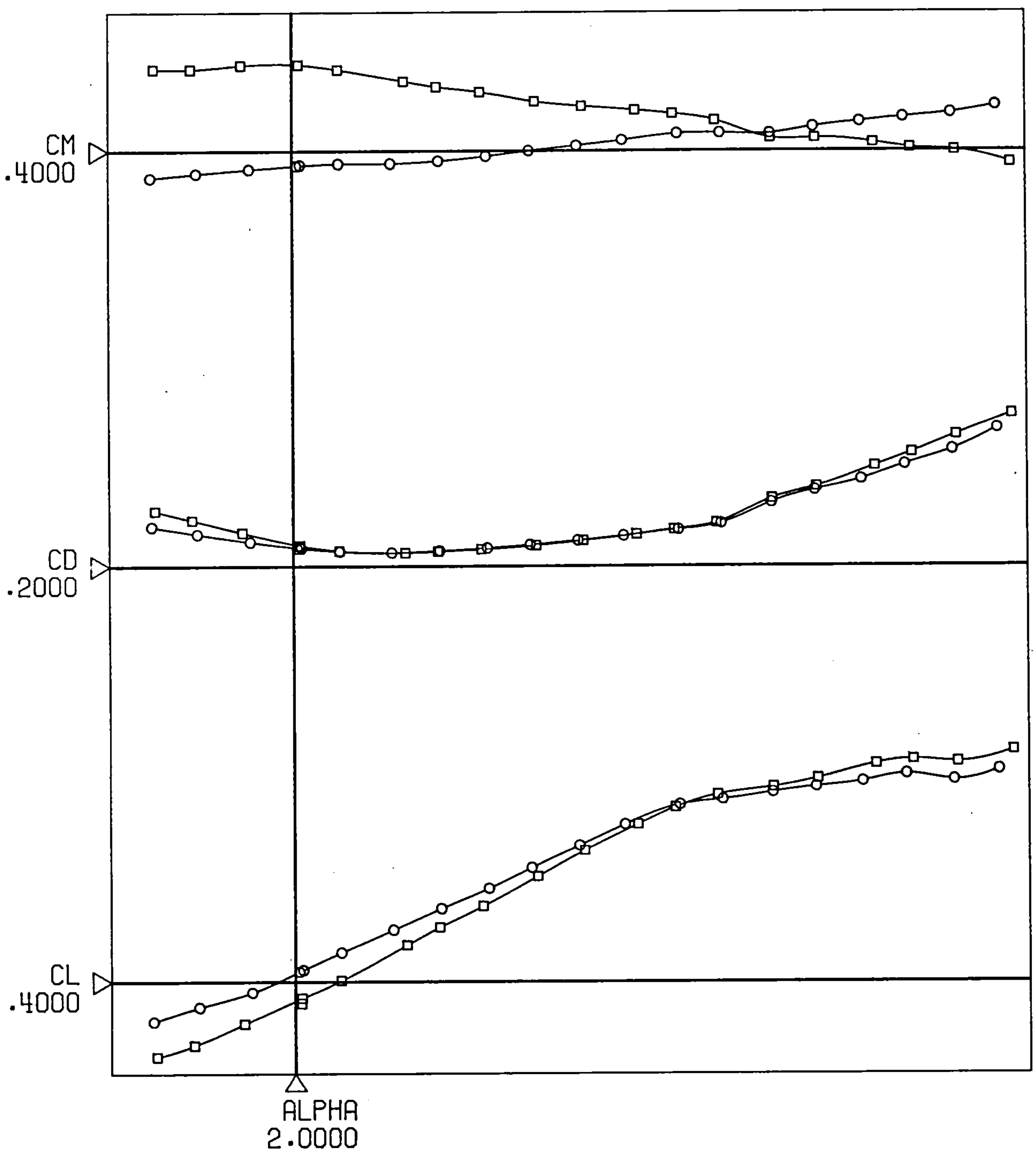
ERROR IN X = .575553E-25 ERROR IN Y = .190000E+00

TABLE VII.- COMPUTER CODE FOR CASE 1

```
C       DEFINE HEIGHT OF COEFFICIENTS
        HT=0.4
C       DEFINE HEIGHT OF AXIS LABELS
        SZ=0.75*HT
C       PROVIDE A BLANK SPACE AT BOTTOM OF PLOT FOR FIGURE TITLE
        CALL CALPLT (0.0,4.0,-3)
C       DRAW GRID LINES
        CALL GRIDLN (0.0,0.0,20.0,23.0,1)
C       DRAW ZERO LINES
        CALL ZEROLN (4.0,0.0,23.0,2)
        CALL ZEROLN (0.0,2.0,20.0,1)
        CALL ZEROLN (0.0,11.0,20.0,1)
        CALL ZEROLN (0.0,20.0,20.0,1)
C       DRAW AND LABEL X-AXIS  ( ALPHA )
        CALL AXISLB (1,0.0,0.0,-8.0,2.0,20.0,SZ,-1)
        XO=(20.0-3.5*HT)/2.0
        YO=-1.0
        CALL COEFSY (XO,YO,HT,8)
C       DRAW AND LABEL LOWER PORTION OF Y-AXIS   ( CL )
        CALL AXISLB (2,0.0,0.0,-0.8,0.4,10.0,SZ,1)
        XO=-1.8
        YO=5.0
        CALL COEFSY (XO,YO,HT,1)
C       DRAW AND LABEL MIDDLE PORTION OF Y-AXIS   ( CD )
        CALL AXISLB (2,0.0,11.0,0.0,0.2,5.0,SZ,1)
        XO=-1.8
        YO=13.5
        CALL COEFSY (XO,YO,HT,2)
C       DRAW AND LABEL UPPER PORTION OF Y-AXIS   ( CM )
        CALL AXISLB (2,0.0,17.0,-1.2,0.4,6.0,SZ,1)
        XO=-1.8
        YO=20.0
        CALL COEFSY (XO,YO,HT,3)
C       DRAW AND LABEL RUN KEY
        XO=2.0
        YO=HT+23.0
        CALL RUNKEY (XO,YO,RUN,LSYM,ISIZE,HT,10)
```

TABLE VIII. - COMPUTER CODE FOR CASE 2

```
C       DEFINE HEIGHT OF COEFFICIENTS
        HT=0.25
C       DEFINE HEIGHT OF AXIS LABELS
        SZ=0.75*HT
C       PROVIDE A BLANK SPACE AT BOTTOM OF PLOT FOR FIGURE TITLE
        CALL CALPLT (0.0,3.0,-3)
C
C       LOWER LEFT-HAND PLOT   ( CM VS CL )
C
C          DRAW GRID LINES AND ZERO LINES
        CALL GRIDLN (0.0,0.0,5.0,5.0,1)
        CALL ZEROLN (1.0,0.0,5.0,2)
        CALL ZEROLN (0.0,2.0,5.0,1)
C          DRAW AND LABEL X-AXIS
        CALL AXISLB (1,0.0,0.0,-0.8,0.8,5.0,SZ,1)
        XO=(5.0-2.0*HT)/2.0
        YO=-0.8
        CALL COEFSY (XO,YO,HT,1)
C          DRAW AND LABEL Y-AXIS
        CALL AXISLB (2,0.0,0.0,-0.8,0.4,5.0,SZ,1)
        XO=-1.0
        YO=2.5
        CALL COEFSY (XO,YO,HT,3)
C
C       LOWER RIGHT-HAND PLOT  ( L/D VS ALPHA )
C
C          DRAW GRID LINES AND ZERO LINES
        CALL GRIDLN (7.0,0.0,5.0,5.0,1)
        CALL ZEROLN (8.0,0.0,5.0,2)
        CALL ZEROLN (7.0,1.0,5.0,1)
C          DRAW AND LABEL X-AXIS
        CALL AXISLB (1,7.0,0.0,-10.0,10.0,5.0,SZ,-1)
        XO=7.0+(5.0-3.5*HT)/2.0
        YO=-0.8
        CALL COEFSY (XO,YO,HT,8)
C          DRAW AND LABEL Y-AXIS
        CALL AXISLB (2,7.0,0.0,-4.0,4.0,5.0,SZ,-1)
        XO=6.0
        YO=2.5
        CALL COEFSY (XO,YO,HT,7)
C
C       UPPER LEFT-HAND PLOT   ( CL VS ALPHA )
C
C          DRAW GRID LINES AND ZERO LINES
        CALL GRIDLN (0.0,7.0,5.0,5.0,1)
        CALL ZEROLN (1.0,7.0,5.0,2)
        CALL ZEROLN (0.0,8.0,5.0,1)
C          DRAW AND LABEL X-AXIS
        CALL AXISLB (1,0.0,7.0,-10.0,10.0,5.0,SZ,-1)
        XO=(5.0-3.5*HT)/2.0
        YO=6.2
        CALL COEFSY (XO,YO,HT,8)
C          DRAW AND LABEL Y-AXIS
        CALL AXISLB (2,0.0,7.0,-0.8,0.8,5.0,SZ,1)
        XO=-1.0
        YO=9.5
        CALL COEFSY (XO,YO,HT,1)
C
C       UPPER RIGHT-HAND PLOT  ( CD VS CL )
C
C          DRAW GRID LINES AND ZERO LINES
        CALL GRIDLN (7.0,7.0,5.0,5.0,1)
        CALL ZEROLN (8.0,7.0,5.0,2)
        CALL ZEROLN (7.0,7.0,5.0,1)
C          DRAW AND LABEL X-AXIS
        CALL AXISLB (1,7.0,7.0,-0.8,0.8,5.0,SZ,1)
        XO=7.0+(5.0-2.0*HT)/2.0
        YO=6.2
        CALL COEFSY (XO,YO,HT,1)
C          DRAW AND LABEL Y-AXIS
        CALL AXISLB (2,7.0,7.0,0.0,0.2,5.0,SZ,1)
        XO=6.0
        YO=9.5
        CALL COEFSY (XO,YO,HT,2)
C
C       DRAW AND LABEL RUN KEY
C
        XO=1.0
        YO=12.0+HT
        CALL RUNKEY (XO,YO,RUN,LSYM,ISIZE,HT,10)
```



TEST 496 ,SHEET 206 ,RUNS 39 59 0 0 0 0 0

Figure 1. - Plot of Case 1 input with standard symbols and cubic-spline curve fairing.

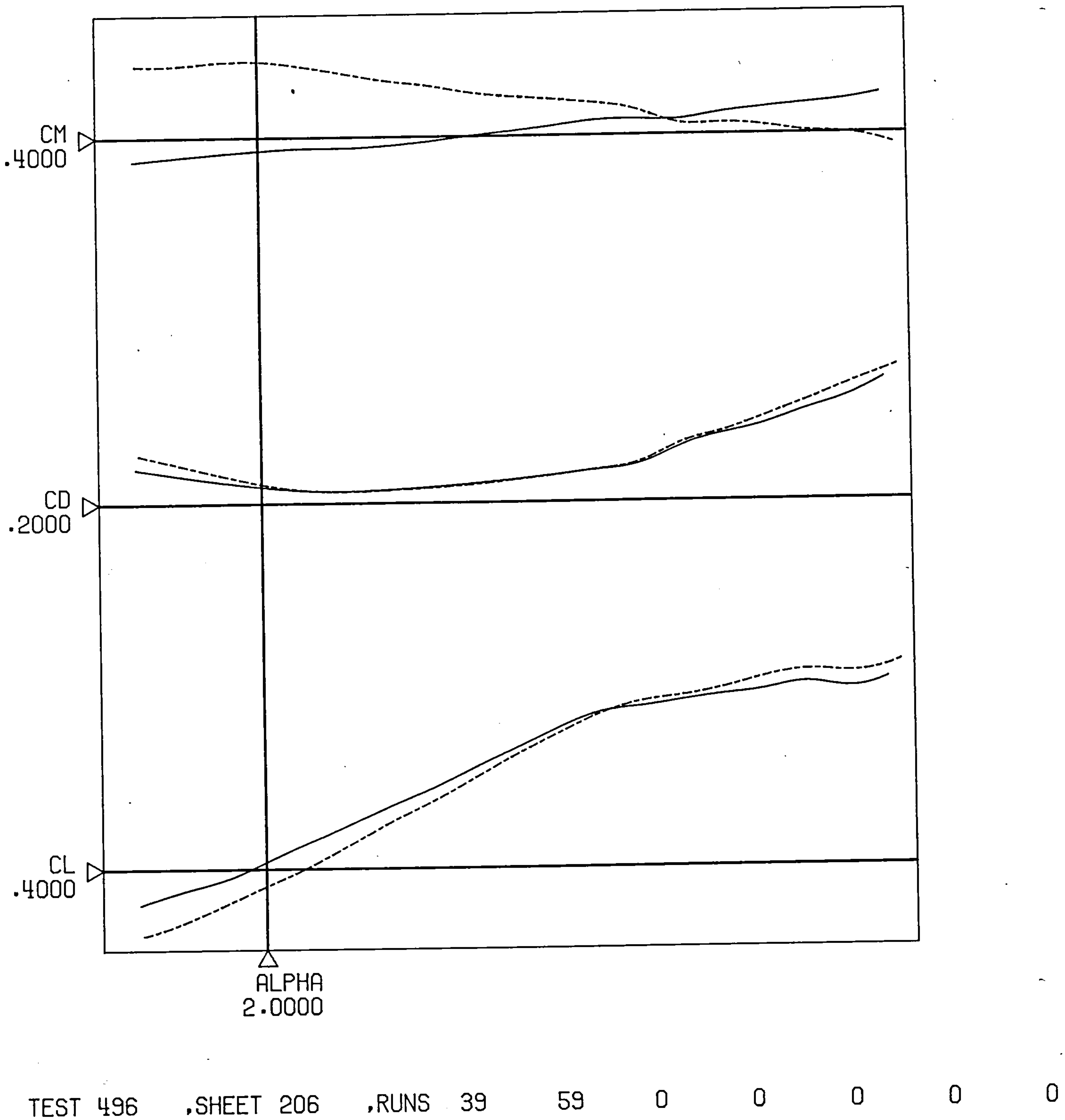
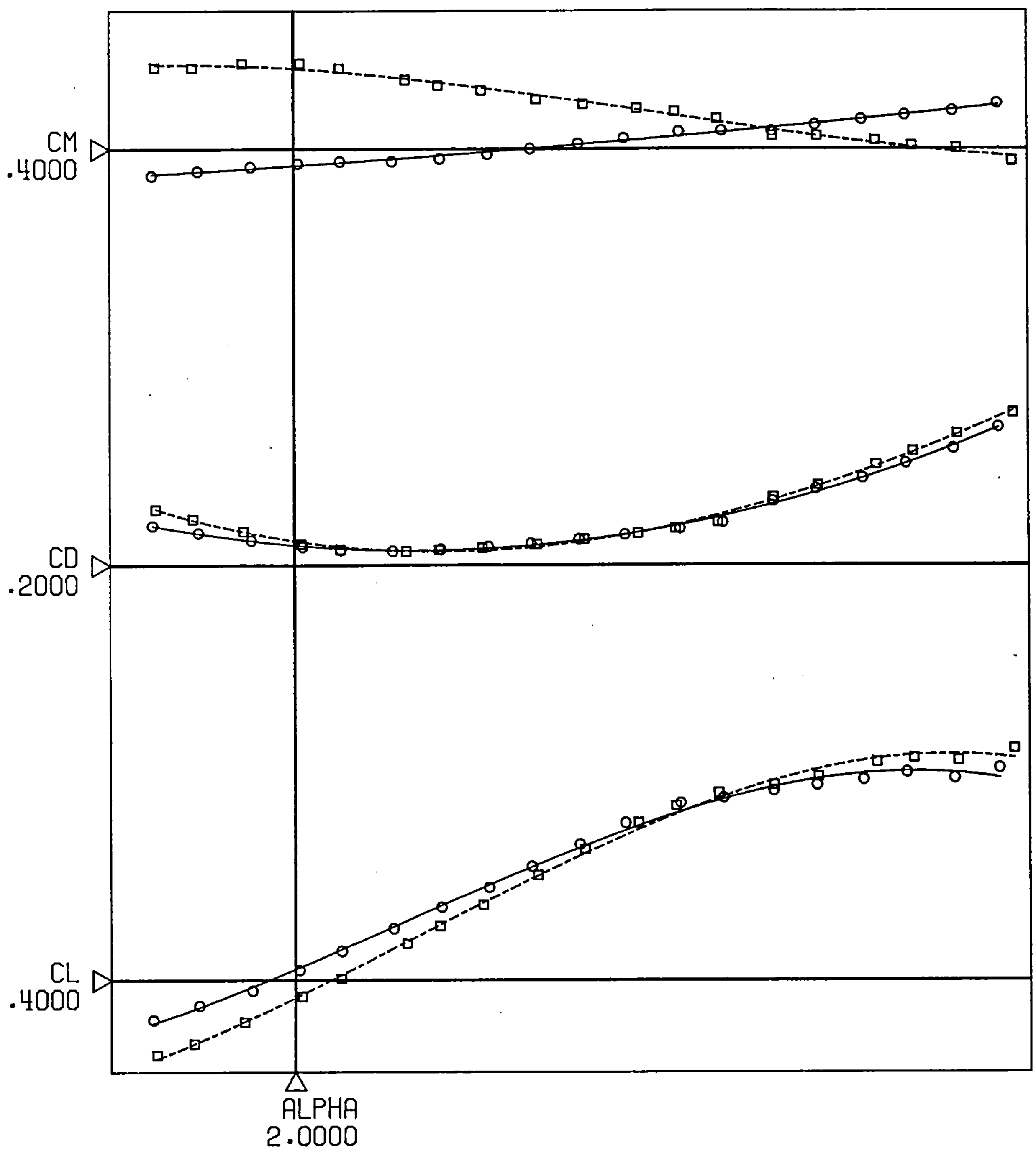


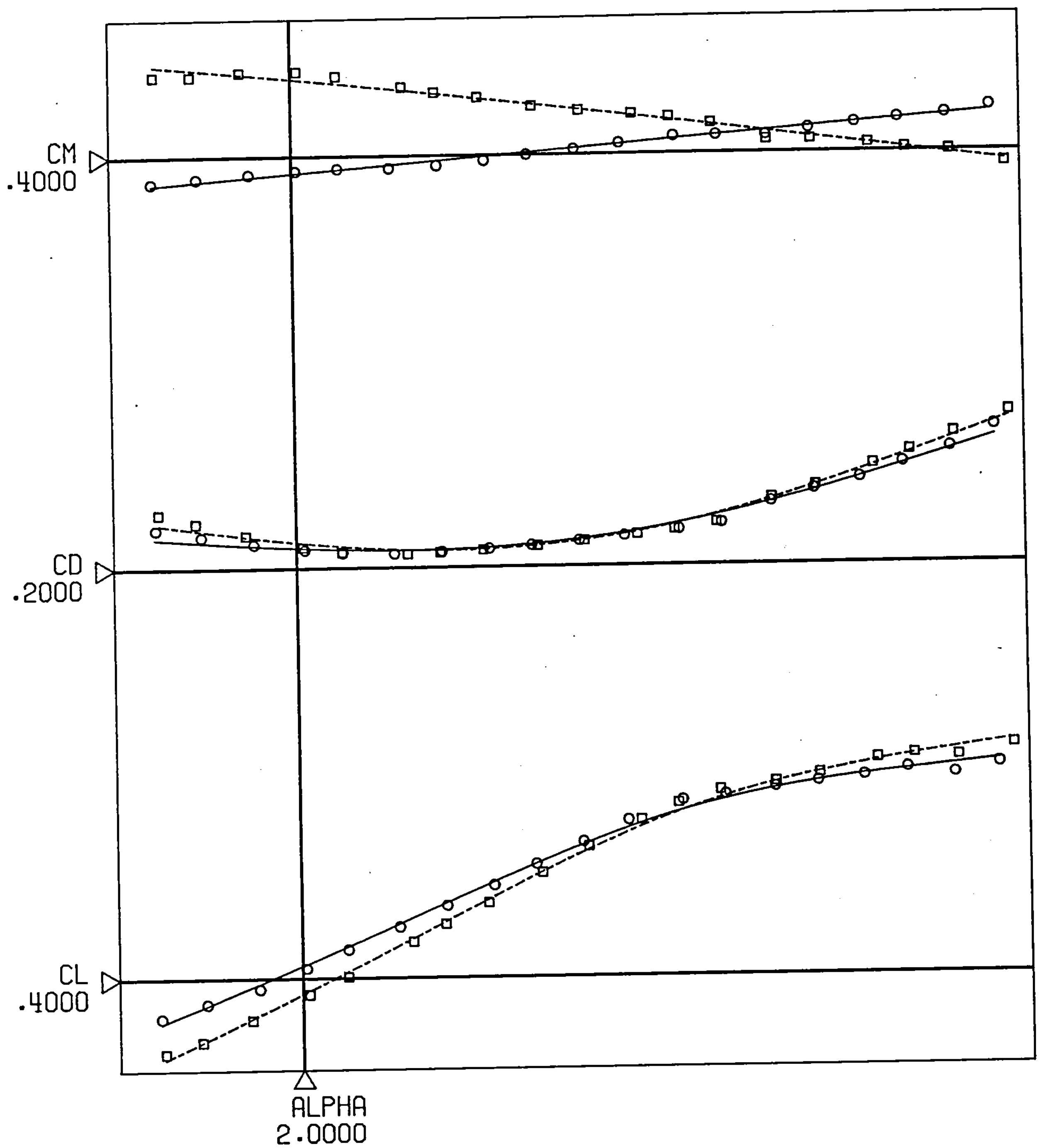
Figure 2. - Plot of Case 1 input with standard dashed line combination and cubic-spline curve fairing.



TEST 496 ,SHEET 206 ,RUNS 39 59 0 0 0 0 0

a) Least-Squares Polynomial

Figure 3. - Plot of Case 1 input with least-squares curve fairing.



TEST 496 ,SHEET 206 ,RUNS 39 59 0 0 0 0 0

b) Least-Squares Cubic Spline

Figure 3. - Concluded.

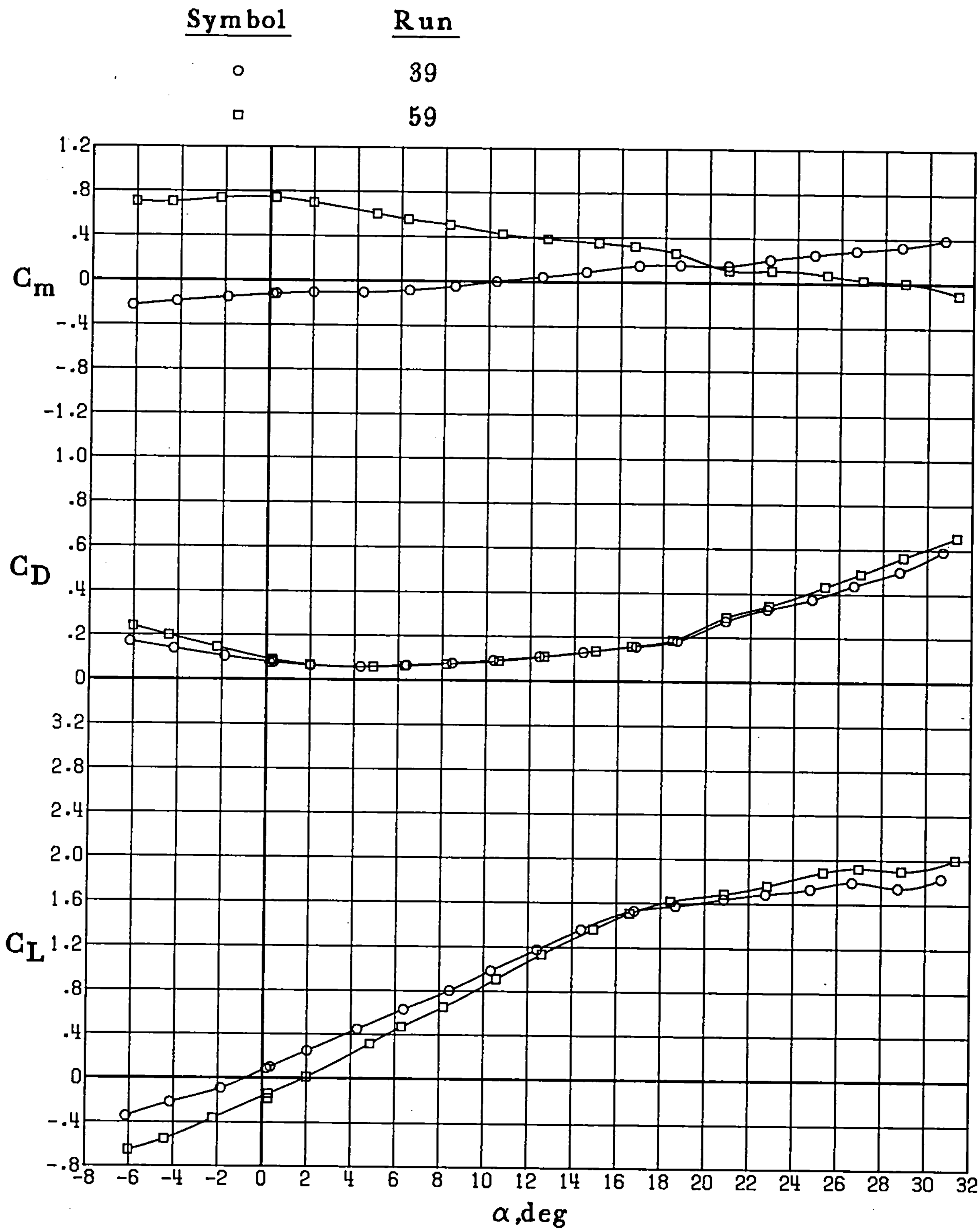


Figure 4. - Plot of Case 1 input with grid lines, axis scale values and labels, and legend drawn.

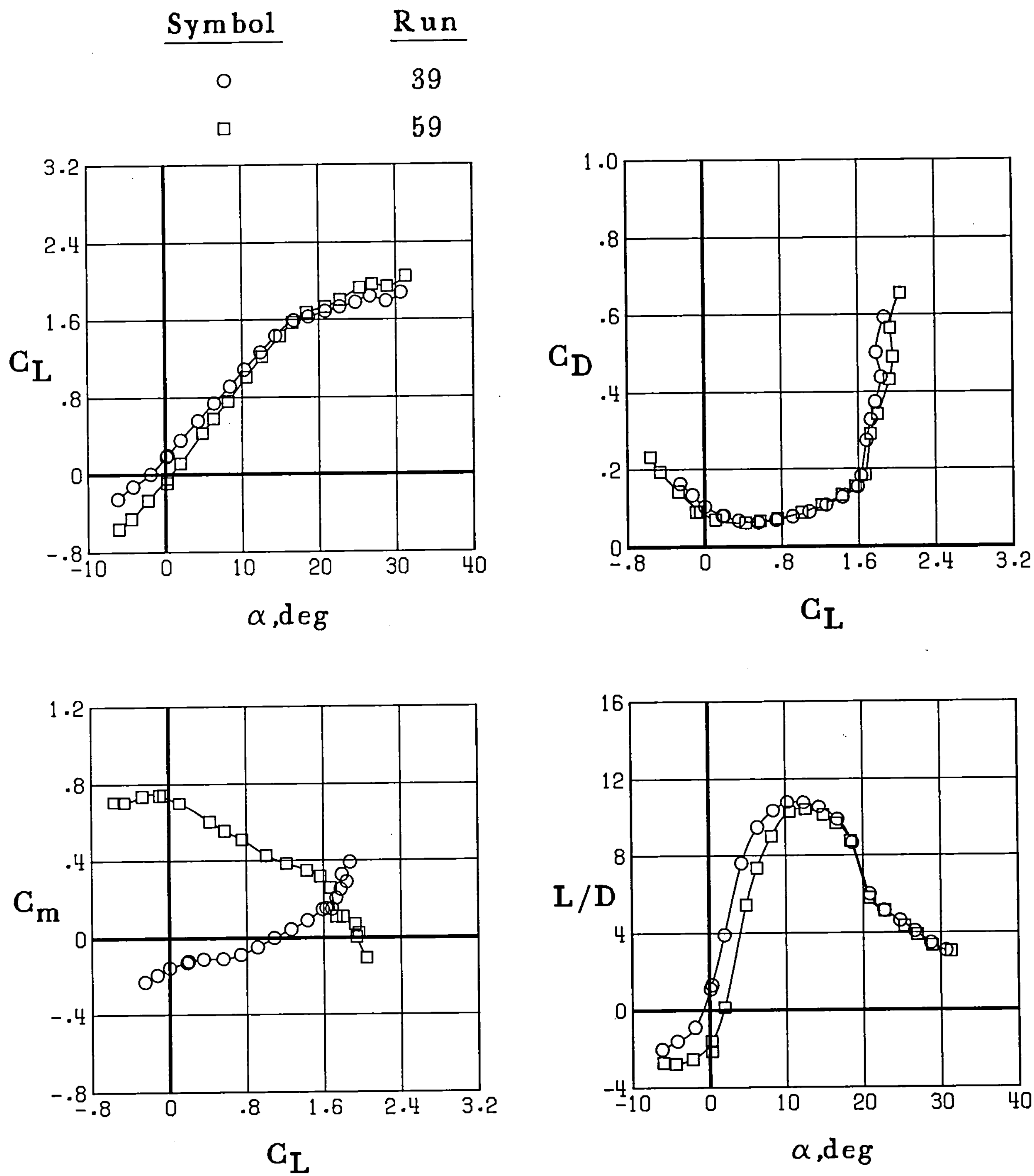


Figure 5.- Plot of Case 2 input with grid lines, axis scale values and labels, and legend drawn.

1. Report No.: NASA TM-85648
2. Government Accession No.:
3. Recipient's Catalog No.:
4. Title and Subtitle: Computer Program for Plotting and Fairing Wind-Tunnel Data
5. Report Date: July 1983
6. Performing Organization Code: 505-31-23-06
7. Author(s): Harry L. Morgan, Jr.
8. Performing Organization Report No.:
9. Performing Organization Name and Address: NASA Langley Research Center, Hampton, VA 23665
10. Work Unit No.:
11. Contract or Grant No.:
12. Sponsoring Agency Name and Address: National Aeronautics and Space Administration, Washington, DC 20546
13. Type of Report and Period Covered: Technical Memorandum
14. Sponsoring Agency Code:
15. Supplementary Notes:

16. Abstract

This report contains a detailed description of the Langley computer program PLOTWD which plots and fairs experimental wind-tunnel data. The program was written for use primarily on the Langley CDC computer and CALCOMP plotters. The fundamental operating features of the program are that the input data are read and written to a random-access file for use during program execution, that the data for a selected run can be sorted and edited to delete duplicate points, and that the data can be plotted and faired using tension splines, least-squares polynomial, or least-squares cubic-spline curves. The most noteworthy feature of the program is the simplicity of the user-supplied input requirements. Several subroutines are also included that can be used to draw grid lines, zero lines, axis scale values and labels, and legends. A detailed description of the program operational features and each subprogram are presented. The general application of the program is also discussed together with the input and output for two typical plot types. A listing of the program code, user-guide, and output description are presented in appendices. The program has been in use at Langley for several years and has proven to be both easy to use and versatile.

17. Key Words (Suggested by Author(s)):
Plotting and fairing
Wind-tunnel data
Computer program
Tension splines

18. Distribution Statement:
[redacted]
Subject Category 02

19. Security Classif. (of this report): Unclassified
20. Security Classif. (of this page): Unclassified
21. No. of Pages: 131
22. Price*:

Available: NASA's Industrial Applications Centers

DO NOT REMOVE SLIP FROM MATERIAL

Delete your name from this slip when returning material to the library.

NAME	MS
Jim Byrd	413

NASA Langley (Rev. May 1988) RIAD N-75