COMPUTER PROGRAM FOR PLOTTING
AND FAIRING WIND-TUNNEL DATA

Harry L. Morgan, Jr.

July 1983

Because of its significant early commercial potential, this
information, which has been developed under a U.S. Gov-
ernment program, is being disseminated within the United
States in advance of general publication. This information
may be duplicated and used by the recipient with the ex-
press limitation that it not be published. Release of this
information to other domestic parties by the recipient shall
be made subject to these limitations.

Foreign release may be made only with prior NASA ap-
proval and appropriate export licenses. This legend shall
be marked on any reproduction of this information in whole
or in part.

LIBRARY COPY

Review for general release July 31, 1983

NASA
National Aeronautics and
Space Administration
Langley Research Center
Hampton, Virginia 23665

LANGLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA
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.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_0, a_1, a_2, \ldots, a_m$</td>
<td>polynomial coefficients</td>
</tr>
<tr>
<td>$c_i, d_i, e_i, f_i$</td>
<td>coefficients of spline equation</td>
</tr>
<tr>
<td>$h_i$</td>
<td>increment in spline independent variable such as $(x - x_i), (y - y_i)$, or $(t - t_i)$</td>
</tr>
<tr>
<td>$M$</td>
<td>order of polynomial</td>
</tr>
<tr>
<td>$N$</td>
<td>number of points</td>
</tr>
<tr>
<td>$P(x)$</td>
<td>polynomial equation</td>
</tr>
<tr>
<td>$S$</td>
<td>sum-of-squares of differences</td>
</tr>
<tr>
<td>$SP$</td>
<td>cubic-spline smoothing parameter</td>
</tr>
<tr>
<td>$W$</td>
<td>weighting factor</td>
</tr>
<tr>
<td>$t$</td>
<td>independent variable</td>
</tr>
<tr>
<td>$x$</td>
<td>$x$-axis dependent variable</td>
</tr>
<tr>
<td>$x''$</td>
<td>$\frac{d^2x}{dt^2}$</td>
</tr>
<tr>
<td>$y$</td>
<td>$y$-axis dependent variable</td>
</tr>
<tr>
<td>$y''$</td>
<td>$\frac{d^2y}{dt^2}$</td>
</tr>
<tr>
<td>$\delta$</td>
<td>spline tension factor</td>
</tr>
<tr>
<td>$\delta$</td>
<td>increment</td>
</tr>
</tbody>
</table>

**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:

\[
\text{WRITE(1) IRUN, ITEST, IPT, (DATA(I), I = 1, IMAX)}
\]

where \(\text{IRUN}\) is the run number, \(\text{ITEST}\) is the test number, \(\text{IPT}\) is the data point number, \(\text{DATA}\) is an array containing the wind-tunnel data, and \(\text{IMAX}\) is the maximum number of elements in the \(\text{DATA}\) array. This input data requirement 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 coefficient form to the equivalent $X_p$ and $Y_p$ locations on the plotting sheet using equations

$$X_p = \frac{x_{wt}}{X_{sf}} - x_o$$

$$Y_p = \frac{y_{wt}}{Y_{sf}} - y_o'$$

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
yp 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 xwt and ywt variables
XOFFSET, YOFFSET arrays containing origin x₀ and y₀ locations on plotting sheet, in.
XSCHOOL, YSCALE arrays containing axis scale factors xsf and ysf, 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 xp and yp 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

NPMAKX 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 arrowheads 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 $\sigma$
IEDIT data sort-and-edit option
TOLR editing tolerance of independent variable
IW DATA array index of independent variable
NPLLOT 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*NPLLOT+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 pro-
gram 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 faiured 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 faiered 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 TVAR 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

\[
\delta = \frac{t_{i+1} - t_i}{(NH-1) I(t)}
\]

where \( NH = 101 \) and \( I(t) \) is the integer portion of \( (t_{i+1} - t_i) \). Spline coordinates are then computed at consecutive \( \delta \) 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 \( \text{FUNC} \) and then the straight line length and angular location computed from the center of the symbol. The angular value is then input to subroutine \( \text{SYMBOL} \) which determines the corresponding 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 \( \text{EPS} \).

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 \( \text{LINE} \) subroutine, which is part of the \( \text{CALCOMP} \) 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:

- \( \text{IVAR} \)
  - if = 0, independent variable \( t \) is plotted on \( x \)-axis
  - if = 1, a dependent variable is plotted on \( x \)-axis

- \( \text{IOP} \)
  - 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

\[ e_i y_{i-1} + d_i y_i + f_i y_{i+1} = c_i \]  \hspace{1cm} (4)

for \( i = 2, 3, \ldots, N-1 \) where for the cubic spline

\[ e_i = \frac{h_{i-1}}{6}, \hspace{0.5cm} d_i = \frac{h_{i-1} + h_i}{3}, \hspace{0.5cm} \text{and} \hspace{0.5cm} f_i = \frac{h_i}{6} \]  \hspace{1cm} (5)
and for the tension spline

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

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

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

For both cubic and tension spline,

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

\[ h_i = x_{i+1} - x_i, \]

and \( h_{i-1} = x_i - x_{i-1} \).

For the tension spline, the tension parameter \( \sigma \) is the average tension per length of the input \( x \) values and determined using the equation

\[ \sigma = \frac{\bar{\sigma} (N-1)}{(x_N - x_1)} \]

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

\[ y = a_1 x^2 + a_2 x + a_3 \]

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

\[ y'' = 2a_1 \quad (10) \]

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

\[ y''_N = c_N \quad \text{for} \quad i = N \quad (11) \]

and

\[ y''_i = \frac{c_i - \bar{f}_i y''_{i+1}}{\bar{d}_i} \quad \text{for} \quad i = N-1, N-2, \ldots, 1 \]

where

\[ \bar{d}_i = d_i - e_i \bar{f}_{i-1}, \]

\[ \bar{f}_i = f_i / \bar{d}_i \quad (12) \]

and

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

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 \( \sigma \)
- **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

\[ y(x) = y_i'' \left[ \frac{(x_i+1 - x)^3}{6h_i} - \frac{(x_i+1 - x) h_i}{6} \right] + \]

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

and for the tension spline is

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

\[ \frac{y_i''}{\sigma^2} \left[ \frac{\sinh \left[ \sigma(x - x_i) \right]}{\sinh (\sigma h_i)} - \frac{(x - x_i)}{h_i} \right] + \left[ y_i (x_{i+1} - x) + y_{i+1} (x - x_i) \right] \frac{h_i}{h_i} \] (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:
\[ x = x_{i-1} + \delta (x_i - x_{i-1}) \]
\[ y = y_{i-1} + \delta (y_i - y_{i-1}) \]  
(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

\[ p(x_i) = a_0 + a_1 x_i + a_2 x_i^2 + \ldots + a_M x_i^M \]  \hspace{1cm} (16)

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

\[ S = \sum_{i=1}^{N} (P(x_i) - w_i y_i)^2 \]  \hspace{1cm} (17)
The least or minimum error is
\[
\frac{\partial E}{\partial a_j} = 2 \sum_{i=1}^{N} \left[ P(x_i) - w_i y_i \right] x_i^j = 0
\]  
(18)
where \( j = 0, 1, 2, \ldots, M \). This reduces to a set of simultaneous equations in the form
\[
a_0 \sum x_i^j + a_1 \sum x_i^{j+1} \ldots + a_M \sum x_i^{j+M} = \sum w_i y_i x_i^j
\]  
(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
\[
P(x) = a_0 + a_1 h_i + a_2 h_i^2 + a_3 h_i^3
\]  
(20)
for \( i = 1, 2, 3, \ldots, N - 1 \) where \( h_i = (x - x_i) \) and \( N \) is the number of input points. The coefficients \( a_0, a_1, a_2, \) and \( a_3 \) are computed in a least-squares manner so that
\[
\sum_{i=1}^{N} \left[ \frac{P(x_i) - y_i}{\delta y_i} \right] \leq SP
\]  \hspace{1cm} (21)

and
\[
\int_{x_1}^{x_N} \left[ \frac{a^2P(x)}{dx^2} \right] dx \text{ is a minimum}
\]  \hspace{1cm} (22)

where the smoothing parameter \( SP \) is in the interval \( N - \sqrt{2N} \leq SP \leq N + \sqrt{2N} \) and \( \delta y_i \) 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
The 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 "7" 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:

- \( X_0, Y_0 \): \( X_p \) and \( Y_p \) values for the lower left-hand corner of the frame, in.
- \( X_L \): width of frame, in.
- \( Y_H \): height of frame, in.
- \( N_DIV \): 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, \( X_M \), 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:

- \( X_0, Y_0 \): 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, ISAVERS, TSAVERS, TEDIT, TEDIT, IUN, and TUN in subroutine PRTEDIT; 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 NMAX 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:

\begin{verbatim}
GET, PLOTWD/UN = 621288N
UPDATE,F,P = PLOTWD, S = SOURCE, I = 0, L = 0.
\end{verbatim}

REFERENCES


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.
# Listing of Deck: Plot

**Card No.**

<table>
<thead>
<tr>
<th>Card</th>
<th>Description</th>
<th>Format</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test - TFS number</td>
<td>(F10.0)</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Sheetw - Width of plotting sheet (inches)</td>
<td>(3F10.0)</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>SheetH - Height of plotting sheet (inches)</td>
<td>*</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Space - Space between plotting sheets (inches)</td>
<td>*</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>Iop - Plotting option IOP=0 plot data only</td>
<td>(2F10.0)</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Iop - Plotting option IOP=1 plot and fair data</td>
<td>*</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>Tension - Spline tension factor (for IOP=1 only)</td>
<td>(2F10.0)</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>Iedit - Edit option IEDIT=0 no editing</td>
<td>(2F10.0)</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>Iedit - Edit option IEDIT=1 edit with given tolerance</td>
<td>(2F10.0)</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>Tolerance - Edit tolerance for independent variable</td>
<td>*</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>Iw - Data array location of independent variable (F10.0)</td>
<td>*</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>Nplot - Number of plots per sheet</td>
<td>(F10.0)</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>Ytape - Array location of Y variable</td>
<td>(2F310.0,A10)</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>Yoffset - Origin offset of Y variable (inches)</td>
<td>*</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>Yscale - Scale value per inch for Y variable</td>
<td>*</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>Ylabel - Label for Y variable</td>
<td>*</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>Xtape - Array location of X variable</td>
<td>*</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>Xoffset - Origin offset of X variable (inches)</td>
<td>*</td>
<td>1</td>
</tr>
</tbody>
</table>
LISTING OF DECK: PLOT

CARD NO.

41  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 NPLT TIMES * PT 43
C * -- DEFINITION OF SHEET NAMELIST PARAMETERS -- * PT 44
C * SHEET - NAMELIST LABEL * PT 45
C * NO - SHEET NUMBER * PT 46
C * RUN - RUN NUMBERS TO BE PLOTTED ON SHEET (MAXIMUM OF 10) * PT 47
C * NEWCASE - IF NEW SETUP DECK follows SET NEWCASE=1 * PT 48
C * END - END OF NAMELIST * PT 49
C
C ***********************************************
C
C 55  C  INTEGER RUN
C  C  DIMENSION ARRAY LSYM WHICH CONTAINS ORDER OF PLOTTING SYMBOLS
C  C  DIMENSION LSYM(22)
C  60  C  DIMENSION RUN(NRNMAX) WHERE NRNMAX IS THE MAXIMUM NUMBER OF
C     ALLOWSABLE RUNS PER SHEET
C  C  DIMENSION RUN(10)
C  65  C  DIMENSION NPT(NPMax) WHERE NPMax IS THE MAXIMUM NUMBER OF DATA
C     POINTS FOR TOTAL PLOT
C     NOTE -- DO NOT DESTROY THIS ARRAY DURING PROGRAM EXECUTION
C  C  DIMENSION NPT(5000)
C  70  C  DIMENSION EACH OF THE FOLLOWING ARRAYS BY NPLMIX WHICH IS THE
C     MAXIMUM NUMBER OF ALLOWABLE PLOTS PER SHEET
C  C  DIMENSION XPLT(10), YPLT(10), YTAPE(10), YOFFSET(10), YSCALE(10),
C      1YLABEL(10), XTAP(10), XOFFSET(10), XSCALE(10), XLABEL(10)
C  C  DIMENSION EACH OF THE FOLLOWING ARRAYS BY NDMAK WHICH IS THE
C     MAXIMUM NUMBER OF ALLOWABLE POINTS FOR A GIVEN RUN
C  80  C

39
<table>
<thead>
<tr>
<th>CARD NO.</th>
<th>LISTING OF DECK: PLOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>81</td>
<td>DIMENSION X(50), Y(50), T(50)</td>
</tr>
<tr>
<td></td>
<td>COMMON /WORK/ ARRAY1(500)</td>
</tr>
<tr>
<td></td>
<td>COMMON /PLT/ ARRAY2(150)</td>
</tr>
<tr>
<td>85</td>
<td>DIMENSION VAR(NDMAX,NVAR) AND NLOC(NVAR) WHERE NVAR=2*NPLMAX+1</td>
</tr>
<tr>
<td></td>
<td>COMMON /WORK/ ARRAY1(500)</td>
</tr>
<tr>
<td></td>
<td>COMMON /PLT/ ARRAY2(150)</td>
</tr>
<tr>
<td>90</td>
<td>SHEET NAMELIST PARAMETERS</td>
</tr>
<tr>
<td></td>
<td>NAMELIST /SHEET/ NO,RUN,NEWCASE</td>
</tr>
<tr>
<td>95</td>
<td>NAMELIST /SHEET/ NO,RUN,NEWCASE</td>
</tr>
<tr>
<td></td>
<td>DEFINE INPUT TAPE NUMBER</td>
</tr>
<tr>
<td></td>
<td>DATA JREAD/5/</td>
</tr>
<tr>
<td>100</td>
<td>DEFINE ORDER OF PLOTTING SYMBOLS</td>
</tr>
<tr>
<td></td>
<td>DATA LSYM/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22</td>
</tr>
<tr>
<td>105</td>
<td>DEFINE MAXIMUM NUMBER OF ALLOWABLE RUNS PER SHEET</td>
</tr>
<tr>
<td></td>
<td>DATA NRNMAX/10/</td>
</tr>
<tr>
<td>110</td>
<td>DEFINE MAXIMUM NUMBER OF ALLOWABLE PLOTS PER SHEET</td>
</tr>
<tr>
<td></td>
<td>DATA NPLMAX/10/</td>
</tr>
<tr>
<td>115</td>
<td>DEFINE MAXIMUM NUMBER OF ALLOWABLE DATA POINTS FOR A GIVEN RUN</td>
</tr>
<tr>
<td></td>
<td>DATA NDMAX/50/</td>
</tr>
<tr>
<td>120</td>
<td>DEFINE MAXIMUM NUMBER OF ALLOWABLE TOTAL DATA POINTS</td>
</tr>
<tr>
<td></td>
<td>NMAX=5000</td>
</tr>
</tbody>
</table>
LISTING OF DECK: PLOT

CARD NO.

121  C DEFINE PROGRAM OPTIONS
C     IF IPRT=0, PRINT EDITED DATA
C     IF IPRT=1, DO NOT PRINT EDITED DATA
125  C 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
130  C     NORG=1, DO NOT DRAW ARROWHEAD AT ORIGIN OF PLOTTING VARIABLES
NORG=0
C     SET PROGRAM CONSTANTS
135  C IDISK=0
C     INITIALIZE PLOTTING DEVICE
C     CALL PSEUDO
CALL LEROY
C     READ SETUP DECK
140  C CALL SETUP (ITEST, SHEETW, SHEETH, SPACE, ISYM, ISIZE, IOP, TENSION, IEDIT
C     TCR, IT, YOFFSET, SCALE, YLABEL, XTAPEx, XOFFSET, XSCALE, XL
C     LABEL, JREAD, NPLMX, IERR)
C     IF (IERR, NE=0) GO TO 23
C REWIND 7
REWIND 11
KEDT=0
C     LOAD DATA DISK
150  C IF (IDISK) 3, 2, 3
C     CALL DATADK (0, ITEST, IRUN, VAR, NVAR, NLOC, NP, NPT, NPLMX, NDMAX, IERR)
C     IF (IFRR=NE.0) GO TO 23
C IDISK=1
160  C PRINT 24
LISTING OF DECK: PLOT

CARD NO.

161
165
170
175
180
185
190
195
200

LOAD PLOTTING VARIABLES
DO 4 I=1,NPLOT
J2=2*I
J1=J2-1
NL0C(J1)=IFIX(YTAPF(I)+0.0001)
NL0C(J2)=IFIX(XTAPF(I)+0.0001)
NVAR=2*NPLOT+1
NL0C(NVAR)=I

CHECK FOR DUPLICATE PLOTTING VARIABLES
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 (XTAPF(J)-XTAPF(I)) >5
IF (XOFFSET(I).EQ.XOFFSET(I).AND.XSCALE(J).EQ.XSCALE(I)) XPLT(I)=1
1=0
IF (YTAPF(J)-YTAPF(I)) >5
IF (YOFFSET(I).EQ.YOFFSET(I).AND.YSCALE(J).EQ.YSCALE(I)) YPLT(I)=1
1=0
CONTINUE
9 CONTINUE

READ SHEET NUMBER AND RUNS TO BE PLOTTED
DO 11 I=1,NRNMAX
RUN(I)=0
NEWCASE=NO=0

READ (JREAD,SHEFT)
IF (EOF(JREAD)) GT,12
CONTINUE

PRINT 25, NO,RUN

PERFORM GRID CHFCK
LISTING OF DECK: PLOT

CARD NO.

201 C
CALL CALPLT (O,,O,,3)
CALL GRIDCK
C
205 C
POSITION PLOTTING PEN AT LOWER LEFT HAND CORNER OF PLOTTING SHEET
C
CALL CALPLT (2,,2,,3)
C
210 C
LABEL TEST, SHEET, AND RUN NUMBERS
C
215 C
SZ=0,15
D=-1,9
LABEL TEST NUMBER
CALL NOTATE (D,-2,0,SZ,5HTEST,0,,5)
D=30*SZ/7,9+D
TP=FLOAT(ITEST)
C
CALL NUMBER (D,-7,0,SZ,TP,0,,1)
D=36*SZ/7,9+D
C
LABEL SHEET NUMBER
CALL NOTATE (D,-2,0,SZ,7THSHEET,0,,7)
D=42*SZ/7,9+D
TP=FLOAT(NO)
C
CALL NUMBER (D,-2,0,SZ,TP,0,,1)
D=36*SZ/7,9+D
C
LABEL RUN NUMBERS
CALL NOTATE (D,-2,0,SZ,6THRUNS,0,,6)
D=42*SZ/7,9+D
do 13 in=rnmmax
TP=FLOAT(RUN(I))
C
CALL NUMBER (D,-2,0,SZ,TP,0,,1)
D=42*SZ/7,9+D
IX=1FIX(D)
D=MAX=FLOAT(I)+SPACE+2,0
C
DRAW PLOTTING SHEET FRAME
C
235 C
CALL CALPLT (O,,0,,3)
C
CALL CALPLT (SHEET,0,,2)
C
CALL CALPLT (SHEET,2,,2)
C
CALL CALPLT (0,,SHEET,,2)
LISTING OF DECK: PLOT

CARD NO.

241
CALL CALPLT (0.,0.,2)
C
C
LABEL ORIGIN, VARIABLE DESCRIPTION, AND SCALE
C

245.
DO 15 I=1,NPLOT
IF (YPLOT(I).EQ.1.) GO TO 14
D=YOFFSET(I)
IF (NORG.EQ.0) CALL PARROW (-0.3,0.0,.D2,-2.0,.3)
CALL NOTATE (-1.8,D,0.15,YLABEL(I),0.,10)

14.
IF (XPLT(I).EQ.1.) GO TO 15
D=XOFFSET(I)
IF (NORG.EQ.0) CALL PARROW (0.0,0.0,.D2,-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

260.
DRAW ZERO LINES
C
IF (LZERO.NE.0) GO TO 18
DO 17 I=1,NPLOT
IF (YPLOT(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

270.
READ DATA FROM DATA DISK AND PLOT
C
KSYM=ISYM-1
C

275.
DO 21 IR=1,NRMAX
   DEFINE RUN NUMBER
IRUN=RUN(IR)
   DEFINE PLOTTING SYMBOL
KSYM=KSYM+1
280.
IF (KSYM.GT.22) KSYM=ISYM
LISTING OF DECK: PLOT

CARD NO.

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

PAGE 8
CONTINUE
ADVANCE TO ORIGIN OF NEXT FRAME
D=SHEFW+SPACE+2.*O
IF (D.LT.DMAX) D=DMAX
CALL NFRAME (D,O*)
C
READ NEXT SHEET NAMELIST UNLESS NEW SETUP DECK FOLLOWS
C
IF (NEWCASE.NE.0) GO TO 22
GO TO 10
C
PRINT EDITED DATA
C
IF (IEDIT.NE.0.AND.IPRF.EQ.0) CALL PRFEDIT
PRINT 27
GO TO 1
IF (IEDIT.NE.0.AND.IPRF.EQ.0) CALL PRFEDIT
C
FINALIZE PLOTTING DEVICE
C
CALL NOTATE (2.,2.,0.5,15) END OF PLOTTING,90.,15)
CALL NFRAME (4.,O*)
CALL CALPLT (0.,0.,999)
C
PRINT CLOSING MESSAGE
C
PRINT 28
STOP
C
FORMAT (1H1,3X,124DATA PLOTTED)
FORMAT (/4X,10HShefT NO =,15,5X,6HRUNS =,9(I7,1H),17/(30X,9(I7,1H)
1,) I7)
FORMAT (/4X,9HFOR RUN =,I7,3H ; ,I4,55H POINTS WERE NOT WITHIN PLO
ITING SHEET FRAME FOR PLOT =,I3)
FORMAT (/4X,2HNEW CASE SHOULD FOLLOW)
FORMAT (/4X,2HALL PLOTTING COMPLETED)
LISTING OF DECK: SETUP

CARD NO.

1

SUBROUTINE SETUP (ITEST, SHEETW, SHEETH, SPACE, ISYM, ISIZE, IOP, TENSION) ST 1
1, IEDIT, TOLR, IW, NPLT, YTAPE, YOFFSET, YSCALE, XSCALE, XLABEL, XTAPE, XOFFSET, XSC ST 2
2ALE, XLABEL, JREAD, NPLMAX, IERR) ST 3

5

ROUTINE TO READ PLOTTING SETUP DECK ST 5

C Coded by -- HARRY L. MORGAN NASA/LARC/TAD/AAB 1983 ST 7

10

PARAMETER DEFINITION ST 9

ITEST - TEST NUMBER ST 10
SHEETW - WIDTH OF PLOTTING SHEET (IN) ST 11
SHEETH - HEIGHT OF PLOTTING SHEET (IN) ST 12
SPACE - SPACE BETWEEN PLOTTING SHEETS (IN) ST 13
ISYM - STARTING SYMBOL NUMBER ST 14
IOP - PLOTTING OPTION - IOP=0, PLOT SYMBOLS ONLY ST 15
IOP=1, PLOT AND FAIR DATA ST 16
TENSION - SPLINE TENSION FACTOR ST 17
EDIT - EDIT OPTION - IEDIT=0, DO NOT SORT AND EDIT DATA ST 18
EDIT=1, SORT AND EDIT DATA ST 19
TOLR - EDIT TOLERANCE OF INDEPENDENT VARIABLE ST 20
IW - DATA ARRAY LOCATION OF INDEPENDENT VARIABLE ST 21
NPLT - NUMBER OF PLOTS PER SHEET ST 22
YTAPE - DATA ARRAY LOCATION OF Y VARIABLE ST 23
YOFFSET - ORIGIN OF Y VARIABLE (IN) ST 24
YSCALE - VALUE PER INCH FOR Y VARIABLE ST 25
YLABEL - LABEL FOR Y VARIABLE ST 26
XTAPE - DATA ARRAY LOCATION OF X VARIABLE ST 27
XOFFSET - ORIGIN OF X VARIABLE (IN) ST 28
XSCALE - VALUE PER INCH FOR X VARIABLE ST 29
XLABEL - LABEL FOR X VARIABLE ST 30
JREAD - TAPE NUMBER OF INPUT FILE ST 31
IERR - ERROR INDICATOR - IERR=0, NO ERRORS OCCURRED ST 32
IERR=1, ERRORS OCCURRED ST 33

35

XTAPE, XOFFSET, XSCALE, XLABEL, YTAPE, YOFFSET, YSCALE, AND YLABEL ST 35
MUST BE PROPERLY DIMENSIONED IN CALLING PROGRAM ST 36
DIMENSION XTAPE(1), XOFFSET(1), XSCALE(1), XLABEL(1), YTAPE(1), YO ST 37
OFFSET(1), YSCALE(1), YLABEL(1) ST 38

30

DIMENSION AND DEFINE SYMBOL SIZE ARRAY AND NMAX ST 39

40

DIMENSION IS(3) ST 40

PAGE 1

47
LISTING OF DECK: SETUP

DATA IS/6HSMALL, 5Hmedium, 6Hlarge /

INITIALIZE ERROR INDICATOR
IERP=0

READ AND PRINT TEST NUMBER
READ (JREAD,6) D1
IF (EOF(JREAD)) 4,1
ITEST=IFIX(D1+0.0001)
PRINT 7, ITEST

READ AND PRINT SHEETW, SHEETH, AND SPACE
READ (JREAD,6) SHEETW, SHEETH, SPACE
IF (SHEETW.LT.11.0) SHEETW=11.0
PRINT 8, SHEETW, SHEETH, SPACE

READ AND PRINT ISYM AND ISIZE
READ (JREAD,6) D1, D2
ISYM=IFIX(D1+0.0001)
ISIZE=IFIX(D2+0.0001)
IF (ISYM.LT.1.0 OR ISYM.GT.22) ISYM=1
IF (ISIZE.LT.1.0 OR ISIZE.GT.3) ISIZE=2
PRINT 9, ISYM, ISIZE

READ AND PRINT IOP AND TENSION
READ (JREAD,6) D1, TENSION
IOP=IFIX(D1+0.0001)
IF (IOP.LT.0.0 OR IOP.GT.1.0) IOP=0
IF (IOP.EQ.0) PRINT 10
IF (IOP.EQ.1) PRINT 11, TENSION

READ AND PRINT IEDIT AND TOLR
READ (JREAD,6) D1, TOLR
IEDIT=IFIX(D1+0.0001)
LISTING OF DECK: SFTUP

CARD NO.

81 IF (EDIT.LT.0, OR. EDIT.GT.1) IFINIT*0 ST R1
IF (EDIT.EQ.1) PRINT 12, TOLR
C
85 READ AND PRINT IW
C
86 READ (JREAD, 6) D1
IW=IFIX(D1+0.0001)
PRINT 13, IW
IF (IW.LE.0) GO TO 4
C
90 READ AND PRINT NPLOT
C
91 READ (JREAD, 6) D1
NPLOT=IFIX(D1+0.0001)
PRTN 14, NPLOT
IF (NPLOT.LE.0.OR.NPLOT.GT.NPLMAX) GO TO 3
C
95 READ AND PRINT YTAPE, YOFFSET, YSCALE, YLABEL, XTAPE, XOFFSET, XSCALE,
AND XLABEL
C
100 DO 2 I=1, NPLOT
READ (JREAD, 15) YTAPE(I), YOFFSET(I), YSCALE(I), YLABEL(I), YTAPE(I), X
1OFFSET(I), XSCALE(I), XLABEL(I)
2 CONTINUE
GO TO 5
C
105 PRINT 16, I, YTAPE(I), YOFFSET(I), YSCALE(I), YLABEL(I), X, XOFFSET(I), YSCALE
I(I), XLABEL(I)
C
110 PRINT ERROR MESSAGES THAT MAY HAVE OCCURRED
C
115 PRINT 17, NPLOT
IERP=1
GO TO 5
IERP=1
C
120 RETURN TO CALLING PROGRAM
C
125 RETURN
LISTING OF DECK: SETUP

CARD NO.

121  C  FORMAT (4F10,4)  ST 121
6  FORMAT (1H1,3X,30P) PLOTTING SETUP DECK FOR TEST = I5)  ST 122
7  FORMAT (4X,13HSHEET WIDTH = F8.2,5X,14HSHEET HEIGHT = F8.2,5X,22MS  ST 123
8  FORMAT (4X,13HSHEET WIDTH = F8.2,5X,14HSHEET HEIGHT = F8.2,5X,22MS  ST 124
125  9  FORMAT (4X,24HSHEET WIDTH = F8.2)  ST 125
10  FORMAT (4X,25HSHEET WIDTH = F8.2)  ST 126
11  FORMAT (4X,25HSHEET WIDTH = F8.2)  ST 127
130  11  FORMAT (4X,59HSHEET WIDTH = F8.2)  ST 128
12  FORMAT (4X,59HSHEET WIDTH = F8.2)  ST 129
13  FORMAT (4X,59HSHEET WIDTH = F8.2)  ST 130
135  14  FORMAT (4X,59HSHEET WIDTH = F8.2)  ST 131
15  FORMAT (4X,59HSHEET WIDTH = F8.2)  ST 132
16  FORMAT (4X,59HSHEET WIDTH = F8.2)  ST 133
17  FORMAT (4X,59HSHEET WIDTH = F8.2)  ST 134
18  FORMAT (4X,59HSHEET WIDTH = F8.2)  ST 135
19  FORMAT (4X,59HSHEET WIDTH = F8.2)  ST 136
20  FORMAT (4X,59HSHEET WIDTH = F8.2)  ST 137
21  FORMAT (4X,59HSHEET WIDTH = F8.2)  ST 138
22  FORMAT (4X,59HSHEET WIDTH = F8.2)  ST 139
LISTING OF DECK: DATADK

CARD NO.

1  SUBROUTINE DATADK (ICODE, ITEST, IPUN, VAR, NVAR, NLOC, NP, NPT, NPMAX, NOM) DK 1
1AX, IERR) DK 2
C
C ROUTINE TO READ FROM OR WRITE TO A DATA DISK DK 3
DK 4
DK 5
DK 6
C Coded by -- HARRY L. MORGAN NASA/LARC/TAD/AAK 1983 DK 7
C
C PARAMETER DEFINITION DK 8
C
C ICODE - READ OR WRITE OPTION DK 9
C ITEST - TEST NUMBER DK 10
C ICODE=0, WRITE IN DATA DISK DK 11
C ICODE=1, READ DATA FROM DATA DISK DK 12
C IRUN - RUN NUMBER TO BE READ FROM DATA DISK DK 13
C VAR(I,J) - TWO DIMENSIONAL ARRAY CONTAINING DATA READ FROM DK 14
C DATA DISK, (I=1,....,NP AND J=1,....,NVAR) DK 15
C NVAR - NUMBER OF VARIABLES READ FROM DISK FOR EACH POINT OF DK 16
C DATA DK 17
C NLOC(J) - ARRAY CONTAINING DATA ARRAY LOCATIONS OF DESIRED DK 18
C VARIABLES, (J=1,....,NVAR) DK 19
C NP - NUMBER OF DATA POINTS READ FROM DISK FOR DESIRED RUN DK 20
C NPT - ARRAY CONTAINING COUNT NUMBER OF DATA POINTS STORED ON DK 21
C THE DATA DISK DK 22
C NPMAX - MAXIMUM NUMBER OF DATA POINTS DK 23
C NDMAX - MAXIMUM NUMBER OF ALLOWABLE DATA POINTS FOR ANY GIVEN DK 24
C RUN DK 25
C IERR - ERROR INDICATOR DK 26
C IERR=0, NO ERRORS OCCURRED DK 27
C IERR=1, ERROR OCCURRED DURING READ OR WRITE DK 28
C IERR=2, IRUN NOT ON DATA DISK DK 29
C
C NOTE -- VAR, NLOC, AND NPT MUST BE PROPERLY DIMENSIONED IN THE DK 30
C CALLING PROGRAM DK 31
C
C DIMENSION VAR(NDMAX,1), NLOC(1), NPT(1) DK 32
C
C WORK ARRAY DK 33
C C
C COMMON /PLT/ NPT(10) DK 34

C

51
LISTING OF DECK DATADK

CARD NO.

41
C DIMENSION DATA(IMAX+3) WHERE IMAX IS THE MAXIMUM NUMBER OF DATA
C VARIABLES STORED ON THE INPUT DATA TAPE

45
C COMMON /WORK/ DATA(303)

50
C DEFINE ITAPE (INPUT DATA TAPE NUMBER) AND NDISK (DATA DISK
C NUMBER)

C DATA ITAPE/1/,NDISK/2/

55
C DEFINE IMAX
C DATA IMAX/300/

C COMPUTE ROUTINE CONSTANTS

C IERR=0
C I3=IMAX+3
C REWIND 12

C SELECT READ OR WRITE PROCEDURE

60
C IF (ICODE) 1,1,16
C WRITE DATA ON DATA DISK

65
C INITIALIZE NPT
C DO 2 I=1,NMAX
C NPT(I)=0
C OPEN DATA DISK
C CALL OPENHS (NDISK,NPT,NMAX,0)

70
C COMPUTE NEEDED CONSTANTS
C II=IMAX+1
C K=0
C IRUNP=0
C IC=0

75
C REWIND INPUT TAPE
C REWIND ITAPE
C READ DATA FROM INPUT TAPE
C READ (ITAPE) IRUN,ITESTC,IPT,(DATA(I),I=1,IMAX)
C IF (EOF(ITAPE)) 10,4

80
C CONTINUE
LISTING OF DECK: DATADK

CARD NO.

81 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
85   LOAD DATA ARRAY
89   DO 5 I=1,IMAX
      J=1-I
      J3=J+3
   5 DATA (J3)=DATA(J)
   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-NP MAX) 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
   C
   WRITE (12) IRUN,K
   IRUNP=IRUN
   9 CALL WRITS (NDISK,DATA,I3,K,O,O)
   C
   READ NEXT DATA POINT ON INPUT TAPE
   C
   GO TO 3
   C
   WRITE ON TAPE 12 RUN NUMBER AND COUNT NUMBER OF LAST DATA
   C
   POINT + 1 OF RUN
   3 K=K+1
   C
   WRITE (12) IRUNP,K
   C
   END FILE 12
   C
   10 PRINT SUMMARY OF RUNS STORED ON DATA DISK
   C
   PRINT 29, ITESTC
   IF (IC.EQ.0) GO TO 13
   C
   REWIND 12
   C
   K=0

53
LISTING OF DECK: DATADK

CARD NO.

121  DO 12 I=1,IC
     READ (12) IRUN,JS
     K=K+1
     NPRT(K)=IRUN
     IF (K-10) 12,11,11
     PRINT 30, (NPRT(J),J=1,10)
     K=0
     11  CONTINUE
     IF (K.GT.0.AND.K.LT.10) PRINT 30, (NPRT(J),J=1,K)
     READ (12) IRUN,JS
     K=K-1
     PRINT 31, KS
     REWIND 12
     GO TO 20
     12
     C  PRINT ERROR THAT MAY HAVE OCCURRED DURING WRITE TO DISK
     C
     13  PRINT 32
     IERR=1
     GO TO 28
     14
     15  PRINT 34, ITEST, ITESTC
     IERR=1
     GO TO 28
     16
     C  READ DATA FROM DATA DISK
     C
     17  READ TAPE 12 TO FIND POINT COUNT OF FIRST DATA POINT FOR DESIRED RUN
     16
     READ (12) J,K
     IF (EOF(12)) 23,17
     IF (J-IRUN) 16,18,16
     17
     18  READ TAPE 12 TO FIND POINT COUNT OF FIRST DATA POINT FOR NEXT RUN ON DATA DISK
     C
     READ (12) J,JS
     IF (EOF(12)) 24,19
     C
     19  COMPUTE NUMBER OF DATA POINTS ON DATA DISK FOR DESIRED RUN
     NP=JS-K
     20
     21
LISTING OF DECK: DATADK

CARD NO.

161 C IF (NP GT NDMAX) GO TO 26
   READ DATA FROM DATA DISK
   DO 22 I=1, NP
   CALL READMS (NDISK,DATA,I3,K)
165 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
170 J3=NLOC(J)+3
   IF (J3-I3) 20, 27
20 VAR(I,J)=DATA(J3)
21 CONTINUE
22 K=K+1
175 GO TO 28

C C PRINT ERROR THAT MAY HAVE OCCURRED DURING READ FROM DATA DISK
C
180 23 PRINT 35, IRUN
   IERR=2
   GO TO 28
24 PRINT 36, IRUN
   IERR=1
   GO TO 28
185 25 PRINT 37, ITEST, ITESTC
   IERR=1
   GO TO 28
26 PRINT 38, IRUN, NDMAX
   IERR=1
   GO TO 28
190 27 PRINT 39, JS, IRUN, IMAX
   IERR=1
C C RETURN TO CALLING PROGRAM
195 C
28 RETURN
C
29 FORMAT (1H1, 5X, 71H THE FOLLOWING LIST OF RUNS ARE AVAILABLE ON THE
   1DATA DISK FOR TEST NO., I5/)
200 30 FORMAT (5X, 101I0)
LISTING OF DECK: DATADK

PAGE 6

CARD NO.

201 31  FORMAT (/5X,29HTOTAL NUMBER OF DATA POINTS = ,I7)  DK 201
201 32  FORMAT (/5X,67ERROR OCCURRED DURING WRITE TO DATA DISK --- NO DATA ) DK 202
201 33  1TA ON INPUT TAPE ) DK 203
205 34  FORMAT (/5X,92ERROR OCCURRED DURING WRITE TO DATA DISK --- POINT DK 204
205 35  1 COUNT HAS EXCEEDED ALLOWABLE MAXIMUM OF ,I7) DK 205
205 36  FORMAT (/5X,71ERROR OCCURRED DURING WRITE TO DATA DISK --- REQUE DK 206
205 37  1STFD TEST NUMBR IS ,I5,34H AND TEST NUMBR ON INPUT TAPE IS ,I5) DK 207
210 38  FORMAT (/5X,5ERROR OCCURRED DURING READ FROM DATA DISK --- RUN DK 208
210 39  1NUMBER ,I7,15H IS NOT ON DISK) DK 209
210 40  FORMAT (/5X,91ERROR OCCURRED DURING READ FROM DATA DISK --- SEAR DK 210
210 41  1CH FOR STARTING POINT FOR RUN FOLLOWING ,I7,14H WAS NOT FOUND) DK 211
215 42  FORMAT (/5X,72ERROR OCCURRED DURING READ FROM DATA DISK --- REQU DK 212
215 43  1TESTED TEST NUMBR IS ,I5,78H AND TEST NUMBR ON DISK IS ,I5) DK 213
220 44  FORMAT (/5X,47ERROR OCCURRED DURING READ FROM DATA DISK --- /5X, DK 214
220 45  125HNUMBER OF POINTS FOR RUN ,I7,30H EXCEEDS ALLOWABLE MAXIMUM OF ,I4) DK 215
220 46  END DK 219

56
SUBROUTINE EDIT (T,X,Y,N,NSYM,ISIZE,IRUN,TOLR,KEDT,IERR)

ROUTINE TO SORT AND EDIT DATA

CODED BY -- HARRY L. MORGAN
NASA/LARC/TAD/AAR 1983

PARAMETER DEFINITION

T - ARRAY CONTAINING INDEPENDENT VARIABLE
X - ARRAY CONTAINING X VARIABLE
Y - ARRAY CONTAINING Y VARIABLE
N - NUMBER OF POINTS TO BE SORTED AND EDITED
NSYM - SYMBOL NUMBER TO BE USED TO PLOT UNEDITED POINTS
ISIZE - SYMBOL SIZE
IRUN - RUN NUMBER
TOLR - EDIT TOLERANCE OF INDEPENDENT VARIABLE
KEDT - NUMBER OF RUNS ALREADY EDITED AND STORED FOR FUTURE PRINT OUT
IERR - EDIT ERROR INDICATOR

IERR=0, NO ERRORS OCCURRED
IERR=1, DURING EDIT OF FIRST SORT POINT ALL GREATER POINTS ARE WITHIN EDIT TOLERANCE
IERR=2, DURING EDIT OF LAST SORT POINT ALL LESSER POINTS ARE WITHIN EDIT TOLERANCE
IERR=3, DURING EDIT OF INTERMEDIATE SORT POINT ALL GREATER POINTS ARE WITHIN EDIT TOLERANCE
IERR=4, DURING EDIT OF INTERMEDIATE SORT POINT ALL LESSER POINTS ARE WITHIN EDIT TOLERANCE

T, X, AND Y MUST BE PROPERLY DIMENSIONED IN CALLING PROGRAM

DIMENSION T(1), X(1), Y(1)

WORK ARRAYS
COMMON /WORK/ TIN(50),TEDT(50),TSAVE(50),YSAVE(50),IEDIT(50),ITEMP(150),ISAVE(50)

CHECK TO SEE IF THIS RUN HAS ALREADY BEEN EDITED AND STORED FOR FUTURE PRINT OUT

IERR=0

IF (N.LT.1) RETURN
LISTING OF DECK: EDIT

CARD NO.

41  NSTORE=1
    REWIND 1
    IF (KEDT) 3,3,1
    READ (11) NRUN
    IF (EOF(11)) 3,2
    1
    IF (IRUN-NRUN) 1,4,1
    KEDT=KEDT+1
    NSTORE=0
    IF (KEDT GT 1) BACKSPACE 11
    WRITE (11) IRUN
    END FILE 11

45  IN=N

50  C
    SET INITIAL VALUES OF Tin, TEdT, IEdT, ANDITEM
    ED 54
    ED 55
    ED 56
    ED 57
    ED 58
    ED 59
    ED 60
    ED 61
    ED 62
    ED 63
    ED 64
    ED 65
    ED 66
    ED 67
    ED 68
    ED 69
    ED 70
    ED 71
    ED 72
    ED 73
    ED 74
    ED 75
    ED 76
    ED 77
    ED 78
    ED 79
    ED 80

75  K1=K1-1
    GO TO 8
    7
    TMN=TMN
    8
    JSAVE=1
    DO 11 J=1,K1
    IF (I.EQ.1) TEdT(J)=T(J)

70  DO 6 JJ=1,JSAVE
    IF (JJ .EQ. JSAVE) GO TO 6
    TMN=TEDT
    IEDT(K)=ITEM(JJ)
    CONTINUE
    TMN=TEDT(1)
    K1=K1-1
    GO TO 8
    7
    TMN=TMN
    8
    JSAVE=1
    DO 11 J=1,K1
    IF (I.EQ.1) TEdT(J)=T(J)
LISTING OF DECK: EDIT

CARD NO.

81 IF (I,EQ.,1) IEDIT(J)=ITEMP(J)
82 IF (ITEMP(J),LE.,TMIN) GO TO 9
83 GO TO 10
84 JSAVE=J
85 TMIN=TEDT(J)
86 T(J)=TEDT(J)
87 ITEMP(J)=IEDIT(J)
88 CONTINUE
89 ISAVE(I)=IEDIT(JSAVE)
90 TSAVE(I)=TMIN
91 DO 13 I=1,N1
92 K1=ISAVE(I)
93 YSAVE(I)=Y(K1)
94 C
95 C EDIT DATA
96 C
97 N1=N-1
98 IF (N1,LT.,1.,OR.,TOLR,LE.,.0.,.0.) GO TO 31
99 K1=0
100 DO 15 I=1,N1
101 ITEMP(I)=0
102 DIFF=ABS(TSAVE(I+1)-TSAVE(I))
103 IF (DIFF,LE.,TOLR) GO TO 14
104 GO TO 15
105 14 ITEMP(I)=I
106 K1=1
107 CONTINUE
108 IF (.NOT.(K1,EQ.,0.,)) GO TO 31
109 N=1
110 I=1
111 K1=1
112 J=0
113 IF (I,GT.,N1) GO TO 27
114 IF (ITEMP(I),NE.,0.,) GO TO 17
115 IEDIT(N)=ISAVE(I)
116 N=N+1
117 I=I+1
118 IEDIT(N)=ISAVE(I)
119 GO TO 16
120 17 I=I+1
LISTING OF DECK: EDIT

CARD NO.

121    J=J+1
       IF (I1.EQ,IN) GO TO 21
       IF (ITEMP(I1).EQ.0) GO TO 18
       GO TO 17

125.   18    K2=K1+J
       IF (K1.EQ.1) GO TO 19
       I1=K1-1
       I2=K2+1
       GO TO 23

130.   19    I1=K2
       I1=I1+1
       I2=I1+1
       IF (I2.GT,IN) IERR=1
       IF (I2.GT,IN) GO TO 29

135.   20    DIFF=ABS(TSAVE(I2)-TSAVE(I1))
       IF (DIFF.LE.TOLR) GO TO 20
       GO TO 23

140.   21    I2=K1
       I2=I2-1
       I1=I2-1
       IF (I1.LE.T1) IERR=2
       IF (I1.LE.T1) GO TO 29
       DIFF=ABS(TSAVE(I2)-TSAVE(I1))
       IF (DIFF.LE.TOLR) GO TO 22

145.   22    K2=K1+J
       IF (I1.LE.T1) IERR=3
       IF (I1.LE.T1) GO TO 29
       IF (I2.GT,IN) IERR=4
       IF (I2.GT,IN) GO TO 29

150.   23    IEDT(K)=TSAVE(K1)
       T(1)=YSAVE(I1)*TSAVE(I2)-YSAVE(I2)*TSAVE(I1)
       T(2)=YSAVE(I2)-YSAVE(I1)
       T(3)=TSAVE(I2)-TSAVE(I1)
       TMIN=ABS(YSAVE(K1)-(T(1)+T(2)+TSAVE(K1))/T(3))

155.   24    K1=K1+1
       DO 25 K=K1,K2
       DIFF=ABS(YSAVE(K)-(T(1)+T(2)+TSAVE(K))/T(3))
       IF (DIFF.LE.TMIN) GO TO 24
       GO TO 25

       TMIN=DIFF

PAGE 4
LISTING OF DECK: EDIT

CARD NO.

161 25 IEDT(N)*ISAVE(K)
    CONTINUE
    IF (K2.EQ.IN) GO TO 27
    IF (I.EQ.N1) GO TO 26
    N=N+1
    I=I+1
    GO TO 16
    N=N+1
    IEDT(N)=ISAVE(IN)
    DO 28 K=1,N
    K1=IEDT(K)
    TENT(K)=TIN(K1)
    GO TO 33
    IF (NSSTORE.EQ.O) PRINT 40, IRUN, TOLR, IERR
    DO 30 I=1,N
    K1=ISAVE(I)
    CALL PNTPLT (X(K1), Y(K1), NSYM, SIZE)
    IF (NSSTORE.EQ.O) PRINT 40, I, K1, TSAVE(I), X(K1), Y(K1)
    CONTINUE
    RETURN
    DO 32 I=1,N
    K1=ISAVE(I)
    TEDIT(I)=TIN(K1)
    IEDT(I)=K1
    WRITE (7) IRUN, IN, N, TOLR
    WRITE (7) (TIN(I), I=1,N)
    WRITE (7) (ISAVE(I), TSAVE(I), I=1,N)
    WRITE (7) (TEDIT(I), TEDIT(I), I=1,N)
    C
    C     WRITE EDITED DATA ON TAPE 7 FOR USE IN SUBROUTINE PRTEDT
    C
    33 IF (NSSTORE.EQ.O) GO TO 34
    WRITE (7) IRUN, IN, N, TOLR
    WRITE (7) (TIN(I), I=1,N)
    WRITE (7) (ISAVE(I), TSAVE(I), I=1,N)
    WRITE (7) (TEDIT(I), TEDIT(I), I=1,N)
    C
    C     REORDER T, X, AND Y FOR PLOTTING
    C
    34 DO 35 I=1,N
   35 T(I)=TEDIT(I)
    DO 36 I=1,N
    TSAVE(I)=X(I)
    TEDIT(I)=Y(I)
LISTING OF DECK: EDIT

CARD NO.

201 DO 37 I=1,IN
   K=IEDT(I)
   X(I)=TSAVE(K)
   Y(I)=TEDT(K)
37 CONTINUE

205 C PLCT DATA POINTS NOT EDITED
   C IF (N.EQ.IN) RETURN
   DO 39 I=1,IN
   K1=0
210   DO 38 J=1,N
   IF (IEDT(J).EQ.1) K1=1
38   CONTINUE
   IF (K1.EQ.0) CALL PHTPLT (TSAVE(I),TECT(I),NSYM,ISTIZE)
   IF ((K1.EQ.0).AND.(NSTORE.EQ.0)) WRITE (7) I,TIN(I)
39   CONTINUE

220 C RETURN TO CALLING PROGRAM
   C RETURN
C
40 FORMAT (/2X,194ATTEMPT TO EDIT RUN,1X,I7,1X,29HFAILED FOR AN EDIT TO 1LERANCE=,F8.3,10X,9HERROR NO.,I3//4X,1HI,4X,2HIS,6X,1HT,9X,1HX,9X, 21HY/)

225 C
41 FORMAT (2I5,3F10.4)
END
SUBROUTINE PRTEDT

ROUTINE TO PRINT EDITED DATA STORED ON TAPE 7

CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AAB 1983

WORK ARRAYS
COMMON /WORK/ TIN(50),ISAVE(50),TSAVE(50),IEDITOR(50),EDITOR(50),TURN(50)

END FILE 7
REWORD 7
PRINT 7
READ (7) IRUN,TIN,N,TOLR
IF (EOF(7)) 6,2
NR=IN-N
READ (7) (TIN(I),I=1,N)
READ (7) (ISAVE(I),TSAVE(I),I=1,N)
READ (7) (EDITOR(I),EDITOR(I),I=1,N)
IF (NR.LE.0) GO TO 4
DO 3 I=1,NR
READ (7) IUN(I),TURN(I)
PRINT 8,IRUN,TOLR
DO 5 I=1,N
IF ((I.LE.N),AND.(I.LT.N)) PRINT 9, I,TIN(I),ISAVE(I),TSAVE(I),IE
10 I=I+1
10 I=I+1
10 I=I+1
10 I=I+1
CONTINUE
GO TO 1
RETURN

FORMAT (1H1,10X,3H THE FOLLOWING RUNS HAVE BEEN EDITED)
FORMAT (/1X,9H RUN NO. = ,7,10X,11H TOLERANCE = ,F8.3,7/9X,5H INPUT,19
1X,6H SORTED,19X,6H EDITED,20X,7H DELETE,7F4(1X,9H INPUT NO. = ,3X,7H-VAL
2UE5X))
FORMAT (4(5X,T3,2X,F10.4,5X))
LISTING OF DECK: PRTEOT

CARD NO.

<table>
<thead>
<tr>
<th>Number</th>
<th>Format</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>FORMAT (3(5X, I3, 2X, F10.4, 5X))</td>
<td>PR 41</td>
</tr>
<tr>
<td>10</td>
<td>FORMAT (2(5X, I3, 2X, F10.4, 5X), 30X, I3, 2X, F10.4)</td>
<td>PR 42</td>
</tr>
<tr>
<td>11</td>
<td>FORMAT (2(5X, I3, 2X, F10.4, 5X))</td>
<td>PR 43</td>
</tr>
<tr>
<td>12</td>
<td>END</td>
<td>PR 44</td>
</tr>
</tbody>
</table>

64
LISTING OF DECK: ZEROLN

CARD NO.

1 SUBROUTINE ZEROLN (X0,Y0,DIST,LINE)

2 ROUTINE TO DRAW ZERO LINES

3 CODED BY -- HARRY L. MORGAN NASA/LARC/TAN/AAR 1983

5 PARAMETER DEFINITION

6 X0,Y0 - STARTING LOCATION OF ZERO LINE

8 DIST - LENGTH OF ZEROLINE

10 LINE - AXIS OPTION

12 LINE=1, ZERO LINE PARALLEL TO X-AXIS

14 LINE=2, ZERO LINE PARALLEL TO Y-AXIS

15 NOTE - THICKNESS OF ZERO LINE = 2*D

19 D=0.014

20 CALL CALPLT (X0,Y0,3)

21 IF (LINE.EQ.2) GO TO 1

25 X=X0+DIST

26 CALL CALPLT (X,Y0,2)

27 Y=Y0+D

30 CALL CALPLT (X,Y,2)

34 CALL CALPLT (X0,Y,2)

38 CALL CALPLT (X,Y,2)

41 GO TO 1

45 Y=Y0+DIST

50 CALL CALPLT (X0,Y,2)

54 X=X0+D

58 CALL CALPLT (X,Y,2)

62 CALL CALPLT (X0,Y,2)

66 CALL CALPLT (X,Y,2)

70 CALL CALPLT (X0,Y,3)

74 RFTRN

78 END
SUBROUTINE CURPLT (T,X,Y,N,IVAR,NSYM,ISIZE,IOP,IRUN,TENSION)
ROUTINE TO PLOT AND FAIR DATA
Coded by -- Harry L. Morgan
NASA/LARC/TAD/AAB 1983

PARAMETER DEFINITION
T - ARRAY CONTAINING INDEPENDENT VARIABLE
X - ARRAY CONTAINING X VARIABLE
Y - ARRAY CONTAINING Y VARIABLE
N - NUMBER OF POINTS TO BE PLOTTED
IVAR - VARIABLE CODE IVAR=0 IF X AND T ARE THE SAME
        IVAR=1 IF X AND T ARE DIFFERENT
NSYM - SYMBOL NUMBER
ISIZE - SYMBOL SIZE
IOP - PLOTTING OPTION IOP=0 PLOT SYMBOLS ONLY
        IOP=1 PLOT CURVE AND SYMBOLS
IRUN - RUN NUMBER
TENSION - SPLINE TENSION

T, X, AND Y MUST BE PROPERLY DIMENSIONED IN CALLING ROUTINE
DIMENSION T(N), X(N), Y(N), H(3)

WORK ARRAYS
COMMON /WORK/ DS1(105),DS2(105),DUMX(105),DUMY(105)
COMMON /PLT/ MX(50),MY(50),A(50)

REAL MX,MY

ROUTE CONSTANTS
DATA NH/101/,YNT/100/,EPS/0.0001/,NTA/100/,HF/02.,03..04/

SELECT PLOTTING OPTION
IF (IOP.EQ.0.0.OR.NLT.3) GO TO 1
    GO TO 3

PLOT SYMBOLS ONLY
DO 2 I=1,N
    CALL PNTPLT (X(I),Y(I),NSYM,ISIZE)
2 CONTINUE
RETURN
LISTING OF DECK: CURPLT

CARD NO.

41 C PLOT CURVE AND SYMBOL

45 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
50 CONTINUE

5  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),Y(I),I=1,N)
C PLOT SYMBOLS ONLY IF X OR T NOT STRICTLY INCREASING
5 GO TO 1
C FIT SPLINE CURVE THROUGH DATA POINTS
7 IF (IVAR.EQ.0) CALL CUBSPL (X,Y,N,M,TENSION,TENS,A)
   IF (IVAR.EQ.1) CALL CUBSPL (T,X,N,M,TENSION,TENS,A)
   IF (IVAR.EQ.1) CALL CUBSPL (T,Y,N,M,TENSION,TENS,A)
C PLOT FIRST POINT
65 CALL PNTPLT (X(I),Y(1),NSYM,ISIZE)
C PLOT AND FAIR REMAINING POINTS
66 DO 25 I=1,N
   NN=N-1
   DO 25 I=1,NN
   COMPUTE STRAIGHT LINE DISTANCE BETWEEN TWO POINTS. IF DISTANCE
   LESS THAN SYMBOL DIAMETER, PLOT POINTS ONLY.
70 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)
75 DSS1=SQRT(XS1*XS1+YS1*YS1)
   X1=--X1
   Y1=--Y1
   T1=ATANF(Y1,X1)
   CALL SYMBS (NSYM,ISIZE,XS2,YS2,T1)
80 DSS2=SQRT(XS2*X2+YS2*YS2)
LISTING OF DECK: CURPLT

CARD NO.

81 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(YNT*(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)

85 DUMX(I)=X(I)
     DUMY(I)=Y(I)
     DSI(I)=0.0
     IF (IVAR.EQ.0) DA=X(I)
     IF (IVAR.EQ.1) DA=T(I)

90 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)

95 GO TO 9

100 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)

105 DO 10 J=1,NT
     K=NT+1-J
     DS1(J)=SQRT((DUMX(K)-X(I+1))**2+(DUMY(K)-Y(I+1))**2)
     FIND X AND Y LOCATION WHERE SYMBOL AND CURVE INTERSECT

110 DS2(J)=SQRT((DUMX(K)-X(I+1))**2+(DUMY(K)-Y(I+1))**2)
     DELTA=H(I*SIZE)
     IF (IVAR.EQ.0) DA=X(I)
     IF (IVAR.EQ.1) DA=T(I)

115 DO 11 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)

120 GO TO 12
121  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)  
     DS=SORT(X1+X1+Y1+Y1)  
125.  T1=ATANF(Y1+X1)  
     CALL SYMBS (NSYM,ISIZE,XS1,YS1,T1)  
     DSS1=SORT(XS1*YS1+YS1*YS1)  
     IF (ABS(DS-DSS1)*LE.EPS) GO TO 15  
     IF (DS.GT.DSS1) GO TO 13  
130  GO TO 14  
13.  DA=DA-DELTA  
     DELTA=DELTA/2.  
14.  CONTINUE  
135.  X1=XS1+X(I)  
     YS1=YS1+Y(I)  
     DELTA=HEISIZE  
     IF (IVAR.EQ.O) DA=X(I+1)  
     IF (IVAR.EQ.1) DA=T(I+1)  
140.  CONTINUE  
145.  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  
150.  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  
155.  IF (ABS(DS-DSS2)*LE.EPS) GO TO 20  
     IF (DS.GT.DSS2) GO TO 19  
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=SORT(X1+X1+Y1+Y1)  
     T1=ATANF(Y1+X1)  
     CALL SYMBS (NSYM,ISIZE,XS2,YS2,T1)  
     DSS2=SORT(XS2*YS2+YS2*YS2)  
18.  CONTINUE  
19.  CONTINUE
LISTING OF DECK: CURPLT

CARD NO.

161 C
162 20 X22=XS2+X(I+1)
163 Y22=YS2+Y(I+1)
164 NP=1
165 NDS1=0
166 DO 21 J=2,NT
167 IF ((DS1(J),LT.,DSS2),AND.(NDS1,EQ.0)) GO TO 21
168 IF ((DS1(J),GE.,DSS1),AND.(NDS1,EQ.0)) NDS1=1
169 NP=NP+1
170 DUMX(NP)=DUMX(J)
171 DUMY(NP)=DUMY(J)
172 CONTINUE
21 C
173 PLOT CURVE BETWEEN POINTS
174 DUMX(1)=XS1
175 DUMY(1)=YS1
176 DO 22 J=1,NT
177 IF (DS2(J),LE.,DSS2) GO TO 22
178 GO TO 23
179 180 22 NP=NP-1
181 23 NP=NP+1
182 DUMX(NP)=XS2
183 DUMY(NP)=YS2
184 DUMX(NP+1)=DUMX(NP+1)=0.0
185 DUMY(NP+2)=DUMY(NP+2)=1.0
186 CALL LINE (DUMX,DUMY,NP,1,0,0,0)
187 C
188 PLOT SYMBOL AT POINT I+1
189 24 CALL PNTPLT (X(I+1),Y(I+1),NSYM,ISIZF)
190 C
191 ADVANCE TO NEXT POINT
192 193 CONTINUE
194 25 C
195 C
196 RETURN TO CALLING PROGRAM
197 C
198 RETURN
199 C
200 FORMAT (1H1//5X,36HX IS NOT STRICTLY INCREASING FOR RUN,17/16X,1HX
201 1,13X,1HY//)
202 1,13X,13X,1HY/)
203 C
204 RETURN
205 C
206 FORMAT (1H1//5X,36HX IS NOT STRICTLY INCREASING FOR RUN,17/16X,1HT
207 1,13X,1HX,13X,1HY/)
LISTING OF DECK: CURPLT

CARD NO.

201  29  FORMAT (5X,3F15.4)
END
FUNCTION ATANF (DY,DX)

ROUTINE TO COMPUTE ARCTANGENT FOR ANGLES FROM 0 TO 360 DEGREES

CODED BY -- HARRY L. MORGAN

NASA/LARC/TAD/AAR 1983

C IF (DX.EQ.0.0) GO TO 1
C ATANF=ABS(DY/DX)
C ATANF=ATAN(ATANF)*57.2957795131
C IF ((DX.GT.0.0).AND.(DY.LT.0.0)) ATANF=360.-ATANF
C IF ((DX.LT.0.0).AND.(DY.GT.0.0)) ATANF=180.-ATANF
C IF ((DX.LT.0.0).AND.(DY.LT.0.0)) ATANF=180.+ATANF
C RETURN
C IF (DY.LT.0.0) ATANF=270.
C IF (DY.GE.0.0) ATANF=90.
C RETURN
END
LISTING OF DECK: SYMBS

CARD NO.

1

SUBROUTINE SYMBS (NO,IS,X,Y,T)

C

ROUTINE TO COMPUTE THE X AND Y INTERCEPTS OF A RADIAL LINE DRAWN
FROM THE CENTER OF A GIVEN SYMBOL WITH THE SIDE OF THAT SYMBOL

C

CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AA8 1983

C

PARAMETER DEFINITION

NO - SYMBOL NUMBER

IS - SYMBOL SIZE 1-SMALL 2-MEDIUM 3-LARGE

X AND Y - INTERCEPT OF SYMBOL AND RADIAL DRAWN FROM SYMBOL

CENTER AT ANGLE T.

T - ANGLE OF RADIAL DRAWN FROM SYMBOL CENTER

DIMENSION SCALE(3)

DATA RAD/57.2957795131/, PI/3.141592654/

DATA DA/1.4142135624/, R/1.7320508076/, SCALE/13,16,19/

DATA T1/213.6900675260/, T2/326.3099324740/, T3/116.5650511771/, T4/3

133.4349488229/, T7/11.3099324740/, T8/168.6900675260/, T9/218.6598082

2541/ , S1/321.3401917459/, S2/185.710593175/, S3/354.2894068625/, S4/1

31,3099324740/, S5/168.6900675260/, S6/218.6598082541/, S7/321.3401917

C

IF ((NO.EQ.1) .OR. (NO.EQ.11)) GO TO 1

IF ((NO.EQ.2) .OR. (NO.EQ.12)) GO TO 2

IF ((NO.EQ.3) .OR. (NO.EQ.13)) GO TO 8

IF ((NO.EQ.4) .OR. (NO.EQ.14)) GO TO 13

IF ((NO.EQ.5) .OR. (NO.EQ.15)) GO TO 17

IF ((NO.EQ.6) .OR. (NO.EQ.16)) GO TO 21

IF ((NO.EQ.7) .OR. (NO.EQ.17)) GO TO 26

IF ((NO.EQ.8) .OR. (NO.EQ.18)) GO TO 31

IF ((NO.EQ.9) .OR. (NO.EQ.19)) GO TO 35

IF ((NO.EQ.10) .OR. (NO.EQ.20)) GO TO 40

IF ((NO.EQ.21) .OR. (NO.EQ.22)) GO TO 45

C

SYMBOL NUMBER 1 OR 11

C

X=SCALE(IS)*.5525*COS(T/RAD)

40

Y=SCALE(IS)*.5525*SIN(T/RAD)
LISTING OF DECK: SYMS

RETURN

SYMBOL NUMBER 2 OR 12

45. IF (T.GE.45.) AND (T.LT.45.) GO TO 3
   IF (T.GE.45.) 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

50. X=SCALE(IS)/2.
    Y=X*TAN(T/RAD)
    RETURN

SYMBOL NUMBER 3 OR 13

55. IF (T.EQ.90.) X=0.0
    Y=SCE(S)2.
    RETURN

60. X=-SCALE(IS)/2.
    Y=X*TAN(T/RAD)
    RETURN

65. Y=-SCALE(IS)/2.
    RETURN

70. IF (T.GE.90.) 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

75. X=SCALE(IS)*DA/2./SQN(T/RAD)+1.
    Y=-X+SCALE(IS)*DA/2.
    RETURN

80. RETURN
| CARD NO. | 81  | 11  | X=SCALE(IS)*DA/2./(TAN(T/RAD)+1.) | SY 81 |
|         |     |     | Y=X-SCALE(IS)*DA/2.             | SY 82 |
|         |     |     | RETURN                         | SY 83 |
|         | 85  | 12  | IF (T.EQ.270.) X=0.0          | SY 84 |
|         |     |     | IF (T.NE.270.) X=-SCALE(IS)*DA/2./(TAN(T/RAD)-1.) | SY 85 |
|         |     |     | Y=X-SCALE(IS)*DA/2.             | SY 86 |
|         |     |     | RETURN                         | SY 87 |
|         |     |     | C                              | SY 88 |
|         |     |     | C                              | SY 89 |
|         |     |     | SYMBOL NUMBER 4 OR 14         | SY 90 |
|         | 90  | 13  | IF ((T.GE.0.)*AND.((T.LT.90.)) GO TO 14 | SY 91 |
|         |     |     | IF ((T.GE.90.)*AND.((T.LT.11)) GO TO 15 | SY 92 |
|         |     |     | IF ((T.GE.11).AND.((T.LT.22)) GO TO 16 | SY 93 |
|         |     |     | X=(2.*/3.)*SCALE(IS)*1.105/(TAN(T/RAD)+2.*) | SY 94 |
|         |     |     | Y=-2.*X+2.*SCALE(IS)*1.105/3.   | SY 95 |
|         |     |     | RETURN                         | SY 96 |
|         |     |     | 15 IF (T.EQ.90.) Y=0.0         | SY 97 |
|         |     |     | IF (T.NE.90.) X=(2.*/3.)*SCALE(IS)*1.105/(TAN(T/RAD)-2.*) | SY 98 |
|         |     |     | Y=2.*X+2.*SCALE(IS)*1.105/3.   | SY 99 |
|         |     |     | RETURN                         | SY 100|
|         | 100 | 16  | Y=SCALE(IS)*1.105/3.           | SY 101|
|         |     |     | IF (T.NE.270.) X=0.0          | SY 102|
|         |     |     | IF (T.NE.270.) X=Y/TAN(T/RAD)  | SY 103|
|         |     |     | RETURN                         | SY 104|
|         |     |     | C                              | SY 105|
|         | 105 | 17  | IF ((T.GE.0.)*AND.((T.LT.3)) GO TO 18 | SY 106|
|         |     |     | IF ((T.GE.3.)*AND.((T.LT.25)) GO TO 19 | SY 107|
|         |     |     | IF ((T.GE.25.)*AND.((T.LT.4)) GO TO 20 | SY 108|
|         |     |     | IF (T.EQ.90.) X=0.0          | SY 109|
|         |     |     | IF (T.NE.90.) X=SCALE(IS)*1.22222/3./(TAN(T/RAD)+1.) | SY 110|
|         |     |     | Y=X-SCALE(IS)*1.22222/3.       | SY 111|
|         |     |     | RETURN                         | SY 112|
|         |     |     | 115 X=SCALE(IS)*1.22222/3.     | SY 113|
|         |     |     | Y=X*TAN(T/RAD)                 | SY 114|
|         |     |     | RETURN                         | SY 115|
|         |     |     | 20 Y=SCALE(IS)*1.22222/3.      | SY 116|
|         |     |     | IF (T.EQ.270.) X=0.0          | SY 117|
|         |     |     | IF (T.NE.270.) X=Y/TAN(T/RAD)  | SY 118|
|         |     |     | RETURN                         | SY 119|
|         |     |     | 120 Y=SCALE(IS)*1.22222/3.      | SY 120|
LISTING OF DECK: SYMAS

CARD NO.

121
C
125
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
130.
IF ((T.GE.0.0).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.90.0) GO TO 23
BB=-2.*A*(1.+TAN(T/RAD))
AA=TAN(T/RAD)**2+1.
CC=2.*AA-(SCALE(IS)*1.22222)**2
X=SORT(BB-BB-4.*AA*CC)/(2.*AA)
135.
IF ((T.GE.0.0).AND.((T.LT.90.0)) X=X+BB/(2.*AA)
IF ((T.GE.90.0).AND.((T.LT.180.)) X=-X+BB/(2.*AA)
140.
IF ((T.GE.270.0).AND.((T.LE.360.)) X=X+BB/(2.*AA)
Y=X*TAN(T/RAD)
RETURN
23 X=0.0
Y=-A*SORT((SCALE(IS)*1.22222)**2-A*A)
145.
RETURN
24 X=-A
Y=X*TAN(T/RAD)
RETURN
25 Y=-A
140.
IF (T.EQ.270.0) X=0.0
IF (T.*NE.270.0) X=Y*TAN(T/RAD)
RETURN
C
C
155
C
155.
26 IF ((T.GE.0.0).AND.((T.LT.T7)) GO TO 27
IF ((T.GE.T7).AND.((T.LT.T81)) GO TO 22
IF ((T.GE.T8).AND.((T.LT.T9)) GO TO 29
IF ((T.GE.T9).AND.((T.LT.S1)) GO TO 30
160.
27 X=SCALE(IS)/2.
LISTING OF DECK: SYMS

CARD NO.

161

Y=X*TAN(T/RAD)
RETURN

SY 161

28

X=SCALE(IS)*.1*SIN(T/RAD)*COS(T/RAD)+SCALE(IS)*COS(T/RAD)*SORT(.25
1-(.1*COS(T/RAD))*2)

SY 163

165.

Y=X*TAN(T/RAD)
RETURN

SY 165

29

X=SCALE(IS)/2.
Y=X*TAN(T/RAD)
RETURN

SY 166

170

Y=SCALE(IS)*.4
IF (T.EQ.270.0) X=0.0
IF (T.EQ.270.0) X=Y/TAN(T/RAD)
RETURN

SY 170

C

SYMBOL NUMBER 8 OR 18

C

31

IF ((T.GE.0.0)*AND.((T.LE.1.52)) GO TO 32
IF ((T.GE.1.52)*AND.((T.LE.270.0)) GO TO 33
IF ((T.GT.270.0)*AND.((T.LE.359.0)) GO TO 34

SY 171

180.

X=-05*SCALE(IS)*SIN(T/RAD)*COS(T/RAD)+SCALE(IS)*COS(T/RAD)*SORT(.125-(.05*COS(T/RAD))*2)

SY 180

185

Y=X*TAN(T/RAD)
RETURN

SY 182

33

IF (T.EQ.270.0) X=0.0
IF (T.EQ.270.0) Y=SCALE(IS)*.55
IF (T.EQ.270.0) X=SCALE(IS)*.55/(TAN(T/RAD)+1.0)
IF (T.EQ.270.0) Y=X*TAN(T/RAD)
RETURN

SY 184

190

Y=X*TAN(T/RAD)
RETURN

SY 188

C

SYMBOL NUMBER 9 OR 19

C

195

IF ((T.GE.0.0)*AND.((T.LE.90.0)) GO TO 36
IF ((T.GE.90.0)*AND.((T.LE.180.0)) GO TO 37
IF ((T.GE.180.0)*AND.((T.LE.270.0)) GO TO 38
IF ((T.GE.270.0)*AND.((T.LE.360.0)) GO TO 39

SY 197

36

X=SCALE(IS)*R/2./((TAN(T/RAD)+R)

SY 199

200

Y=X*TAN(T/RAD)

SY 200
LISTING OF DECK: SYMS

CARD NO.                  RETURN
201  37 IF (T.EQ.90.)  Y=0.0
     IF (T.NE.90.)  X=SCALE(IS)*R/2./TAN(T/RAD)-R
     IF (T.EQ.90.)  Y=SCALE(IS)*R/2.
     IF (T.NE.90.)  Y=X*TAN(T/RAD)
     RETURN

205.  38 X=SCALE(IS)*R/2./TAN(T/RAD)+R
     Y=X*TAN(T/RAD)
     RETURN

210.  39 IF (T.EQ.270.)  X=0.0
     IF (T.EQ.270.)  Y=-SCALE(IS)*R/2.
     IF (T.NE.270.)  X=-SCALE(IS)*R/2./TAN(T/RAD)-R
     IF (T.NE.270.)  Y=X*TAN(T/RAD)
     RETURN

215 C  SYMBOL NUMBER 10 OR 20
     C
     40 IF ((T.GE.0.0).AND.(T.LT.S4)) GO TO 41
     IF ((T.GE.S4).AND.(T.LT.S5)) GO TO 42
     IF ((T.GE.S5).AND.(T.LT.S6)) GO TO 43
     IF ((T.GE.S6).AND.(T.LT.S7)) GO TO 44
     X=SCALE(IS)/2.
     Y=X*TAN(T/RAD)
     RETURN

220.  41 IF (T.LT.90.)  X=SCALE(IS)*.6/(TAN(T/RAD)+1.)
     IF (T.EQ.90.)  X=0.0
     IF (T.GT.90.)  X=SCALE(IS)*.6/(TAN(T/RAD)-1.)
     IF (T.EQ.90.)  Y=SCALE(IS)*.6
     IF (T.NE.90.)  Y=X*TAN(T/RAD)
     RETURN

224.  43 X=SCALE(IS)/2.
     Y=X*TAN(T/RAD)
     RETURN

225.  44 Y=-SCALE(IS)*.4

229 C  SYMBOL NUMBER 21 OR 22

230 C

235 C  RETURN

C
LISTING OF DECK: SYMBS
CARD NO.

241  45  X = 1 * SCALE(IS) * COS(T/RAD)
     Y = 1 * SCALE(IS) * SIN(T/RAD)
RETURN
END

SY 241
SY 242
SY 243
SY 244-
SUBROUTINE CUBSPL (X,Y,N,YPP,TENSION,TENS,A) CB 1
THIS SUBROUTINE FITS A CUBIC SPLINE TO A SET OF Y VS X INPUT
POINTS CB 2
CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AA9 1983 CB 3
IN CALLING PROGRAM DIMENSION X, Y, YPP, AND A BY N CB 4
DIMENSION X(N), Y(N), YPP(N), A(N) CB 5
COMPUTE TENSION PARAMETER CB 6
TENS=TENSION*FLOAT(N-1)/(X(N)-X(1)) CB 7
IF (TENS.LT.0.0) TENS=0.0 CB 8
COMPUTE SECOND DERIVATIVE AT END POINTS BY FITTING CB 9
Y=A*X**2+B*X+C TO THE LAST THREE POINTS AND SOLVE FOR A. CB 10
SECOND DERIVATIVE AT END POINT IS THEN EQUAL TO 2*A CB 11
H1=X(2)-X(3) CB 12
H2=X(3)-X(1) CB 13
H3=X(1)-X(2) CB 14
YPP(1)=2.*((Y(1)*H1+Y(2)*H2+Y(3)*H3)/(H1*X(1)**2+H2*X(2)**2+H3*X(3)) **2) CB 15
H1=X(N-1)-X(N) CB 16
H2=X(N)-X(N-2) CB 17
H3=X(N-2)-X(N-1) CB 18
YPP(N)=2.*((Y(N-2)*H1+Y(N-1)*H2+Y(N)*H3)/(H1*X(N-2)**2+H2*X(N-1)**2+H3*X(N)**2) **2) CB 19
PERFORM FORWARD ELIMINATION CB 20
IF (TENS.NE.0.0) GO TO 1 CB 21
CON=6. CB 22
E=1 CB 23
GO TO 2 CB 24
1 CON=TENS*TENS CB 25
41  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

45  A(I)=0.0  CB  45
   DD 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

50  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

60  YPP(I)=(C-E*YPP(I-1))/D  CB  60
   E=F  CB  61

5  H1=H2  CB  62

C C PERFORM BACK SUBSTITUTION  CB  63
C

65  C
   J=N  CB  65
   DD 6 I=2,N1  CB  66
   J=J-1  CB  67

6  YPP(J)=YPP(J)-A(J)*YPP(J+1)  CB  68

70  C C RETURN TO CALLING PROGRAM  CB  69
C
C RETURN  CB  70
C
END  CB  71

FUNCTION FUNC (X, X1, X2, Y1, Y2, YPP1, YPP2, TENS)

ROUTINE TO COMPUTE Y-VALUE AT A GIVEN X-VALUE ALONG SPLINE CURVE

CODED BY -- HARRY L. MORGAN    NASA/LARC/TAD/AAB    1983

1. FUNCTION FUNC (X, X1, X2, Y1, Y2, YPP1, YPP2, TENS)

2. IF (TENS.EQ.0.0) GO TO 1

3. F1= YPP1/(TENS*TENS)

4. F2= YPP2/(TENS*TENS)

5. FUNC=(F1*FINH(TENS*DX1)+F2*FINH(TENS*DX2))/FINH(TENS*DH)+((Y1-F1)*

6. 10X1+(Y2-F2)*DX2)/DH

7. RETURN

8. IF (TENS.EQ.0.0) GO TO 1

9. FUNC=YPP1*DX1**3/(6.*DH)+YPP2*DX2**3/(6.*DH)+((Y1-YPP1*DH/DH/6.)*DX

10. 11/DH+(Y2-YPP2*DH/DH/6.)*DX2/DH

11. RETURN

12. END
LISTING OF DECK: DASHLN

CARD NO.

1 SUBROUTINE DASHLN (T, X, Y, N, NSYM, IOP, ISIZE, IRUN, TENSION) DH 1
2 C ROUTINE TO FAIR DATA WITH DASH LINES DH 2
3 C CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AAB 1983 DH 3
4 C PARAMETER DEFINITION DH 4
5 C T = ARRAY CONTAINING INDEPENDENT VARIABLE DH 7
6 C X = ARRAY CONTAINING X VARIABLE DH 8
7 C Y = ARRAY CONTAINING Y VARIABLE DH 9
8 C N = NUMBER OF POINTS TO BE PLOTTED DH 10
9 C NSYM = LINE OR SYMBOL NUMBER DH 11
10 C IOP = PLOTTING OPTION IOP=0 PLOT SYMBOLS ONLY DH 12
11 C IOP=1 PLOT DASH LINE DH 13
12 C ISIZE = SYMBOL SIZE DH 14
13 C IRUN = RUN NUMBER DH 15
14 C TENSION = SPLINE TENSION DH 16
15 C T, X, AND Y MUST BE PROPERLY DIMENSIONED IN CALLING ROUTINE DH 17
16 C DIMENSION T(1), X(1), Y(1), NL(22), NS(22) DH 18
17 C WORK ARRAYS DH 19
18 C COMMON /WORK/ S(166), YI(166), YI(166) DH 20
19 C COMMON /PLT/ MX(50), MY(50), A(50) DH 21
20 C REAL MX, MY DH 22
21 C ROUTINE CONSTANTS DH 23
22 C NL = NUMBER OF LONG DASHES AND NS = NUMBER OF SHORT DASHES DH 24
23 C DATA NL/1,1,1,1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4,4,4,5,5,5,5/ DH 25
24 C DATA NS/0,1,1,1,2,2,3,3,4,4,4,4,1,2,2,3,3,4,1,2,3,3,4,5/ DH 26
25 C ROUTINE CONSTANTS DH 27
26 C NP=166 DH 28
27 C EPS=.02 DH 29
28 C SL=.3 DH 30
29 C IF (NSYM*EO.2) SL=.2 DH 31
30 C SS=.1 DH 32
31 C SP=.1 DH 33
32 C SELECT PLOTTING OPTION DH 34
33 C
C IF (IDP.EQ.0,.NOT.N.LE.3) GO TO 1
   GO TO 3
C PLOT SYMBOLS ONLY
1 DO 2 I=1,N
2 CALL PNTPLT (X(I),Y(I),NSYM,ISIZE)
   RETURN
C PLOT DASH LINE
C CHECK TO SEE IF T IS STRICTLY INCREASING
3 DO 4 I=2,N
4 IF (T(I).LT.T(I-1)) GO TO 5
   CONTINUE
   GO TO 6
C PRINT ERROR MESSAGE IF T IS NOT STRICTLY INCREASING
   PRINT 15, JUN
   PRINT 16, (T(I),X(I),Y(I),I=1,N)
5 IF T IS NOT STRICTLY INCREASING, PLOT SYMBOLS ONLY
   GO TO 1
C FIT SPLINE CURVE THROUGH DATA POINTS
6 CALL CSBSPL (T,X,N,MX,TENSION,TENS,A)
   CALL CSBSPL (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)
70 S(1)=0.
   M=2
   DO 9 I=1,N
      IF (TI.LT.T(I)) TI=T(I)
      IF (TI.GT.T(N)) TI=T(N)
   9 DO 7 J=M,N
      K=J-1
      IF (TI.GE.T(K).AND.TI.LE.T(J)) GO TO 8
      CONTINUE
   7 CONTINUE
   M=K
80 IF (M.LE.1) M=2
LISTING OF DECK: DASHLN

CARD NO.

81

X1(I) = FUNC(T1, T(K), T(K+1), X(K), X(K+1), MX(K), MX(K+1), TFNS)
Y1(I) = FUNC(T1, T(K), T(K+1), Y(K), Y(K+1), MY(K), MY(K+1), TENS)
IF (I.EQ.1 OR NSYM.EQ.1) GO TO 9
S(I)=S(I-1)+SORT((XI(I)-XI(I-1))**2+(YI(I)-YI(I-1))**2)

85

TI=TI+DT
ST=ST(5)
C IF NSYM=1, PLOT CONTINUOUS LINE
C IF (NSYM.EQ.1) GO TO 13
C COMPUTE LENGTH OF LONG AND SHORT DASH AND SPACE BETWEEN EACH

90

NLG=NL(NSYM)
NSG=NS(NSYM)
DT=FLOAT(NLG)*SL+FLOAT(NSG)*SS+FLOAT(NLG+NSG)*SP
NLT=IFIX((ST+SP)/DT)
IF (NLT.LE.0) NLT=1

95

DTN=(ST+SP)/FLOAT(NLT)
TI=DTN/DT
SL=SL*TI
SS=SS*TI
SP=SP*TI

100

C POSITION PFN AT START OF LINE
SI=0
CALL CALPLT (XI(1), YI(1), 3)
JSTART=1

105

C PLOT NL LONG DASHES
DO 11 I=1, NLG
SI=SI+SL
IF (SI.GE.ST) SI=ST
CALL LINEAR (SI, XX, YY, NP, S, XI, YI, JSTART)
CALL CALPLT (XX, YY, 2)

110

SI=SI+SP
IF (SI.GE.ST) GO TO 14
CALL LINEAR (SI, XX, YY, NP, S, XI, YI, JSTART)
CALL CALPLT (XX, YY, 3)
CONTINUE

115

C PLOT NS SHORT DASHES
IF (NSG.EQ.0) GO TO 10
DO 12 I=1, NSG
SI=SI+SS
IF (SI.GE.ST) SI=ST
CALL LINEAR (SI, XX, YY, NP, S, XI, YI, JSTART)
LISTING OF DECK: DASHLN

CARD NO.

121 CALL CALPLT (XX, YY, 2)
   SI = SI + SP
   IF (SI GE ST) GO TO 14
   CALL LINEAR (SI, XX, YY, NP, S, XI, YI, JSTART)

125 CALL CALPLT (XX, YY, 3)

12 C CONTINUE
   C CONTINUE PLOTTING DASH LINE
   GO TO 10
   C PLOT CONTINUOUS LINE

130 XI(NP+1) = 0.0
   YI(NP+1) = 0.0
   XI(NP+2) = YI(NP+2) = 1.0
   CALL LINE (XI, YI, NP, 1, 0, 0, 0)

135 C C RETURN TO START OF CURVE
   C C

140 C C CALL CALPLT (XI(1), YI(1), 3)
   C C
   C C RETURN TO CALLING PROGRAM

145 C C

15 FORMAT (1H1/5X, 36HT IS NOT STRICTLY INCREASING FOR RUN, I7/16X, 1HT
       1, 13X, 1Hx, 13X, 1HY/)

16 FORMAT (5X, 3F15.4)

END
SUBROUTINE LINEAR (TINT,XINT,YINT,N,T,X,Y,JSTART)

LINEAR INTERPOLATION METHOD

CODED BY -- HARRY L. MORGAN

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
1 CONTINUE
IF (T(J)-TINT)'1,J=2
JEND=J

2 DT=(TINT-T(J-1))/(T(J)-T(J-1))
XINT=Y(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
LISTING OF DECK: LSQPLT

CARD NO.

1

SUBROUTINE LSQPLT (T,X,Y,N,NSYM,ISIZE,IOP,IRUN,TENSION,ILSQ,NDOL,D
1F)

5

ROUTINE TO FAIR DATA WITH LEAST SQUARES CURVE FIT

CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AAR 1983

PARAMETER DEFINITION

T = ARRAY CONTAINING INDEPENDENT VARIABLE

X = ARRAY CONTAINING X VARIABLE

Y = ARRAY CONTAINING Y VARIABLE

N = NUMBER OF POINTS TO BE PLOTTED

NSYM = LINE AND SYMBOL NUMBER

ISIZE = SYMBOL SIZE

IOP = PLOTTING OPTION

IOP=0 PLOT SYMBOLS ONLY

IOP=1 PLOT SYMBOLS AND CURVE

IRUN = RUN NUMBER

TENSION = SPLINE TENSION

ILSQ = LEAST SQUARES CURVE FIT OPTION

ILSQ=0 LEAST SQUARES POLYNOMIAL CURVE FIT

ILSQ=1 LEAST SQUARES CUBIC SPLINE FIT

NDOL = ORDER OF LEAST SQUARES POLYNOMIAL CURVE FIT

DF = STANDARD DEVIATION FOR LEAST-SQUARES CUBIC-SPLINE FIT

DIMENSION T, X, AND Y IN CALLING PROGRAM

DIMENSION T(1), X(1), Y(1)

WORK ARRAYS

DIMENSION XNEW(50), YNEW(50)

DIMENSION COEF(50,4)

COMMON /WORK/ WK(360), SDV(50), CX(11), CY(11)

ROUTINE CONSTANTS

NMAX=50

WT=1.0

IPRINT=0

SELECT PLOTTING OPTION

IF (IOP.EQ.0 .OR. N.LT.3) GO TO 1

GO TO 3
LISTING OF DECK: LSQPLT

CARD NO.

41  
C  PLOT SYMBOLS ONLY
C  
DO 2 I=1,N
1  CALL PNTPLT (X(I),Y(I),NSYM,ISIZE)
RETURN
2
C  CHECK TO SEE IF T IS STRICTLY INCREASING
C  
DO 4 I=2,N
3  IF (T(I).LT.T(I-1)) GO TO 5
IF (T(I).EQ.T(I-1)) GO TO 6
CONTINUE
4  CONTINUE
GO TO 6
5  PRINT 20, IRUN
PRINT 21, (T(I),X(I),Y(I),I=1,N)
GO TO 1
C  PLOT DATA POINTS WITH SYMBOLS
C  
DO 7 I=1,N
6  SDV(I)=WT
IF (ILSQ.EQ.1) SDV(I)=DF
CALL PNTPLT (X(I),Y(I),NSYM,ISIZE)
CONTINUE
7
C  COMPUTE LEAST SQUARES CURVE
C  
IF (ILSQ.GT.0) GO TO 10
C  COMPUTE COEFFICIENTS OF LEAST SQUARES POLYNOMIAL
C  
MPOL=NPOL
IF (MPOL.GT.N-1) MPOL=N-1
IF (MPOL.LE.0) MPOL=1
IF (MPOL.GT.10) MPOL=10
M=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
C  
EPRX=ERRY=0.0
DO 9 I=1,N
9  YNEW(I)=CX(I)

PAGE 2
LISTING OF DECK: LSQPLT

CARD NO.

81  YNEW(I)=CY(I)
    DELTA=1.0
    DO 8 J=2,M1
    DELTA=DELTA*T(I)
85  YNEW(I)=YNEW(I)+DELTA*CY(J)
    CONTINUE
     XNEW(I)=XNEW(I)+DELTA*CY(I)
    ERRX=ERRX+(X(I)-XNEW(I))**2
     ERRY=ERRY+(Y(I)-YNEW(I))**2
89  CONTINUE
     GO TO 17
90  C  COMPUTE COEFFICIENTS OF LEAST SQUARES CUBIC SPLINE
     FIT X
10  K=-1
    SM=FLOAT(N)
    CALL CSOS (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
     XNEW(I)=COEF(I,1)
    GO TO 12
11  K=I-1
    DELTA=T(I)-T(K)
105  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
    CONTINUE
     FIT Y
11  K=-1
    SM=FLOAT(N)
    CALL CSOS (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
12  K=I-1

90
LISTING OF DECK1 LSQPLT

CARD NO.

121

DELTAT(I)-T(K)
YNEW(I)=((COEF(K,4)*DELTACOE(K,9))**DELTACOE(K,2))**DELTACOE(K,1)
1
15
ERRY=ERRY+(Y(I)-YNEW(I))**2
16
CONTINUE
17
C
C
C
PL DASHED LINE THRU NEW X AND Y VALUES
TEN=TENSION
C
C
IF (ILSQ,EQ,1) TEN=0.0
CALL DASHLN (T,YNEW,YNEW,N,NSYM,ISIP,ISIZE,IRUN,TEN)
C
C
PRINT RESULTS OF LEAST SQUARES CURVE FIT
C
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 10 I=1,N
DELTAY(I)=YNEW(I)
18
PRINT 24, I,T(I),Y(I),YNEW(I),DELTAY(I),YNEW(I),SM
CONTINUE
PRINT 25, ERRY,ERRY
RETURN
C
C
PRINT ERROR MESSAGE
C
C
PRINT 26, IERR,IRUN,NSYM
GO TO 1
C
C
FORMAT (1H1,1/5,36HT IS NOT STRICTLY INCREASING FOR RUN,I7/16X,1HT)
1,13X,1HX,13X,1HY/)
1
121
15
16
17
18
19
20
21
22
23
24

FORMAT (1H1/F5,36HT IS NOT STRICTLY INCREASING FOR RUN,I7/16X,1HT)
1,13X,1HX,13X,1HY/)
1
121
15
16
17
18
19
20
21
22
23
24

FORMAT (5X,3F15.4)

FORMAT (/5X,55HERESULTS OF LEAST SQUARES POLYNOMIAL CURVE FIT FOR"
1RUN = I4,3X,6HNSYM = I2,3X,6HNPOL = 13//4X,1HI,6X,1HT,9X,1HX,9X,4H"
2XNEWX,4X,6HX,NEWX,6X,1HY,9X,4HYNEW,4X,6HY-YNEW)

FORMAT (/5X,55HERESULTS OF LEAST SQUARES CUBIC SPLINE CURVE FIT FOR"
1RUN = I4,3X,6HNSYM = I2,3X,4DF = 110,5//4X,1HI,6X,1HT,9X,1HX,9X"
2,4HXNEW,4X,6HX,NEWX,6X,1HY,9X,4HYNEW,4X,6HY-YNEW)

FORMAT (I5,7F10.4)
25 FORMAT (/10X,12HERROR IN X =,E15.6,10X,12HERROR IN Y =,E15.6)  LS 161
26 FORMAT (/5X,13HERROR NUMBER *13,42H OCCURRED IN CS0S CALL IN LSQPL LS 162
                   IT FOR RUN =,I4,11H AND NSYM =,I3)  LS 163
                  END  LS 164-
LISTING OF DECK: LSQ

CARD NO.

1

SUBROUTINE LSQ (X,Y,W, NP, N, C)

2

LEAST SQUARES WEIGHTED POLYNOMIAL CURVE FIT ROUTINE

3

COPED BY — HARRY L. MORGAN NASA/LARC/TAD/AAB 1983

4

PARAMETER DEFINITION

5

X — ARRAY CONTAINING INDEPENDENT VARIABLE

6

Y — ARRAY CONTAINING DEPENDENT VARIABLE

7

W — ARRAY CONTAINING WEIGHTING VALUES

8

NP — NUMBER OF POINTS

9

N — ORDER OF POLYNOMIAL

10

C — COEFFICIENTS OF POLYNOMIAL

11

NEW Y VALUE = C(1)+C(2)*X+C(3)*X**2+...+C(N+1)*X**N

12

DIMENSION X(NP), Y(NP), W(NP), AND C(N+1) IN CALLING PROGRAM

13

DIMENSION X(1), Y(1), W(1), C(1)

14

DIMENSION A(N+1,N+2) N=10 FOR THIS VERSION

15

COMMON /WORK/ A(11,12)

16

COMPUTE LEAST SQUARES MATRIX

17

N1=N+1

18

N2=N+2

19

DO 1 I=1,N1

20

DO 1 J=1,N2

21

A(I,J)=0.

22

DO 3 K=1,NP

23

Ti=1.

24

DO 3 J=1,N1

25

T2=Ti

26

DO 2 I=1,N1

27

A(J,I)=A(J,I)+T2*W(K)

28

T2=T2*X(K)

29

A(J,N2)=A(J,N2)-Y(K)*Ti*W(K)

30

T1=T1*X(K)

31

SOLVE FOR COEFFICIENTS OF LEAST SQUARES POLYNOMIAL

32

C

33

DO 2 J=1,N2

34

T2=T2*X(K)

35

A(J,N2)=A(J,N2)-Y(K)*Ti*W(K)

36

T1=T1*X(K)

37

C

38

DO 1 J=1,N1

39

T2=Ti

40

A(J,I)=A(J,I)+T2*W(K)

41

Ti=1.

42

DO 3 K=1,NP

43

A(I,J)=0.

44

DO 1 I=1,N1
LISTING OF DECK: LSQ

CARD NO.

41

DO 4 K=1,N
DO 4 J=K,N
T1=A(J+1,K)/A(K,K)
DO 4 I=K,N2

45

4

A(J+1,I)*A(J+1,I)-A(K,I)*T1
C(N1)=-A(N1,N2)/A(N1,N1)
DO 5 I=2,N1
K=N2-I
C(K)=-A(K,N2)/A(K,K)

50

L=K+1
DO 5 J=L,N1
C(K)=-C(J)*A(K,J)/A(K,K)
RETURN
END

PAGE 2
LISTING OF DECK: CSDS

CARD NO.

1 SUBROUTINE CSDS (MAX,IX,X,F,DF,S,IPT,COEF,WK,IERR) CS 1
C********************************************************************** CS 2
C* PURPOSE: SUBROUTINE CSDS FITS A SMOOTH CUBIC SPLINE TO A * CS 3
C* UNIVARIATE FUNCTION. DATA MAY BE UNEQUALLY SPACED. * CS 4
C* USE: CALL CSDS(MAX,IX,X,F,DF,S,IPT,COEF,WK,IERR) * CS 5
C* MAX INPUT INTEGER SPECIFYING THE MAXIMUM NUMBER OF DATA * CS 6
C* POINTS FOR THE INDEPENDENT VARIABLE. * CS 7
C* IX INPUT INTEGER SPECIFYING THE ACTUAL NUMBER OF DATA * CS 8
C* POINTS FOR THE INDEPENDENT VARIABLE. IX>MAX. * CS 9
C* X ONE-DIMENSIONAL INPUT ARRAY DIMENSIONED AT LEAST * CS 10
C* IX IN THE CALLING PROGRAM. UPON ENTRY TO CSDS, * CS 11
C* X(I) MUST CONTAIN THE VALUE OF THE INDEPENDENT * CS 12
C* VARIABLE AT POINT I. * CS 13
C* F ONE-DIMENSIONAL INPUT ARRAY DIMENSIONED AT LEAST * CS 14
C* IX IN THE CALLING PROGRAM. UPON ENTRY TO CSDS, * CS 15
C* F(I) MUST CONTAIN THE VALUE OF THE FUNCTION AT * CS 16
C* POINT X(I). * CS 17
C* I IPT INPUT INITIALIZATION PARAMETER. THE USER MUST * CS 18
C* SPECIFY IPT=-1 WHENEVER A NEW X ARRAY IS * CS 19
C* INPUT. THE ROUTINE WILL ALSO CHECK TO INSURE THAT * CS 20
C* THE X ARRAY IS IN STRICTLY INCREASING ORDER. * CS 21
C* ...

95
<table>
<thead>
<tr>
<th>CARD NO.</th>
<th>COEF</th>
<th>A TWO-DIMENSIONAL OUTPUT ARRAY DIMENSIONED (MAX, 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td></td>
<td>IN THE CALLING PROGRAM. UPON RETURN, COEF(I, J)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONTAINS THE J-TH COEFFICIENT OF THE SPLINE FOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>THE INTERVAL BEGINNING AT POINT X(I). THE</td>
</tr>
<tr>
<td>45</td>
<td></td>
<td>FUNCTIONAL VALUE OF THE SPLINE AT ABSCISSA X(I),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WHERE X(I) &lt; X(I+1), IS GIVEN BY:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F(X(I)) = ((COEF(I, 4) * H + COEF(I, 3)) * H +</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COEF(I, 2)) * H + COEF(I, 1) * H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WHERE H = X(I+1) - X(I)</td>
</tr>
<tr>
<td>50</td>
<td>WK</td>
<td>A ONE-DIMENSIONAL WORK AREA ARRAY DIMENSIONED AT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LEAST (7 * IX + 9) IN THE CALLING PROGRAM.</td>
</tr>
<tr>
<td>55</td>
<td>IERR</td>
<td>OUTPUT ERROR PARAMETER:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 0 NORMAL RETURN. NO ERROR DETECTED.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= J THE J-TH ELEMENT OF THE X ARRAY IS NOT IN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STRICTLY INCREASING ORDER.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= -1 THERE ARE LESS THAN FOUR VALUES IN THE X ARRAY</td>
</tr>
<tr>
<td>60</td>
<td></td>
<td>UPON RETURN FROM CSDD, THIS PARAMETER SHOULD BE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TESTED IN THE CALLING PROGRAM.</td>
</tr>
<tr>
<td>65</td>
<td></td>
<td>REQUIRED ROUTINES - NONE</td>
</tr>
<tr>
<td>70</td>
<td></td>
<td>LANGUAGE - FORTRAN</td>
</tr>
<tr>
<td>75</td>
<td></td>
<td>DATE RELEASED - SEPTEMBER 5, 1973</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LATEST REVISION - MARCH 1975</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DIMENSION X(I), F(I), DF(I), COEF (MAX, 4), WK(1)</td>
</tr>
<tr>
<td>80</td>
<td></td>
<td>SET UP WORKING AREAS</td>
</tr>
</tbody>
</table>
LISTING OF DECK: CSDS

CARDO NO.

81 C
IERR=0
IF (IPT.NE.-1) GO TO 4
IPT=0
85.
IF (IX.LT.4) GO TO 1
GO TO 2
1 IERR=-1
RETURN
IX1=IX-1
90 DO 3 I=1,IX1
IF (X(I+1)-X(I),GT.0) GO TO 3
IERR=I+1
RETURN
3 CONTINUE
95 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(IB5)=0.
IJK5=IB5+NP1
WK(IJK5)=0.
CONTINUE
4 P=0.
H=X(2)-X(1)
F2=-S
FF=(F(2)-F(1))/H
120 IF (IX.LT.3) GO TO 10

CS 81
CS 82
CS 83
CS 84
CS 85
CS 86
CS 87
CS 88
CS 89
CS 90
CS 91
CS 92
CS 93
CS 94
CS 95
CS 96
CS 97
CS 98
CS 99
CS 100
CS 101
CS 102
CS 103
CS 104
CS 105
CS 106
CS 107
CS 108
CS 109
CS 110
CS 111
CS 112
CS 113
CS 114
CS 115
CS 116
CS 117
CS 118
CS 119
CS 120

97
LISTING OF DECK: CSDS

CARD NO.

121  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)+WK(IJK1)*WK(IJK2+1)
     COEF(I-1,4)=WK(I)*WK(IJK2+2)
     6  CONTINUE

145  NEXT ITERATION

C
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(IJK0)
     IJK2=IB2*I-2
     IJKO=I-2
     WK(IJK2)=G*WK(IJK0)
     IJKO=I
     IJK3=IB3*I
     WK(IJK0)=1./(P*COFF(I-1,2)+WK(IJK3)-FF*WK(IJK1)+G*WK(IJK2))
     8  CONTINUE

150  IJK5=IB5*I
     IJKN=IJK5-1
     IJKO=IJKN-1

160.
LISTING OF DECK: CSDS

CARD NO.

161  WK(IJK5)=COEF(I-1,1)-WK(IJK1)*WK(IJK5)-WK(IJK2)*WK(IJK0)
      IJK4=IB4+1
      FF=P*COEF(I-1,3)+WK(IJK4)-H*WK(IJK1)
      G=H

165  H=COEF(-1,4)*P
      CONTINUE
      DO 9 I=3,IX
      J=IX-1+3
      IJK5=IB5+J
      170  IJK6=IJK5+1
      IJK7=IJK6+1
      IJK1=IB1+J
      IJK2=IB2+J
      WK(IJK5)=WK(IJK5)+WK(IJK1)*WK(IJK6)-WK(IJK2)*WK(IJK7)
      175  CONTINUE
      DO 9 E=0
      H=O
      CONTINUE

180  COMPUTE U AND ACCUMULATE E
      CONTINUE
      DO 11 I=2,IX
      G=H
      IJK5=IB5+I
      H=WK(IJK5)+1-WK(IJK5)/(X(I)-X(I-1))
      185  IJK6=IB6+I
      WK(IJK6)=(H-G)*DF(I-1)*DF(I-1)
      E=E+WK(IJK6)*(H-G)
      CONTINUE
      G=-H*DF(IX)*DF(IX)
      190  IJK6=IB6+NPI
      WK(IJK6)=G
      E=E-G*H
      G=F2
      F2=E+F*P
      IF (F2+GE+SOR+F+L+6) GO TO 14
      FF=O
      IJK6=IB6+2
      H=WK(IJK6)+1-WK(IJK6)/(X(2)-X(1))
      IF (IX=LT,3) GO TO 13
      200  CONTINUE
      DO 12 I=3,IX
LISTING OF DECK: CSDS

CARD NO.

201

G=H
IJK6=IB6+I
H=(WK(IJK6+1)-WK(IJK6))/(X(I)-X(I-1))
IJK1=IB1+I-1
IJK2=IB2+I-2

205

G=G-WK(IJK1)*WK(I-1)-WK(IJK2)*WK(I-2)
FF=FF+G*WK(I)*G
WK(I)=G
CONTINUE

210

H=E-P*FF
IF (H*LE=0) GO TO 14

C
C

C

215

C

C

C

C

C

C

P=P+(S-F2)/((SQR(S/E)+P)*H)
GO TO 7

C

C

C

C

C

C

C

C

DO 15 I=2,NP1
IJK6=IB6+I
COEF(I-1,1)=F(I-1)-P*WK(IJK6)

225

IJK5=IB5+I
COEF(I-1,3)=WK(IJK5)
CONTINUE

15

DO 16 I=2,IX
H=X(I)-X(I-1)
COEF(I-1,4)=(COEF(I,3)-COEF(I-1,3))/(3.*H)

230

COEF(I-1,2)=(COEF(I,1)-COEF(I-1,1))/H-(H*COEF(I-1,4)+COEF(I-1,3))*H
CONTINUE

16

RETURN
END

CS 201
CS 202
CS 203
CS 204
CS 205
CS 206
CS 207
CS 208
CS 209
CS 210
CS 211
CS 212
CS 213
CS 214
CS 215
CS 216
CS 217
CS 218
CS 219
CS 220
CS 221
CS 222
CS 223
CS 224
CS 225
CS 226
CS 227
CS 228
CS 229
CS 230
CS 231
CS 232
CS 233
CS 234
CS 235
SUBROUTINE AXISLB (IAxis,XOrg,YOrg,Org,Scale,Dist,HT,NDig) XL 1
C ROUTINE TO DRAW SCALE VALUES ON AXIS XL 2
C CODFD BY -- HARRY L. MORGAN NASA/LARC/TAO/AAB 1983 XL 3
C PARAMETER DEFINITION XL 4
C IAxis = 1 - DRAW X-AXIS SCALES XL 5
C IAxis = 2 - DRAW Y-AXIS SCALES XL 6
C XOrg,YOrg = X,Y COORDINATES OF BEGINNING OF AXIS XL 7
C Org = SCALE VALUE AT BEGINNING OF AXIS XL 8
C Scale = SCALE VALUE PER INCH XL 9
C Dist = LENGTH OF AXIS XL 10
C HT = HEIGHT OF VALUES DRAWN ON AXIS XL 11
C NDig = NUMBER OF SIGNIFICANT FIGURES TO RIGHT OF DECIMAL POINT XL 12
C NDig=1 WILL DROP DECIMAL POINT XL 13
C COMPUTE ROUTING CONSTANTS XL 14
WADJ=0.0 XL 15
IF (HT.LE.0.1) WADJ=0.01 XL 16
Cons=6.*HT/7.*WADJ XL 17
Cons2=2.*HT/7.* XL 18
Adj=0.4*HT XL 19
IF (Adj*LT.0.1) Adj=0.1 XL 20
NL=IFIX(Dist+0.0001)+1 XL 21
XO=XOrg XL 22
YO=YOrg XL 23
Value=Org XL 24
C POSITION PEN AT BEGINNING OF AXIS XL 25
CALL CALPLT (YOrg,YOrg,3) XL 26
C DRAW AXIS VALUE AT EACH INCH MARK ALONG AXIS XL 27
DO P I=1,NL XL 28
Value=Scale*Float(I-1)*Org XL 29
C COMPUTE NUMBER OF SIGNIFICANT FIGURES IN VALUE XL 30
NDig=NDig XL 31
V=ABS(Value) XL 32
IF (V-1.0E-5) 1,1,2 XL 33
1 NDig=-1 XL 34
NSig=1 XL 35
V=0.*0 XL 36
LISTING OF DECK: AXISLB

CARD NO.

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

VALUE=0.0

GO TO 5

NSIG=NDG+1

S=V+SIGN(1,E-7,V)

N=IFIX(S)

IF (N) 5*5,4

NSIG=NSIG+1

V=0.10*V

GO TO 3

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=XT-0.5*S

YT=YT-HT-ADJ

XO=XO+1*

GO TO 7

XT=XT-S-ADJ

YT=YT-0.5*HT

YO=YO+1*

CALL NUMBER (XT,YT,HT,VALUE,0.,NDG)

C ADVANCE TO NEXT INCH MARK

C CONTINUE

C RETURN TO BEGINNING OF AXIS

CALL CALPLT (XORG,YORG,3)

C RETURN TO CALLING PROGRAM

RETURN

END
LISTING OF DECK: COFESY

CARD NO.

1

SUBROUTINE COFESY (X0, Y0, HT, ISYM)

C ROUTINE TO DRAW AERODYNAMIC COEFFICIENTS

C Coded by -- Harry L. Morgan

C NASA/LARC/TAD/ASI 1983

5

C PARAMETER DEFINITION

C X0 - X COORDINATE OF LEFT-HAND EDGE OF SYMBOL

C Y0 - Y COORDINATE OF CENTERLINE OF MAIN LETTER OF SYMBOL

C HT - SYMBOL HEIGHT

C ISYM = 1 - LIFT

C ISYM = 2 - DRAG

C ISYM = 3 - PITCHING MOMENT

C ISYM = 4 - ROLLING MOMENT

C ISYM = 5 - YAWING MOMENT

C ISYM = 6 - SIDE FORCE

C ISYM = 7 - LIFT/DRAG

C ISYM = 8 - ALPHA, DEG.

C ISYM = 9 - BETA, DEG

10

C INITIALIZE CHARACTER SETS

C CALL CHARST1

C CALL CHARST2

C CALL CHARST7

C CALL CHARST8

25

C POSITION PEN AT START OF SYMBOL

C CALL CALPLT (X0, Y0, 3)

C Y1=Y0-0.5*HT

C IF (ISYM.EQ.7) GO TO 1

C IF (ISYM.EQ.8) OR (ISYM.EQ.9) GO TO 2

C IF (ISYM,GT.9) GO TO 3

30

C DRAW LETTER C

C CALL CHARACT (X0, Y1, HT, 2H)C,0..2,0.0

C CALL CHARWH (WD, X1, Y1, HT, 2H)C,2,0.0

C X1=X0+1.15*WD

C Y1=Y0-HT

35

C DRAW SUBSCRIPT OF C

C IF (ISYM.EQ.1) CALL CHARACT (X1, Y1, HT, 2H) l,0..2,0.0

C IF (ISYM.EQ.2) CALL CHARACT (X1, Y1, HT, 2H)D,0..2,0.0

C IF (ISYM.EQ.3) CALL CHARACT (X1, Y1, HT, 2H)H,0..2,0.0

C IF (ISYM.EQ.4) CALL LAM (X1, Y1, HT)
LISTING OF DECK: COEFSY

CARD NO.

41
IF (ISYM.EQ.5) CALL CHARACT (X1,Y1,H,T,2H(N=0.,2,0.,0.) CY 41
IF (ISYM.EQ.6) CALL CHARACT (X1,Y1,H,T,2H(Y=0.,2,0.,0.) CY 42
GO TO 4 CY 43
C DRAW L/D CY 44
45
1 CALL CHARACT (X0,Y1,H,T,4H)/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,H,T,11H$7(A$2(,DEG,0.,11,0.20)) CY 48
IF (ISYM.EQ.9) CALL CHARACT (X0,Y1,H,T,11H$7(B$2(,DEG,0.,11,0.20) CY 49
GO TO 4 CY 50
C ADD ADDITIONAL SYMBOLS HERE AS DESIRED CY 51
3 CONTINUE CY 52
C CY 53
55
C RETURN PEN TO START OF SYMBOL CY 56
4 CALL CALPLT (X0,YM,3) CY 57
CALL CHNGSET (1) CY 58
C RETURN TO CALLING PROGRAM CY 59
60.
RETURN CY 60
END CY 61-
LISTING OF DECK: LAM

CARD NO.

1

SUBROUTINE LAM (XO,YO,HT)

ROUTINE TO DRAW ROLLING MOMENT SUBSCRIPT

CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AAB 1983

PARAMETER DEFINITION
XO,YO - X,Y COORDINATES OF LOWER LEFT-HAND CORNER OF SYMBOL
HT - HEIGHT OF SYMBOL

DIMENSION XM(17), YM(17)

DATA XM/3,13,0,0,4,0,13,14,13,12,2,10,4,1,1,14,4,3,6/ L

DATA YM/43,43,7,2,0,0,2,5,5,2,6,1,1,2,6,7,44,44,43,1/ L

HTH=HT/44.

CALL CALPLT (XO,YO,3)

DO 1 I=1,17

K=2

IF (I.EQ.1) K=3

X=XO+(XM(I)+2)*HTH

Y=YO+YM(I)*HTH

CALL CALPLT (X,Y,K)

CALL CALPLT (XO,YO,3)

RETURN

END

PAGE 1

105
LISTING OF DECK: GRIDLN

CARD NO.

1 SUBROUTINE GRIDLN (XO,YO,XL,YH,NDIV)

5 ROUTINE TO DRAW AREA OF GRID LINES

10 CODED BY -- HARPY L. MORGAN NASA/LARC/TAD/AAB 1983

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 COMPUTE NUMBER OF GRID LINES

C XD=FLOAT(NDIV)

15 IF (XD.LE.0.0) XD=1.

20 NGX=IFIX(XD*XL+0.0001)+1

25 NGY=IFIX(XD*YH+0.0001)+1

30 C DRAW GRID LINES PARALLEL TO Y-AXIS

35 CALL CALPLT (XN,YO,3)

40 C DRAW GRID LINES PARALLEL TO X-AXIS

45 CALL CALPLT (YO,XP,3)
LISTING OF DECK: GRIDLN

CARD NO.

41  D=0.
    IF (NGY*GT.1) D=Y4/FL0AT(NGY-1)
    Y1=Y0
    X1=X0
    X2=X0+XL
    L=0
    DO 6 I=1,NGY
    L=L+1
    Y2=Y1+D
   4   IF (L-1) 4,4,5
        CALL CALPLT (X2,Y1,2)
        IF (I.EQ.NGY) GO TO 6
        CALL CALPLT (X2,Y2,3)
        GO TO 6
50
   5   CALL CALPLT (X1,Y1,2)
        IF (I.EQ.NGY) GO TO 6
        CALL CALPLT (X1,Y2,3)
        L=0
    Y1=Y2
60
   6   C FRAME GRID AREA
        X1=X0+XL
        Y1=Y0+YM
        CALL CALPLT (X0,Y0,3)
        CALL CALPLT (X1,Y0,2)
        CALL CALPLT (X1,Y1,2)
        CALL CALPLT (X0,Y1,2)
        CALL CALPLT (X0,Y0,2)
        CALL CALPLT (X0,Y1,3)
   C RETURN TO CALLING PROGRAM
65
70     RETURN
     END

   6 C FRAME GRID AREA
        X1=X0+XL
        Y1=Y0+YM
        CALL CALPLT (X0,Y0,3)
        CALL CALPLT (X1,Y0,2)
        CALL CALPLT (X1,Y1,2)
        CALL CALPLT (X0,Y1,2)
        CALL CALPLT (X0,Y0,2)
        CALL CALPLT (X0,Y1,3)
   C RETURN TO CALLING PROGRAM
70     RETURN
     END

107
SUBROUTINE RUNKEY (X0, Y0, RUN, LSYM, ISIZE, HT, NRNMAX)

ROUTINE TO DRAW TITLE BLOCK CONTAINING SYMBOLS AND RUN NUMBERS

CODED BY -- HARRY L. MORGAN  NASA/LARC/TAD/AAB  1983

PARAMETER DEFINITION
X0, Y0 - X,Y COORDINATES OF LOWER LEFT-HAND CORNER OF KEY
RUN - RUN NUMBERS
LSYM - SYMBOL ORDER
ISIZE - SYMBOL SIZE
HT - SYMBOL HEIGHT
NRNMAX - MAXIMUM NUMBER OF ALLOWABLE RUNS PER SHEET

DIMENSION RUN(1), LSYM(1)
WORK ARRAYS
COMMON /PLT/ KRUN(10), NRUN(10), LRUN(22), JSYM(22)

INTEGER RUN

DATA XM/1.0/, XMARG/0.1/, XNSP/2.5/, SPC/0.2/

DEFINE LETTERING SIZE
SZ=HT*X0

INITIALIZE CHARACTER SET
CALL CHARS2

DETERMINE THE NUMBER OF NON-ZERO RUNS AND CORRESPONDING SYMBOLS

NR=0
DO 2 I=1,NRNMAX
IF (RUN(I)) 1,2,1
1 NR=NR+1
L Run(NR)=RUN(I)
J Sym(NP)=LSYM(I)
2 CONTINUE
IF (NR.EQ.0) RETURN

COMPUTE WIDTH OF WORDS SYMBOL AND 99999

CALL CHARWH (SYM, HA, HBS, SZ, SPC)
LISTING OF DECK: RUNKEY

CARD NO.

41 CALL CHARWH (WD,HA,HB,SZ,5H99999,5,SPC) KY 41
XS=X0+0.5*SYM+XMarg KY 42
XR=X0+2.5*SYM+0.5*WD+XMarg KY 43
YSR=Y0+XMarg KY 44

45 C
C DRAW SYMBOL AND RUN NUMBER
C

D0 6 L=1,NR
I=NR+1-L
ENCODE (10,7,MRUN) LRUN(I) KY 46
DECODE (10,8,MRUN) KRUN KY 47

50 C
C FIND NUMBER OF DIGITS IN RUN NUMBER
C

D0 3 J=1,10
NRUN(J)=1H KY 48

55 C
KDIG=0
D0 4 J=1,10
IF (KRUN(J) .EQ. 1H ) GO TO 4
KDIG=KDIG+1
C
SHIFT RUN NUMBER FROM RIGHT TO LEFT JUSTIFIED
C

60 NRUN(KDIC)=KRUN(J) KY 49
CONTINUE KY 50

65 C
C REFORMAT RUN NUMBER
C
ENCODE (10,8,MRUN) NRUN KY 51
DECODE (10,9,MRUN) IRUN KY 52

69 CALL CHARWH (WD,HA,HB,SZ,SPC) KY 53
IF (HB .LT. HBS) HBS=HB KY 54

70 CALL CHARWH (WD,HA,HB,SZ,5H)R(UN,5,SPC) KY 55
XX=XR-0.5*WD KY 56

5 CALL CHARACT (XX,YSR,SZ,IRUN,0.,KDIG,SPC) KY 57
YSR=YSR+XNSP+SZ KY 58
CONTINUE KY 59

75 C
C DRAW AND UNDERLINE WORD SYMBOL AND RUN
C

C
CALL CHARWH (WD,HA,HB,SZ,5H)R(UN,5,SPC) KY 60
IF (HB .LT. HBS) HBS=HB KY 61
XX=X0-0.5*(SYM+XMarg) KY 62

PAGE 2
LISTING OF DECK: RUNKEY

CARD NO.

81  CALL CALPLT (XX, YSR, 3)                  KY 81
XX=XS+0.5*(SYM+XMARG)                      KY 82
CALL CALPLT (XX, YSR, 2)                  KY 83
XX=XR-0.5*(WD+XMARG)                      KY 84
85  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*SYM                              KY 89
90  CALL CALPLT (XX, YSR, 3)                  KY 90
CALL CHARACT (XX, YSR, SZ, 8H)S(YMBOL, O, 8, SPC) KY 91
XX=XR-0.5*WD                               KY 92
CALL CHARACT (XX, YSR, SZ, 5H)R(UN, O, 5, SPC) KY 93
95  C                                      KY 94
   RETURN TO TITLE BLOCK ORIGIN             KY 95
   C                                      KY 96
   CALL CALPLT (XO, YO, 3)                 KY 97
   CALL CHARST1                             KY 98
100 C                                      KY 99
    RETURN                                 KY 100
    C                                      KY 101
    7 FORMAT (I10)                          KY 102
    8 FORMAT (I10A1)                        KY 103
    9 FORMAT (A10)                          KY 104
105 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

<table>
<thead>
<tr>
<th>CARD</th>
<th>VARIABLE</th>
<th>VALUE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TEST</td>
<td>-</td>
<td>Test number</td>
</tr>
<tr>
<td>2</td>
<td>SHEETW</td>
<td>-</td>
<td>Total width of plotting sheet, in.</td>
</tr>
<tr>
<td></td>
<td>SHEETH</td>
<td>-</td>
<td>Total height of plotting sheet, in.</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
<td>-</td>
<td>Space between plotting sheets, in.</td>
</tr>
<tr>
<td>3</td>
<td>ISYM</td>
<td>-</td>
<td>Starting symbol number (See table I.)</td>
</tr>
<tr>
<td></td>
<td>ISIZE</td>
<td>1</td>
<td>Small symbol size</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Medium symbol size</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>Large symbol size</td>
</tr>
<tr>
<td>4</td>
<td>IOP</td>
<td>0</td>
<td>Plot data points only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Plot data points and fair with tension spline</td>
</tr>
<tr>
<td></td>
<td>TENSION</td>
<td>-</td>
<td>Tension factor, ( \delta )</td>
</tr>
<tr>
<td>5</td>
<td>IEDIT</td>
<td>0</td>
<td>Do not sort or edit data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Sort and edit data</td>
</tr>
<tr>
<td></td>
<td>TOLR</td>
<td>-</td>
<td>Edit tolerance of independent test variable (TOLR=0 for sort only)</td>
</tr>
<tr>
<td>6</td>
<td>IW</td>
<td>-</td>
<td>Data array location of independent test variable</td>
</tr>
</tbody>
</table>
Card 8 is input NPLOT times and NPLOT is limited to a value of 10 for each setup.

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

Plotting Sheet Namelist

<table>
<thead>
<tr>
<th>NAMELIST VARIABLE</th>
<th>VALUE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>$SHEET</td>
<td>-</td>
<td>Namelist label</td>
</tr>
<tr>
<td>NO</td>
<td>-</td>
<td>Sheet number</td>
</tr>
<tr>
<td>RUN</td>
<td>-</td>
<td>Run numbers to be plotted on sheet</td>
</tr>
<tr>
<td>NEWCASE</td>
<td>0 or default 1</td>
<td>A new SHEET namelist follows A new setup deck follows</td>
</tr>
<tr>
<td>$END</td>
<td>-</td>
<td>End of namelist</td>
</tr>
</tbody>
</table>

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 NMAX 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>
<thead>
<tr>
<th>ISYM</th>
<th>SYMBOL</th>
<th>DASHED LINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>◆</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>△</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>▽</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>◈</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>◊</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>✡</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>✍</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>▽</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>◆</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>△</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>▽</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>◆</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>△</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>▽</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>+</td>
<td></td>
</tr>
</tbody>
</table>
TABLE II. - LIST OF STANDARD AERODYNAMIC SYMBOLS DRAWN BY SUBROUTINE COEFSY

<table>
<thead>
<tr>
<th>ISYM</th>
<th>COEFFICIENT</th>
<th>SYMBOL</th>
<th>WIDTH/HEIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lift</td>
<td>$C_L$</td>
<td>2.0</td>
</tr>
<tr>
<td>2</td>
<td>Drag</td>
<td>$C_D$</td>
<td>2.1</td>
</tr>
<tr>
<td>3</td>
<td>Pitching Moment</td>
<td>$C_m$</td>
<td>2.2</td>
</tr>
<tr>
<td>4</td>
<td>Rolling Moment</td>
<td>$C_\ell$</td>
<td>1.7</td>
</tr>
<tr>
<td>5</td>
<td>Yawing Moment</td>
<td>$C_n$</td>
<td>1.9</td>
</tr>
<tr>
<td>6</td>
<td>Side Force</td>
<td>$C_Y$</td>
<td>2.1</td>
</tr>
<tr>
<td>7</td>
<td>Lift-Drag Ratio</td>
<td>$L/D$</td>
<td>2.6</td>
</tr>
<tr>
<td>8</td>
<td>Angle of Attack</td>
<td>$\alpha,\text{deg}$</td>
<td>3.5</td>
</tr>
<tr>
<td>9</td>
<td>Angle of Sideslip</td>
<td>$\beta,\text{deg}$</td>
<td>3.4</td>
</tr>
</tbody>
</table>
### TABLE III. - LISTING OF SAMPLE INPUT CASES

#### CASE 1 - Single Plot with Three Variables

<table>
<thead>
<tr>
<th>496.</th>
<th>20.</th>
<th>23.</th>
<th>10.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>3.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>2.</td>
<td>.4</td>
<td>CL</td>
</tr>
<tr>
<td>7.</td>
<td>11.</td>
<td>.2</td>
<td>CD</td>
</tr>
<tr>
<td>8.</td>
<td>20.</td>
<td>.4</td>
<td>CM</td>
</tr>
</tbody>
</table>

$\text{SHEET NO}=206, \text{RUN}=39,59$

#### CASE 2 - Four Plots with One Variable

<table>
<thead>
<tr>
<th>496.</th>
<th>12.</th>
<th>12.</th>
<th>10.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>2.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>4.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>2.</td>
<td>.4</td>
<td>CM</td>
</tr>
<tr>
<td>9.</td>
<td>1.</td>
<td>.4</td>
<td>L/D</td>
</tr>
<tr>
<td>2.</td>
<td>8.</td>
<td>.8</td>
<td>CL</td>
</tr>
<tr>
<td>3.</td>
<td>7.</td>
<td>.2</td>
<td>CD</td>
</tr>
</tbody>
</table>

$\text{SHEET NO}=206, \text{RUN}=39,59$

$\text{SHEET NO}=206, \text{RUN}=39,59$
### TABLE IV. - SAMPLE OUTPUT FOR CASE 1 INPUT

#### PAGE 1 OUTPUT

<table>
<thead>
<tr>
<th>PLOT NO.</th>
<th>YTAPE</th>
<th>YOFFSET</th>
<th>YSCALE</th>
<th>LABEL</th>
<th>XTAPE</th>
<th>XOFFSET</th>
<th>XSCALE</th>
<th>LABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>2.00</td>
<td>.4000</td>
<td>CL</td>
<td>1</td>
<td>4.00</td>
<td>2.0000</td>
<td>ALPHA</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>11.00</td>
<td>.2000</td>
<td>CD</td>
<td>1</td>
<td>4.00</td>
<td>2.0000</td>
<td>ALPHA</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>20.00</td>
<td>.4000</td>
<td>CM</td>
<td>1</td>
<td>4.00</td>
<td>2.0000</td>
<td>ALPHA</td>
</tr>
</tbody>
</table>

#### PAGE 2 OUTPUT

The following list of runs are available on the data disk for Test No. 496:

<table>
<thead>
<tr>
<th>RUNS</th>
<th>XSC</th>
<th>XOF</th>
<th>FS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>230</td>
<td>119</td>
<td>231</td>
</tr>
<tr>
<td>2</td>
<td>232</td>
<td>120</td>
<td>232</td>
</tr>
<tr>
<td>3</td>
<td>233</td>
<td>121</td>
<td>233</td>
</tr>
<tr>
<td>4</td>
<td>234</td>
<td>122</td>
<td>234</td>
</tr>
<tr>
<td>5</td>
<td>235</td>
<td>123</td>
<td>235</td>
</tr>
<tr>
<td>6</td>
<td>236</td>
<td>124</td>
<td>236</td>
</tr>
<tr>
<td>7</td>
<td>237</td>
<td>125</td>
<td>237</td>
</tr>
<tr>
<td>8</td>
<td>238</td>
<td>126</td>
<td>238</td>
</tr>
<tr>
<td>9</td>
<td>239</td>
<td>127</td>
<td>239</td>
</tr>
<tr>
<td>10</td>
<td>240</td>
<td>128</td>
<td>240</td>
</tr>
<tr>
<td>11</td>
<td>241</td>
<td>129</td>
<td>241</td>
</tr>
<tr>
<td>12</td>
<td>242</td>
<td>130</td>
<td>242</td>
</tr>
<tr>
<td>13</td>
<td>243</td>
<td>131</td>
<td>243</td>
</tr>
<tr>
<td>14</td>
<td>244</td>
<td>132</td>
<td>244</td>
</tr>
<tr>
<td>15</td>
<td>245</td>
<td>133</td>
<td>245</td>
</tr>
<tr>
<td>16</td>
<td>246</td>
<td>134</td>
<td>246</td>
</tr>
<tr>
<td>17</td>
<td>247</td>
<td>135</td>
<td>247</td>
</tr>
<tr>
<td>18</td>
<td>248</td>
<td>136</td>
<td>248</td>
</tr>
<tr>
<td>19</td>
<td>249</td>
<td>137</td>
<td>249</td>
</tr>
<tr>
<td>20</td>
<td>250</td>
<td>138</td>
<td>250</td>
</tr>
<tr>
<td>21</td>
<td>251</td>
<td>139</td>
<td>251</td>
</tr>
<tr>
<td>22</td>
<td>252</td>
<td>140</td>
<td>252</td>
</tr>
<tr>
<td>23</td>
<td>253</td>
<td>141</td>
<td>253</td>
</tr>
<tr>
<td>24</td>
<td>254</td>
<td>142</td>
<td>254</td>
</tr>
</tbody>
</table>

**Total number of data points = 4901**

#### PAGE 3 OUTPUT

Data plotted

<table>
<thead>
<tr>
<th>SHEET NO.</th>
<th>RUNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>206</td>
<td>39, 59, 0, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
</tbody>
</table>
**TABLE IV** - CONCLUDED

**PAGE 4 OUTPUT**

**THE FOLLOWING RUNS HAVE BEEN EDITED**

<table>
<thead>
<tr>
<th>RUN NO.</th>
<th>TOLERANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>.500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>POINT NO.</th>
<th>T-VALUE</th>
<th>POINT NO.</th>
<th>T-VALUE</th>
<th>POINT NO.</th>
<th>T-VALUE</th>
<th>POINT NO.</th>
<th>T-VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-6.1700</td>
<td>1</td>
<td>-6.1700</td>
<td>1</td>
<td>-6.1700</td>
<td>4</td>
<td>.3500</td>
</tr>
<tr>
<td>2</td>
<td>-4.1800</td>
<td>2</td>
<td>-4.1800</td>
<td>2</td>
<td>-4.1800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-1.8800</td>
<td>3</td>
<td>-1.8800</td>
<td>3</td>
<td>-1.8800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>.3500</td>
<td>20</td>
<td>.1700</td>
<td>20</td>
<td>.1700</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2.0100</td>
<td>4</td>
<td>.3500</td>
<td>5</td>
<td>2.0100</td>
<td>6</td>
<td>4.2800</td>
</tr>
<tr>
<td>6</td>
<td>4.2800</td>
<td>5</td>
<td>2.0100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6.3700</td>
<td>6</td>
<td>4.2800</td>
<td>7</td>
<td>6.3700</td>
<td>8</td>
<td>8.4500</td>
</tr>
<tr>
<td>8</td>
<td>8.4500</td>
<td>7</td>
<td>6.3700</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10.3200</td>
<td>8</td>
<td>8.4500</td>
<td>9</td>
<td>10.3200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>12.4100</td>
<td>9</td>
<td>10.3200</td>
<td>10</td>
<td>12.4100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>14.4000</td>
<td>10</td>
<td>12.4100</td>
<td>11</td>
<td>14.4000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>18.6700</td>
<td>12</td>
<td>16.8000</td>
<td>13</td>
<td>18.6700</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>20.8600</td>
<td>13</td>
<td>18.6700</td>
<td>14</td>
<td>20.8600</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>22.7500</td>
<td>14</td>
<td>20.8600</td>
<td>15</td>
<td>22.7500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>24.7800</td>
<td>15</td>
<td>22.7500</td>
<td>16</td>
<td>24.7800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>26.6700</td>
<td>16</td>
<td>24.7800</td>
<td>17</td>
<td>26.6700</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>28.7500</td>
<td>17</td>
<td>26.6700</td>
<td>18</td>
<td>28.7500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>30.7000</td>
<td>18</td>
<td>28.7500</td>
<td>19</td>
<td>30.7000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>.1700</td>
<td>19</td>
<td>30.7000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RUN NO.</th>
<th>TOLERANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>59</td>
<td>.500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>POINT NO.</th>
<th>T-VALUE</th>
<th>POINT NO.</th>
<th>T-VALUE</th>
<th>POINT NO.</th>
<th>T-VALUE</th>
<th>POINT NO.</th>
<th>T-VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-6.0200</td>
<td>1</td>
<td>-6.0200</td>
<td>1</td>
<td>-6.0200</td>
<td>4</td>
<td>.2800</td>
</tr>
<tr>
<td>2</td>
<td>-4.4000</td>
<td>2</td>
<td>-4.4000</td>
<td>2</td>
<td>-4.4000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-2.2200</td>
<td>3</td>
<td>-2.2200</td>
<td>3</td>
<td>-2.2200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>.2800</td>
<td>4</td>
<td>.2800</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2.0000</td>
<td>5</td>
<td>2.0000</td>
<td>5</td>
<td>2.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4.8700</td>
<td>6</td>
<td>4.8700</td>
<td>6</td>
<td>4.8700</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6.3000</td>
<td>7</td>
<td>6.3000</td>
<td>7</td>
<td>6.3000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10.5800</td>
<td>9</td>
<td>10.5800</td>
<td>10</td>
<td>12.6400</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>12.6400</td>
<td>10</td>
<td>12.6400</td>
<td>11</td>
<td>14.9700</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>14.9700</td>
<td>11</td>
<td>14.9700</td>
<td>12</td>
<td>16.6100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>16.6100</td>
<td>12</td>
<td>16.6100</td>
<td>13</td>
<td>18.4600</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>20.8900</td>
<td>13</td>
<td>18.4600</td>
<td>14</td>
<td>20.8900</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>22.8300</td>
<td>14</td>
<td>20.8900</td>
<td>15</td>
<td>22.8300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>25.3700</td>
<td>15</td>
<td>22.8300</td>
<td>16</td>
<td>25.3700</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>26.9800</td>
<td>16</td>
<td>25.3700</td>
<td>17</td>
<td>26.9800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>28.9200</td>
<td>17</td>
<td>26.9800</td>
<td>18</td>
<td>28.9200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>31.3400</td>
<td>18</td>
<td>28.9200</td>
<td>19</td>
<td>31.3400</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>.2900</td>
<td>19</td>
<td>31.3400</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ALL PLOTTING COMPLETED**

120
### Table V. - Sample of Least-Squares Polynomial Curve Fit

<table>
<thead>
<tr>
<th>DATA PLOTTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHEET NO = 206</td>
</tr>
<tr>
<td>RUNS = 39, 59, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RESULTS OF LEAST SQUARES POLYNOMIAL CURVE FIT FOR RUN = 39</th>
<th>NSYM = 1</th>
<th>NPOL = 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>X</td>
<td>XNEW</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>-6.1700</td>
<td>.9150</td>
</tr>
<tr>
<td>2</td>
<td>-4.1800</td>
<td>1.9100</td>
</tr>
<tr>
<td>3</td>
<td>-1.8800</td>
<td>3.0600</td>
</tr>
<tr>
<td>4</td>
<td>.1700</td>
<td>4.0850</td>
</tr>
<tr>
<td>5</td>
<td>2.0100</td>
<td>5.0050</td>
</tr>
<tr>
<td>6</td>
<td>4.2800</td>
<td>6.1400</td>
</tr>
<tr>
<td>7</td>
<td>6.3700</td>
<td>7.1850</td>
</tr>
<tr>
<td>8</td>
<td>8.4500</td>
<td>8.2250</td>
</tr>
<tr>
<td>9</td>
<td>10.3200</td>
<td>9.1600</td>
</tr>
<tr>
<td>10</td>
<td>12.4100</td>
<td>10.2050</td>
</tr>
<tr>
<td>13</td>
<td>18.6700</td>
<td>13.3350</td>
</tr>
<tr>
<td>15</td>
<td>22.7500</td>
<td>15.3750</td>
</tr>
<tr>
<td>16</td>
<td>24.7800</td>
<td>16.3900</td>
</tr>
<tr>
<td>17</td>
<td>26.6700</td>
<td>17.3350</td>
</tr>
<tr>
<td>18</td>
<td>28.7500</td>
<td>18.3750</td>
</tr>
</tbody>
</table>

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, 0, 0, 0, 0 |

**RESULTS OF LEAST SQUARES CUBIC SPLINE CURVE FIT FOR RUN = 39**

| NSYM = 1 | DF = .10000 |

<table>
<thead>
<tr>
<th>I</th>
<th>T</th>
<th>X</th>
<th>XNEW</th>
<th>X-XNEW</th>
<th>Y</th>
<th>YNEW</th>
<th>Y-YNEW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-6.1700</td>
<td>.9150</td>
<td>.9150</td>
<td>0.0000</td>
<td>1.1329</td>
<td>1.0207</td>
<td>.1122</td>
</tr>
<tr>
<td>2</td>
<td>-4.1800</td>
<td>1.9100</td>
<td>1.9100</td>
<td>0.0000</td>
<td>1.4455</td>
<td>1.4164</td>
<td>.0291</td>
</tr>
<tr>
<td>3</td>
<td>-1.8800</td>
<td>3.0600</td>
<td>3.0600</td>
<td>0.0000</td>
<td>1.7651</td>
<td>1.8793</td>
<td>-.1142</td>
</tr>
<tr>
<td>4</td>
<td>1.1700</td>
<td>4.0850</td>
<td>4.0850</td>
<td>0.0000</td>
<td>2.2144</td>
<td>2.3021</td>
<td>-.0877</td>
</tr>
<tr>
<td>5</td>
<td>2.0100</td>
<td>5.0050</td>
<td>5.0050</td>
<td>0.0000</td>
<td>2.6249</td>
<td>2.6911</td>
<td>-.0662</td>
</tr>
<tr>
<td>6</td>
<td>4.2800</td>
<td>6.1400</td>
<td>6.1400</td>
<td>0.0000</td>
<td>3.1164</td>
<td>3.1807</td>
<td>-.0643</td>
</tr>
<tr>
<td>7</td>
<td>6.3700</td>
<td>7.1850</td>
<td>7.1850</td>
<td>0.0000</td>
<td>3.5735</td>
<td>3.6354</td>
<td>-.0619</td>
</tr>
<tr>
<td>8</td>
<td>8.4500</td>
<td>8.2250</td>
<td>8.2250</td>
<td>0.0000</td>
<td>4.0115</td>
<td>4.0837</td>
<td>-.0722</td>
</tr>
<tr>
<td>9</td>
<td>10.3200</td>
<td>9.1600</td>
<td>9.1600</td>
<td>0.0000</td>
<td>4.4649</td>
<td>4.4744</td>
<td>-.0095</td>
</tr>
<tr>
<td>10</td>
<td>12.4100</td>
<td>10.2050</td>
<td>10.2050</td>
<td>0.0000</td>
<td>4.9459</td>
<td>4.8855</td>
<td>.0604</td>
</tr>
<tr>
<td>11</td>
<td>14.4000</td>
<td>11.2000</td>
<td>11.2000</td>
<td>0.0000</td>
<td>5.4040</td>
<td>5.2396</td>
<td>.1644</td>
</tr>
<tr>
<td>12</td>
<td>16.8000</td>
<td>12.4000</td>
<td>12.4000</td>
<td>0.0000</td>
<td>5.8305</td>
<td>5.6095</td>
<td>.2250</td>
</tr>
<tr>
<td>13</td>
<td>18.6700</td>
<td>13.3350</td>
<td>13.3350</td>
<td>0.0000</td>
<td>5.9419</td>
<td>5.8400</td>
<td>.1018</td>
</tr>
<tr>
<td>14</td>
<td>20.8600</td>
<td>14.4300</td>
<td>14.4300</td>
<td>0.0000</td>
<td>6.0995</td>
<td>6.0624</td>
<td>.0371</td>
</tr>
<tr>
<td>15</td>
<td>22.7500</td>
<td>15.3750</td>
<td>15.3750</td>
<td>0.0000</td>
<td>6.2170</td>
<td>6.2162</td>
<td>.0009</td>
</tr>
<tr>
<td>16</td>
<td>24.7800</td>
<td>16.3900</td>
<td>16.3900</td>
<td>0.0000</td>
<td>6.3325</td>
<td>6.3511</td>
<td>-.0186</td>
</tr>
<tr>
<td>17</td>
<td>26.6700</td>
<td>17.3350</td>
<td>17.3350</td>
<td>0.0000</td>
<td>6.4900</td>
<td>6.4569</td>
<td>.0331</td>
</tr>
<tr>
<td>18</td>
<td>28.7500</td>
<td>18.3750</td>
<td>18.3750</td>
<td>0.0000</td>
<td>6.3650</td>
<td>6.5606</td>
<td>-.1956</td>
</tr>
<tr>
<td>19</td>
<td>30.7000</td>
<td>19.3500</td>
<td>19.3500</td>
<td>0.0000</td>
<td>6.5800</td>
<td>6.6538</td>
<td>-.0738</td>
</tr>
</tbody>
</table>

**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 (4.0,0.0,13.0,2)  
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 (1.0,0.0,0.0,-8.0,2.0,20.0,SZ,-1)  
XT=(20.0-3.5*HT)/2.0  
YT=-1.0  
CALL COEFSY (XT,YT,HT,8)  
C DRAW AND LABEL LOWER PORTION OF Y-AXIS (CL)  
CALL AXISLB (2.0,0.0,0.0,-0.8,0.4,10.0,SZ,1)  
XT=-1.8  
YT=5.0  
CALL COEFSY (XT,YT,HT,1)  
C DRAW AND LABEL MIDDLE PORTION OF Y-AXIS (CD)  
CALL AXISLB (2.0,0.0,11.0,0.0,0.2,5.0,SZ,1)  
XT=-1.8  
YT=13.5  
CALL COEFSY (XT,YT,HT,2)  
C DRAW AND LABEL UPPER PORTION OF Y-AXIS (CM)  
CALL AXISLB (2.0,0.0,17.0,-1.2,0.4,6.0,SZ,1)  
XT=-1.8  
YT=20.0  
CALL COEFSY (XT,YT,HT,3)  
C DRAW AND LABEL RUN KEY  
XT=2.0  
YT=HT+23.0  
CALL RUNKFY (XT,YT,RE,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
C DRAW AND LABEL X-AXIS
CALL AXISLB (1,0.0,0.0,-0.8,0.8,5.0,5Z,1)
X0=15.0-2.0*HT)/2.0
YO=0.8
CALL COEFSY (X0,Y0,HT,1)
C
C DRAW AND LABEL Y-AXIS
CALL AXISLB (2,0.0,0.0,-0.8,0.8,5.0,5Z,1)
X0=1.0
YO=2.5
CALL COEFSY (X0,Y0,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 (6.0,0.0,5.0,2)
CALL ZEROLN (7.0,1.0,5.0,1)
C
C DRAW AND LABEL X-AXIS
CALL AXISLB (1,7.0,0.0,-10.0,10.0,5.0,5Z,-1)
X0=7.0*(5.0-3.0*HT)/2.0
YO=0.8
CALL COEFSY (X0,Y0,HT,8)
C
C DRAW AND LABEL Y-AXIS
CALL AXISLB (2,7.0,0.0,-4.0,4.0,5.0,5Z,-1)
X0=6.0
YO=2.5
CALL COEFSY (X0,Y0,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 (8.0,8.0,5.0,1)
C
C DRAW AND LABEL X-AXIS
CALL AXISLB (1,0.0,7.0,-10.0,10.0,5.0,5Z,-1)
X0=15.0-3.0*HT)/2.0
YO=6.2
CALL COEFSY (X0,Y0,HT,8)
C
C DRAW AND LABEL Y-AXIS
CALL AXISLB (2,0.0,7.0,-0.8,0.8,5.0,5Z,1)
X0=1.0
YO=9.5
CALL COEFSY (X0,Y0,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
C DRAW AND LABEL X-AXIS
CALL AXISLB (1,7.0,7.0,-0.8,0.8,5.0,5Z,1)
X0=7.0*(5.0-2.0*HT)/2.0
YO=6.2
CALL COEFSY (X0,Y0,HT,1)
C
C DRAW AND LABEL Y-AXIS
CALL AXISLB (2,7.0,7.0,0.0,0.2,5.0,5Z,1)
X0=6.0
YO=9.5
CALL COEFSY (X0,Y0,HT,2)
C
C DRAW AND LABEL RUN KEY
C
X0=1.0
YO=12.0+HT
CALL RUNKEY (X0,Y0,RUNLSYM,ISIZE,HT,10)
```
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.
Figure 3. - Plot of Case 1 input with least-squares curve fairing.
b) Least-Squares Cubic Spline

Figure 3. - Concluded.
\begin{tabular}{|c|c|}
\hline
Symbol & Run \\
\hline
\circ & 39 \\
\square & 59 \\
\hline
\end{tabular}

Figure 4. - Plot of Case 1 input with grid lines, axis scale values and labels, and legend drawn.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Run</th>
</tr>
</thead>
<tbody>
<tr>
<td>○</td>
<td>39</td>
</tr>
<tr>
<td>□</td>
<td>59</td>
</tr>
</tbody>
</table>

Figure 5. - Plot of Case 2 input with grid lines, axis scale values and labels, and legend drawn.
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 subroutine 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.
DO NOT REMOVE SLIP FROM MATERIAL

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

<table>
<thead>
<tr>
<th>NAME</th>
<th>MS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jim, Langley</td>
<td>413</td>
</tr>
</tbody>
</table>

NASA Langley (Rev. May 1998)