

Research Series No. 3

INSTRUCTIONS FOR USING
THE AJ 832 PLOTTING SOFTWARE PACKAGE
AS MODIFIED FOR THE CDC 6500 COMPUTER

Compiled and Edited
By Marcia Carlyn

Published By

The Institute for Research on Teaching
252 Erickson Hall
Michigan State University
East Lansing, Michigan 48824

Printed and Distributed
by the
College of Education
Michigan State University

November, 1977

This work is sponsored in part by the Institute for Research on Teaching, College of Education, Michigan State University. The Institute for Research on Teaching is funded primarily by the Program for Teaching and Instruction of the National Institute of Education, United States Department of Health, Education, and Welfare. The opinions expressed in this publication do not necessarily reflect the position, policy, or endorsement of the National Institute of Education. (Contract No. 400-76-0073)

Institute for Research on Teaching

The **Institute for Research on Teaching** was founded at Michigan State University in 1976 by the National Institute of Education. Following a nationwide competition in 1981, the NIE awarded a second contract to the IRT, extending work through 1984. Funding is also received from other agencies and foundations for individual research projects.

The IRT conducts major research projects aimed at improving classroom teaching, including studies of classroom management strategies, student socialization, the diagnosis and remediation of reading difficulties, and teacher education. IRT researchers are also examining the teaching of specific school subjects such as reading, writing, general mathematics, and science, and are seeking to understand how factors outside the classroom affect teacher decision making.

Researchers from such diverse disciplines as educational psychology, anthropology, sociology, and philosophy cooperate in conducting IRT research. They join forces with public school teachers, who work at the IRT as half-time collaborators in research, helping to design and plan studies, collect data, analyze and interpret results, and disseminate findings.

The IRT publishes research reports, occasional papers, conference proceedings, a newsletter for practitioners, and lists and catalogs of IRT publications. For more information, to receive a list or catalog, and/or to be placed on the IRT mailing list to receive the newsletter, please write to the IRT Editor, Institute for Research on Teaching, 252 Erickson Hall, Michigan State University, East Lansing, Michigan 48824-1034.

Co-Directors: Jere E. Brophy and Andrew C. Porter

Associate Directors: Judith E. Lanier and Richard S. Prawat

Editorial Staff

Editor: Janet Eaton

Assistant Editor: Patricia Nischan

Abstract

This is a manual for using the Anderson Jacobson AJ 832 Plotting Software Package, as modified for the CDC 6500 computer at Michigan State University. At present, the package is restricted to use with the AJ 832 keyboard printer terminal and cannot be used with an AJ 830 terminal.

PREFACE

This manual includes instructions for using the Anderson Jacobson AJ 832 Plotting Software Package, as modified for the CDC 6500 computer at Michigan State University.

Before reading these instructions, users are advised to read the AJ 832 Keyboard Printer Terminal Operator's Manual (Section 5) and the AJ 830/832 Plotting Software Operator's Manual (Sections 1 and 2). Many of the instructions in this manual are quite similar to instructions included in the CalComp Basic Software package, and familiarity with the CalComp routines should prove helpful in using the AJ 832 package. (See the MSU Computer Laboratory User's Guide, Volume VII, Chapter 2).

At the present time, the modified AJ 832 Plotting Software Package is restricted to the Anderson Jacobson AJ 832 keyboard printer terminal, and cannot be used with an AJ 830 terminal.

*
Marcia Carlyn

*Presently a research associate in the Department of Psychiatry, Marcia Carlyn was an instructor with the Office of Research Consultation while completing this project.

CONTENTS

PREFACE

INTRODUCTION.	1
SAMPLE PLOTTING PROGRAM	2
SUBROUTINES	4
PLOTS.	5
PLOT	6
SETUP	8
FACTOR	10
WHERE.	11
SET	12
HLINE	13
VLINE	14
SYMBOL	15
SCALE.	17
AXIS	19
OFFSET	21
APPENDICES	
A. Translation of Special Characters	22
B. Translation of Graphic Characters	23

INTRODUCTION

The original AJ 832 Plotting Software Package was written for an IBM FORTRAN compiler, and numerous modifications were required for it to run successfully with the CDC 6500 FORTRAN compiler at MSU. Major changes included translating 32-bit words to 60-bit words, and translating EBCDIC variables into the particular ASCII binary code which is recognized by the AJ832 terminal. To ease the task of future translations, ASCII binary representations of characters recognizable by the AJ 832 terminal are included in Appendices A and B.

In order to use the modified AJ 832 Plotting Software Package, the following on-line procedure should be followed at an AJ 832 terminal:

1. Write a FORTRAN program which includes calls to the plotting routines. The first statement in the FORTRAN program must be a PROGRAM statement. The program may have any name, but the program name must be followed by these required parameters: (ZZZZAS=1,TAPE61=ZZZZAS,OUTPUT).
2. Successfully compile the FORTRAN program.
3. Attach the AJ 832 plotting subroutines.
4. Load and execute the program and subroutines.

An example of this procedure is shown below:

```
SYSTEM,FORTRAN.  
10=PROGRAM ANYNAME(ZZZZAS=1,TAPE61=ZZZZAS,OUTPUT)  
20=CALL PLOTS  
.  
.  
.  
180=CALL PLOT(-5.0,-5.0,999)  
190=STOP  
200=END  
FIN.  
ATTACH,SUBS,AJSUBS.  
LOAD,LGO,SUBS.
```

SAMPLE PLOTTING PROGRAM

To illustrate the use of the modified AJ 832 Plotting Software Package, the sample program that produced the graph shown in Figure 1 is given below with annotations.

```
PROGRAM ANYNAME(ZZZZAS=1,TAPE61=ZZZZAS,OUTPUT)
```

Identify the program and include the required parameters.

```
DIMENSION XYARRAY(12)
```

Define an array containing 4 (X,Y) data points plus 4 additional words required by SCALE, AXIS, and OFFSET.

```
DATA XYARRAY/1.0,3.0,4.0,9.5,7.0,-5.0,10.0,14.5,4*0.0/
```

Store values to be plotted in the array.

```
CALL PLOTS
```

Initialize the plotting sequence.

```
CALL PLOT(-5.0,0.0,-3)
```

Drive the print wheel to the left margin.

```
CALL PLOT(2.0,-9.0,-3)
```

Establish the origin 2 inches from the left margin and 9 inches down the page.

```
CALL SCALE(XYARRAY(1),5.0,4,2)
```

```
CALL SCALE(XYARRAY(2),8.0,4,2)
```

Scale the X-values first, then the Y-values.

```
CALL AXIS(0.0,0.0,4HTIME,-204,5.0,0.0,XYARRAY(9),XYARRAY(11))
```

```
CALL AXIS(0.0,0.0,5HSCORE,205,8.0,90.0,XYARRAY(10),XYARRAY(12))
```

Draw the X-axis first, then the Y-axis.

```
CALL OFFSET(XYARRAY(9),XYARRAY(11),XYARRAY(10),XYARRAY(12))
```

Prepare to plot the data in the area defined by the axes.

```
CALL PLOT(XYARRAY(1),XYARRAY(2),3)
```

Position the print head at the first data point.

```
DO 100 I=1,7,2  
CALL PLOT(XYARRAY(I),XYARRAY(I+1),2)  
100 CONTINUE
```

Draw lines connecting the data points.

```
CALL OFFSET(0.0,1.0,0.0,1.0)
```

Return to normal X and Y plotting values.

```
CALL SYMBOL(1.5,7.0,16HPERFORMANCE TEST,0.0,16)  
CALL SYMBOL(1.5,6.6,11H      N = 4,0.0,11)
```

Put titles on the graph.

```
CALL PLOT(-5.0,-5.0,999)
```

Close the plot file and skip down the page.

```
STOP  
END
```

Terminate program execution.

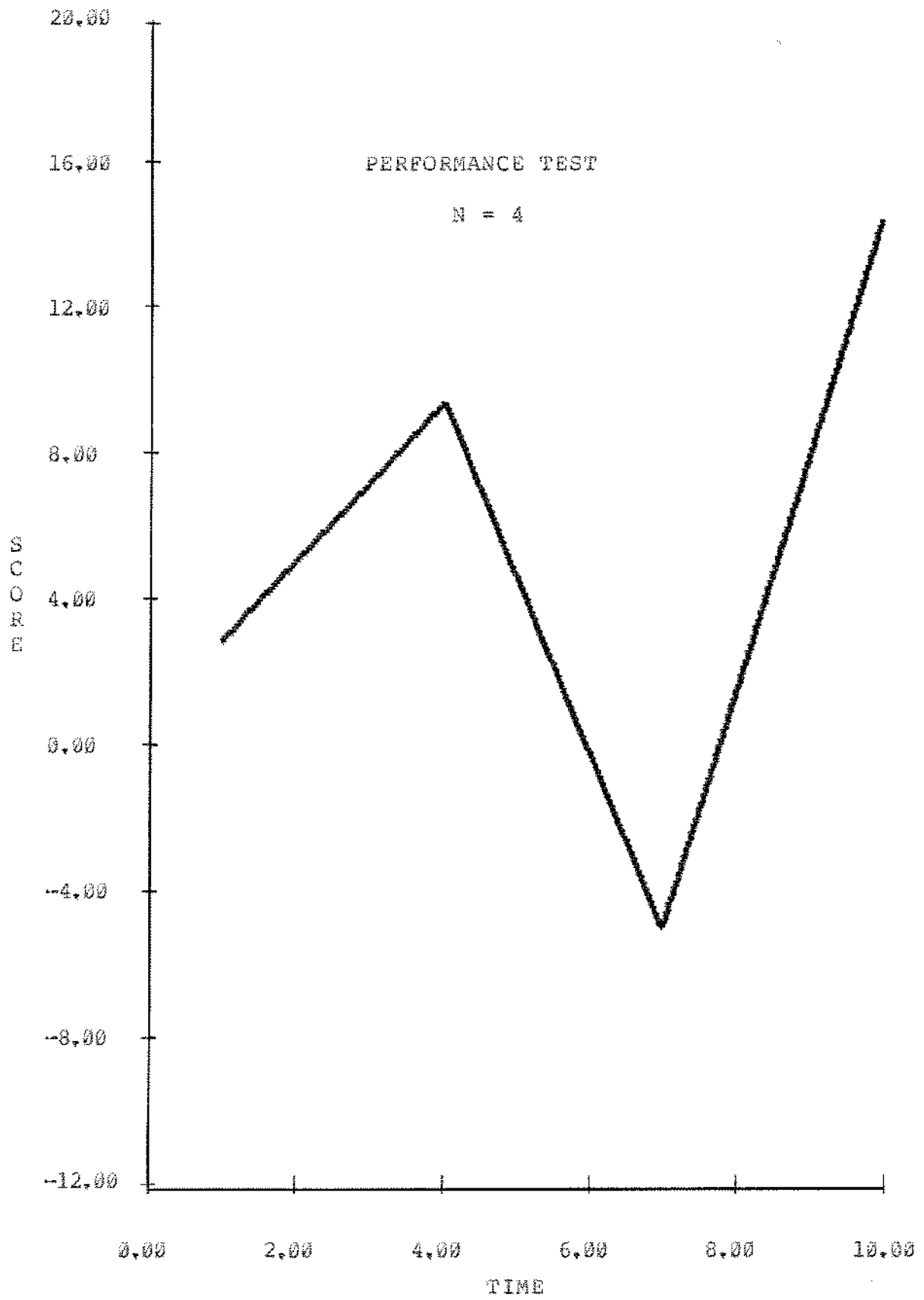


Figure 1

SUBROUTINES

The following pages describe the plotting routines of the AJ 832 Plotting Software Package as modified for the CDC 6500 computer at MSU. This basic set of FORTRAN Extended callable subroutines is comprised of twelve routines: PLOTS, PLOT, SETUP, FACTOR, WHERE, SET, HLINE, VLINE, SYMBOL, SCALE, AXIS, and OFFSET. A particular application may not need all of these routines, depending on the user's requirements.

- PLOTS initializes a plotting sequence.
- PLOT moves the print head in a straight line to a new position.
- SETUP specifies the print pitch, the plot print character, the line density, and the print mode.
- FACTOR enlarges or reduces the size of the entire plot.
- WHERE determines the current print head location and scale factor.
- SET resets the origin without moving the print head or the paper.
- HLINE draws a horizontal line.
- VLINE draws a vertical line.
- SYMBOL prints one or more characters of text anywhere on a plot.
- SCALE scans the user's data array to determine the optimum starting value and number of data units per inch along the axis.
- AXIS uses the information obtained by SCALE in order to produce a scaled, annotated axis.
- OFFSET uses the information obtained by SCALE in order to change the scale of the points to be plotted to correspond to the scale of the axes.

PLOTS

PLOTS initializes a plotting sequence. It puts the terminal in Ultraplot mode, sets the origin at the present print head location, and selects the dot as the plotting character.

The call to PLOTS must be made before any other plotting calls. If the terminal is taken out of plotting mode (by specifying that option in PLOT), then PLOTS must be called again before any other plotting calls.

The PLOTS calling statement has no arguments:

```
CALL PLOTS
```

PLOT

PLOT is used to move the print head in a straight line to a new position. The line may or may not be printed, depending on the option specified.

A series of points in the form of a curve may be plotted with a DO loop using a single PLOT calling statement in which X and Y are used to reference an array containing both horizontal and vertical coordinates. This array should also be referenced in SCALE, AXIS, and OFFSET.

The PLOT calling statement has three arguments:

```
CALL PLOT (X,Y,±IPEN)
```

- X The X coordinate for the end point of the line, specified as a real variable. Determines the length of the horizontal movement relative to the origin. May be given in either inches or plot units, depending on the value of IPEN.
- Y The Y coordinate for the end point of the line, specified as a real variable. Determines the length of the vertical movement relative to the origin. May be given in either inches or plot units, depending on the value of IPEN.
- ±IPEN Integer variable which specifies pen up/down status, origin definition, line density, whether print head movements are to be measured in inches or plot units, and whether the terminal is to be taken out of plotting mode.

If IPEN = +2, the pen is down during movement. A straight line will be printed from the current print head location to location (X,Y), with a density of 1. X and Y are given in inches.

If IPEN = +3, the pen is up during movement. The print head will be transferred to (X,Y) with no line being drawn. X and Y are given in inches.

If IPEN = +4, the pen is down during movement. A straight line will be printed from the current print head location to location (X,Y), with a density determined by a previous call to SETUP. X and Y are given in inches.

If IPEN = +92, same as IPEN = +2, except X and Y are given in plot units.

If IPEN = +93, same as IPEN = +3, except X and Y are given in plot units.

If IPEN = +94, same as IPEN = +4, except X and Y are given in plot units.

If IPEN = +999, the pen is up during movement. The print head will be transferred to (X,Y) with no line being drawn. X and Y are given in inches. After movement, the terminal will be taken out of plotting mode. Note: The last plotting call in a program must be a call to PLOT with IPEN = +999, or the program will not execute properly.

If IPEN = -2, -3, -4, -92, -93, or -94, a new origin is defined at the end point (X,Y) after the movement is completed. Otherwise, same as if IPEN were positive.

Examples of various calls to PLOT are shown below:

```
CALL PLOT(-5.0,0.0,-3)
CALL PLOT(2.0,-1.0,-3)
CALL PLOT(0.0,0.0,2)
CALL PLOT(3.0,2.0,2)
CALL PLOT(0.0,-4.0,-3)
CALL SETUP(4,7)
CALL PLOT(3.0,2.0,4)
CALL PLOT(0.0,-4.0,-3)
CALL PLOT(180.0,96.0,92)
CALL PLOT(0.0,-192.0,-93)
CALL PLOT(-5.0,-2.0,999)
```

SETUP

SETUP is used to specify the print pitch, the plot print character (if other than a dot), the line density, and whether print enhancement mode is desired.

The SETUP calling statement has two arguments:

```
CALL SETUP (IOPT,IVAL)
```

IOPT Integer variable which specifies the option to be exercised during this call to SETUP.

IVAL Integer variable (except in the case of IOPT=3) which corresponds to the particular option selected.

If IOPT = 1, IVAL defines the horizontal pitch. This feature permits character strings along the horizontal axis to be condensed or expanded at will. Values may range from -99 to +99, the distance between characters varying from 1/60 inch to 99/60 inch on the horizontal axis. The print head will move to the right if IVAL is positive and to the left if IVAL is negative.

If IOPT = 2, IVAL defines the vertical pitch. This feature permits character strings along the vertical axis to be condensed or expanded at will. Values may range from -99 to +99, the distance between characters varying from 1/48 inch to 99/48 inch on the vertical axis. The paper will move down if IVAL is positive and up if IVAL is negative.

If IOPT = 3, IVAL is the name of a real variable which contains the binary representation of the new plot print character. See Appendix B for specific format information.

If IOPT = 4, IVAL defines the new plot line density. This feature permits the printing of plot characters to be condensed or expanded at will. Values may range from 1 to 7, the distance between characters varying from 1/60 inch to 7/60 inch on the horizontal axis and varying from 1/48 inch to 7/48 inch on the vertical axis.

If IOPT = 5, IVAL indicates whether the print character is to be struck once or twice by the print hammer. Print enhancement mode is two strikes.

IVAL = 1 Normal print mode
IVAL = 2 Print enhancement mode

Note: If SETUP is not called, the following default values are used:

Horizontal pitch = +6 (10 characters per inch)
Vertical pitch = -8 (6 lines per inch)
Plot character = dot
Plot density = 1
Normal print mode

Examples of various calls to SETUP are shown below:

```
DATA AST/00520000000000000000B/  
DATA DOT/00560000000000000000B/  
CALL SETUP(1,12)  
CALL SETUP(2,-16)  
CALL SETUP(3,AST)  
CALL SETUP(4,6)  
CALL SETUP(5,2)  
CALL PLOT(2.0,3.0,4)  
CALL SETUP(3,DOT)  
CALL PLOT(4.0,0.0,2)  
CALL SETUP(5,1)
```

FACTOR

FACTOR is used to enlarge or reduce the size of the entire plot. During debugging of a plotting application program, computer and plotting time can be saved by reducing the size of the entire plot. However, users are warned to exercise caution when reducing plot size because SYMBOL and AXIS cannot reduce the size of the print characters or numbers and this may cause overprinting.

The FACTOR calling statement has one argument:

CALL FACTOR (F)

F The scale factor, which indicates the ratio of the desired plot size to the normal plot size, specified as a real variable. Default value is 1.0.

For example, if $F = 2.0$, all subsequent print head movements which are given in inches will be twice their normal size. If $F = 0.5$, all subsequent print head movements given in inches will be half their normal size. When F is reset to 1.0, plotting returns to normal size. FACTOR has no effect on print head movements given in plot units.

An example of a call to FACTOR is shown below:

```
CALL FACTOR(0.75)
```


WHERE

WHERE is used to determine the current print head location and scale factor. This permits the user to optimize print head movement.

For example, at the end of a plotting sequence or after printing a long character string, the user might wish to obtain the current location of the print head as well as the current scale factor.

The WHERE calling statement has three arguments:

```
CALL WHERE (X,Y,RFACT)
```

- X User-supplied name of the real variable which will later contain the X coordinate for the current print head location, relative to the last established origin.
- Y User-supplied name of the real variable which will later contain the Y coordinate for the current print head location, relative to the last established origin.
- RFACT User-supplied name of the real variable which will contain the current scale factor upon return.

The value of RFACT at the time the subroutine is called determines whether X and Y will be returned in inches or plot units (90 = inches, 90 = plot units).

An example of a call to WHERE is shown below:

```
DATA SCALE/89.0/  
CALL WHERE (XVALUE,YVALUE,SCALE)
```

SET

SET is used to reset the origin without moving the print head or the paper.

The SET calling statement has three arguments:

CALL SET (X,Y,J)

- X The new value given to the X coordinate of the current print head location, specified as a real variable.
- Y The new value given to the Y coordinate of the current print head location, specified as a real variable.
- J Integer variable which specifies whether X and Y values are given in inches or plot units, and which governs the status of PLOT.

If J = 2, X and Y are in inches, and a point will not be plotted at the current location with the next call to PLOT.

If J = 3, X and Y are in inches, and a point may be plotted at the current location with the next call to PLOT.

If J = 92, X and Y are in plot units, and a point will not be plotted at the current location with the next call to PLOT.

If J = 93, X and Y are in plot units, and a point may be plotted at the current location with the next call to PLOT.

An example of a call to SET is shown below:

CALL SET(0.0,0.0,3)

HLINE

HLINE is used to draw a horizontal line using the underscore character. Lines can be printed either from left to right or from right to left.

The HLINE calling statement has three arguments:

```
CALL HLINE (X,Y,XLEN)
```

- | | |
|------|----------------------------------------------------------------------------------------------------------------------------|
| X | The X coordinate for the starting location of the line, given in inches and specified as a real variable. |
| Y | The Y coordinate for the starting location of the line, given in inches and specified as a real variable. |
| XLEN | The length of the line in inches, specified as a real variable. A negative value produces a line drawn from right to left. |

An example of a call to HLINE is shown below:

```
CALL HLINE(1.0,-7.0,6.0)
```

VLINE

VLINE is used to draw a vertical line using the vertical bar character. Lines can be drawn either upward or downward.

The VLINE calling statement has three arguments:

```
CALL VLINE (X,Y,YLEN)
```

- X The X coordinate for the starting location of the line, given in inches and specified as a real variable.
- Y The Y coordinate for the starting location of the line, given in inches and specified as a real variable.
- YLEN The length of the line in inches, specified as a real variable. A negative value produces a line drawn downward.

An example of a call to VLINE is shown below:

```
CALL VLINE(1.0,-7.0,-6.0)
```

SYMBOL

SYMBOL is used to print one or more characters of text anywhere on a plot. The subroutine is typically used for annotating the X and Y axes and other areas of the diagram. A string of characters can be printed at four different angles: 0, 90, 180, or 270 degrees. The character string can be printed starting either at the location (X,Y) or immediately following the last character of the text previously printed by SYMBOL.

The SYMBOL calling statement has five arguments:

```
CALL SYMBOL (X,Y,TEXT,ANGLE,NCHAR)
```

- X The X coordinate for the starting location of the text, given in inches and specified as a real variable.
- If X = 999.0, the text will be printed in the X-location immediately following the last character previously printed by SYMBOL.
- Y The Y coordinate for the starting location of the text, given in inches and specified as a real variable.
- If Y = 999.0, the text will be printed in the Y-location immediately following the last character previously printed by SYMBOL.
- TEXT The character string to be printed, specified as a real variable in display code representation (H-type format). The character(s) must be left-justified and contiguous within a single variable, an array, or a Hollerith literal.
- ANGLE The angle at which the text will be printed, specified as a real variable. Allowable values are 0 degrees and 270 degrees, measured counterclockwise from the positive X-axis (90 degrees or 180 degrees can be achieved by changing the sign on the horizontal or vertical pitch through SETUP).
- NCHAR Integer variable which specifies the exact number of characters to be printed.

Examples of various calls to SYMBOL are shown below:

```
DIMENSION HEADER(2)
DATA HEADER/10HA SAMPLE T,4HTITLE/
CALL SYMBOL(3.0,4.0,HEADER,0.0,14)
CALL SYMBOL(3.0,2.0,9HSUB-TITLE,0.0,9)
CALL SYMBOL(3.0,2.0,16HLONGER SUB-TITLE,0.0,16)
CALL SYMBOL(3.0,0.0,8HVERTICAL,270.0,8)
CALL SYMBOL(999.0,999.0,6H TITLE,270.0,6)
```

SCALE

SCALE is used to scan the user's data, which is generally in the form of an array of (X,Y) data points. The subroutine then picks scale values in such a way that the data will fall entirely within a specified axis length. It also calculates the scale so that the axis can be marked in convenient, rounded numbers at one-inch intervals.

The output of SCALE is FIRSTV and DELTAV. FIRSTV is the starting value which will appear as the first annotation on the axis at the origin. DELTAV is the number of data units per inch along the axis. SCALE adjusts DELTAV so that it is always an interval of 1, 2, 4, 5, or 8×10^n (where n is an exponent consistent with the original unadjusted scale factor).

FIRSTV and DELTAV for X values and Y values are stored by SCALE at the end of the user's data array. These values are used by AXIS to produce annotated axes. They are also used by OFFSET to scale the plot points so they will correspond to the scale of the axes.

A data array normally contains both X and Y data points stored alternately, so that X and Y coordinates for each point are paired, with the X values in the array having subscripts 1, 3, 5, ...n-1, and the Y values having subscripts 2, 4, y, ... n. The X and Y values are usually passed to subroutine SCALE in two calls. SCALE thus treats the data array as two "separate" arrays. To pass the "X-array" and the "Y-array" separately, an increment of 2 is specified in each call so that only every second data point is scanned. The "X-array" is passed to SCALE using starting subscript 1, and the "Y-array" is passed to SCALE using starting subscript 2.

SCALE stores FIRSTV and DELTAV values at the end of the user's data array in the following manner: If the array contains n number of alternating X and Y data values, then

```

FIRSTV for X values is stored in location n + 1,
FIRSTV for Y values is stored in location n + 2,
DELTAV for X values is stored in location n + 3,
DELTAV for Y values is stored in location n + 4.

```

To allow for these values, the FORTRAN dimension statement for the data array must include four additional words at the end of the array.

The SCALE calling statement has four arguments:

```
CALL SCALE (ARRAY,AXLEN,NPTS,INC)
```

ARRAY The first element of the array of data points to be examined, specified as a real variable.

AXLEN The length of the desired axis in inches, specified as a real variable.

NPTS Integer variable which specifies the number of data points to be scanned in the array.

INC Integer variable which specifies the increment between consecutive data points which are to be scanned.

If INC is positive, FIRSTV will contain a minimum scale value upon return and DELTAV will be positive.

If INC is negative, FIRSTV will contain a maximum scale value upon return and DELTAV will be negative.

Examples of calls to SCALE are shown below:

```
DIMENSION XYARRAY(12)
DATA XYARRAY/1.0,3.0,4.0,9.5,7.0,-5.0,10.0,14.5,4*0.0/
CALL SCALE(XYARRAY(1),5.0,4,2)
CALL SCALE(XYARRAY(2),8.0,4,2)
```


AXIS

AXIS is used to produce a scaled, annotated axis. The axis can be horizontal or vertical and positively or negatively directed. The clockwise or counterclockwise placement of the annotation in relation to the axis is always determined from the starting point of the axis (X,Y).

The AXIS calling statement has eight arguments:

```
CALL AXIS (X,Y,TITLE,NCHAR,AXLEN,ANGLE,FIRSTV,DELTAV)
```

- X The X coordinate for the starting location of the axis, given in inches and specified as a real variable.
- Y The Y coordinate for the starting location of the axis, given in inches and specified as a real variable.
- TITLE The character string to be printed as the axis title, specified as a real variable in display code representation (H-type format). The characters must be left-justified and contiguous within a single variable, an array, or a Hollerith literal.

- NCHAR Integer variable which specifies the exact number of characters to be printed (nm) and the type of annotation.

If NCHAR = nm or lnm, the axis will be annotated with scale values and a scale factor.

If NCHAR = 2nm, the axis will be annotated with scale values but without a scale factor.

If NCHAR = 3nm, neither scale values nor a scale factor will be printed on the axis.

For any of the three options listed above:

If nm = 0, no title will be printed for the axis.

If xnm is positive, all annotations will be printed on the counterclockwise side of the axis.

If xnm is negative, all annotations will be printed on the clockwise side of the axis.

- AXLEN** The length of the axis in inches, specified as a real variable. A negative value reverses the angle specified in **ANGLE**.
- ANGLE** The angle at which the axis will be drawn, specified as a real variable. Allowable values are 0 degrees and 90 degrees, measured counterclockwise from the positive X-axis (180 degrees and 270 degrees can be achieved by specifying a negative value for **AXLEN**).
- FIRSTV** The starting value which will appear as the first annotation on the axis at the graph origin (X,Y), specified as a real variable. **FIRSTV** may be calculated by **SCALE** and stored at the end of the user's data array, or it may be determined by the user and stored anywhere.
- DELTAV** The number of data units per inch along the axis, specified as a real variable. **DELTAV** may be calculated by **SCALE** and stored at the end of the user's data array, or it may be determined by the user and stored anywhere. The value of **DELTAV** is added to **FIRSTV** for each succeeding one-inch division along the axis.

Examples of calls to **AXIS** are shown below:

```

DIMENSION XYARRAY(12)
DATA XYARRAY/1.0,3.0,4.0,9.5,7.0,-5.0,10.0,14.5,4*0.0/
DATA XTITLE/4HTIME/
DATA YTITLE/5HSCORE/
CALL SCALE(XYARRAY(1),5.0,4,2)
CALL SCALE(XYARRAY(2),8.0,4,2)
CALL AXIS(1.0,-8.0,XTITLE,-4,6.0,0.0,XYARRAY(9),XYARRAY(11))
CALL AXIS(1.0,-8.0,YTITLE,5,8.0,90.0,XYARRAY(10),XYARRAY(12))

```

OFFSET

OFFSET is used whenever the scale of the axes has been changed by SCALE. OFFSET causes the scale of the points to be plotted to correspond to the scale of the axes. The scaled plotting values produced by OFFSET are then used by PLOT to plot the user's data in the area specified by the axes.

The OFFSET calling statement has four arguments:

```
CALL OFFSET (XO,XF,YO,YF)
```

- XO The starting X-value which will appear as the first annotation on the X-axis at the graph origin, specified as a real variable. XO is equivalent to FIRSTV for X-values, which is calculated by SCALE and stored at the end of the user's data array.
- XF The number of data units per inch along the X-axis, specified as a real variable. XF is equivalent to DELTAV for X-values, which is calculated by SCALE and stored at the end of the user's data array.
- YO The starting Y-value which will appear as the first annotation on the Y-axis at the graph origin, specified as a real variable. YO is equivalent to FIRSTV for Y-values, which is calculated by SCALE and stored at the end of the user's data array.
- YF The number of data units per inch along the Y-axis, specified as a real variable. YF is equivalent to DELTAV for Y-values, which is calculated by SCALE and stored at the end of the user's data array.

Examples of calls to OFFSET are shown in the sample plotting program at the beginning of the manual.

APPENDIX A

Translation of Special Characters

To be recognized by the AJ 832 terminal, a character must be specified in octal ASCII code and must be right-justified within one of the five 12-bit components of a 60-bit word. The special characters listed below are not automatically translated by the FORTRAN compiler from display code to ASCII. To ease the system programmer's task of translating characters into this particular format, the following translation table is included:

Special Character	ASCII Binary Representation
Idle	/00000000000000000000B/
Backspace	/00100000000000000000B/
Line feed	/00120000000000000000B/
Vertical tab	/00130000000000000000B/
Escape	/00330000000000000000B/
Delete	/01770000000000000000B/

Note that for documentation purposes, only the first of the five 12-bit components of each 60-bit word were used, and the remaining four components were filled with idle characters. Such a strategy may simplify programming, but processing time is improved if five distinct characters are sent in each word.

In addition to the special characters listed above, the AJ 832 terminal recognizes the graphic characters translated in Appendix B.

APPENDIX B

Translation of Graphic Characters

The AJ 832 terminal recognizes the graphic characters listed below. The FORTRAN compiler translates most of these characters automatically if they are specified in the user's program in display code representation (H-type format). In certain cases, however, the user must translate the character into its ASCII binary equivalent. For example, if a new plot print character is desired (using a call to SETUP with IOPT = 3), the character must be specified in binary representation within the user's program, using the following translation table:

Graphic Character	ASCII Binary Representation
A	/01010000000000000000B/
B	/01020000000000000000B/
C	/01030000000000000000B/
D	/01040000000000000000B/
E	/01050000000000000000B/
F	/01060000000000000000B/
G	/01070000000000000000B/
H	/01100000000000000000B/
I	/01110000000000000000B/
J	/01120000000000000000B/
K	/01130000000000000000B/
L	/01140000000000000000B/
M	/01150000000000000000B/
N	/01160000000000000000B/
O	/01170000000000000000B/
P	/01200000000000000000B/
Q	/01210000000000000000B/
R	/01220000000000000000B/
S	/01230000000000000000B/
T	/01240000000000000000B/
U	/01250000000000000000B/
V	/01260000000000000000B/
W	/01270000000000000000B/
X	/01300000000000000000B/
Y	/01310000000000000000B/
Z	/01320000000000000000B/
0	/00600000000000000000B/
1	/00610000000000000000B/
2	/00620000000000000000B/
3	/00630000000000000000B/

Graphic Character	ASCII Binary Representation
4	/00640000000000000000B/
5	/00650000000000000000B/
6	/00660000000000000000B/
7	/00670000000000000000B/
8	/00700000000000000000B/
9	/00710000000000000000B/
+	/00530000000000000000B/
-	/00550000000000000000B/
*	/00520000000000000000B/
/	/00570000000000000000B/
(/00500000000000000000B/
)	/00510000000000000000B/
\$	/00440000000000000000B/
=	/00750000000000000000B/
blank	/00400000000000000000B/
,	/00540000000000000000B/
.	/00560000000000000000B/
#	/00430000000000000000B/
[/01330000000000000000B/
]	/01350000000000000000B/
%	/00450000000000000000B/
"	/00420000000000000000B/
'	/01370000000000000000B/
!	/00410000000000000000B/
&	/00460000000000000000B/
'	/00470000000000000000B/
?	/00770000000000000000B/
<	/00740000000000000000B/
>	/00760000000000000000B/
@	/01000000000000000000B/
\	/01340000000000000000B/
>	/01360000000000000000B/
;	/00730000000000000000B/
:	/00720000000000000000B/
]	/00340000000000000000B/
+	/00350000000000000000B/
	/01740000000000000000B/