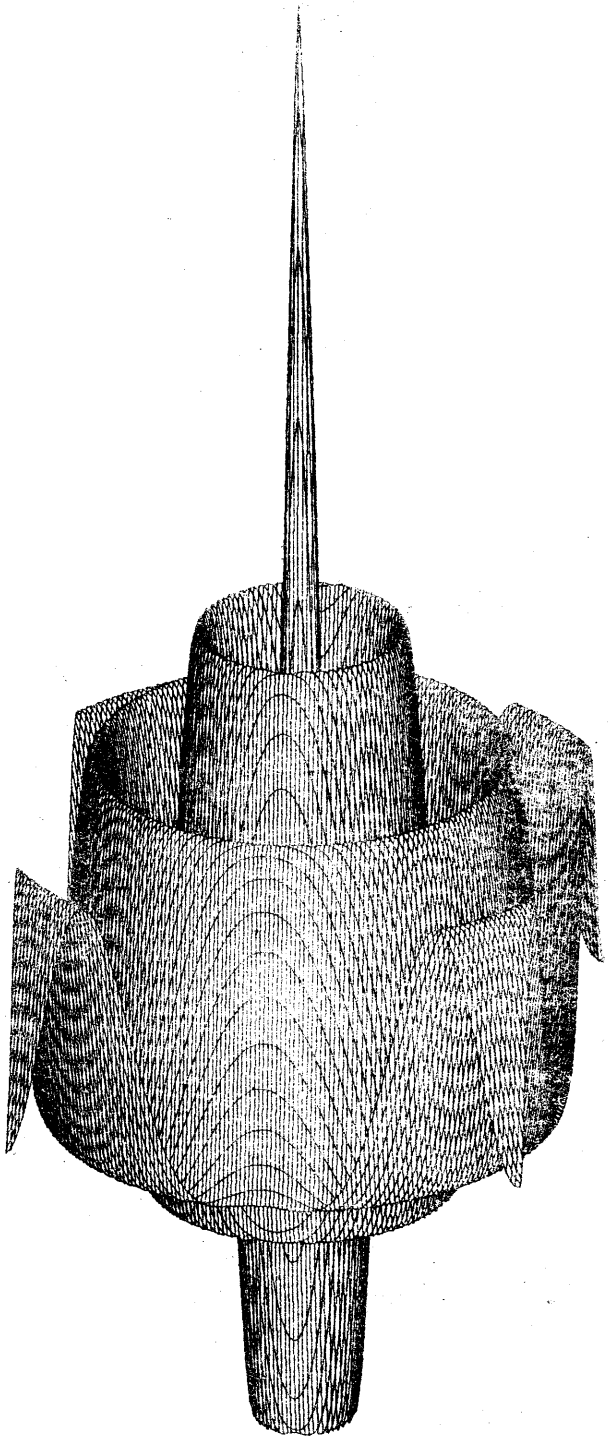
GRAPHICAL DISPLAY SYSTEM



April 1969

Computer Center

University of California
Berkeley

Graphical Display System

April 1969

Arthur R. Paradis
David F. Hussey

Computer Center

University of California
Berkeley

PREFACE

When one thinks of computer-oriented graphics or displays, one can conjure up various images, from simple graphs of data to flow charts to orthographic projections to dynamic real-time portrayal of information or data to complicated animated movies. Behind these exciting visual pictures which are drawn or projected is a maze of programs and routines which transform raw data into the exacting movements of the display device.

These routines fall into a broad spectrum of capability, complexity and ease of use, from complete plot packages with little user decisions on one end to routines which move the drawing instrument one increment. The Graphical Display System (GDS) stands above and in the middle of these two extremes. It is a set of subroutines which produce fundamental graphic entities independent of any given display device.

By being concerned with the basic elements of a display, such as

a grid in data representation graphics,

a rectangle in diagramatic graphics,

or a projection in three-dimensional graphics,

GDS allows the user freedom to choose his plotting parameters without resorting to detailed descriptions of the movements of the display tool. The routines may then be used as tools in constructing display packages for particular applications.

By being display-device independent, the Graphical Display System is above the routines of the plotting spectrum which are usually designed to work with one specific display device. Programs written using GDS routines may be used within restrictions (most are imposed by the surface area of a display device) on one display device or another with little or no change in the program.

Display-device independence is achieved by describing the graphic entities in terms of basic plotter movements or directives which are placed in an intermediate storage device or file. These movements are then transformed to actual plotter commands by a second independent program, called a post-processor, which is designed for a particular plotter.

Experience has shown that the GDS philosophy is sound. GDS has been successfully implemented on several computer systems and for a number of display devices. The Graphical Display System has proven to be flexible yet stable in the fast-changing computer world.

Acknowledgement must be given to Messrs. Charles Dickens and Don Ruchman of the Boeing AeroSpace Division who originally developed the GDS concept with their Numerical Plotting System. Messrs. David Hussey and Art Paradis have brought GDS to its current stage of development at the University of California, Berkeley and are currently expanding Graphical Display System to include interactive graphics.

Introduction to Plotting

A. Introduction

Welcome! The purpose of this section is to demonstrate how to use the "Graphical Display System" (GDS) by explaining what it does, how to use this manual and by presenting a complete example.

B. Purpose of "Graphical Display System"

The "Graphical Display System" is a flexible method of data display via automatic plotting machines. In proper perspective it is but a topic under the general subject of drafting. A few familiar examples of graphical representation of data are:

1. Rectilinear graphs
2. Logarithmic graphs
3. Semi-logarithmic graphs
4. Polar graphs
5. Nomographs
6. Contour Maps

Traditionally, such graphs and maps are produced by a combination of manual and semi-automatic methods. With due respect to those conventional practices, the Graphical Display System affords its users a competitive plotting operation, one that complements a computer-centered data reduction operation.

C. Anatomy of a Plot

In general, six basic components of a plot must be considered and specified in order to construct a suitable finished plot:

1. An appropriate coordinate system must be selected, whether it be polar, rectangular, or of some other type.

2. If the selected coordinate system is rectangular, the scale type for each axis must be chosen, whether it be linear, logarithmic, or some other function scale.

3. The coordinate system frame may be either an axes pair or a grid, plus any supplementary axes.

4. The scale range for each scale must be defined.

5. The data representation on the graph may be either by plot lines, curves, and/or plot symbols.

6. Ample annotation is needed to finish the plot.

D. "Graphical Display System" Organization

The Graphical Display-System employs two steps for generating plots, namely, the Graphical Display subroutines and the plotter-oriented post-processing programs. The Graphical Display subroutines are independent of the plotter. These routines reduce the plot specifications into general construction codes for input to the post-processing program(s). The purpose of the post-processing program is to translate the general construction codes into a series of plotter-oriented commands that drive the plotter to construct the plot.

Graphical Display Subroutines (Step 1)

The Graphical Display subroutines constitute an open-ended set of graphic routines (routines may be added) written on many levels and for many types of computer graphics.

The description of the routines are first separated into different levels (complete plot routines, component plot routines, and basic routines) which will enable the user to select the degree of versatility he wants and the degree of effort he wishes to expend. These levels are then grouped into different areas of graphics (general routines, contouring, diagramming, 3-D plots, etc.). The following table explains more fully the purpose and scope of each level.

Complete Plot Routines -- These routines are for the person who wants an easy to use, painless plot program. Each call produces a complete plot. Most of the decisions such as axis length, type and amount of annotation, etc. are predefined. These complete plot routines are grouped into areas of plotting (general, contouring, diagramming, 3-D, etc.). The general index and the chapter entitled "Short Guide to Plotting Routines Available" contains a list of these routines. The write-ups for these routines will be described in a separate volume (extended chapter).

Component Plot Routines -- These routines offer a great deal more versatility than the "complete plot routines". Each call produces one component of the plot (i.e., grid, data line, axis annotation, title, etc.). These routines are combined into one coherent set of routines by the use of a general plot specifications array (SPECS). These routines are grouped into different areas of plotting (general, contouring, diagramming, 3-D, etc.). See the general index and the chapter entitled "Short Guide to Plotting Routines Available" for a list of these routines. This manual contains the descriptions for the General User routines. The routines in the other areas of plotting will appear in separate volumes (extended chapters).

Basic Plot Routines -- These consist of the subroutines used by the "component plot routines". There are two general sub-classifications -- utility routines (character packing. etc.) and low-level component routines. The low-level component routines are characterized by more complicated calling sequences and control over certain plot specifications (i.e., length of tick marks). See the general index and the chapter entitled "Short Guide to Plotting Routines Available" for a list of these routines. These routines will appear in a separate volume (extended chapter).

Post-Processing Programs (Step 2)

These programs translate the general plot construction codes generated by the above routines into plotter oriented commands. There is one post-processing routine for each plotter available. See the chapter entitled "Post-processor" for a detailed description of each post-processor (this volume).

E. How to use the GDS manual

The manual is divided into the following chapters:

Chapter	Title	Purpose
I	Introduction	Introduction; guide for use of manual; complete example.
II	The *SPECS* Array	Detailed description of the *SPECS* Array: The *SPECS* array is used to specify the various plot parameters (i.e., axis length, scale range, etc.).
III	Short Guide to Plotting Routines Available	A list of plotting routines available: The routines are grouped according to function and a small description of purpose is given for each.
IV	Graphical Display Subroutines	Write-ups for the individual subroutines grouped according to function: Each group has an introduction which gives the purpose, method, and restrictions of each routine. Each individual write-up states the purpose, calling statement, explains the call list arguments, and gives an example or diagram.
V	Post-Processors	This section has one write-up for each type of plotting device.
VI	Control Card Usage	This section shows which control cards are necessary to run your program.
VII	Diagnostics	Description of error checking (in post-processors) and diagnostic aids.

Although the manual is large, it is intended to be easy to use. When using this manual, the user should follow these steps:

1. Formulate his plot requirements (i.e., plot type, size, annotation required, etc.).
2. Make a sketch of what he wants the result to look like.
3. Check the post-processor write-ups to see which plotting device best suits his purposes. Note any restrictions or options associated with that particular plotter.
4. Check through Chapter III (Short Guide to Plotting Routines Available) to find out which routines he needs to produce his plot.
5. Then turn to the individual write-ups. Refer to the main section associated with that write-up for a description of the method or restrictions. See the individual write-up for calling sequence and an example of usage.
6. The individual write-ups list which elements of the array *SPECS* are needed. Turn to Chapter II (The *SPECS* Array) for a detailed description of any element of the *SPECS* array.

7. **Setup and make the calls for the different plot components (i.e., grids, plot lines, annotation, etc.).**

8. **Refer to Chapter VI for control card usage.**

9. **Diagnostics are discussed in Chapter VII.**

Good Luck when writing your program!

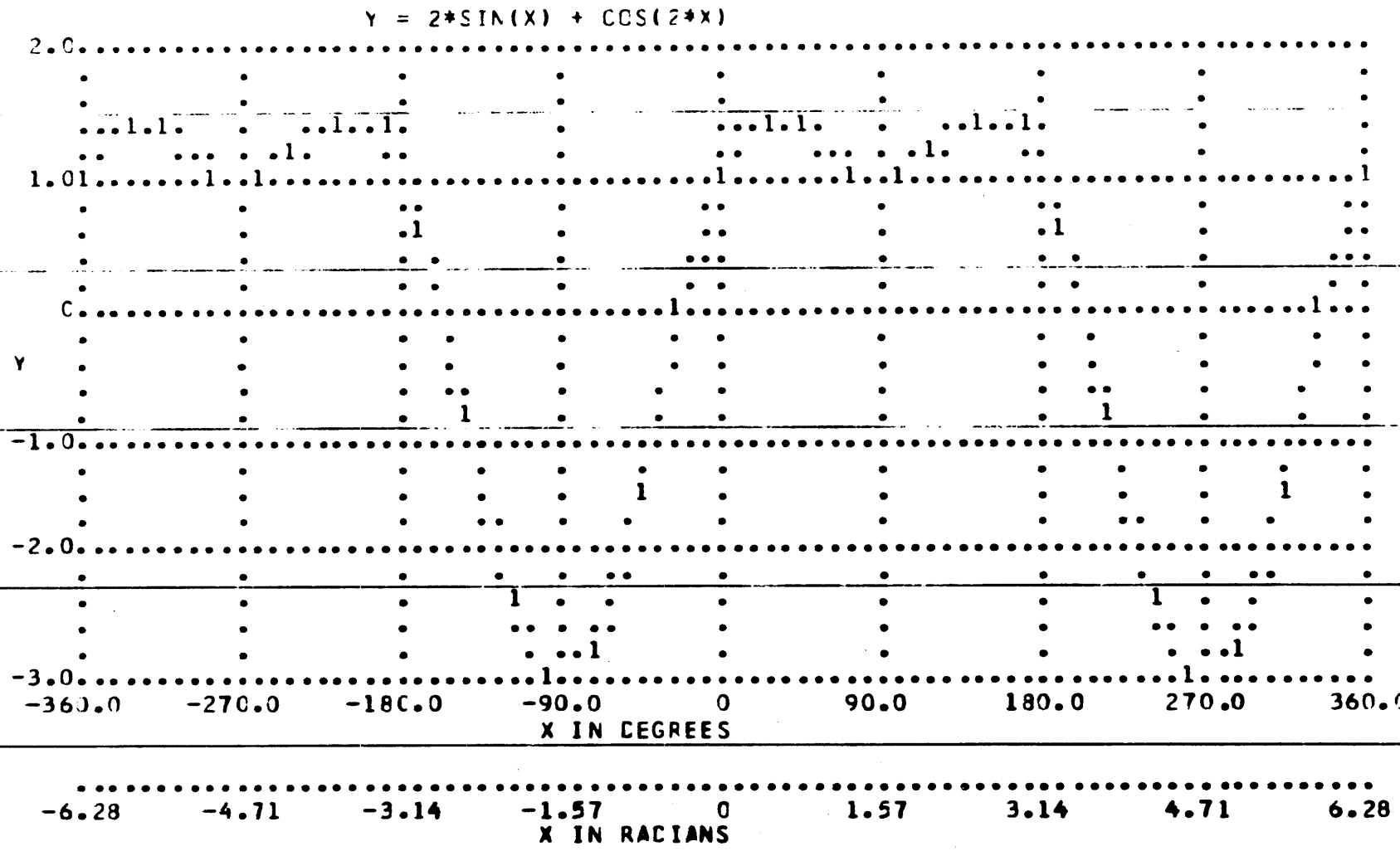
F. Complete Example

```
      PROGRAM PLOT(OUTPUT,TAPE99)
      REAL SPECS(30),X(50),Y(50),BUFX(500),BUFY(500),GIVEN(3)
C        SETUP SPECS ARRAY
C        SET XDIST,YDIST
      SPECS(1) = 1.5
      SPECS(2) = 3.3
C        SET XRIGHT, XLEFT
      SPECS(3) = 360.
      SPECS(4) = -360.
C        SET XLNGTH, YLNGTH
      SPECS(7) = 8.
      SPECS(8) = 4.
C        SET XDIV  (YDIV WILL BE CALCULATED BY FABLIY)
      SPECS(9) = 8.
C        SPECIFY TOOL
      SPECS(11) = 1.
C        SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C        POINTS WILL BE SPECIFIED LATER
C        SET XSKIPS, YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C        GENERATE SOME DATA
      DO 10 I = 1,31
      DEG = -384 + 24*I
      RAD = DEG*0.0174532925
      X(I) = DEG
      Y(I) = 2.*SIN(RAD)+COS(2.*RAD)
   10 CONTINUE
C        SPECIFY POINTS
      SPECS(13) = 31.
C        FIND THE MIN AND MAX
      YMIN = Y(1)
      YMAX = Y(1)
      DO 20 I = 2,31
      IF(Y(I)-YMIN) 12,20,15
   12 YMIN = Y(I)
      GO TO 20
   15 IF(Y(I)-YMAX) 20,20,17
   17 YMAX = Y(I)
   20 CONTINUE
C        SETUP THE GIVEN ARRAY FOR FABLIY
      GIVEN(1) = YMAX
      GIVEN(2) = YMIN
      GIVEN(3) = 6.
```

```
C        FABLIY FINDS A NICE Y SCALE RANGE
      CALL FABLIY(GIVEN,SPECS)
C        GDLILI DRAWS THE GRID -- NO TICK MARKS
      CALL GDLILI(SPECS)
C        SPECIFY THE SYMBOL NUMBER
      SPECS(16) = 1.
C        SPECIFY CHARACTER SIZE
      SPECS(17) = 0.1
      SPECS(18) = 0.1
      SPECS(19) = 0.0
C        SPECIFY THE ANGLE OF LINE ROTATION IN DEGREES
      SPECS(20) = 0.0
C        SPECIFY THE FONT TYPE
      SPECS(21) = 1.
C        SPECIFY THE NUMBER OF DECIMAL PLACES
      SPECS(28) = 1.
C        INITIALIZE THE ZONE VALUES
      SPECS(24) = 0.05
      SPECS(25) = 0.05
      SPECS(26) = 0.05
      CALL NODLIE(SPECS)
C        NODLIL NUMBERS THE LEFT AXIS TO SPECIFIED NUMBER OF DECIMAL PLACES
      CALL NODLIL(SPECS)
C        RESET THE NUMBER OF DIVISIONS FOR TICK MARKS
      SPECS(9) = 72.
      SPECS(10) = SPECS(10)*10.
C        AXLILI DRAWS AN AXIS PAIR WITH TICK MARKS
      CALL AXLILI(SPECS)
C        PSLILI PLOTS A SYMBOL AT EACH DATA POINT
      CALL PSLILI(X,Y,SPECS)
C        PFLILI DRAWS A SMOOTH CURVE THROUGH THE DATA POINTS
      CALL PFLILI(X,Y,BUFX,BUFY,SPECS)
C        RESET THE CHARACTER SIZE
      SPECS(17) = 0.2
      SPECS(18) = 0.2
C        TITLEB CONSTRUCTS A BOTTOM CENTERED TITLE
      CALL TITLEB(12HX IN DEGREES,SPECS)
C        TITLEL CONSTRUCTS A LEFT CENTERED TITLE
      CALL TITLEL(1HY,SPECS)
C        TITLET CONSTRUCTS A TOP CENTERED TITLE
      CALL TITLET(23HY = 2*SIN(X) + COS(2*X),SPECS)
C        RESET THE X DIVISIONS FOR A SUPPLEMENTARY AXIS
      SPECS(9) = 16.
C        RESET THE NUMBER OF DECIMAL PLACES
      SPECS(28) = 2.
C        SAXLIB CONSTRUCTS A LINEAR AXIS BELOW THE PLOTTING AREA
      CALL SAXLIE(SPECS)
C        RESET XRIGHT AND XLEFT
      SPECS(3) = 6.283195
      SPECS(4) = -6.283195
C        RESET THE CHARACTER SIZE
      SPECS(17) = 0.1
      SPECS(18) = 0.1
      CALL NODLIE(SPECS)
```

```
      SPECS(17) = 0.2
      SPECS(18) = 0.2
      CALL TITLEB(12HX IN RADIANS,SPECS)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C        SET XSTART AND YSTART
      SPECS(22) = 5.25
      SPECS(23) = 1.5
      RULE = 1.0
C        SYMKEY PLOTS A SYMBOL AND A LINE OF TEXT
      CALL SYMKEY(1.,18H CALCULATED POINTS,SPECS)
C        RESET YSTART
      SPECS(23) = 1.3
C        TITLEG CONSTRUCTS A RANDOMLY PLACED LINE OF TEXT
      CALL TITLEG(1.,30HPFLILI IS USED IN THIS EXAMPLE,SPECS)
      SPECS(23) = 1.1
      CALL TILEG(1.,30HTO DRAW A SMOOTH CURVE THROUGH,SPECS)
      SPECS(23) = 0.9
      CALL TITLEG(1.,21HTHE CALCULATED POINTS,SPECS)
C        GDSEND MUST BE THE LAST PLOTTING CALL
      CALL GDSEND(SPECS)
      STOP
      END
```

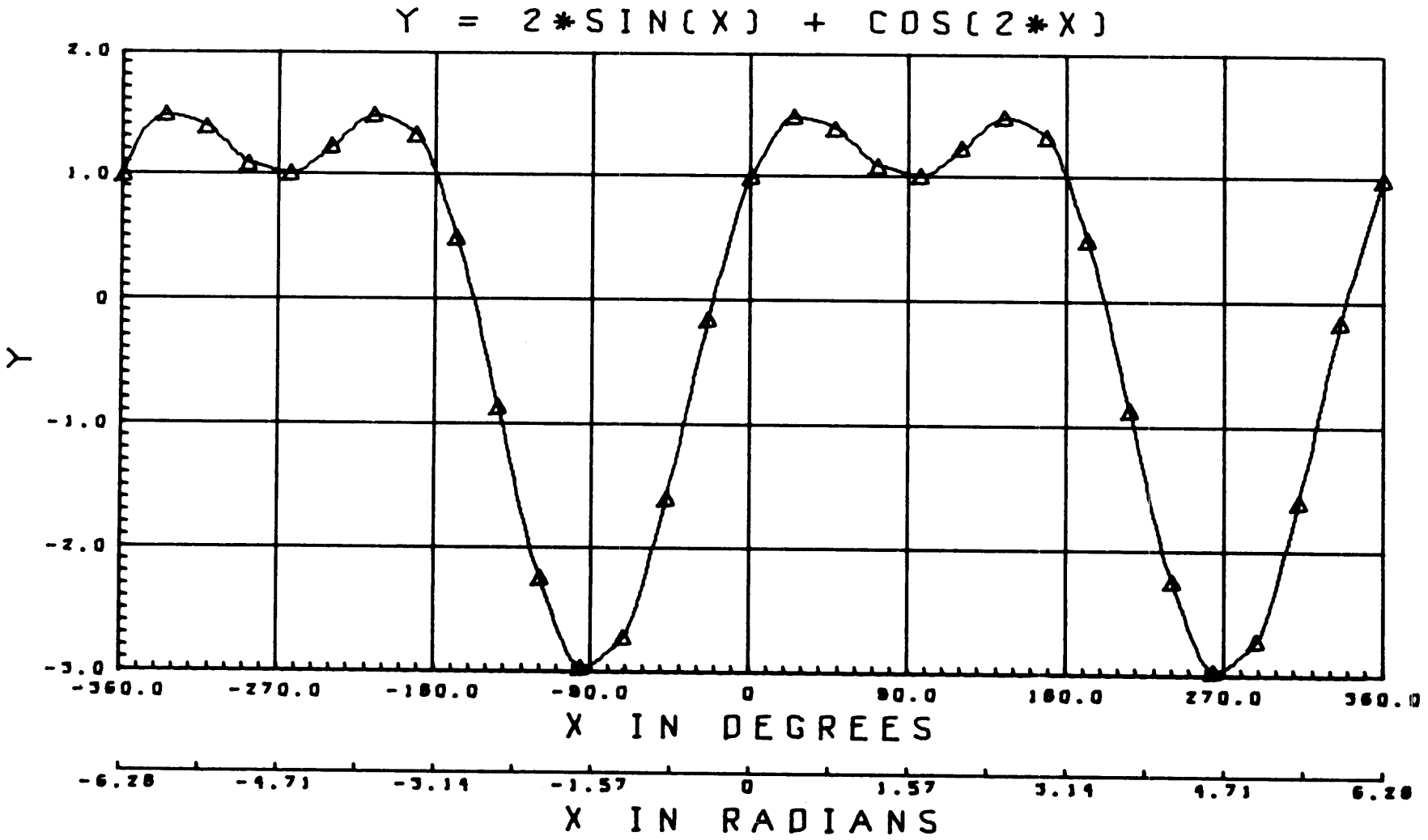
Example of plot from PRINTR post-processor



8-I

1 CALCULATED POINTS
PFLILI IS USED IN THIS EXAMPLE
TO DRAW A SMOOTH CURVE THROUGH
THE CALCULATED POINTS

Example of plot from CALCOMP (Model 565) post-processor



△ CALCULATED POINTS
PFLILI IS USED IN THIS EXAMPLE
TO DRAW A SMOOTH CURVE THROUGH
THE CALCULATED POINTS

The *SPECS* Array

SPECS is an array of floating-point values that specify the construction parameters necessary to produce the various plot components. Table 1 presents the format of *SPECS* and defines each construction parameter; Figure 1 graphically illustrates the various construction parameters.

The use of this standard array by all the system subroutines greatly simplifies their integrated use.

Table 1. DEFINITION OF PARAMETERS IN THE ARRAY *SPECS*

Parameter	Suggested Mnemonic	Definition
SPECS(1)	XDIST	Parameter specifies X component of the vector distance (inches) from a reference point to the lower left corner of the defined plotting area.
SPECS(2)	YDIST	Parameter specifies Y component of the vector distance (inches) from a reference point to the lower left corner of the defined plotting area.
SPECS(3)	XRIGHT	Parameter specifies the extreme X ordinate on the right of the X-scale of a rectangular coordinate system. Parameter specifies one extreme of the ordinate X-scale range. XRIGHT may be larger or smaller than XLEFT.
SPECS(4)	XLEFT	Parameter specifies the extreme X ordinate on the left of the X-scale of a rectangular coordinate system. Parameter specifies one extreme of the ordinate X-scale range.
SPECS(5)	YTOP	Parameter specifies the extreme Y ordinate at the top of the Y-scale of a rectangular coordinate system. Parameter specifies one extreme of the ordinate Y-scale range. YTOP may be larger or smaller than YBOT.
SPECS(6)	YBOT	Parameter specifies the extreme Y ordinate at the bottom of the Y-scale of a rectangular coordinate system. Parameter specifies one extreme of the ordinate Y-scale range.

Table 1 (Continued)

Parameter	Suggested Mnemonic	Definition
SPECS(7)	XLNGTH	Parameter specifies the length (inches) of X-scale. Parameter defines the horizontal dimension of the plotting area.
SPECS(8)	YLNGTH	Parameter specifies the length (inches) of Y-scale. Parameter defines the vertical dimension of the plotting area.
SPECS(9)	XDIV	Parameter specifies the number of equal divisions of the X-scale associated with the defined plotting area.
SPECS(10)	YDIV	Parameter specifies the number of equal divisions of the Y-scale associated with the defined plotting area.
SPECS(11)	TOOL	Parameter specifies the use of a particular plotter-oriented marking, or printing, tool. Normal setting for this parameter is one.
SPECS(12)	CCTAPE	Parameter specifies the logical file to record construction codes generated by the plotting subroutines.
SPECS(13)	POINTS†	Parameter specifies the total number of data points to be plotted.
SPECS(14)	XSKIPS	Parameter, a positive whole number, specifies that every XSKIPth value of the X array, starting with the first value, is to be plotted. For example, if *SPECS(14)* is 1.0, every value of the X array is to be plotted.
SPECS(15)	YSKIPS	Parameter, a positive whole number, specifies that every YSKIPthe value of the Y array, starting with the first value, is to be plotted. For example, if *SPECS(15)* is 3.0, the first value of the Y array and every third value after the first is to be plotted (until the (POINTS-1)*3 + 1 value of the Y array is reached).

†POINTS is not necessarily the number of values of dimensionality in an X or Y array. (POINTS-1)*[X,Y]SKIPS + 1 must not exceed the dimensionality of an X or Y array.

Table 1 (Continued)

Parameter	Suggested Mnemonic	Definition
SPECS(16)	SYMBOL	Parameter specifies the plotter-oriented plot symbol to be used in symbol plotting.
SPECS(17)	FONTB	Parameter specifies the base dimension (inches) of the rectangle in which a plot symbol or alphanumeric character is defined.
SPECS(18)	FONTH	Parameter specifies the height dimension (inches) of the rectangle in which a plot symbol or alphanumeric character is defined.
SPECS(19)	SPACER	Parameter specifies the additional separation distance (inches) between adjacent alphanumeric characters that form a line of text. Normal setting for this parameter is zero.
SPECS(20)	ROTATE	Parameter specifies the counter-clockwise rotation (degrees) to give a line of alphanumeric text.
SPECS(21)	FONTNO	Parameter specifies which alphanumeric font, i.e., character set, to use for annotation work.
SPECS(22)	XSTART	Parameter specifies X component of the vector distance (inches) from the reference point to a point† on the line of annotation specified. The value is positive if the specified point is right of the reference point; otherwise, value is negative.
SPECS(23)	YSTART	Parameter specifies Y component of the vector distance (inches) from the reference point to a point† on the line of annotation specified. The value is positive if the specified point is above the reference point; otherwise, the value is negative.
SPECS(24)	ZONEB	Parameter specifies the distance (inches) to offset construction below the defined plotting area. (Following the construction, certain routines automatically increase this specified offset value enough to avoid an overlap with the next construction below the defined plotting area.)

† The defined point is either the lower left corner of the first character, the lower center of the line, or the lower right corner of the last character depending on a parameter given in the calling routines.

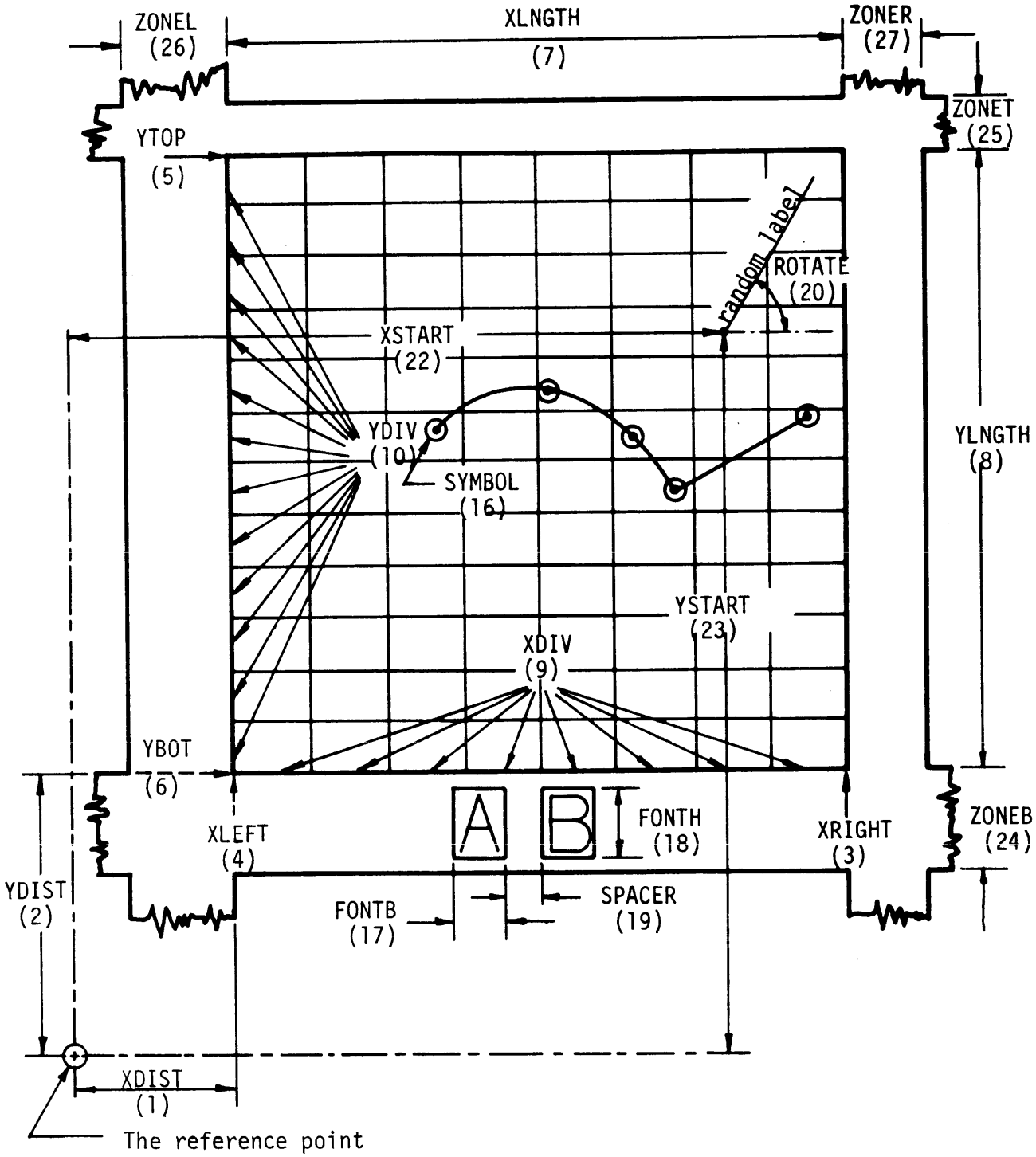
Table 1 (Continued)

Parameter	Suggested Mnemonic	Definition
SPECS(25)	ZONET	Parameter specifies the distance (inches) to offset construction above the defined plotting area. (Following the construction, certain routines automatically increase this specified offset value enough to avoid an overlap with the next construction above the defined plotting area.)
SPECS(26)	ZONEL	Parameter specifies the distance (inches) to offset construction to the left of the defined plotting area. (Following the construction, certain routines automatically increase the specified offset value enough to avoid an overlap with the next construction left of the defined plotting area.)
SPECS(27)	ZONER	Parameter specifies the distance (inches) to offset construction to the right of the defined plotting area. (Following the construction, certain routines automatically increase the specified offset value enough to avoid an overlap with the next construction right of the defined plotting area.)
SPECS(28)	DECPLS	Parameter specifies the number of decimal places to be used when annotating a scale range.
SPECS(29)	SIGFIG	Parameter specifies the maximum number of significant figures to be used when annotating a scale range.
SPECS(30)	STAPE	Parameter specifies the logical file† to be used as a scratch tape by some of the GDS routines.

† The file specified by STAPE must be declared on the program card.

Table 1 (Continued)

Parameter	Suggested Mnemonic	Definition
The following parameters are used only in the contouring routines.		
SPECS(31)	XCOLMS	Parameter specifies the size of the array X which contains the X ordinates of the columns for the XCOLMS by YROWS grid of Z values.
SPECS(32)	YROWS	Parameter specifies the size of the array Y which contains the Y ordinates of the rows for the XCOLMS by YROWS grid of Z values.
SPECS(33)	ZLEVELS	Parameter specifies the number of contour levels to be constructed. The array ZLEVEL, which contains the values of the various contour levels to be constructed, must be dimensioned at least the size of ZLEVELS.
SPECS(34)	CGRIDB	Parameter specifies the base size (inches) of a unit grid division of the program-generated "contour grid".
SPECS(35)	CGRIDH	Parameter specifies the height size (inches) of a unit grid division of the program-generated "contour grid".



See post-processor write-ups for the actual position of the Reference Point on any particular plotter.

Figure 1

A Sample Plot Illustrating *SPECS* Parameters

Short Guide to Plotting Routines Available

Routines to Determine Layout of an Axis' Scale

These subroutines determine a layout of an axis' scale when given a minimum scale range and a maximum number of scale subdivisions. A user of either blank (maximum number of scale subdivisions or less) or gridded forms (fixed size of scale subdivisions) can utilize these routines. Ordinate range and layout limits are given by the list array *GIVEN*. The list array *SPECS* is the output array.

Found in Section	Name	Call List	Function
IV-1.4	FABLIX	(*GIVEN,SPECS*)	To determine for an X ordinate range a linear X-scale for a blank form.
IV-1.6	FABLIY	(*GIVEN,SPECS*)	To determine for a Y ordinate range a linear Y-scale for a blank form.
IV-1.8	FAGLIX	(*GIVEN,SPECS*)	To determine for an X ordinate range a linear X-scale for a gridded form.
IV-1.10	FAGLIY	(*GIVEN,SPECS*)	To determine for a Y ordinate range a linear Y-scale for a gridded form.

Routines to Construct a Grid

These subroutines construct the various types of grids. Construction specifications appear in the call list argument SPECS.

Found in Section	Name	Call List	Function
IV-2.4	GDLILI	(SPECS)	To construct a grid having linear subdivisions for both the X and Y ordinates.
IV-2.6	GDLILG	(SPECS)	To construct a grid having a linear subdivision for the X ordinate and logarithmic (base 10) subdivisions for the Y ordinate.
IV-2.8	GDLGLI	(SPECS)	To construct a grid having logarithmic (base 10) subdivisions for the X ordinate and linear subdivisions for the Y ordinate.
IV-2.10	GDLGLG	(SPECS)	To construct a grid having logarithmic (base 10) subdivisions for both the X and Y ordinates.

Routines to Construct a Pair of Rectangular Axes

These subroutines construct a pair of axes which may be used either in conjunction with, or in lieu of, the grid. Construction specifications appear in the call list argument *SPECS*.

Found in Section	Name	Call List	Function
IV-3.2	AXLILI	(*SPECS*)	To construct an axes pair having linear subdivisions for both the X and Y ordinates.
IV-3.4	AXLILG	(*SPECS*)	To construct an axes pair having linear subdivisions for the X ordinate and logarithmic (base 10) subdivisions for the Y ordinate.
IV-3.6	AXLGLI	(*SPECS*)	To construct an axes pair having logarithmic (base 10) subdivisions for the X ordinate and linear subdivisions for the Y ordinate.
IV-3.8	AXLGLG	(*SPECS*)	To construct an axes pair having logarithmic (base 10) subdivisions for both the X and Y ordinates.

Routines to Construct a Supplementary Axis

Found in Section	Name	Call List	Function
IV-4.2	SAXLIB	(SPECS)	To construct below the defined plotting area an X-axis having linear subdivisions.
IV-4.4	SAXLIT	(SPECS)	To construct above the defined plotting area an X-axis having linear subdivisions.
IV-4.6	SAXLIL	(SPECS)	To construct to the left of the defined plotting area a Y-axis having linear subdivisions.
IV-4.8	SAXLIR	(SPECS)	To construct to the right of the defined plotting area a Y-axis having linear subdivisions.
IV-4.10	SAXLGB	(SPECS)	To construct below the defined plotting area an X-axis having logarithmic (base 10) subdivisions.
IV-4.12	SAXLGT	(SPECS)	To construct above the defined plotting area an X-axis having logarithmic (base 10) subdivisions.
IV-4.14	SAXLGL	(SPECS)	To construct to the left of the defined plotting area a Y-axis having logarithmic (base 10) subdivisions.
IV-4.16	SAXLGR	(SPECS)	To construct to the right of the defined plotting area a Y-axis having logarithmic (base 10) subdivisions.

Routines to Symbol-Plot Data

These subroutines plot data according to the coordinate system and construct about the plotted point a plot symbol. Plotting specifications of type and size of symbol appear in the call list array *SPECS*. X and Y coordinate values are given in the call list arrays *X* and *Y* (*R* and *THETA* for PSPOLR).

Found in Section	Name	Call List	Function
IV-5.4	PSLILI	(*X*,*Y*,*SPECS*)	To symbol-plot data onto a rectangular coordinate system having linear function scales for both X and Y ordinates.
IV-5.6	PSLILG	(*X*,*Y*,*SPECS*)	To symbol-plot data onto a rectangular coordinate system having a linear function scale for the X ordinate and a logarithmic (any base) function scale for the Y ordinate.
IV-5.8	PSLGLI	(*X*,*Y*,*SPECS*)	To symbol-plot data onto a rectangular coordinate system having a logarithmic (any base) function scale for the X ordinate and a linear function scale for the Y ordinate.
IV-5.10	PSLGLG	(*X*,*Y*,*SPECS*)	To symbol-plot data onto a rectangular coordinate system having logarithmic (any base) function scales for both the X and Y ordinates.
IV-5.12	PSPOLR	(*R*,*THETA*,*SPECS*)	To symbol-plot data onto a rectangular coordinate system with data given in polar coordinates.

Routines to Construct a Trend Line through Data

Since there is more than one way to pass a trend line through a given set of points, several sets of routines fall under this heading.

Routines to Construct Straight Lines through Data

These subroutines construct straight lines between consecutive data points. No restrictions are imposed on the nature of the input data by the routines. Also, any line that intersects the boundaries of the defined plotting area is constructed for only that portion of the line lying within or on the boundaries. Plotting specifications appear in the call list array *SPECS*. The coordinate values are given in the call list arrays *X* and *Y* (*R* and *THETA* for SLPOLR).

Found in Section	Name	Call List	Function
IV-6.2	SLLILI	(*X*,*Y*,*SPECS*)	To construct straight lines between consecutive data points on a rectangular coordinate system having linear function scales for both the X and Y ordinates.
IV-6.4	SLLILG	(*X*,*Y*,*SPECS*)	To construct straight lines between consecutive data points on a rectangular coordinate system having a linear function scale for the X ordinate and a logarithmic (any base) function scale for the Y ordinate.
IV-6.6	SLLGLI	(*X*,*Y*,*SPECS*)	To construct straight lines between consecutive data points on a rectangular coordinate system having a logarithmic (any base) function scale for the X ordinate and a linear function scale for the Y ordinate.
IV-6.8	SLLGLG	(*X*,*Y*,*SPECS*)	To construct straight lines between consecutive data points on a rectangular coordinate system having logarithmic (any base) function scales for both the X and Y ordinates.
IV-6.10	SLPOLR	(*R*,*THETA*,*SPECS*)	To construct straight lines between consecutive data onto a rectangular coordinate system with data given in polar coordinates.

Routines employing Parabolic Fairing Technique to Construct a Trend Curve through Data

These subroutines construct through the data points a trend curve generated by the non-restrictive, general-purpose, parabolic fairing technique.* Hence, no restrictions are imposed on the nature of the input data by these routines. In addition, should the curve intersect the boundaries of the defined plotting area, then only that portion of the curve lying within or on the boundaries is constructed. Plotting specifications appear in the call list array *SPECS*. *BUFX* and *BUFY* are work regions. The coordinate values are given in the call list arrays *X* and *Y* (*R* and *THETA* for PFPOLR).

Found in Section	Name	Call List	Function
IV-7.4	PFLILI	(*X,Y,BUFX, BUFY,SPECS*)	To construct a trend curve through data points on a rectangular coordinate system having linear function scales for both the X and Y ordinates.
IV-7.6	PFLILG	(*X,Y,BUFX, BUFY,SPECS*)	To construct a trend curve through data points on a rectangular coordinate system having a linear function scale for the X ordinate and a logarithmic (any base) function scale for the Y ordinate.
IV-7.8	PFLGLI	(*X,Y,BUFX, BUFY,SPECS*)	To construct a trend curve through data points on a rectangular coordinate system having a logarithmic (any base) function scale for the X ordinate and a linear function scale for the Y ordinate.
IV-7.10	PFLGLG	(*X,Y,BUFX, BUFY,SPECS*)	To construct a trend curve through data points on a rectangular coordinate system having logarithmic (any base) function scales for both the X and Y ordinates.
IV-7.12	PFPOLR	(*R,THETA,BUFX, BUFY,SPECS*)	To construct a trend curve onto a rectangular coordinate system with data given in polar coordinates.

* See section 4.7 for method used.

Routines to Construct Numerical Values along an Axis' Scale

These subroutines compute and layout a scale of numerical values for an ordinate range. Scale layout and number format are specified in the list array *SPECS*.

Found in Section	Name	Call List	Function
IV-8.4	NODLIB	(*SPECS*)	To construct below the defined plotting area and for a linear X ordinate range a numeric scale when the number of decimal places is specified.
IV-8.6	NODLIT	(*SPECS*)	To construct above the defined plotting area and for a linear X ordinate range a numeric scale to the number of decimal places specified.
IV-8.8	NODLIL	(*SPECS*)	To construct to the left of the defined plotting area and for a linear Y ordinate range a numeric scale to the number of decimal places specified.
IV-8.10	NODLIR	(*SPECS*)	To construct to the right of the defined plotting area and for a linear Y ordinate range a numeric scale to the number of decimal places specified.
IV-8.12	NOSLIB	(*SPECS*)	To construct below the defined plotting area and for a linear X ordinate range a numeric scale to the number of significant figures specified.
IV-8.14	NOSLIT	(*SPECS*)	To construct above the defined plotting area and for a linear X ordinate range a numeric scale to the number of significant figures specified.
IV-8.16	NOSLIL	(*SPECS*)	To construct to the left of the defined plotting area and for a linear Y ordinate range a numeric scale to the number of significant figures specified.
IV-8.18	NOSLIR	(*SPECS*)	To construct to the right of the defined plotting area and for a linear Y ordinate range a numeric scale to the number of significant figures specified.

Routines to Construct Numerical Values along an Axis' Scale (continued)

Found in Section	Name	Call List	Function
IV-8.20	NOLGB	[SPECS]	To construct below the defined plotting area and for a logarithmic (base 10) X ordinate range the logarithmic scale.
IV-8.22	NOLGT	[SPECS]	To construct above the defined plotting area and for a logarithmic (base 10) X ordinate range the logarithmic scale.
IV-8.24	NOLGL	[SPECS]	To construct to the left of the defined plotting area and for a logarithmic (base 10) Y ordinate range the logarithmic scale.
IV-8.26	NOLGR	[SPECS]	To construct to the right of the defined plotting area and for a logarithmic (base 10) Y ordinate range the logarithmic scale.

Routines to Construct Alphanumeric Titles

These subroutines layout and construct alphanumeric text to annotate the plot. Scale layout, character format and text orientation are specified in the call list array *SPECS*. Separate lines of alhanumeric text are given in the several call list arrays (implied by the FORTRAN Hollerith Literals). *VALUE* is a number to be plotted.

Found in Section	Name	Call List	Function
IV-9.2	TITLEB	(#H...,#H..., ...,#H...,SPECS)	To construct consecutive lines of alphanumeric text centered below and parallel to the bottom side of the defined plotting area.
IV-9.4	TITLET	(#H...,#H..., ...,#H...,SPECS)	To construct consecutive lines of alphanumeric text centered above and parallel to the top side of the defined plotting area.
IV-9.6	TITLEL	(#H...,#H..., ...,#H...,SPECS)	To construct consecutive lines of alphanumeric text centered to the left of and parallel to the left side of the defined plotting area.
IV-9.8	TITLER	(#H...,#H..., ...,#H...,SPECS)	To construct consecutive lines of alphanumeric text centered to the right of and parllel to the right side of the defined plotting area.
IV-9.10	TITLEG	(RULE,#H..., SPECS)	To construct a single line of alphanumeric text above an arbitrary reference line.
IV-9.13	SYMKEY	(RULE,#H..., SPECS)	To construct a plot symbol together with a line of alphanumeric text to identify the plot symbol.
IV-9.16	DECVAL	(RULE,VALUE, SPECS)	To construct, when the number of decimal places is specified, a numerical value along an arbitrary reference line.
IV-9.18	SIGVAL	(RULE,VALUE, IPOWER,SPECS)	To construct, when the number of significant figures is specified, a numerical value along an arbitrary reference line.

Plotter Control Subroutines

These subroutines control the non-graphic aspects of the plotter(s).

Found in Section	Name	Call List	Function
IV-10.1	GDSEND	(SPECS)	Terminate plotting. Must be called as the last plot subroutine.
IV-10.2	NXTFRM	(SPECS)	Initiate a new frame.

Routines to Construct Contour Lines through Data*

These routines construct contour lines that represent plane intersections with a surface specified by a number of (X,Y,Z) coordinates. The coordinate values are given in the call list arrays *X*, *Y*, and *Z*. The array *BUFXYZ* is a work region. *LENGTH* is the size of *BUFXYZ*, a variable-sized work region.

Found in Section	Name	Call List	Function
	CONLI	(*X,Y,Z,ZLEVEL, BUFXYZ,LENGTH, SPECS*)	To construct, using only linear interpolation, the contour lines of one or more contour levels from a rectangular grid of values.
	CONBVS	(*X,Y,Z,ZLEVEL, BUFXYZ,LENGTH, SPECS*)	To construct, using a smooth surface interpolation technique (bi-variate spline), the contour lines of one or more contour levels from a rectangular grid of values.

* These routines are in a separate volume.

Routines to Determine Layout of an Axis' Scale

Routine Names

FABLIX, FABLIY, FAGLIX, FAGLIY

Purpose

FABLIX: To determine for an X ordinate range a linear X-scale for a blank form.

FABLIY: To determine for a Y ordinate range a linear Y-scale for a blank form.

FAGLIX: To determine for an X ordinate range a linear X-scale for a gridded form.

FAGLIY: To determine for a Y ordinate range a linear Y-scale for a gridded form.

Method

The following description applies to either the X or Y axis depending on the routine called.

Blank form: Given the minimum scale range and an upper limit restriction on the number of scale divisions, the routine extends the range and linearly subdivides it in the following way:

a. The routine divides the given range by the maximum number of scale divisions to compute the minimum scale step. This value is expressed in scientific notation, i.e., $A \times 10^M$, where M is an integer.

b. The routine enlarges the scale step until it becomes either 10^N, 2×10^N, or 5×10^N, where N is an integer.

c. Then, the routine expands the given scale range such that the maximum and minimum values become whole multiples of either the scale step, or one-half the scale step.

d. Lastly, the routine divides the expanded scale range by the scale step to compute the number of scale subdivisions.

Gridded form: Given (1) the minimum scale range, (2) the number of principal subdivisions of the linearly divided X-axis, (3) the length of the axis and (4) its location on the gridded form, the routine determines a scale layout in the following way:

a. The routine divides the given range by the given number of principal subdivisions to compute the minimum scale step. This value is expressed in scientific notation, i.e., $A \times 10^M$, where M is an integer.

b. The routine enlarges the scale step until it becomes either 10^N, 2×10^N, or 5×10^N, where N is an integer.

c. The routine then expands the given scale range such that the maximum and minimum values become whole multiples of either the scale step, or one-half the scale step.

d. Next, the routine divides the expanded scale range by the scale step to compute the required number of scale subdivisions.

e. Finally, the routine compues the length and location of the determined scale such that the required length of axis lies quasi-centered on the given axis. The new length is less than or equal to the given length.

Restrictions

Scale Range: Range is limited only by the range of a floating-point number (single precision).

Usage

See the following individual subroutine write-ups for:

A. FORTRAN Call Statement.

B. Description of call list arguments.

C, Routines Used.

D. Example.

Subroutine FABLIX

Purpose

To determine a linear scale for an X-axis when given a minimum X ordinate range and a maximum number of scale subdivisions.

Usage

A. FORTRAN Call Statement:

CALL FABLIX (*GIVEN*,*SPECS*)

B. Description of call list arguments

GIVEN An array of 3 consecutive floating-point words containing raw information.

Parameter	Definitions
GIVEN(1)	Parameter specifies the maximum value of the given X ordinate range
GIVEN(2)	Parameter specifies the minimum value of the given X ordinate range
GIVEN(3)	Parameter specifices the upper limit to the number of scale subdivisions the routine may determine.

SPECS An array into which the results are stored. Reference Table 1 under "The *SPECS* Array" for definitions and array format. In brief, the routine stores the results in the words of the array *SPECS* indicated below by an *.

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)			*SPECS(6)*		
SPECS(2)			*SPECS(7)*		
SPECS(3)	*	XRIGHT	*SPECS(8)*		
SPECS(4)	*	XLEFT	*SPECS(9)*	*	XDIV
SPECS(5)					

Note: The maximum and minimum values of the determined scale are stored by the routine at *SPECS(3)* and *SPECS(4)*, respectively.

C. Routine Used: BSCALE

D. Example

With a maximum value of 17.9 and a minimum value of 1.4 for the X range and a maximum of 5 subdivisions the setup would be:

```
DIMENSION GIVEN(3), SPECS(30)
GIVEN(1) = 17.9
GIVEN(2) =  1.4
GIVEN(3) =  5.
CALL FABLIX(GIVEN,SPECS)
```

The resulting values in the *SPECS* array would be:

```
SPECS(3) = 20.0
SPECS(4) =  0.0
SPECS(9) =  4.
```

Subroutine FABLIY

Purpose

To determine a linear scale for a Y-axis when given a mininum Y ordinate range and a maximum number of scale subdivisions.

Usage

A. FORTRAN Call Statement:

CALL FABLIY (*GIVEN*,*SPECS*)

B. Description of call list arguments

GIVEN An array of 3 consecutive floating-point words containing raw information.

Parameter	Definitions
GIVEN(1)	Parameter specifies the maximum value of the given Y ordinate range
GIVEN(2)	Parameter specifies the minimum value of the given Y ordinate range
GIVEN(3)	Parameter specifies the upper limit to the number of scale subdivisions the routine may determine.

SPECS An array into which the results are stored. Reference Table 1 under "The *SPECS* Array" for definitions and array format. In brief, the routine stores the results in the words of the array *SPECS* indicated below by an *.

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)			*SPECS*(6)	*	YBOT
SPECS(2)			*SPECS*(7)		
SPECS(3)			*SPECS*(8)		
SPECS(4)			*SPECS*(9)		
SPECS(5)	*	YTOP	*SPECS*(10)	*	YDIV

Note: The maximum and minimum values of the determined scale are stored by the routine at *SPECS*(5) and *SPECS*(6), respectively.

C. Routine Used: BSCALE

D. Example

With a maximum value of 7642.2 and a minimum value of -121.7 for the Y range and a maximum of 15 subdivisions the setup would be:

```
DIMENSION GIVEN(3), SPECS(30)
GIVEN(1) = 7642.2
GIVEN(2) = -121.7
GIVEN(3) =   15.
CALL FABLIY(GIVEN,SPECS)
```

The resulting values in the SPECS array would be:

```
SPECS(5)  =  8000.0
SPECS(6)  = -1000.0
SPECS(10) =     9.
```

Subroutine FAGLIX

Purpose

To determine a linear scale for an X-axis when given a minimum X ordinate range and a maximum number of scale subdivisions.

Usage

A. FORTRAN Call Statement:

CALL FAGLIX(*GIVEN*,*SPECS*)

B. Description of call list arguments

GIVEN An array of 5 consecutive floating-point words containing raw information.

Parameter	Definitions
GIVEN(1)	Parameter specifies the maximum value of the given X ordinate range
GIVEN(2)	Parameter specifies the minimum value of the given X ordinate range
GIVEN(3)	Parameter specifies the number of principal subdivisions of the given linearly divided X-axis
GIVEN(4)	Parameter specifies the length (inches) of the given X-axis
GIVEN(5)	Parameter specifies the X component of the vector distance (inches) from a reference point to the left end point of the given X-axis

SPECS An array into which the results are stored. Reference Table 1 under "The *SPECS* Array" for definitions and array format. In brief, the routine stores the results in the words of the array *SPECS* indicated below by an *.

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(6)*		
SPECS(2)			*SPECS(7)*	*	XLNGTH
SPECS(3)	*	XRIGHT	*SPECS(8)*		
SPECS(4)	*	XLEFT	*SPECS(9)*	*	XDIV
SPECS(5)					

Note: The maximum and minimum values of the determined scale are stored by the routine at *SPECS(3)* and *SPECS(4)*, respectively.

C. Routine Used: GSCALE

D. Example

Given a maximum value of 71.6 and a minimum value of 2.2 for the X range and a maximum of 5 subdivisions. The approximate length of the X-axis wanted is 10 inches. The distance from the reference point to the left end point of the given X-axis is approximately 2.0 inches. The setup is:

```
DIMENSION GIVEN(5), SPECS(30)
GIVEN(1) = 71.6
GIVEN(2) =  2.2
GIVEN(3) =  5
GIVEN(4) = 10.0
GIVEN(5) =  2.0
CALL FAGLIX(GIVEN,SPECS)
```

The resulting values in the SPECS array would be:

```
SPECS(1) =  2.0
SPECS(3) = 80.0
SPECS(4) =  0.0
SPECS(7) =  8.0
SPECS(9) =  4.0
```

Subroutine FAGLIY

Purpose

To determine a linear scale for a Y-axis when given a minimum Y ordinate range and a linearly divided Y axis.

Usage

A. FORTRAN Call Statement:

CALL FABLIY(*GIVEN*,*SPECS*)

B. Description of call list arguments

GIVEN An array of 5 consecutive words containing raw information

Parameter	Definition
GIVEN(1)	Parameter specifies the maximum value of the given Y ordinate range
GIVEN(2)	Parameter specifies the minimum value of the given Y ordinate range
GIVEN(3)	Parameter specifies the number of principal subdivisions of the given linearly divided Y-axis.
GIVEN(4)	Parameter specifies the length (inches) of the given Y-axis
GIVEN(5)	Parameter specifies the Y component of the vector distance (inches) from a reference point to the bottom end point of the given Y-axis. The value is negative if the reference point is above the end point; otherwise, the value is positive.

SPECS An array into which the results are stored. Reference Table 1 under "The *SPECS* Array" for definitions and array format. In brief, the routine stores the results in the words of the array *SPECS* indicated below by an *.

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)			*SPECS*(6)	*	YBOT
SPECS(2)	*	YDIST	*SPECS*(7)		
SPECS(3)			*SPECS*(8)	*	YLNGTH
SPECS(4)			*SPECS*(9)		
SPECS(5)	*	YTOP	*SPECS*(10)	*	YDIV

Note: The maximum and minimum values of the determined scale are stored by the routine at *SPECS*(5) and *SPECS*(6), respectively.

C. Routine Used: GSCALE

D. Example

Given a maximum value of 12,946 and a minimum value of 2.369 for the Y range and a maximum of 10 subdivisions. The approximate length of the Y-axis wanted is 5 inches. The distance from the reference point to the bottom end point of the Y-axis is approximately 3.0 inches. The setup is:

```
DIMENSION GIVEN(5), SPECS(30)
GIVEN(1) = 12946.0
GIVEN(2) =     2.369
GIVEN(3) =    10.0
GIVEN(4) =     5.0
GIVEN(5) =     3.0
CALL FAGLIY(GIVEN,SPECS)
```

The resulting values in the SPECS array would be:

```
SPECS(2)  =     3.5
SPECS(5)  = 14000.0
SPECS(6)  =     0.0
SPECS(8)  =     3.5
SPECS(10) =     7.0
```

Routines to Construct a Grid

Routine Names

GDLILI, GDLILG, GDLGLI, GDLGLG

Purpose

GDLILI — To construct a grid having linear subdivisions of both axes.

GDLILG — To construct a grid having linear subdivisions of the X-axis and logarithmic (base 10) subdivisions of the Y-axis.

GDLGLI — To construct a grid having linear subdivisions of the Y-axis and logarithmic (base 10) subdivisions of the X-axis.

GDLGLG — To construct a grid having logarithmic (base 10) subdivisions of both axes.

Method

These routines construct a set of constant X, i.e., vertical, grid lines to depict the X function scale (either linear or logarithmic) and construct a set of constant Y, i.e., horizontal, grid lines to depict the Y function scale.

The following description of the linear and logarithmic function grids apply to either the X-axis or Y-axis depending on the grid routine called:

Linear function grids: Equally spaced grid lines are constructed from the bottom to top or left to right of the defined plotting area depending on grid routine called. Grid lines are also constructed on the extremes of the plotting area.

Logarithmic function grids: Grid lines are constructed bottom to top or left to right of the defined plotting area depending on the grid routine called. The grid lines are constructed at the extremes and at each cycle and subcycle marks at the values

10^N, 2×10^N, 3×10^N, ..., 9×10^N

where N is an integer.

Restrictions

The size of the grid generated is only limited by the size of the allowed plotting area.

Usage

See the following individual subroutine writeups for:

A. FORTRAN Call Statement

B. Description of call list arguments

C. Routines used

D. Example

Subroutine GDLILI

Purpose

To construct a rectangular grid having linear subdivisions of both axes.

Usage

A. FORTRAN call statement:

CALL GDLILI (*SPECS*)

B. Description of call list arguments

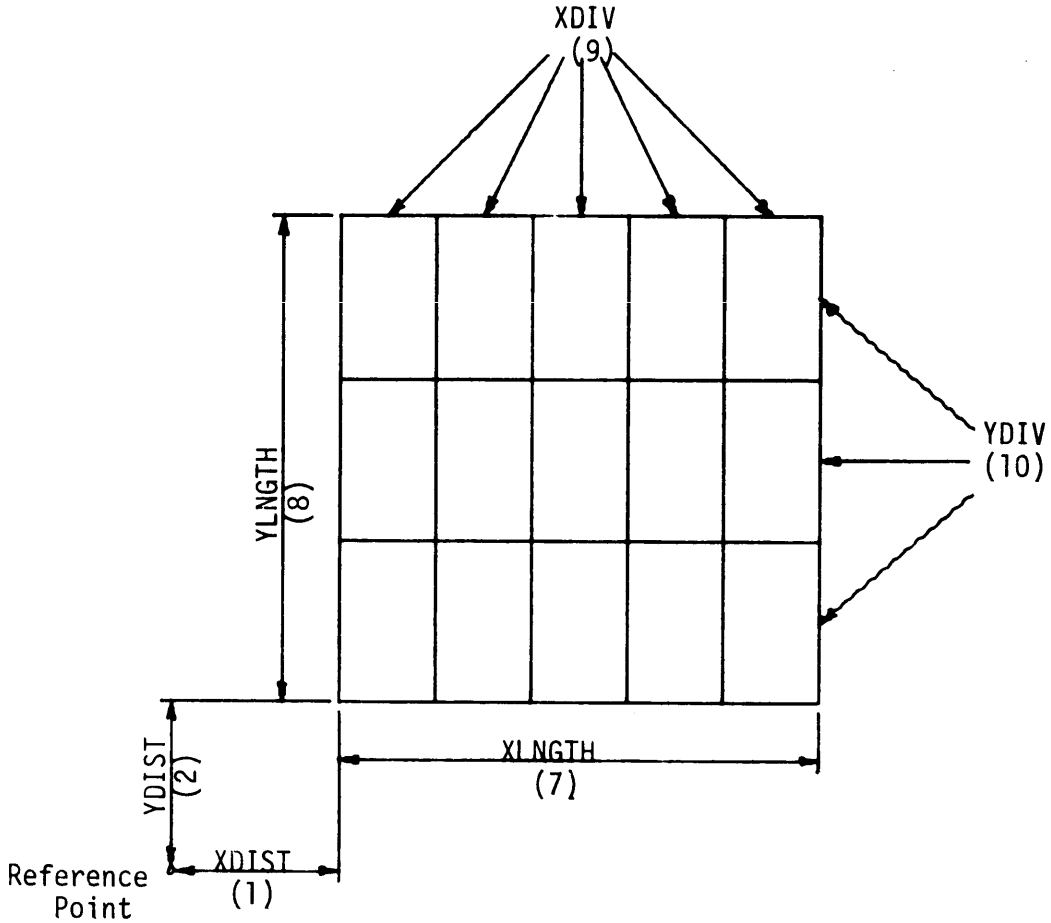
SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(7)*	*	XLNGTH
SPECS(2)	*	YDIST	*SPECS(8)*	*	YLNGTH
SPECS(3)			*SPECS(9)*	*	XDIV
SPECS(4)			*SPECS(10)*	*	YDIV
SPECS(5)			*SPECS(11)*	*	TOOL
SPECS(6)			*SPECS(12)*	*	CCTAPE

C. Routines Used: GDLI, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SET XDIV, YDIV
      SPECS(9)  = 5.
      SPECS(10) = 3.
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     CALL THE LINEAR GRID ROUTINE
      CALL GDLILI (SPECS)
        :
        :
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine GDLILG

Purpose

To construct a rectangular grid having linear subdivisions of the X-axis and logarithmic (base 10) subdivisions of the Y-axis.

Usage

A. FORTRAN call statement

```
CALL GDLILG (SPECS)
```

B. Description of call list arguments

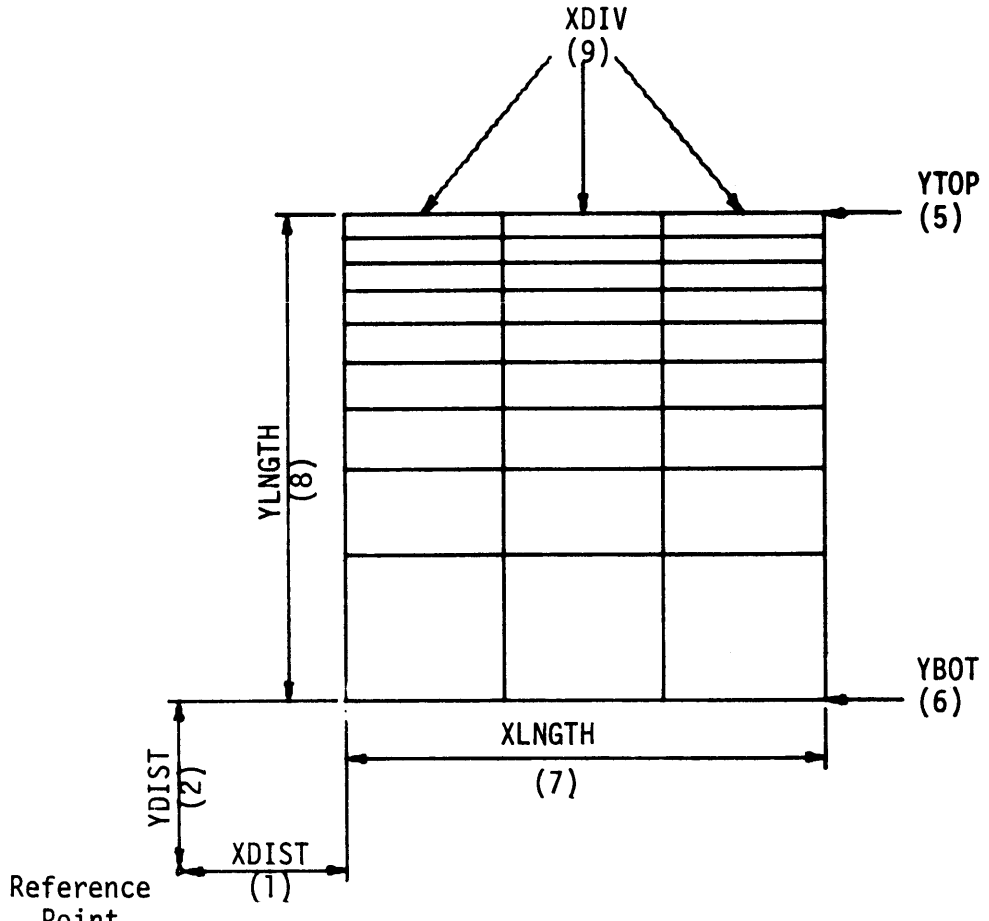
SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(7)*	*	XLNGTH
SPECS(2)	*	YDIST	*SPECS(8)*	*	YLNGTH
SPECS(3)			*SPECS(9)*	*	XDIV
SPECS(4)			*SPECS(10)*		
SPECS(5)	*	YTOP	*SPECS(11)*	*	TOOL
SPECS(6)	*	YBOT	*SPECS(12)*	*	CCTAPE

C. Routines Used: GDLI, GDLG, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
        :
        :
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET YTOP, YBOT
      SPECS(5) = 1.0
      SPECS(6) = 0.1
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SET XDIV
      SPECS(9) = 3.
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     CALL THE LINEAR, LOG GRID ROUTINE
      CALL GDLILG (SPECS)
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine GDLGLI

Purpose

To construct a rectangular grid having logarithmic (base 10) subdivisions of the X-axis and linear subdivisions of the Y-axis.

Usage

A. FORTRAN call statement

```
CALL GDLGLI (SPECS)
```

B. Description of call list arguments

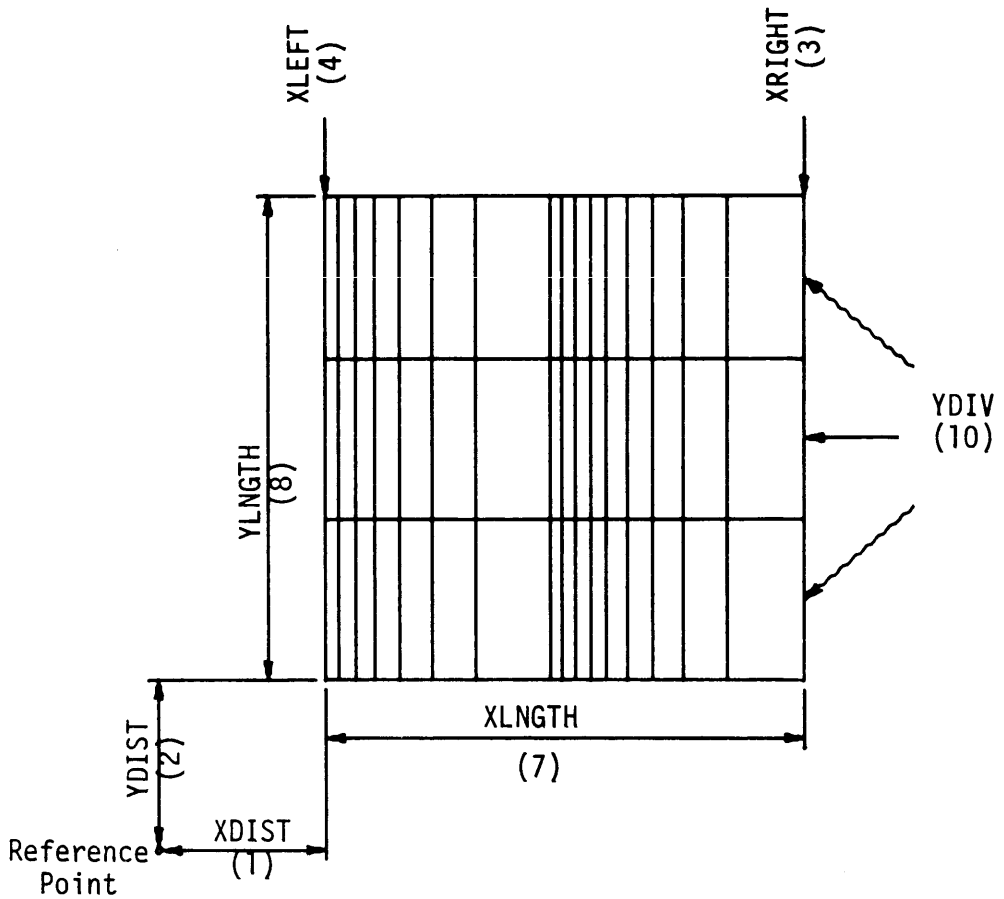
SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(7)*	*	XLNGTH
SPECS(2)	*	YDIST	*SPECS(8)*	*	YLNGTH
SPECS(3)	*	XRIGHT	*SPECS(9)*		
SPECS(4)	*	XLEFT	*SPECS(10)*	*	YDIV
SPECS(5)			*SPECS(11)*	*	TOOL
SPECS(6)			*SPECS(12)*	*	CCTAPE

C. Routine Used: GDLI, GDLG, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET XRIGHT, XLEFT
      SPECS(3) =  1.0
      SPECS(4) = 80.0
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SET YDIV
      SPECS(10) = 3.
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     CALL LOG LINEAR GRID
      CALL GDLGLI (SPECS)
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine GDLGLG

Purpose

To construct a rectangular grid having logarithmic (base 10) subdivisions of both axes.

Usage

A. FORTRAN call statement

CALL GDLGLG (*SPECS*)

B. Description of call list arguments

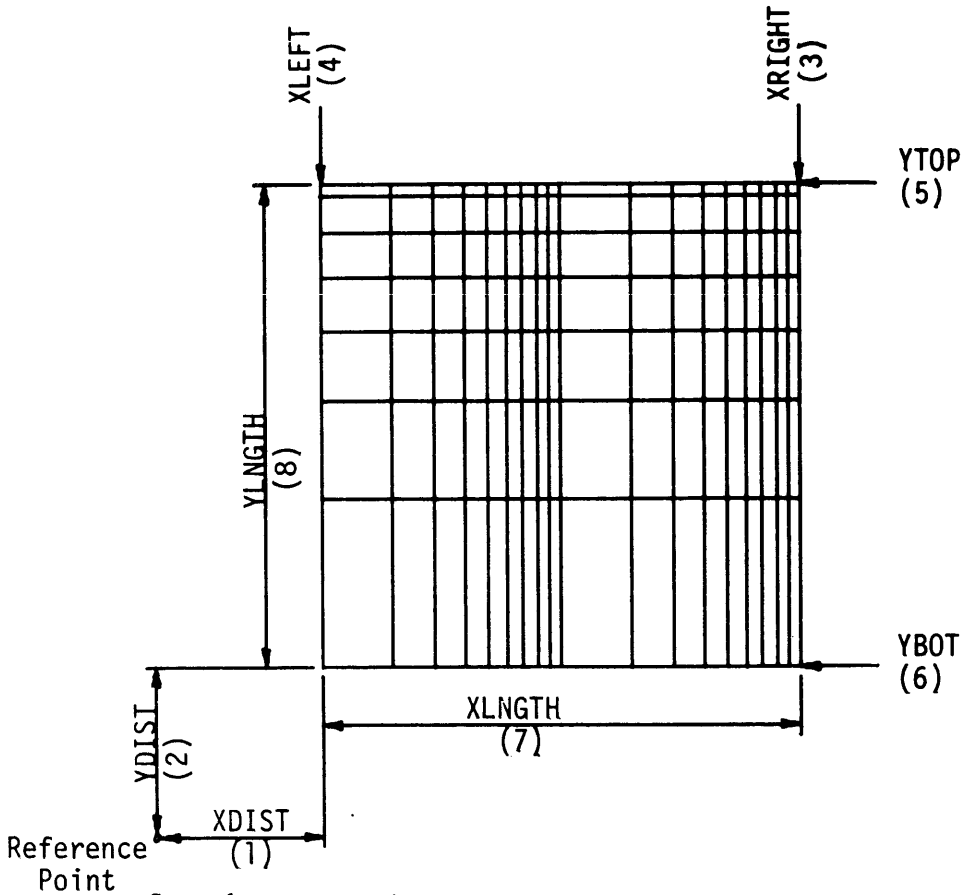
SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(7)*	*	XLNGTH
SPECS(2)	*	YDIST	*SPECS(8)*	*	YLNGTH
SPECS(3)	*	XRIGHT	*SPECS(9)*		
SPECS(4)	*	XLEFT	*SPECS(10)*		
SPECS(5)	*	YTOP	*SPECS(11)*	*	TOOL
SPECS(6)	*	YBOT	*SPECS(12)*	*	CCTAPE

C. Routines Used: GDLG, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET XRIGHT, XLEFT
      SPECS(3) = 100.0
      SPECS(4) =   1.0
C     SET YTOP, YBOT
      SPECS(5) =  73.0
      SPECS(6) =  10.0
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     CALL THE LOG, LOG GRID ROUTINE
      CALL GDLGLG (SPECS)
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Routines to Construct a Pair of Coordinate Axes

Routine Names

AXLILI, AXLILG, AXLGLI, AXLGLG

Purpose

AXLILI — To construct a pair of linear function scales for both axes.

AXLILG — To construct a linear function scale for the X-axis and a logarithmic (base 10) function scale for the Y-axis.

AXLGLI — To construct a linear function scale for the Y-axis and a logarithmic (base 10) function scale for the X-axis.

AXLGLG — To construct a pair of logarithmic (base 10) function scales for both axes.

Method

These routines construct a pair of axes along the bottom and the left side of the defined plotting area.

The following description of the linear and logarithmic function scales apply to either the X-axis or Y-axis depending on the axis routine called.

Linear function scales: equally spaced scale marks (0.05 inches long) are constructed from the appropriate axis projecting into the plotting area. Scale marks are also drawn on the axis extremes.

Logarithmic function scales: scale marks (0.05 inches long) are constructed at each extreme of the axis and within each cycle within the range at the values

10^n, 2×10^n, ..., 9×10^n,

where n is an integer.

Restrictions

The size of the axes generated are only limited by the size of the plotting area.

Usage

See the following individual subroutine write-ups for:

A. FORTRAN call statement

B. Description of call list arguments

C. Routines Used

D. Example

Subroutine AXLILI

Purpose

To construct a pair of axes for a rectangular coordinate system having linear function scales for both axes.

Usage

A. FORTRAN call statement

CALL AXLILI (*SPECS*)

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

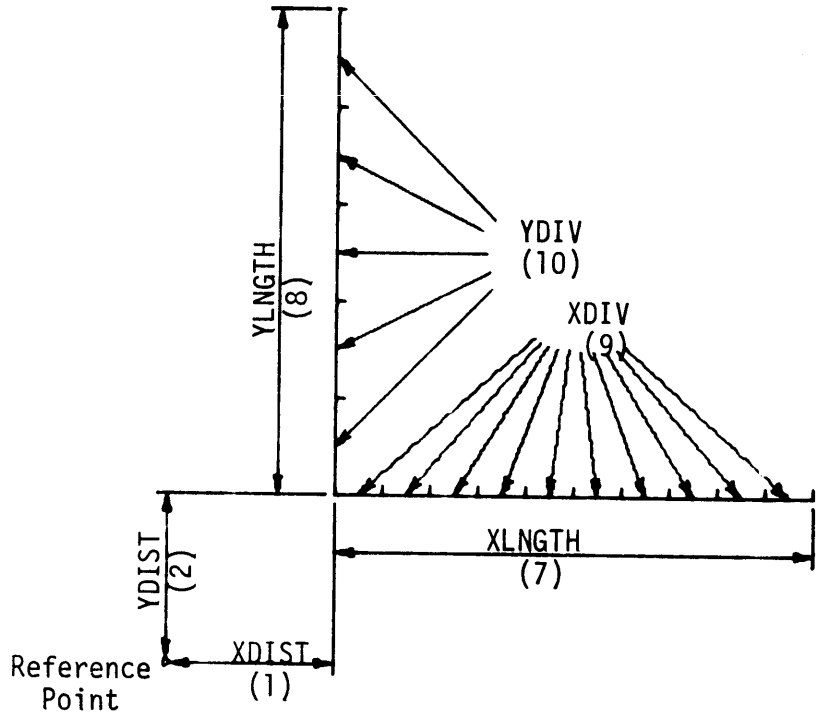
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(*1*)	*	XDIST	*SPECS*(*7*)	*	XLNGTH
SPECS(*2*)	*	YDIST	*SPECS*(*8*)	*	YLNGTH
SPECS(*3*)			*SPECS*(*9*)	*	XDIV
SPECS(*4*)			*SPECS*(*10*)	*	YDIV
SPECS(*5*)			*SPECS*(*11*)	*	TOOL
SPECS(*6*)			*SPECS*(*12*)	*	CCTAPE

C. Routines Used: AXLI, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SET XDIV, YDIV*
      SPECS(9) = 10.
      SPECS(10) = 5.
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     CALL LINEAR, LINEAR AXIS ROUTINE
      CALL AXLILI (SPECS)
          .
          .
          .
```

* These may also be determined as a result of FABLIX, FABLIY.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine AXLILG

Purpose

To construct a pair of axes for a rectangular coordinate system having a linear function scale for the X-axis and a logarithmic (base 10) function scale for the Y-axis.

Usage

A. FORTRAN call statement

```
CALL AXLILG (SPECS)
```

B. Description of call list arguments

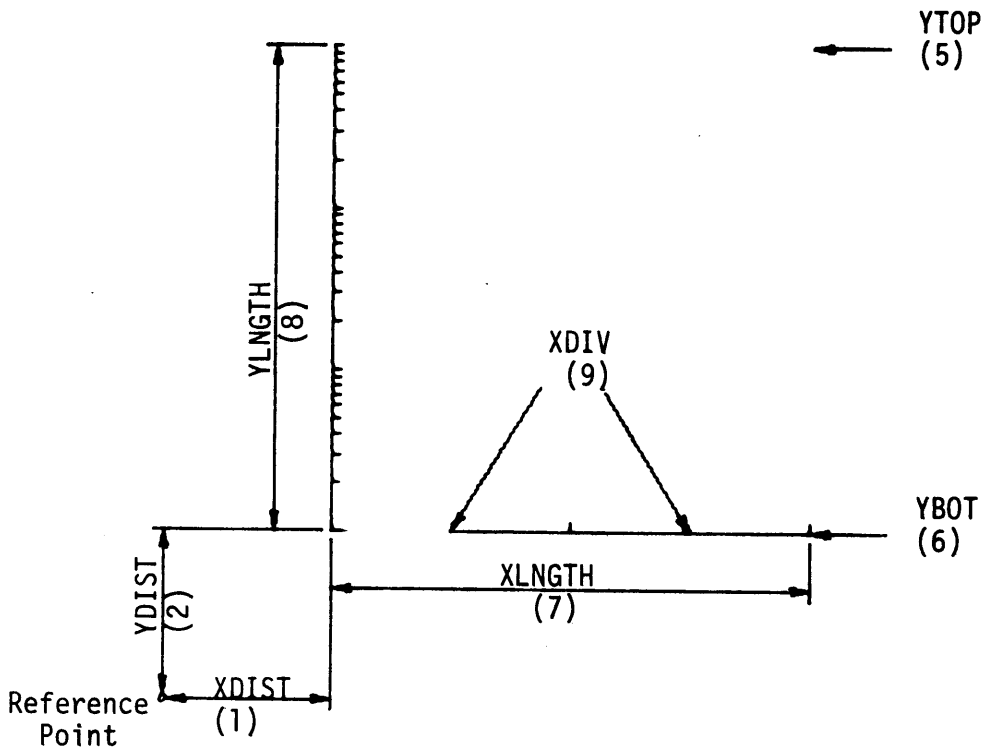
SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(7)*	*	XLNGTH
SPECS(2)	*	YDIST	*SPECS(8)*	*	YLNGTH
SPECS(3)			*SPECS(9)*	*	XDIV
SPECS(4)			*SPECS(10)*		
SPECS(5)	*	YTOP	*SPECS(11)*	*	TOOL
SPECS(6)	*	YBOT	*SPECS(12)*	*	CCTAPE

C. Routines Used: AXLI, AXLG, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET YTOP, YBOT
      SPECS(5) = 1000.0
      SPECS(6) =    1.0
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SET XDIV
      SPECS(9) = 2.
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     CALL LINEAR, LOG AXIS ROUTINE
      CALL AXLILG (SPECS)
         :
         :
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine AXLGLI

Purpose

To construct a pair of axes for a rectangular coordinate system having a logarithmic (base 10) function scale for the X-axis and a linear function scale for the Y-axis.

Usage

A. FORTRAN call statement

CALL AXLGLI (*SPECS*)

B. Description of call list arguments

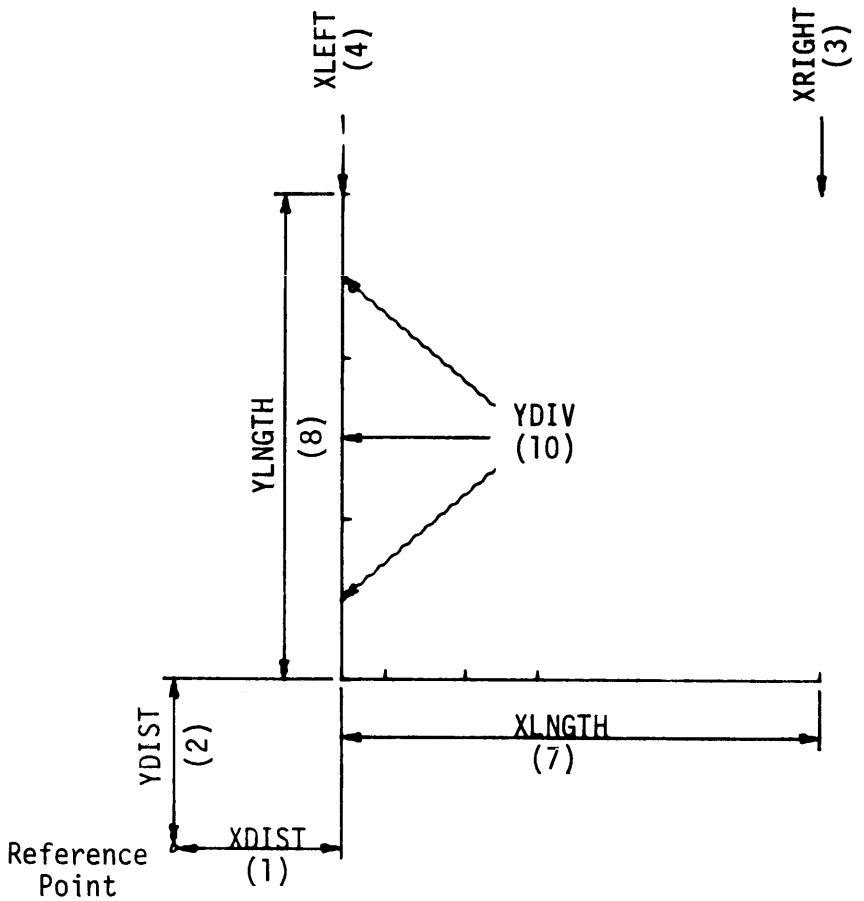
SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(7)*	*	XLNGTH
SPECS(2)	*	YDIST	*SPECS(8)*	*	YLNGTH
SPECS(3)	*	XRIGHT	*SPECS(9)*		
SPECS(4)	*	XLEFT	*SPECS(10)*	*	YDIV
SPECS(5)			*SPECS(11)*	*	TOOL
SPECS(6)			*SPECS(12)*	*	CCTAPE

C. Routines Used: AXLI, AXLG, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET XRIGHT, XLEFT
      SPECS(3) = 15.0
      SPECS(4) =  7.5
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SET YDIV
      SPECS(10) = 3.
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     CALL LOG, LINEAR AXIS ROUTINE
      CALL AXLGLI (SPECS)
          :
          :
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine AXLGLG

Purpose

To construct a pair of axes for a rectangualr coordinate system having logarithmic (base 10) function scales for both axes.

Usage

A. FORTRAN call statement

```
CALL AXLGLG (SPECS)
```

B. Description of call liat arguments

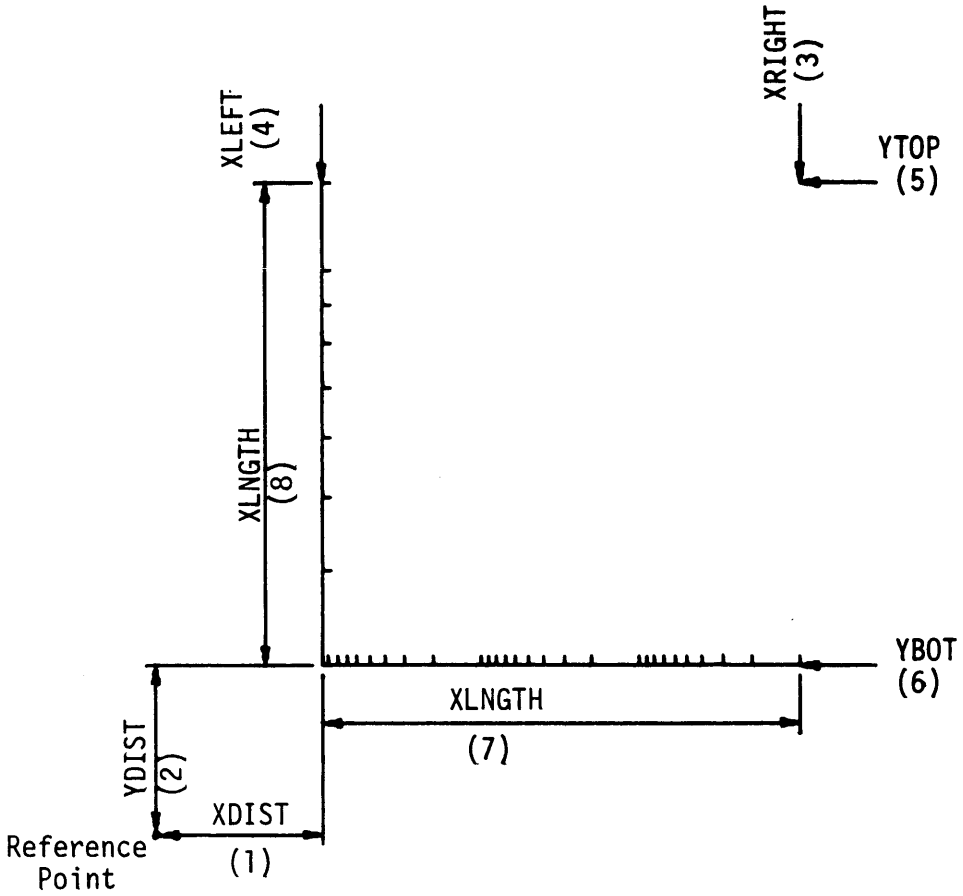
SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(7)*	*	XLNGTH
SPECS(2)	*	YDIST	*SPECS(8)*	*	YLNGTH
SPECS(3)	*	XRIGHT	*SPECS(9)*		
SPECS(4)	*	XLEFT	*SPECS(10)*		
SPECS(5)	*	YTOP	*SPECS(11)*	*	TOOL
SPECS(6)	*	YBOT	*SPECS(12)*	*	CCTAPE

C. Routines Used: AXLG, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET XRIGHT, XLEFT
      SPECS(3) =   0.1
      SPECS(4) = 100.0
C     SET YTOP, YBOT
      SPECS(5) = 13.0
      SPECS(6) =  3.0
C     SET XLNGTH, YLGNTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     CALL LOG, LOG AXIS ROUTINE
      CALL AXLGLG (SPECS)
        .
        .
        .
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Routines to Construct a Supplementary Axis

Routine Names

SAXLIB, SAXLIT, SAXLIL, SAXLIR, SAXLGB, SAXLGT, SAXLGL, SAXLGR

Purpose

To construct a supplementary axis either above, below, to the left, or to the right of the defined plotting area depending on the suffix letter (B,T,L,R) of the subroutine called. The supplementary axis can be constructed with either linear or logarithmic subdivisions (denoted by LI or LG in the subroutine name).

Method

The routines construct a supplementary axis either above, below, to the left or to the right of the defined plotting area. This axis is offset from the defined plotting area by 0.15 inches plus the value of the appropriate ZONE value. To denote the subdivisions, scale marks 0.05 inches in length are constructed from the supplementary axis toward the plotting area. After the construction, the routines automatically add 0.15 to the appropriate ZONE value.

Restrictions

None

Usage

See the following individual subroutine writeups for:

A. FORTRAN Call Statements

B. Description of call list arguments

C. Routines Used

D. Example

Subroutine SAXLIB

Purpose

To construct below the defined plotting area an X-axis having linear subdivisions.

Usage

A. FORTRAN Call Statement:

```
CALL SAXLIB (SPECS)
```

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

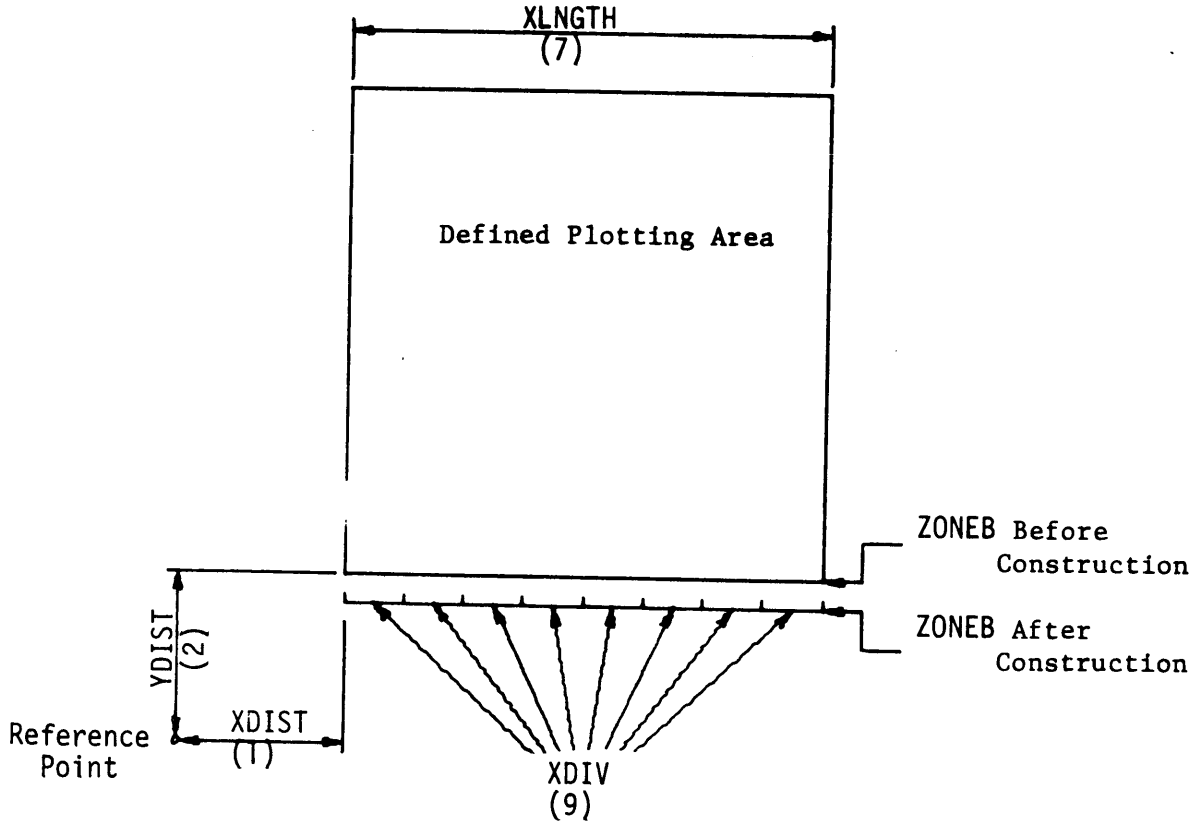
Parameter	Mnemonic	Parameter	Mnemonic
SPECS(1) *	XDIST	*SPECS(13)*	
SPECS(2) *	YDIST	*SPECS(14)*	
SPECS(3)		*SPECS(15)*	
SPECS(4)		*SPECS(16)*	
SPECS(5)		*SPECS(17)*	
SPECS(6)		*SPECS(18)*	
SPECS(7) *	XLNGTH	*SPECS(19)*	
SPECS(8)		*SPECS(20)*	
SPECS(9) *	XDIV	*SPECS(21)*	
SPECS(10)		*SPECS(22)*	
*SPECS(11)**	TOOL	*SPECS(23)*	
*SPECS(12)**	CCTAPE	*SPECS(24)* *	ZONEB

C. Routines Used: AXLI, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET XLNGTH (IN INCHES)
      SPECS(7) = 2.5
C     SET XDIV
      SPECS(9) = 8.
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SET ZONEB
      SPECS(24) = 0.0
C     CALL BOTTOM SUPPLEMENTARY AXIS ROUTINE
      CALL SAXLIB (SPECS)
         :
         :
```

NOTE: ZONEB is incremented after the construction so that the next construction below the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine SAXLIT

Purpose

To construct above the defined plotting area an X-axis having linear subdivisions.

Usage

A. FORTRAN Call Statement:

CALL SAXLIT (*SPECS*)

B. Description of call list arguments:

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

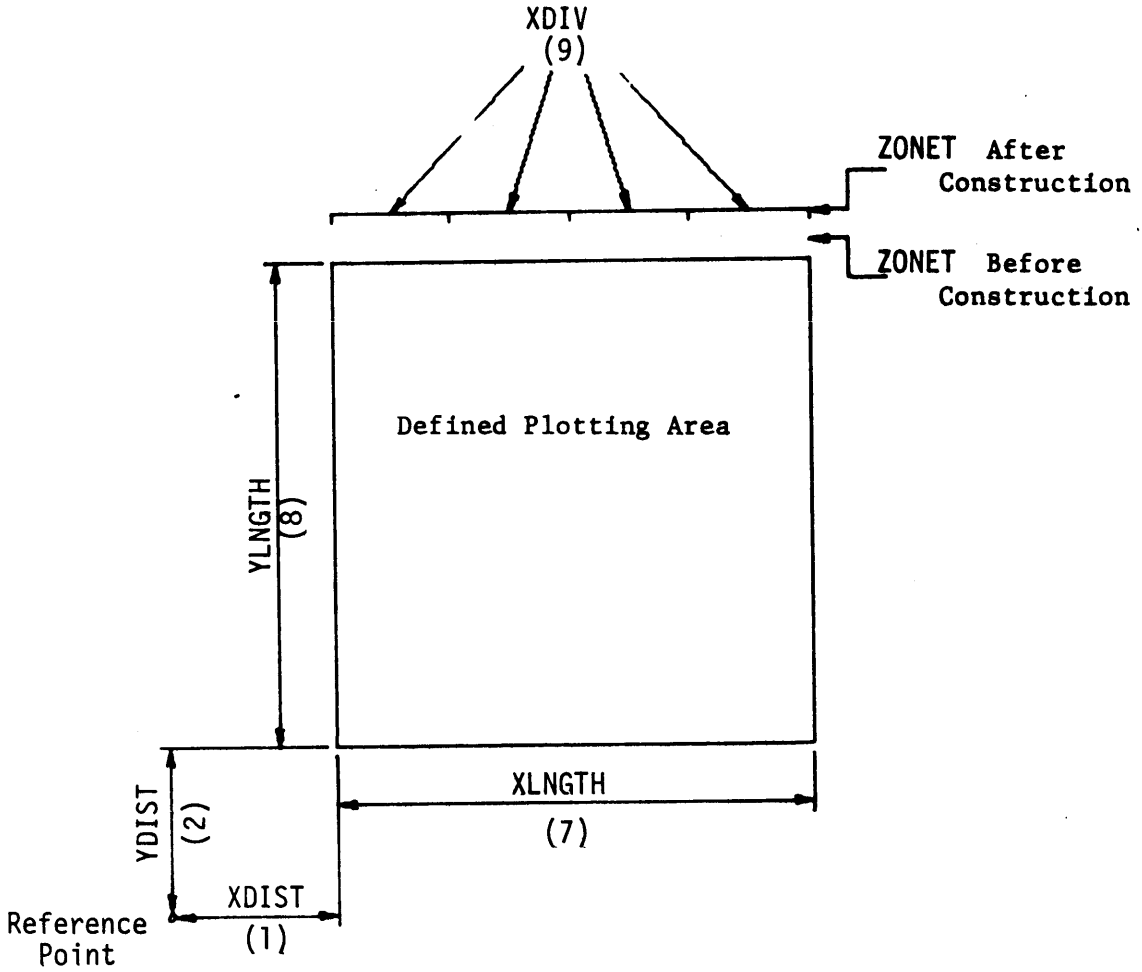
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(14)*		
SPECS(2)	*	YDIST	*SPECS(15)*		
SPECS(3)			*SPECS(16)*		
SPECS(4)			*SPECS(17)*		
SPECS(5)			*SPECS(18)*		
SPECS(6)			*SPECS(19)*		
SPECS(7)	*	XLNGTH	*SPECS(20)*		
SPECS(8)	*	YLNGTH	*SPECS(21)*		
SPECS(9)	*	XDIV	*SPECS(22)*		
SPECS(10)			*SPECS(23)*		
SPECS(11)	*	TOOL	*SPECS(24)*		
SPECS(12)	*	CCTAPE	*SPECS(25)*	*	ZONET
SPECS(13)					

C. Routines Used: AXLI, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SET XDIV
      SPECS(9) = 4.
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SET ZONET
      SPECS(25) = 0.1
C     CALL TOP SUPPLEMENTARY AXIS ROUTINE
      CALL SAXLIT (SPECS)
        .
        .
        .
```

NOTE: ZONET is incremented after the construction so that the next construction above the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine SAXLIL

Purpose

To construct to the left of the defined plotting area a Y-axis having linear subdivisions.

Usage

A. FORTRAN Call Statement:

CALL SAXLIL (SPECS)

B. Description of call list arguments:

SPECS An array containing plot construction parameters. Reference Table 1 under "The SPECS Array" for definitions and array format. This routine requires the SPECS parameters indicated below by an * to be specified prior to calling it:

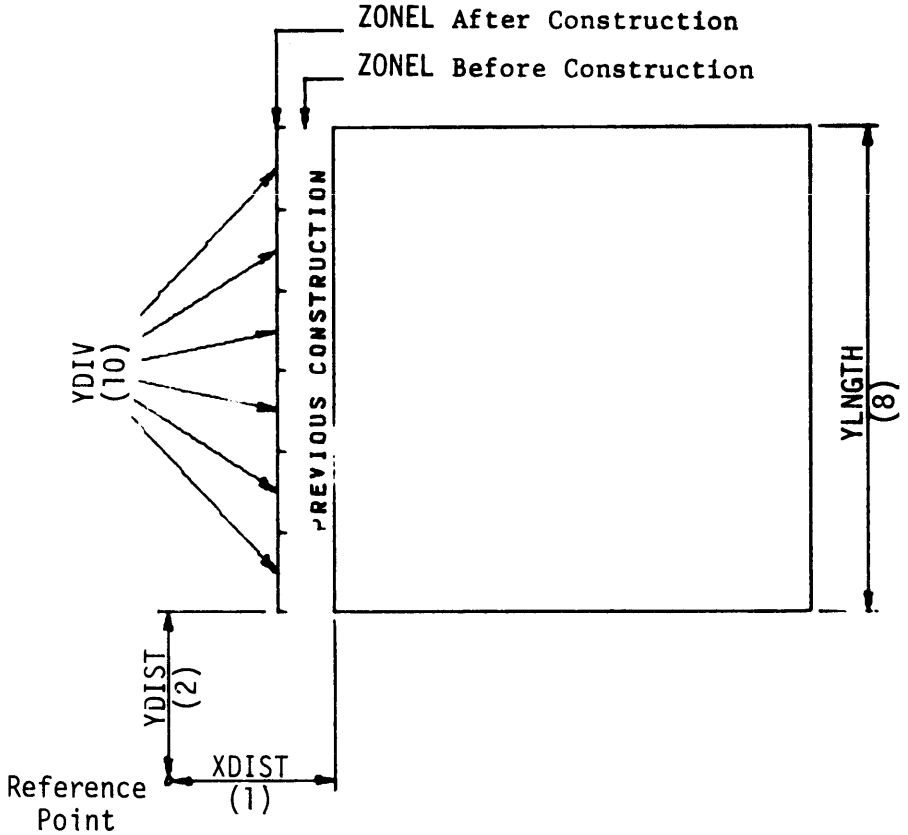
Parameter	Mnemonic	Parameter	Mnemonic
SPECS(1) *	XDIST	SPECS(14)	
SPECS(2) *	YDIST	SPECS(15)	
SPECS(3)		SPECS(16)	
SPECS(4)		SPECS(17)	
SPECS(5)		SPECS(18)	
SPECS(6)		SPECS(19)	
SPECS(7)		SPECS(20)	
SPECS(8) *	YLNGTH	SPECS(21)	
SPECS(9)		SPECS(22)	
SPECS(10)*	YDIV	SPECS(23)	
SPECS(11)*	TOOL	SPECS(24)	
SPECS(12)*	CCTAPE	SPECS(25)	
SPECS(13)		SPECS(26) *	ZONEL

C. Routines Used: AXLI, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SET YDIV
      SPECS(10) = 6.
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SET ZONEL (IN INCHES)
      SPECS(26) = 0.15
C     CALL LEFT SUPPLEMENTARY AXIS ROUTINE
      CALL SAXLIT (SPECS)
         .
         .
         .
```

NOTE: ZONEL is incremented after the construction so that the next construction to the left of the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine SAXLIR

Purpose

To construct to the right of the defined plotting area a Y-axis having linear subdivisions.

Usage

A. FORTRAN call statement:

```
CALL SAXLIR (SPECS)
```

B. Description of call list arguments:

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

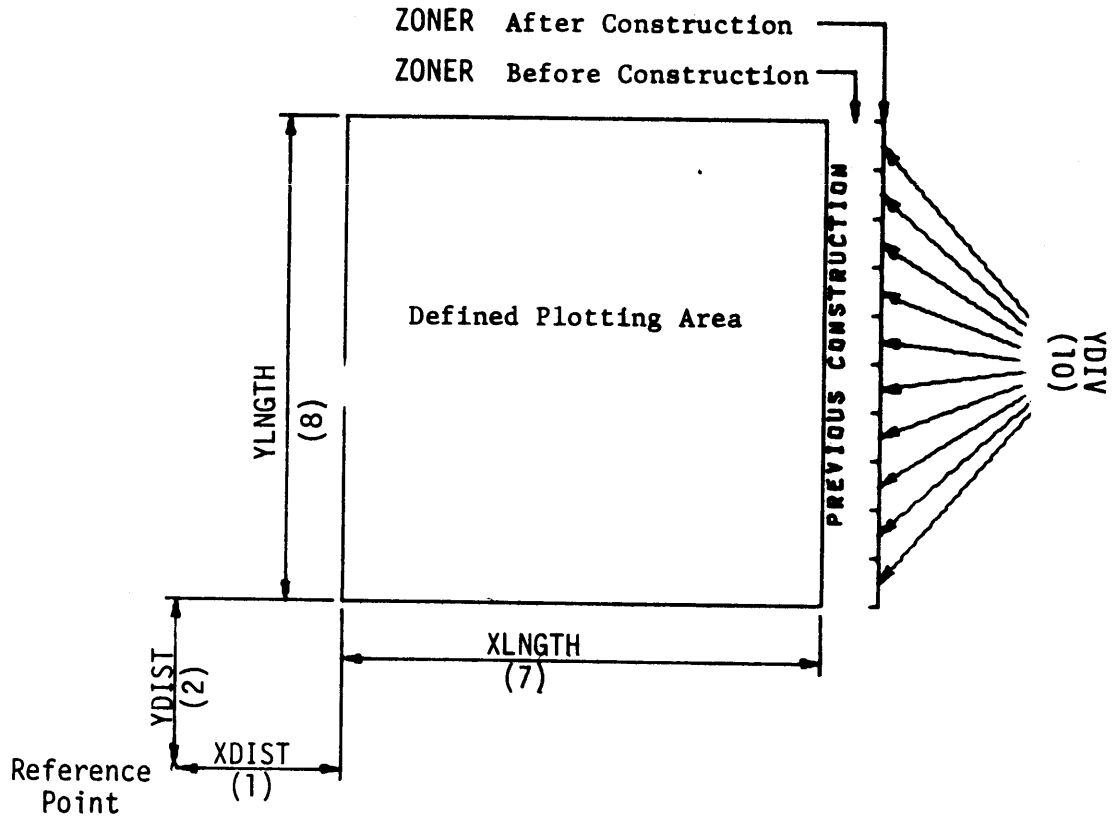
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(15)*		
SPECS(2)	*	YDIST	*SPECS(16)*		
SPECS(3)			*SPECS(17)*		
SPECS(4)			*SPECS(18)*		
SPECS(5)			*SPECS(19)*		
SPECS(6)			*SPECS(20)*		
SPECS(7)	*	XLNGTH	*SPECS(21)*		
SPECS(8)	*	YLNGTH	*SPECS(22)*		
SPECS(9)			*SPECS(23)*		
SPECS(10)	*	YDIV	*SPECS(24)*		
SPECS(11)	*	TOOL	*SPECS(25)*		
SPECS(12)	*	CCTAPE	*SPECS(26)*		
SPECS(13)			*SPECS(27)*	*	ZONER
SPECS(14)					

C. Routines Used: AXLI, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SET YDIV
      SPECS(10) = 6.
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SET ZONER
      SPECS(27) = 0.15
C     CALL RIGHT SUPPLEMENTARY AXIS ROUTINE
      CALL SAXLIR (SPECS)
        .
        .
        .
```

NOTE; ZONER is incremented after the construction so that the next construction to the right of the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine SAXLGB

Purpose

To construct below the defined plotting area an X-axis having logarithmic (base 10) subdivisions.

Usage

A. FORTRAN call statement:

```
CALL SAXLGB (SPECS)
```

B. Description of call list arguments:

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

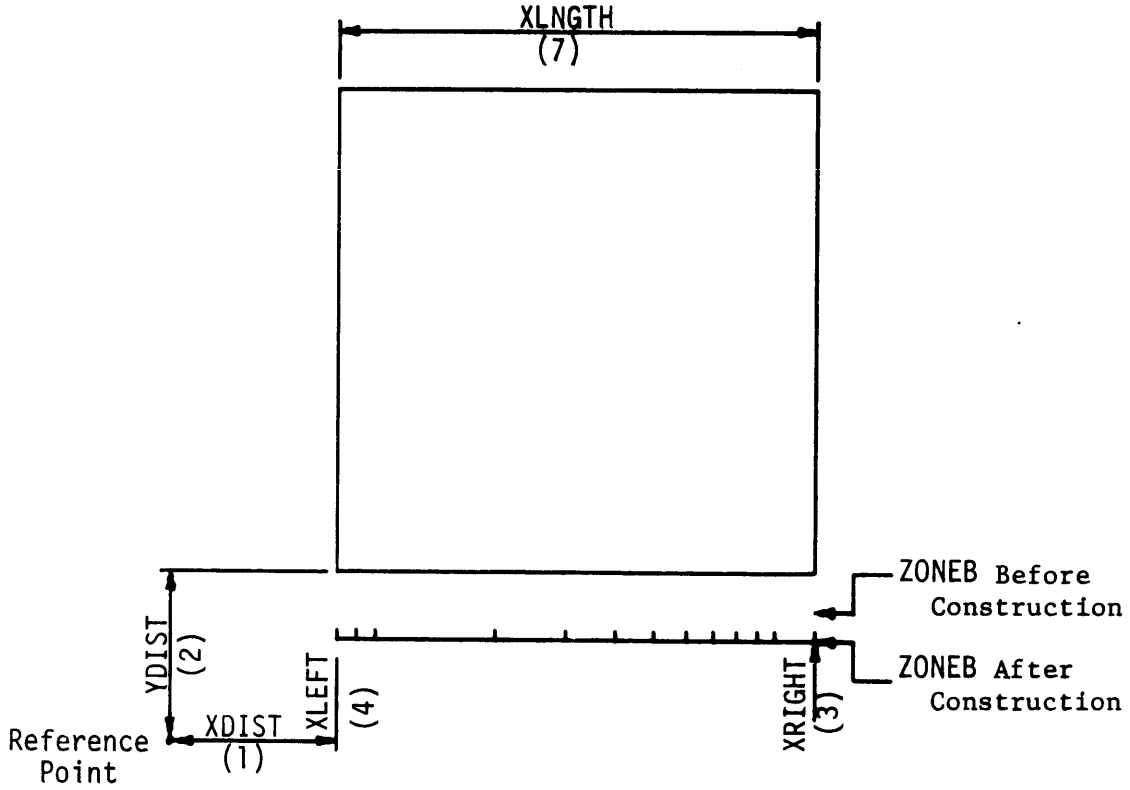
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	SPECS(13)		
SPECS(2)	*	YDIST	SPECS(14)		
SPECS(3)	*	XRIGHT	SPECS(15)		
SPECS(4)	*	XLEFT	SPECS(16)		
SPECS(5)			SPECS(17)		
SPECS(6)			SPECS(18)		
SPECS(7)	*	XLNGTH	SPECS(19)		
SPECS(8)			SPECS(20)		
SPECS(9)			SPECS(21)		
SPECS(10)			SPECS(22)		
SPECS(11)	*	TOOL	SPECS(23)		
SPECS(12)	*	CCTAPE	SPECS(24)	*	ZONEB

C. Routines Used: AXLG, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT, XLEFT
      SPECS(3) = 12.5
      SPECS(4) =  0.8
C     SET XLNGTH (IN INCHES)
      SPECS(7) = 2.5
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SET ZONEB
      SPECS(24) = 0.2
C     CALL BOTTOM LOG SUPPLEMENTARY AXIS ROUTINE
      CALL SAXLGB (SPECS)
         :
         :
```

NOTE; ZONEB is incremented after the construction so that the next construction below the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine SAXLGT

Purpose

To construct above the defined plotting area an X-axis having logarithmic (base 10) subdivisions.

Usage

A. FORTRAN call statement:

```
CALL SAXLGT (SPECS)
```

B. Description of call list arguments:

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

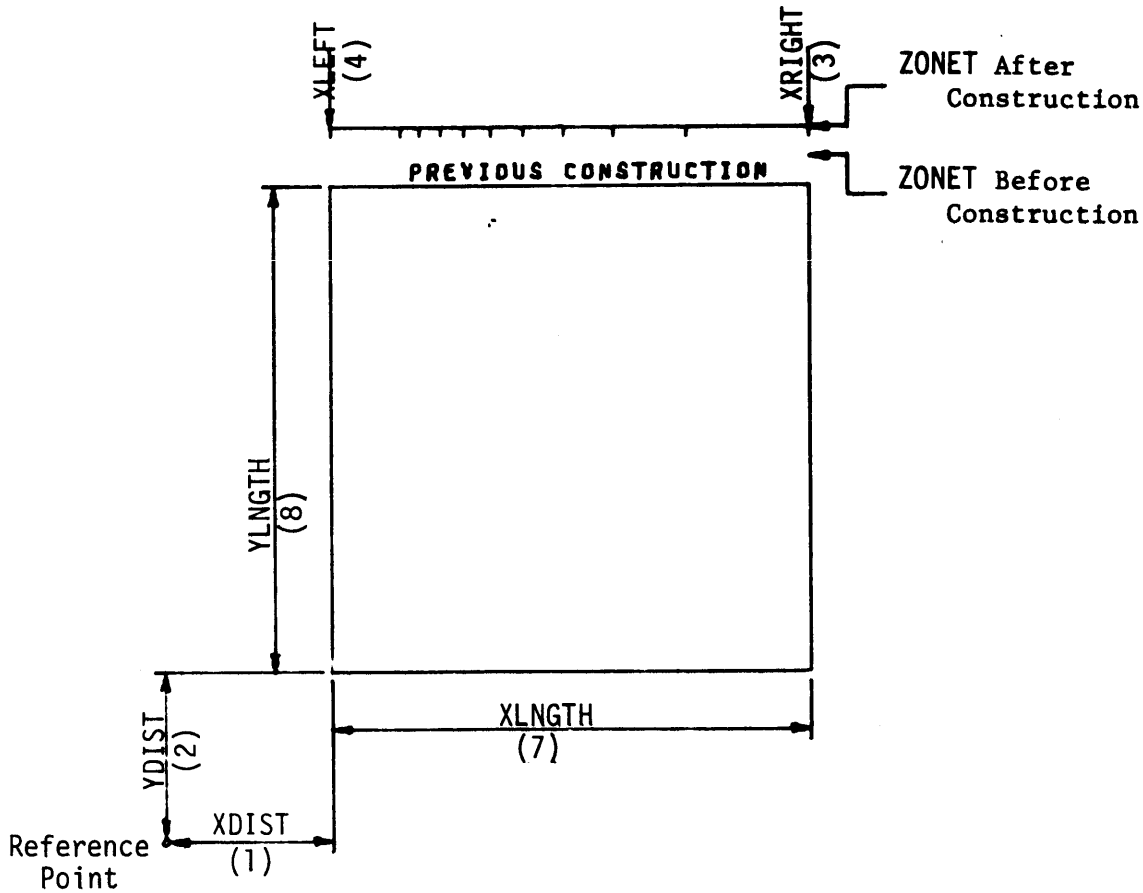
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(14)*		
SPECS(2)	*	YDIST	*SPECS(15)*		
SPECS(3)	*	XRIGHT	*SPECS(16)*		
SPECS(4)	*	XLEFT	*SPECS(17)*		
SPECS(5)			*SPECS(18)*		
SPECS(6)			*SPECS(19)*		
SPECS(7)	*	XLNGTH	*SPECS(20)*		
SPECS(8)	*	YLNGTH	*SPECS(21)*		
SPECS(9)			*SPECS(22)*		
SPECS(10)			*SPECS(23)*		
SPECS(11)	*	TOOL	*SPECS(24)*		
SPECS(12)	*	CCTAPE	*SPECS(25)*	*	ZONET
SPECS(13)					

C. Routines Used: AXLG, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.8777
      SPECS(2) = 0.8777
C     SET XRIGHT, XLEFT
      SPECS(3) =  1.0
      SPECS(4) = 15.0
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SET ZONET
      SPECS(25) = 0.15
C     CALL TOP LOG SUPPLEMENTARY AXIS ROUTINE
      CALL SAXLGT (SPECS)
         .
         .
         .
```

NOTE: ZONET is incremented after the construction so that the next construction above the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine SAXLGL

Purpose

To construct to the left of the defined plotting area a Y-axis having logarithmic (base 10) subdivisions.

Usage

A. FORTRAN call statement:

```
CALL SAXLGL (SPECS)
```

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

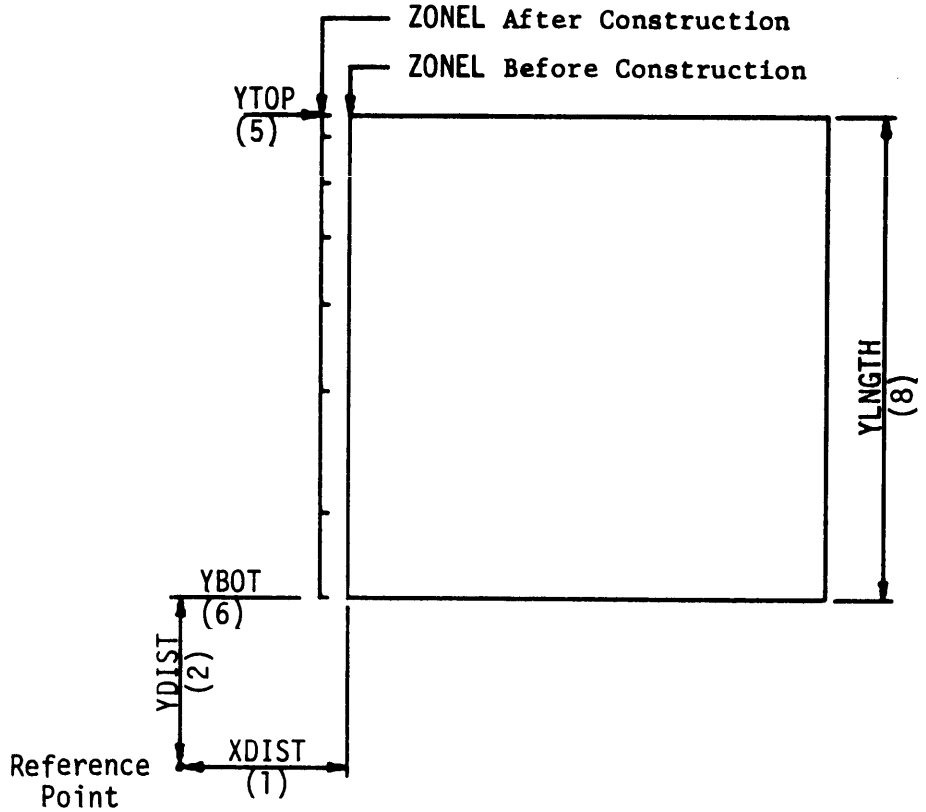
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(14)*		
SPECS(2)	*	YDIST	*SPECS(15)*		
SPECS(3)			*SPECS(16)*		
SPECS(4)			*SPECS(17)*		
SPECS(5)	*	YTOP	*SPECS(18)*		
SPECS(6)	*	YBOT	*SPECS(19)*		
SPECS(7)			*SPECS(20)*		
SPECS(8)	*	YLNGTH	*SPECS(21)*		
SPECS(9)			*SPECS(22)*		
SPECS(10)			*SPECS(23)*		
SPECS(11)	*	TOOL	*SPECS(24)*		
SPECS(12)	*	CCTAPE	*SPECS(25)*		
SPECS(13)			*SPECS(26)*	*	ZONEL

C. Routines Used: AXLG, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET YTOP, YBOT
      SPECS(5) =  7.0
      SPECS(6) = 15.0
C     SET LNGTH (IN INCHES)
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SET ZONEL
      SPECS(26) = 0.0
C     CALL LEFT LOG SUPPLEMENTARY AXIS ROUTINE
      CALL SAXLGL (SPECS)
         .
         .
         .
```

NOTE; ZONEL is incremented after the construction so that the next construction to the left of the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine SAXLGR

Purpose

To construct to the right of the defined plotting area a Y-axis having logarithmic (base 10) subdivisions.

Usage

A. FORTRAN call statement:

CALL SAXLGR (SPECS)

B. Description of call list arguments:

SPECS An array containing plot construction parameters. Reference Table 1 under "The SPECS Array" for definitions and array format. This routine requires the SPECS parameters indicated below by an * to be specified prior to calling it:

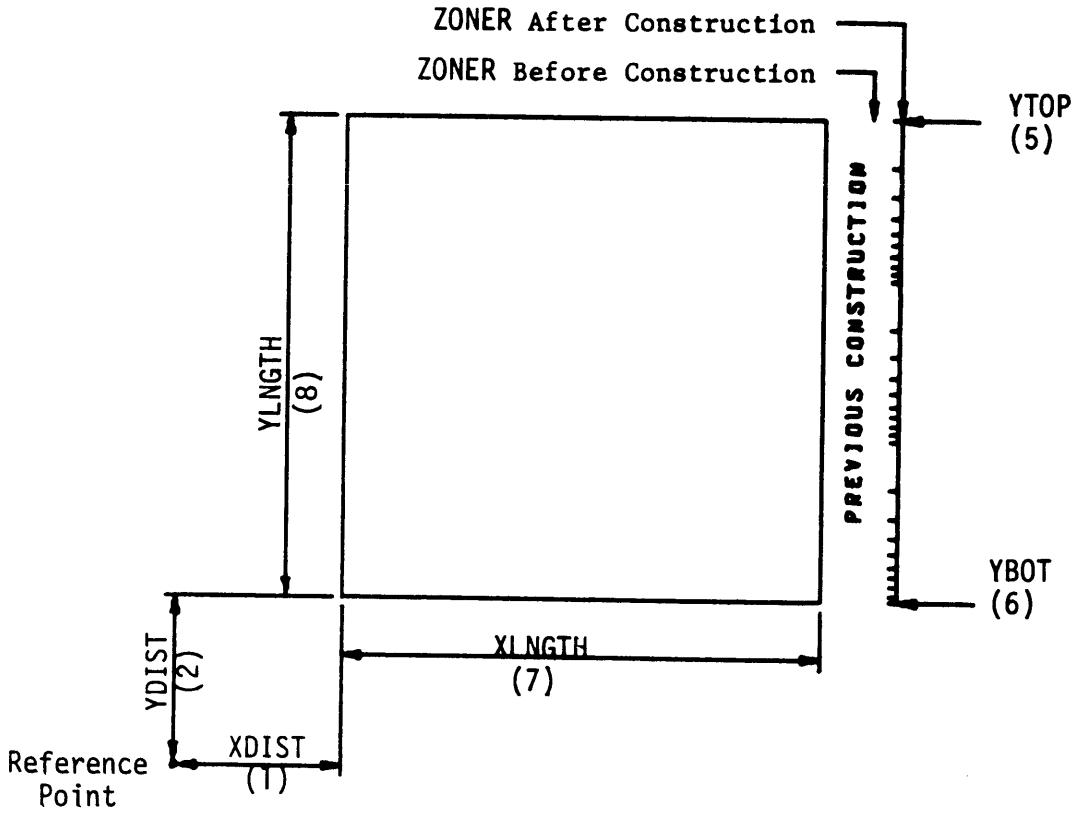
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	SPECS(15)		
SPECS(2)	*	YDIST	SPECS(16)		
SPECS(3)			SPECS(17)		
SPECS(4)			SPECS(18)		
SPECS(5)	*	YTOP	SPECS(19)		
SPECS(6)	*	YBOT	SPECS(20)		
SPECS(7)	*	XLNGTH	SPECS(21)		
SPECS(8)	*	YLNGTH	SPECS(22)		
SPECS(9)			SPECS(23)		
SPECS(10)			SPECS(24)		
SPECS(11)	*	TOOL	SPECS(25)		
SPECS(12)	*	CCTAPE	SPECS(26)		
SPECS(13)			SPECS(27)		
SPECS(14)			SPECS(28)	*	ZONER

C. Routines Used: AXLG, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET YTOP, YBOT
      SPECS(5) =   0.1
      SPECS(6) = 100.0
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SET ZONER
      SPECS(27) = 0.25
C     CALL RIGHT LOG SUPPLEMENTARY AXIS ROUTINE
      CALL SAXLGR (SPECS)
        .
        .
        .
```

NOTE: ZONER is incremented after the construction so that the next construction to the right of the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Routines to Symbol-Plot Data

Routine Names

PSLILI, PSLILG, PSLGLI, PSLGLG, PSPOLR

Purpose

PSLILI — To symbol-plot data onto a rectangular coordinate system having linear function scales for both X and Y ordinates.

PSLILG — To symbol-plot data onto a rectangular coordinate system having a linear function scale for the X ordinate and a logarithmic (any base) function scale for the Y ordinate.

PSLGLI — To symbol-plot data onto a rectangular coordinate system having a logarithmic (any base) function scale for the X ordinate and a linear function scale for the Y ordinate.

PSLGLG — To symbol-plot data onto a rectangular coordinate system having logarithmic (any base) function scales for both the X and Y ordinates.

PSPOLR — To symbol-plot data given in polar coordinates onto a rectangular coordinate system.

Method

Each data point is examined. If the point lies outside the range of either scale, the point is not plotted, nor is the plot symbol constructed. If the point lies within the range of both scales, the point is plotted and the plot symbol is constructed about the point.

Data points are defined as follows:

> The first point is defined by the coordinate pair [X(1),Y(1)]; the next sequential point is [X(1+XSKIPS), Y(1+YSKIPS)], which is followed by [X(1+2*XSKIPS), (Y(1+2*YSKIPS)], etc. POINTS -- *SPECS(13)* -- is the total number of X,Y coordinate pairs to be considered.

Restrictions

A. Point count: One point minumum

B. Data range: For linear function scales, the range is limited only by the range of a floating-point number (single precision). For logarithmic function scales, the range must contain floating-point numbers (single precision) which are all positive, or all negative, and non-zero.

Usage

See the following individual subroutine write-ups for:

A. FORTRAN call statement

B. Description of call list arguments

C. Routines Used

D. Example

Subroutine PSLILI

Purpose

To symbol-plot a set of numerical values (X ordinates) against another set (Y ordinates) in a rectangular coordinate system having linear functions scales for both axes.

Usage

A. FORTRAN call statement

CALL PSLILI (*X*,*Y*,*SPECS*)

B. Description of call list arguments

X An array containing the X ordinates of data points to be processed. Values are floating-point.

Y An array containing the Y ordinates of data points to be processed. Values are floating-point.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

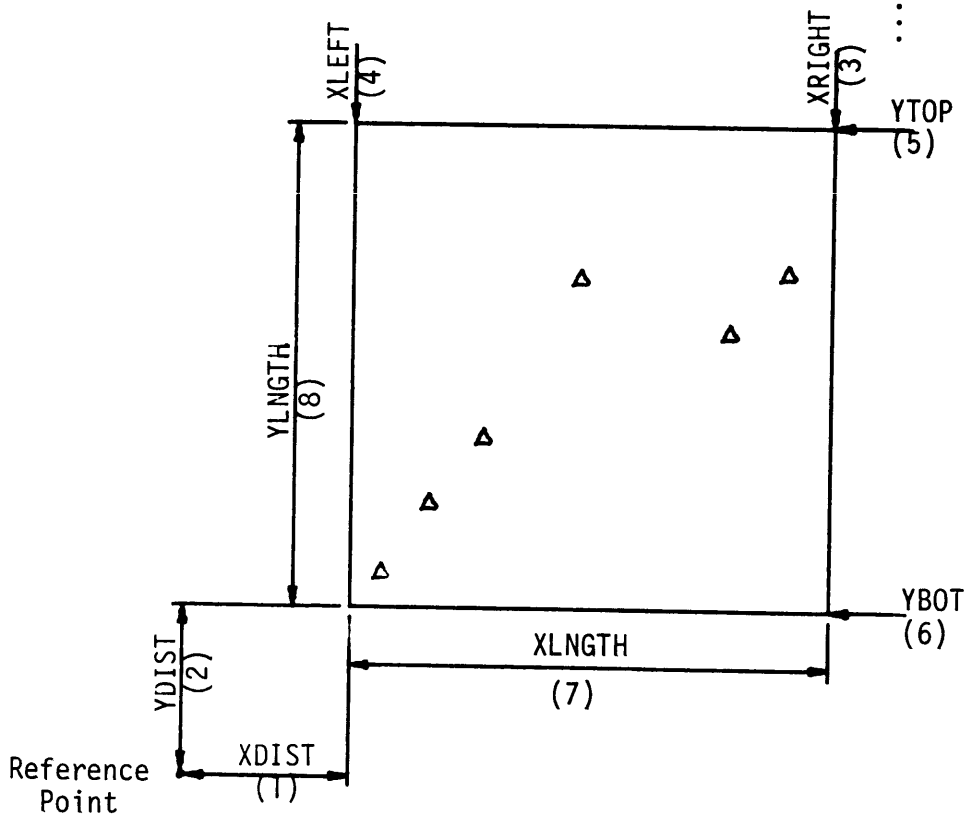
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(10)*		
SPECS(2)	*	YDIST	*SPECS(11)*	*	TOOL
SPECS(3)	*	XRIGHT	*SPECS(12)*	*	CCTAPE
SPECS(4)	*	XLEFT	*SPECS(13)*	*	POINTS
SPECS(5)	*	YTOP	*SPECS(14)*	*	XSKIPS
SPECS(6)	*	YBOT	*SPECS(15)*	*	YSKIPS
SPECS(7)	*	XLNGTH	*SPECS(16)*	*	SYMBOL
SPECS(8)	*	YLNGTH	*SPECS(17)*	*	FONTB
SPECS(9)			*SPECS(18)*	*	FONTH

The first point is defined by the coordinate pair [X(1), Y(1)]; The next sequential point is [X(1+XSKIPS), Y(1+YSKIPS)], which is followed by [X(1+2XSKIPS), Y(1+2YSKIPS)], etc.

C. Routines Used: WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30), X(10), Y(10)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET XRIGHT, XLEFT
      SPECS(3) = 10.0
      SPECS(4) =  1.0
C     SET YTOP, YBOT
      SPECS(5) = 10.0
      SPECS(6) =  1.0
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY NUMBER OF POINTS
      SPECS(13) = 10.
C     SET XSKIPS, YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C     SPECIFY SYMBOL NUMBER
      SPECS(16) = 1.
C     SET FONTB, FONTH (IN INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SETUP X, Y DATA ARRAYS
      X(1) = 0.1
      Y(1) = 0.1
      X(2) = 0.5
      Y(2) = 0.2
      X(3) = 1.2
      Y(3) = 0.7
      X(4) = 1.6
      Y(4) = 1.7
      X(5) = 2.5
      Y(5) = 3.0
      X(6) = 3.5
      Y(6) = 4.2
      X(7) = 5.3
      Y(7) = 7.2
      X(8) = 8.1
      Y(8) = 6.2
      X(9) = 9.2
      Y(9) = 7.3
      X(10) = 10.5
      Y(10) = 10.5
C     CALL SYMBOL PLOT ROUTINE
      CALL PSLILI (X,Y,SPECS)
      .
      .
      .
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine PSLILG

Purpose

To symbol-plot a set of numerical values (X ordinates) against another set (Y ordinates) in a rectangular coordinate system having a linear function scale for the X-axis and a logarithmic function scale for the Y-axis.

Usage

A. FORTRAN call statement

CALL PSLILG (*X*,*Y*,*SPECS*)

B. Description of call list arguments

X An array containing the X ordinates of data points to be processed. Values are floating-point.

Y An array containing the Y ordinates of data points to be processed. Values are floating-point.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

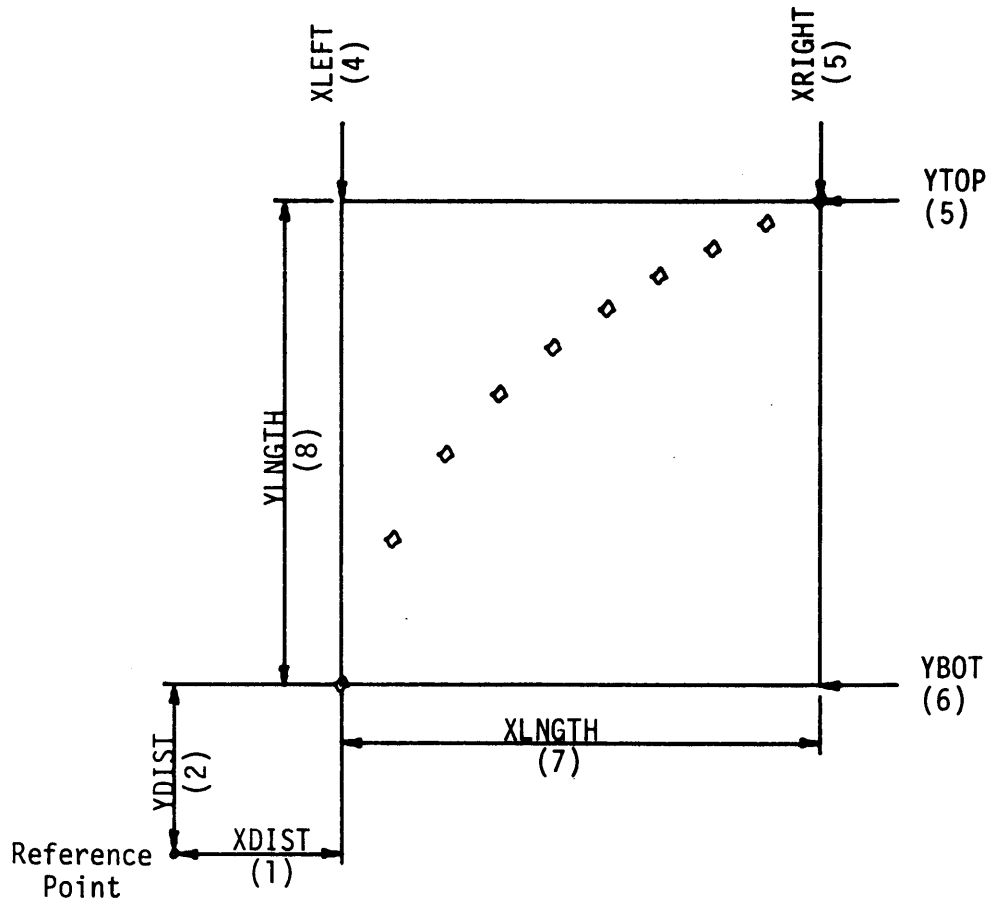
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(16)*	*	SYMBOL
SPECS(2)	*	YDIST	*SPECS(17)*	*	FONTB
SPECS(3)	*	XRIGHT	*SPECS(18)*	*	FONTH
SPECS(4)	*	XLEFT	*SPECS(19)*		
SPECS(5)	*	YBOT	*SPECS(20)*		
SPECS(6)	*	YTOP	*SPECS(21)*		
SPECS(7)	*	XLNGTH	*SPECS(22)*		
SPECS(8)	*	YLNGTH	*SPECS(23)*		
SPECS(9)			*SPECS(24)*		
SPECS(10)			*SPECS(25)*		
SPECS(11)	*	TOOL	*SPECS(26)*		
SPECS(12)	*	CCTAPE	*SPECS(27)*		
SPECS(13)	*	POINTS	*SPECS(28)*		
SPECS(14)	*	XSKIPS	*SPECS(29)*		
SPECS(15)	*	YSKIPS	*SPECS(30)*	*	STAPE

The first point is defined by the coordinate pair [X(1), Y(1)]; the next sequential point is [X(1+XSKIPS), Y(1+YSKIPS)], which is followed by [X(1+2XSKIPS), Y(1+2YSKIPS)], etc.

C. Routines Used: NONLIN, PSLILI, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30), X(10), Y(10)
C     SET XDIST, YIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT, XLEFT
      SPECS(3) = 10.0
      SPECS(4) =  1.0
C     SET YTOP, YBOT
      SPECS(5) = 10.0
      SPECS(6) =  1.0
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY NUMBER OF POINTS
      SPECS(13) = 10.
C     SET XSKIPS, YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C     SPECIFY SYMBOL NUMBER
      SPECS(16) = 2.
C     SET FONTB,FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SCRATCH FILE NUMBER
      SPECS(30) = 97.
C     SETUP SAMPLE X,Y ARRAYS
      DO 10 I = 1,10
      X(I) = I
      Y(I) = I
 10   CONTINUE
C     CALL SYMBOL PLOT ROUTINE
C     LINEAR X, LOG Y
      CALL PSLILG (X,Y,SPECS)
        :
        :
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine PSLGLI

Purpose

To symbol-plot a set of numerical values (X ordinates) against another set (Y ordinates) in a rectangular coordinate system having a logarithmic function scale for the X-axis, and a linear function scale for the Y-axis.

Usage

A. FORTRAN call statement

CALL PSLGLI (X,Y,SPECS)

B, Description of call list arguments

X An array containing the X ordinates of data points to be processed. Values are floating-point. All X values must be all positive, or all negative, and all non-zero.

Y An array containing the Y ordinates of data points to be processed. Values are floating-point.

SPECS An array containing plot construction parameters. Reference Table 1 under "The SPECS Array" for definitions and array format. This routine requires the SPECS parameters indicated below by the * to be specified prior to calling it:

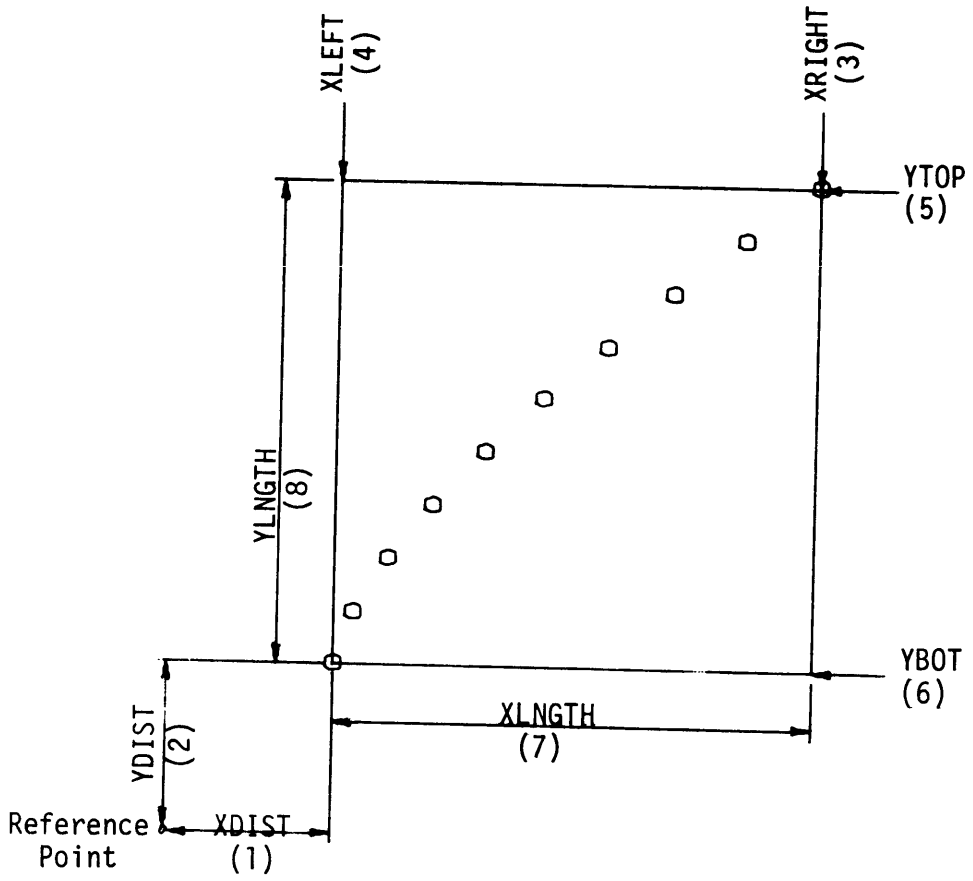
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	SPECS(16)	*	SYMBOL
SPECS(2)	*	YDIST	SPECS(17)	*	FONTB
SPECS(3)	*	XRIGHT	SPECS(18)	*	FONTH
SPECS(4)	*	XLEFT	SPECS(19)		
SPECS(5)	*	YTOP	SPECS(20)		
SPECS(6)	*	YBOT	SPECS(21)		
SPECS(7)	*	XLNGTH	SPECS(22)		
SPECS(8)	*	YLNGTH	SPECS(23)		
SPECS(9)			SPECS(24)		
SPECS(10)			SPECS(25)		
SPECS(11)	*	TOOL	SPECS(26)		
SPECS(12)	*	CCTAPE	SPECS(27)		
SPECS(13)	*	POINTS	SPECS(28)		
SPECS(14)	*	XSKIPS	SPECS(29)		
SPECS(15)	*	YSKIPS	SPECS(30)	*	STAPE

The first point is defined by the coordinate pair [X(1), Y(1)]; the next sequential point is [X(1+XSKIPS), Y(1+YSKIPS)], which is followed by [X(1+2XSKIPS), Y(1+2YSKIPS)], etc.

C, Routines Used: NONLIN, PSLILI, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30), X(10), Y(10)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET XRIGHT, XLEFT
      SPECS(3) = 5**5
      SPECS(4) = .5**.5
C     SET YTOP,YBOT
      SPECS(5) = 10.0
      SPECS(6) =  1.0
C     SET XLNGTH,YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY NUMBER OF POINTS
      SPECS(13) = 10.
C     SET XSKIPS,YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C     SPECIFY SYMBOL NUMBER
      SPECS(16) = 3.
C     SET FONTB,FONTH (IN INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SCRATCH FILE NUMBER
      SPECS(30) = 97.
C     SETUP SAMPLE X,Y ARRAYS
      DO 10  I=1,10
      RI = FLOAT(I)/2.
      X(I) = RI**RI
      Y(I) = I
 10   CONTINUE
C     CALL SYMBOL PLOT ROUTINE
      LOG X, LINEAR Y
C     CALL PSLGLI (X,Y,SPECS)
                .
                .
                .
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine PSLGLG

Purpose

To symbol-plot a set of numerical values (X ordinates) against another set (Y ordinates) in a rectangular coordinate system having logarithmic function scales for both axes.

Usage

A. FORTRAN call statement

```
CALL PSLGLG (X,Y,SPECS)
```

B. Description of call list arguments

X An array containing the X ordinates of data points to be processed. Values are floating-point. All X values must be all positive, or all negative, and all non-zero.

Y An array containing the Y ordinates of data points to be processed. Values are floating-point. All Y values must be all positive, or all negative, and all non-zero.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by the * to be specified prior to calling it:

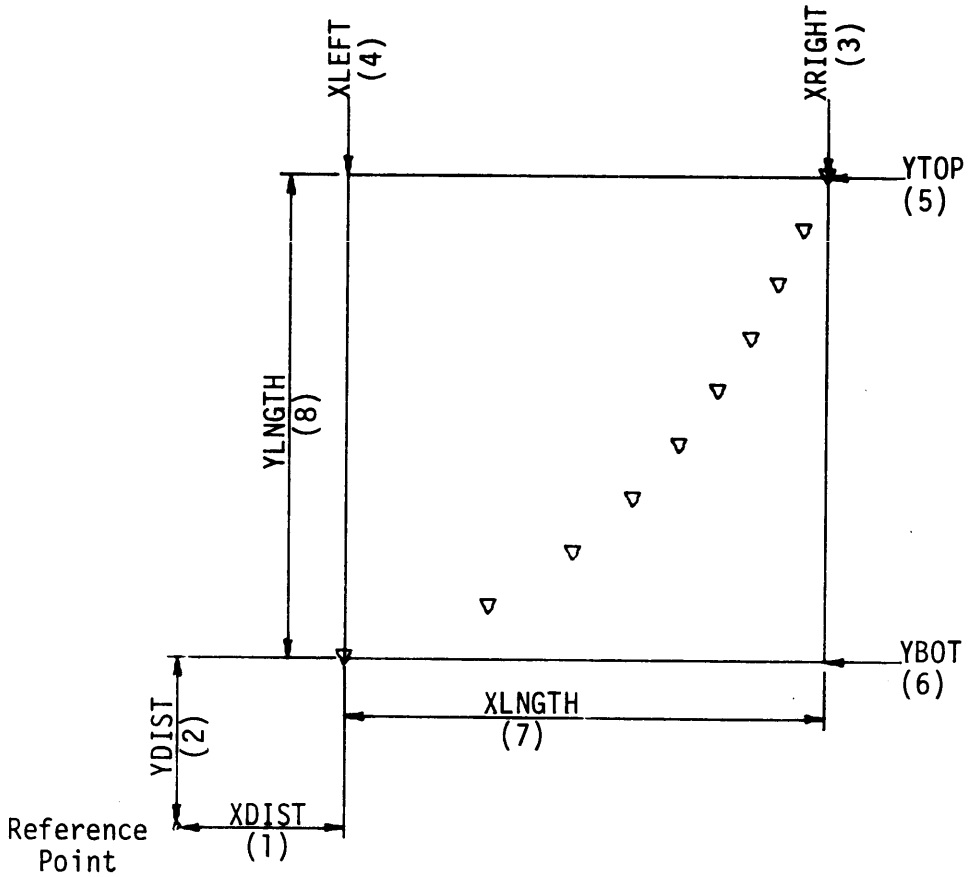
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(16)*	*	SYMBOL
SPECS(2)	*	YDIST	*SPECS(17)*	*	FONTB
SPECS(3)	*	XRIGHT	*SPECS(18)*	*	FONTH
SPECS(4)	*	XLEFT	*SPECS(19)*		
SPECS(5)	*	YTOP	*SPECS(20)*		
SPECS(6)	*	YBOT	*SPECS(21)*		
SPECS(7)	*	XLNGTH	*SPECS(22)*		
SPECS(8)	*	YLNGTH	*SPECS(23)*		
SPECS(9)			*SPECS(24)*		
SPECS(10)			*SPECS(25)*		
SPECS(11)	*	TOOL	*SPECS(26)*		
SPECS(12)	*	CCTAPE	*SPECS(27)*		
SPECS(13)	*	POINTS	*SPECS(28)*		
SPECS(14)	*	XSKIPS	*SPECS(29)*		
SPECS(15)	*	YSKIPS	*SPECS(30)*	*	STAPE

The first point is defined by the coordinate pair [X(1), Y(1)]; the next sequential point is [X(1+XSKIPS), Y(1+YSKIPS)], which is followed by [X(1+2XSKIPS), Y(1+2YSKIPS)], etc.

C. Routines Used: NONLIN, PSLILI, WRGHT, SIGNON

D. Example

```
      DIMENSION SPECS(30), X(10), Y(10)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT, XLEFT
      SPECS(3) = 100.
      SPECS(4) =   1.
C     SET YTOP, YBOT
      SPECS(5) = 2.**10
      SPECS(6) = 2.
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY NUMBER OF POINTS
      SPECS(13) = 10.
C     SPECIFY XSKIPS, YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C     SPECIFY SYMBOL NUMBER
      SPECS(16) = 4.
C     SPECIFY FONTB, FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SCRATCH FILE NUMBER
      SPECS(30) = 97.
C     SETUP SAMPLE X,Y ARRAYS
      DO 10  I = 1,10
      X(I) = I**2
      Y(I) = 2.**I
 10   CONTINUE
C     CALL SYMBOL PLOT ROUTINE
C     LOG X, LOG Y
      CALL PSLGLG (X,Y,SPECS)
            :
            :
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine PSPOLR

Purpose

To symbol-plot data onto a rectangular coordinate system with data given in polar coordinates.†

Usage

A. FORTRAN call statement

CALL PSLILI (*R*,*THETA*,*SPECS*)

B. Description of call list arguments

R An array containing the vector lengths of data points to be processed. Values are floating-point.

THETA An array containing the angles, given in radians of data points to be processed. Values are floating-point.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(10)*		
SPECS(2)	*	YDIST	*SPECS(11)*	*	TOOL
SPECS(3)	*	XRIGHT	*SPECS(12)*	*	CCTAPE
SPECS(4)	*	XLEFT	*SPECS(13)*	*	POINTS
SPECS(5)	*	YTOP	*SPECS(14)*	*	XSKIPS
SPECS(6)	*	YBOT	*SPECS(15)*	*	YSKIPS
SPECS(7)	*	XLNGTH	*SPECS(16)*	*	SYMBOL
SPECS(8)	*	YLNGTH	*SPECS(17)*	*	FONTB
SPECS(9)			*SPECS(18)*	*	FONTH
			and		
			SPECS(30)	*	STAPE

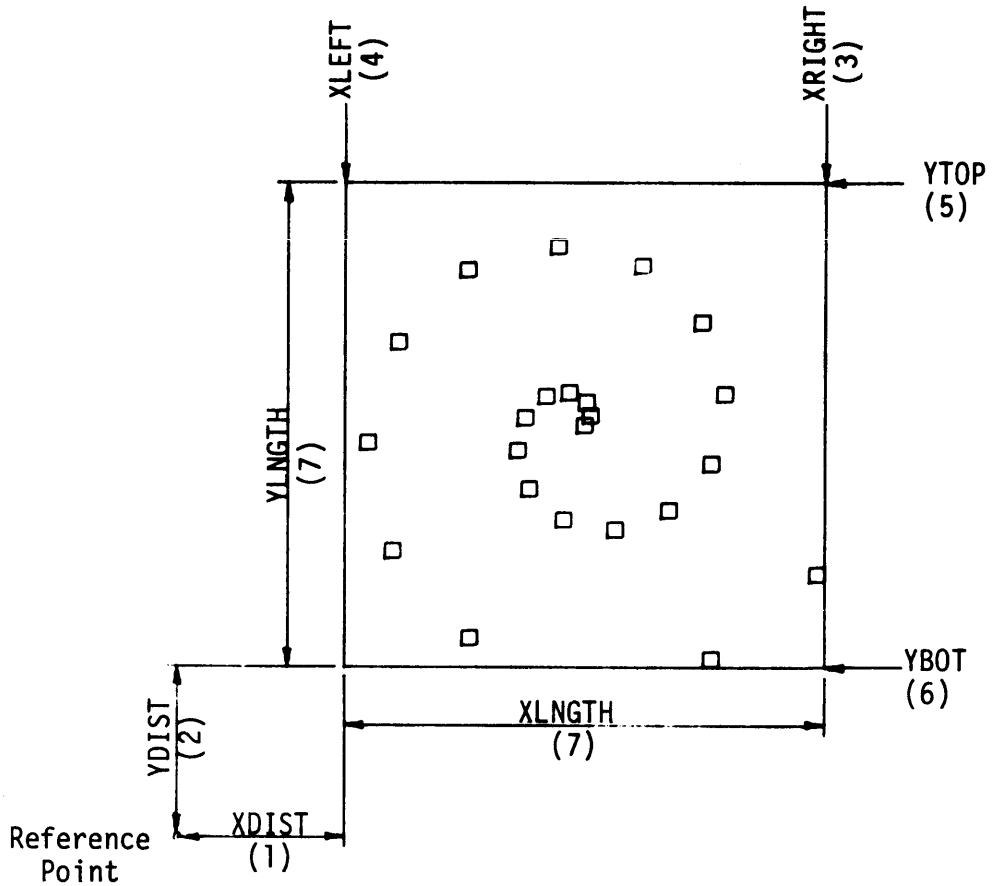
The first point is defined by the coordinate pair [R(1),THETA(1)]; the next sequential points is [R(1+XSKIPS),THETA(1+YSKIPS)], which is followed by [R(1+2XSKIPS),THETA(1+2YSKIPS)], etc.

C. Routines used: POLAR, DATAPE, PSLILI, SIGNON, WRGHT

† The given polar coordinates are not changed.

D. Example

```
      DIMENSION SPECS(30),R(25),THETA(25)
C     SET XDIST,YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT, XLEFT
      SPECS(3) = 100.0
      SPECS(4) = -100.0
C     SET YTOP,YBOT
      SPECS(5) = 100.0
      SPECS(6) = -100.0
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY NUMBER OF POINTS
      SPECS(13) = 25.
C     SET XSKIPS,YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C     SPECIFY SYMBOL NUMBER
      SPECS(16) = 5.
C     SET FONTB,FONTH (IN INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SCRATCH FILE NUMBER
      SPECS(30) = 97.
C     SETUP SAMPLE R,THETA ARRAYS
      DO 10  I=1,25
      RI = I
      R(I) = 5*(I-1)
      THETA(I) = RI/2.
 10   CONTINUE
C     CALL POLAR SYMBOL PLOT ROUTINE
      CALL PSPOLR (R,THETA,SPECS)
            :
            :
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Routines to Construct Straight Lines Through Data

Routine Names

SLLILI, SLLILG, SLLGLI, SLLGLG, SLPOLR

Purpose

To construct straight lines between consecutive data points. The coordinate system may be linear X and linear Y (SLLILI), linear X and logarithmic Y (SLLILG), logarithmic X and linear Y (SLLGLI), or logarithmic X and logarithmic Y (SLLGLG), or polar (SLPOLR).

Method

The routine examines each pair of consecutive points. If both points lie within the defined plotting area, including on its boundaries, the routine constructs a straight line that connects the two points. If either, or both, of the points (lie(s) outside the defined plotting area, the routine constructs between the two points the segment of the line common with the defined plotting area. If the point set consists of a single point, the routine plots the point if the point does not lie outside the defined plotting area.

Restrictions

A. Point count: One point minimum.

B. Data Range: Range is limited only by the range of a floating-point number (single precision).

Usage

See the following individual subroutine writeups for:

A. FORTRAN call statement

B. Description of call list arguments

C. Routines Used

D. Example

Subroutine SLLILI

Purpose

To serially connect with straight lines a set of points in a rectangular coordinate system having linear function scales for both axes.

Usage

A. FORTRAN call statement:

CALL SLLILI (*X*, *Y*, *SPECS*)

B. Description of call list arguments

X An array containing the X ordaintes of data points to be processed. Values are floating-point.

Y An array containing the Y ordinates of data points to be processed. Values are floating-point.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prio to calling it:

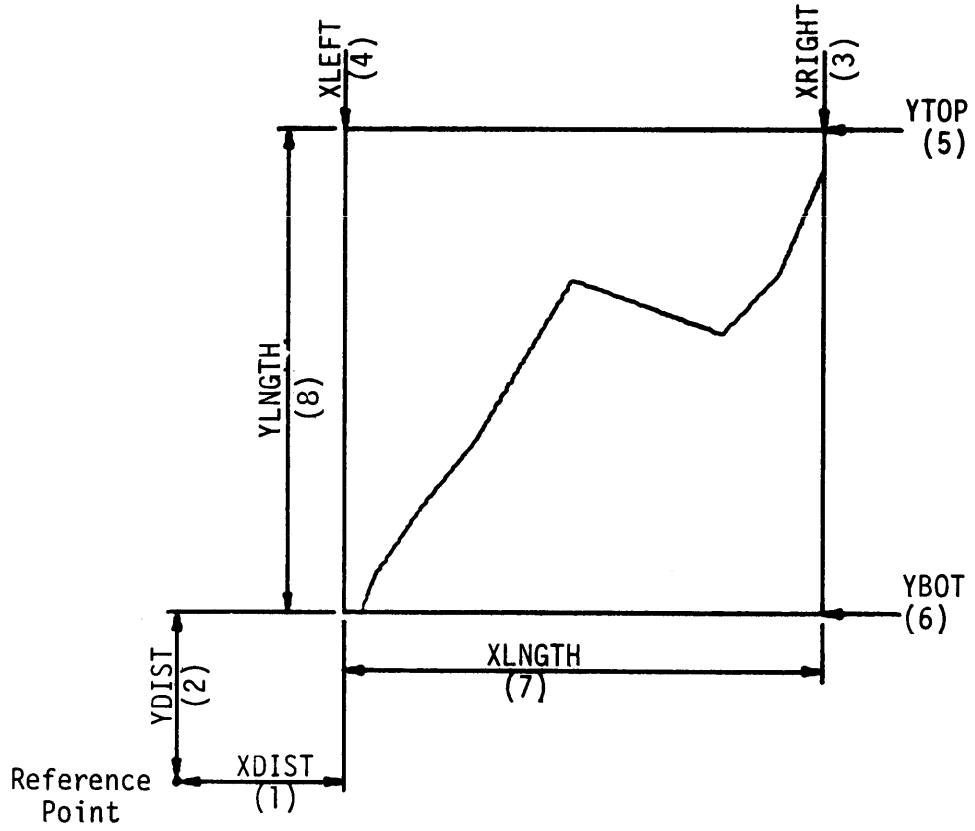
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(9)*		
SPECS(2)	*	YDIST	*SPECS(10)*		
SPECS(3)	*	XRIGHT	*SPECS(11)*	*	TOOL
SPECS(4)	*	XLEFT	*SPECS(12)*	*	CCTAPE
SPECS(5)	*	YTOP	*SPECS(13)*	*	POINTS
SPECS(6)	*	YBOT	*SPECS(14)*	*	XSKIPS
SPECS(7)	*	XLNGTH	*SPECS(15)*	*	YSKIPS
SPECS(8)	*	YLNGTH			

The first point is defined by the coordinate pair [X(1), Y(1)]; the next sequential point is [X(1+XSKIPS), Y(1+YSKIPS)], which is followed by [X(1+2XSKIPS), Y(1+2YSKIPS)], etc.

C. Routines Used: SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30), X(10), Y(10)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET XRIGHT, XLEFT
      SPECS(3) = 10.0
      SPECS(4) =  1.0
C     SET YTOP, YBOT
      SPECS(5) = 10.0
      SPECS(6) =  1.0
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY NUMBER OF POINTS
      SPECS(13) = 10.
C     SET XKSIPS, YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C     SETUP X,Y DATA ARRAYS
      X(1) = 0.1
      Y(1) = 0.1
      X(2) = 0.5
      Y(2) = 0.2
      X(3) = 1.2
      Y(3) = 0.7
      X(4) = 1.6
      Y(4) = 1.7
      X(5) = 2.5
      Y(5) = 3.0
      X(6) = 3.5
      Y(6) = 4.2
      X(7) = 5.3
      Y(7) = 7.2
      X(8) = 8.1
      Y(8) = 6.2
      X(9) = 9.2
      Y(9) = 7.3
      X(10) = 10.5
      Y(10) = 10.5
C     CALL STRAIGHT LINE ROUTINE
C     LINEAR X, LINEAR Y
      CALL SLLILI (X,Y,SPECS)
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine SLLILG

Purpose

To serially connect with straight lines a set of points in a rectangular coordinate system having a linear function scale for the X-axis and a logarithmic function scale for the Y-axis.

Usage

A. FORTRAN call statement:

CALL SLLILG (*X*, *Y*, *SPECS*)

B. Description of call list arguments

X An array containing the X ordinates of data points to be processed. Values are floating-point.

Y An array containing the Y ordinates of data points to be processed. Y ordinate range must contain floating-point numbers (single precision) which are all positive or negative; and non-zero.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

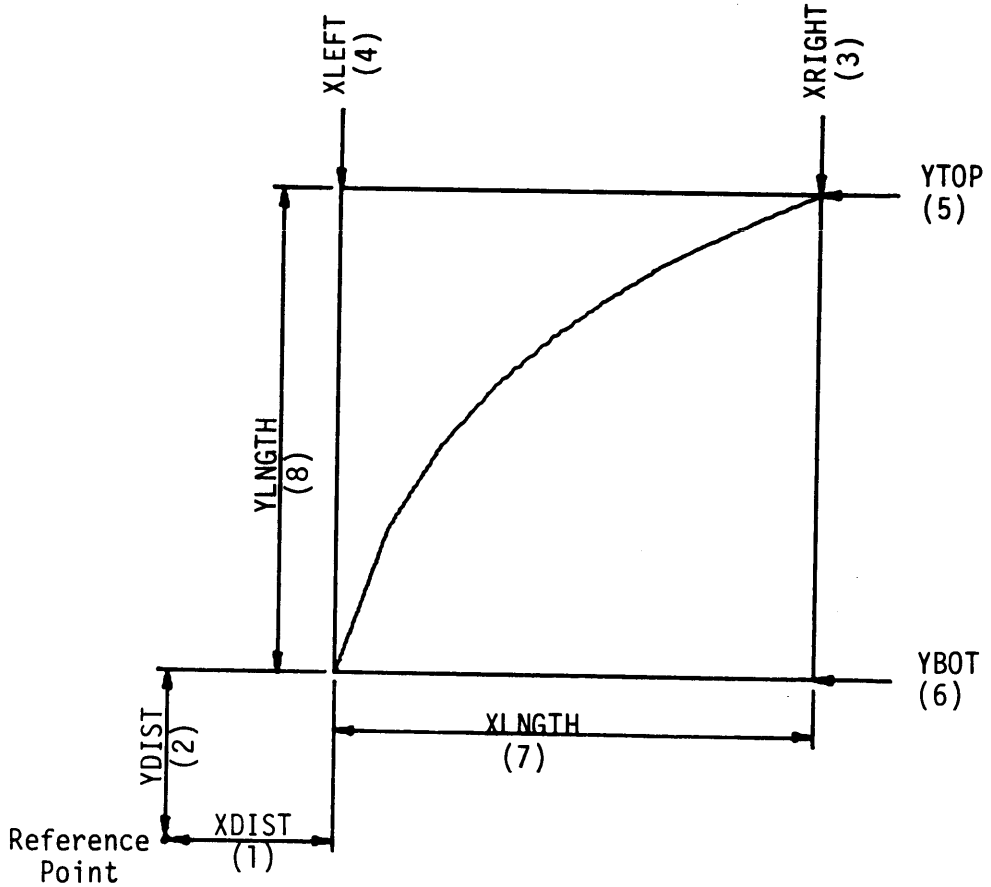
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(9)*		
SPECS(2)	*	YDIST	*SPECS(10)*		
SPECS(3)	*	XRIGHT	*SPECS(11)*	*	TOOL
SPECS(4)	*	XLEFT	*SPECS(12)*	*	CCTAPE
SPECS(5)	*	YTOP	*SPECS(13)*	*	POINTS
SPECS(6)	*	TBOT	*SPECS(14)*	*	XSKIPS
SPECS(7)	*	XLNGTH	*SPECS(15)*	*	YSKIPS
SPECS(8)	*	YLNGTH	⋮		
			SPECS(30)	*	STAPE

Data points are defined as follows: The first point is defined by the coordinate pair [X(1), Y(1)]; the next sequential point is [X(1+XSKIPS), Y(1+YSKIPS)], which is followed by [X(1+2XSKIPS). Y(1+2YSKIPS)], etc.

C. Routines Used: NONLIN, SLLILI, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30), X(10), Y(10)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT, XLEFT
      SPECS(3) = 10.0
      SPECS(4) =  1.0
C     SET YTOP, YBOT
      SPECS(5) = 10.0
      SPECS(6) =  1.0
C     SET XLNGTH, YLNGTH
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY NUMBER OF POINTS
      SPECS(13) = 10.
C     SET XSKIPS, YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C     SPECIFY SCRATCH FILE NUMBER
      SPECS(30) = 97.
C     SETUP SAMPLE X,Y ARRAYS
      DO 10  I = 1,10
      X(I) = I
      Y(I) = I
 10   CONTINUE
C     CALL STRAIGHT LINE ROUTINE
C     LINEAR X, LOG Y
      CALL SLLILG
        :
        :
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine SLLGLI

Purpose

To serially connect with straight lines a set of points in a rectangular coordinate system having a logarithmic function scale for the X-axis and a linear function scale for the Y-axis.

Usage

A. FORTRAN call statement:

CALL SLLGLI (*X*, *Y*, *SPECS*)

B. Description of call list arguments

X An array containing the X ordinates of data points to be processed. X ordinate range must contain floating-point numbers (single precision) which are all positive or all negative, and non-zero.

Y An array containing the Y ordinates of data points to be processed. Values are floating-point.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by the * to be specified prior to callinf it:

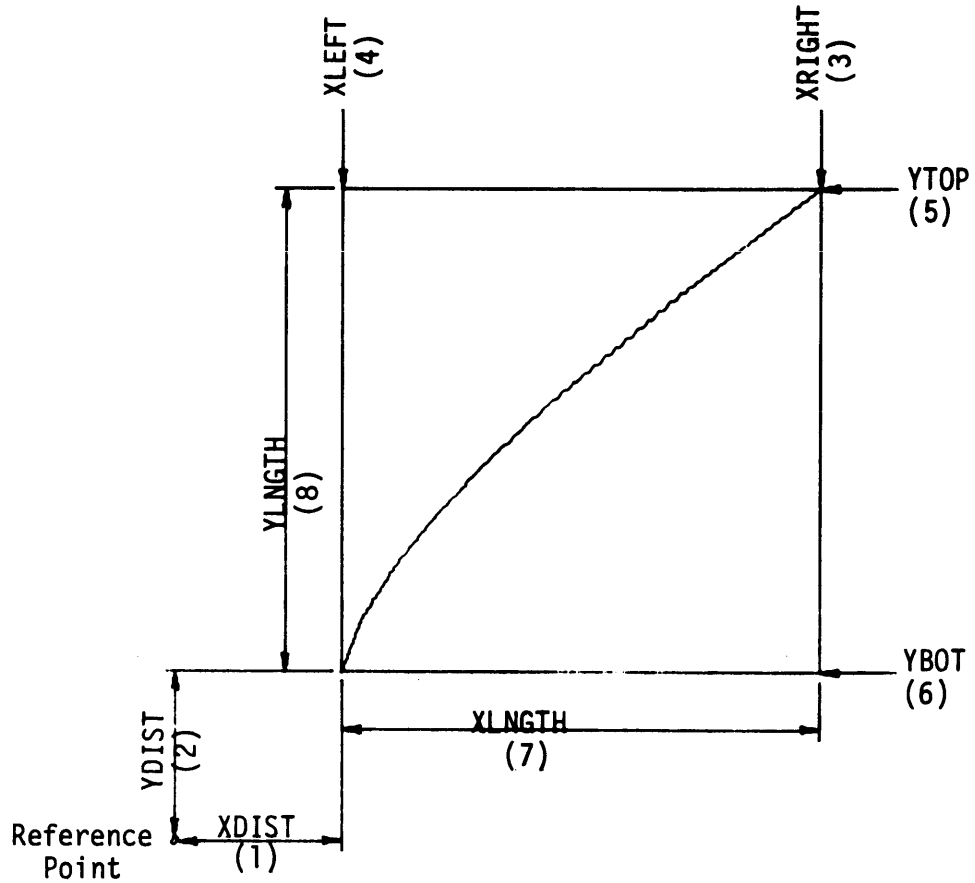
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(9)*		
SPECS(2)	*	YDIST	*SPECS(10)*		
SPECS(3)	*	XRIGHT	*SPECS(11)*	*	TOOL
SPECS(4)	*	XLEFT	*SPECS(12)*	*	CCTAPE
SPECS(5)	*	YTOP	*SPECS(13)*	*	POINTS
SPECS(6)	*	YBOT	*SPECS(14)*	*	XSKIPS
SPECS(7)	*	XLNGTH	*SPECS(15)*	*	YSKIPS
SPECS(8)	*	YLNGTH	⋮		
			SPECS(30)	*	STAPE

Data points are defined as follows: The first point is defined by the coordinate pair [X(1), Y(1)]; the next sequential point is [X(1+XSKIPS), Y(1+YSKIPS)], which is followed by [X(1+2XSKIPS), Y(1+2YSKIPS)], etc.

C. Routines Used: NONLIN, SLLILI, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30), X(10), Y(10)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET XRIGHT, XLEFT
      SPECS(3) = 5**5
      SPECS(4) = .5**.5
C     SET YTOP,YBOT
      SPECS(5) = 10.0
      SPECS(6) =  1.0
C     SET XLNGTH,YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 97.
C     SPECIFY NUMBER OF POINTS
      SPECS(13) = 10.
C     SET XSKIPS,YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C     SPECIFY SCRATCH FILE NUMBER
      SPECS(30) = 97.
C     SETUP SAMPLE X,Y ARRAYS
      DO 10  I=1,10
      RI = FLOAT(I)/2.
      X(I) = RI**RI
      Y(I) = I
 10   CONTINUE
C     CALL STRAIGHT LINE ROUTINE
C     LOG X, LINEAR Y
      CALL SLLGLI (X,Y,SPECS)
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine SLLGLG

Purpose

To serially connect with straight lines a set of points in a rectangular coordinate system having logarithmic function scales for both axes.

Usage

A. FORTRAN call statement:

CALL SLLGLG (*X*, *Y*, *SPECS*)

B. Description of call list arguments

X An array containing the X ordaintes of data points to be processed. X ordainte range must contain floating-point numbers (single precision) which are all positive or all negative, and non-zero.

Y An array containing the Y ordinates of data points to be processed. Y ordinate range must contain floating-point numbers (single precision) which are all positive or negative; and non-zero.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

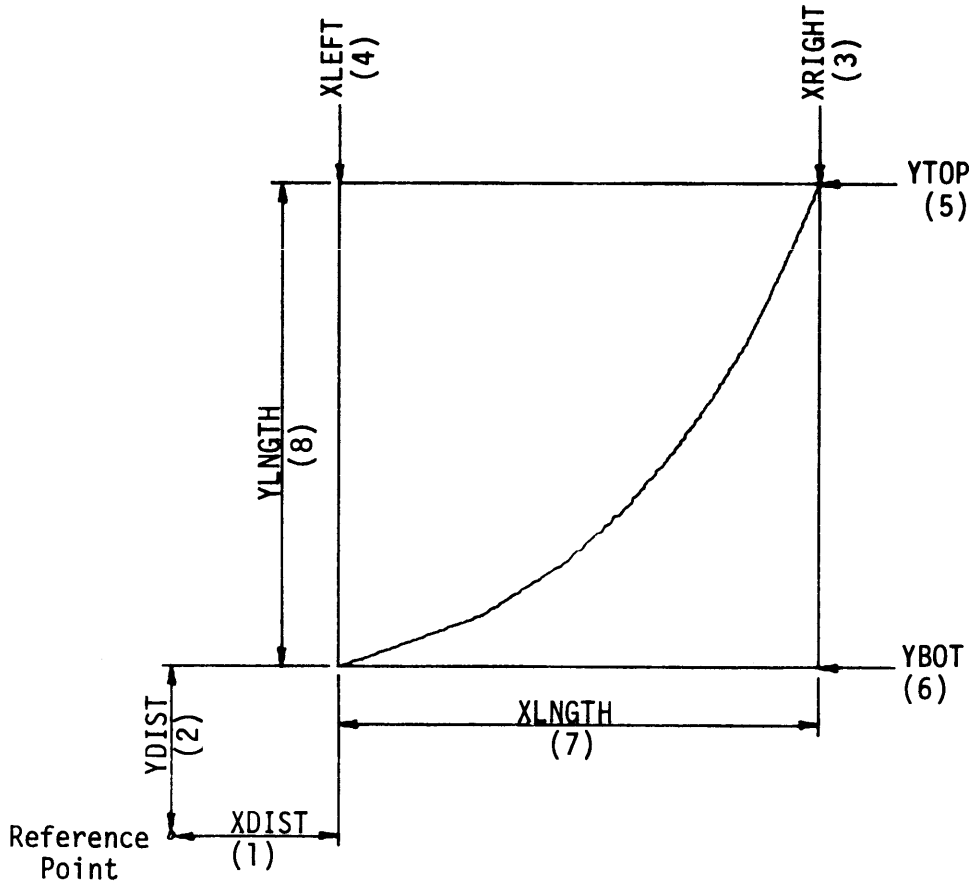
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(9)*		
SPECS(2)	*	YDIST	*SPECS(10)*		
SPECS(3)	*	XRIGHT	*SPECS(11)*	*	TOOL
SPECS(4)	*	XLEFT	*SPECS(12)*	*	CCTAPE
SPECS(5)	*	YTOP	*SPECS(13)*	*	POINTS
SPECS(6)	*	YBOT	*SPECS(14)*	*	XSKIPS
SPECS(7)	*	XLNGTH	*SPECS(15)*	*	YSKIPS
SPECS(8)	*	YLNGTH	⋮		
			SPECS(30)	*	STAPE

Data points are defined as follows: The first point is defined by the coordinate pair [X(1), Y(1)]; the next sequential point is [X(1+XSKIPS), Y(1+YSKIPS)], which is followed by [X(1+2XSKIPS), Y(1+2YSKIPS)], etc.

C. Routines Used: NONLIN, SLLILI, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30), X(10), Y(10)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT, XLEFT
      SPECS(3) = 100.
      SPECS(4) =   1.
C     SET YTOP, YBOT
      SPECS(5) = 2.**10
      SPECS(6) = 2.
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY NUMBER OF POINTS
      SPECS(13) = 10.
C     SPECIFY XSKIPS, YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C     SPECIFY SCRATCH FILE NUMBER
      SPECS(30) = 97.
C     SETUP SAMPLE X,Y ARRAYS
      DO 10  I=1,10
      X(I) = I**2
      Y(I) = 2.**I
 10   CONTINUE
C     CALL STRAIGHT LINE ROUTINE
C     LOG X, LOG Y
      CALL SLLGLG (X,Y,SPECS)
           :
           :
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine SLPOLR

Purpose

To construct straight lines between consecutive data points given in polar coordinates.†

Usage

A. FORTRAN call statement

```
CALL SLPOLR (R,THETA,SPECS)
```

B. Description of call list arguments

R An array containing the vector lengths of data points to be processed. Values are floating-point.

THETA An array containing the angles, given in radians of data points to be processed. Values are floating-point.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(9)*		
SPECS(2)	*	YDIST	*SPECS(10)*		
SPECS(3)	*	XRIGHT	*SPECS(11)*	*	TOOL
SPECS(4)	*	XLEFT	*SPECS(12)*	*	CCTAPE
SPECS(5)	*	YTOP	*SPECS(13)*	*	POINTS
SPECS(6)	*	YBOT	*SPECS(14)*	*	XSKIPS
SPECS(7)	*	XLNGTH	*SPECS(15)*	*	YSKIPS
SPECS(8)	*	YLNGTH	and		
			SPECS(30)	*	STAPE

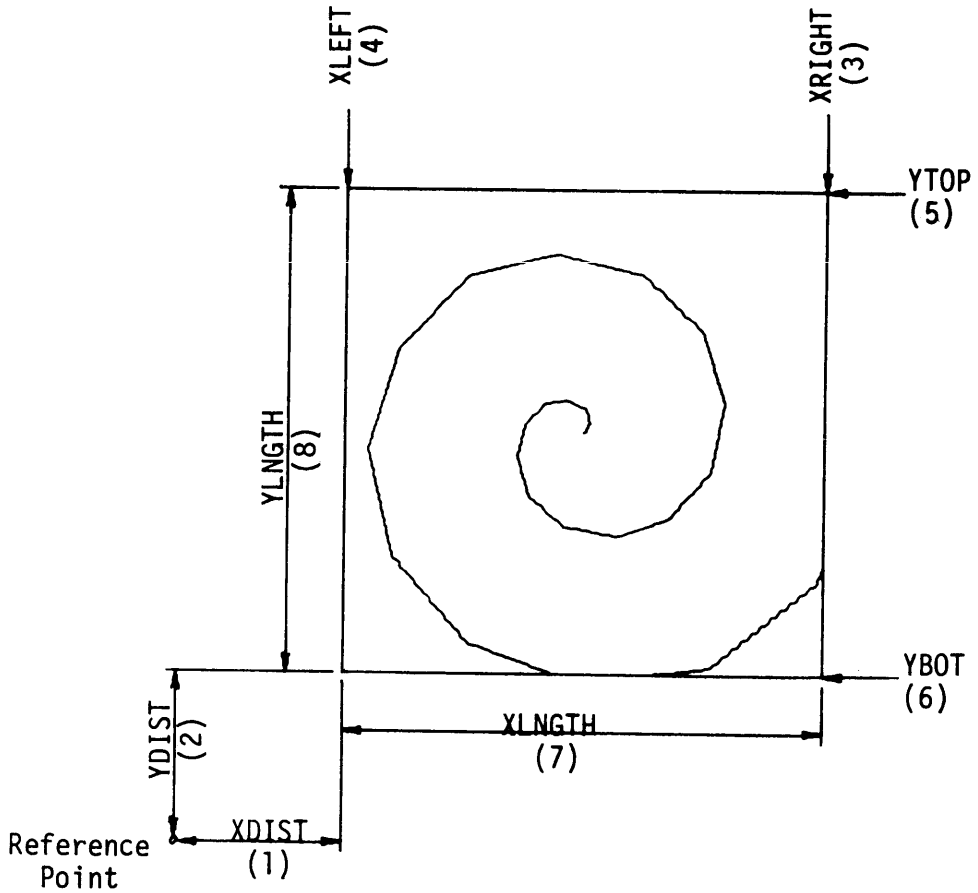
The first point is defined by the coordinate pair [R(1),THETA(1)]; the next sequential point is [R(1+XSKIPS),THETA(1+YSKIPS)], which is followed by [R(1+2XSKIPS),THETA(1+2YSKIPS)], etc.

C. Routines Used: POLAR, DATAPE, SLLILI, SIGNON, WRIGHT

† The given polar coordinates are not changed.

D. Example

```
      DIMENSION SPECS(30),R(25),THETA(25)
C     SET XDIST,YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT,XLEFT
      SPECS(3) =  100.0
      SPECS(4) = -100.0
C     SET YTOP,YBOT
      SPECS(5) =  100.0
      SPECS(6) = -100.0
C     SET XLNGTH,YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY NUMBER OF POINTS
      SPECS(13) = 25.
C     SET XSKIPS, YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C     SPECIFY SCRATCH FILE NUMBER
      SPECS(30) = 97.
C     SETUP SAMPLE R,THETA ARRAYS
      DO 10  I=1,25
      RI = I
      R(I) = 5*(I-1)
      THETA(I) = RI/2.
 10   CONTINUE
C     CALL POLAR STRAIGHT LINE ROUTINE
      CALL SLPOLR(R,THETA,SPECS)
            :
            :
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Routines Employing Parabolic Fairing Technique to Construct a Trend Curve Through Data

Routine Names

PFLILI, PFLILG, PFLGLI, PFLGLG, PFPOLR

Purpose

To construct via a parabolic fairing technique a trend curve through a set of points. The coordinate system may be linear X and linear Y (PFLILI), linear X and logarithmic Y (PFLILG), logarithmic X and linear Y (PFLGLI), logarithmic X and logarithmic Y (PFLGLG) or polar (PFPOLR).

Method

A. Point set contains less that three points:

If the given point set consists of two points, the trend curve is the straight line segment that joins the two points. If the point set consists of a single point, the trend curve is the point. The routine constructs only that portion of the trend curve common within the defined plotting area.

B. Point set contains three or more points:

The routine interpolates a series of points to establish the trend curve. The interpolation scheme employs a group of rules, called "three-point rules", plus a blending rule. The three-point rules pertain to the solution of the two parabolic equations:

$$Y = aX^2 + bX + c$$

$$X = AY^2 + BY + C$$

The blending rule combines double solutions of these equations to produce a well-behaved trend curve such that there is maintained a one-to-one correspondence between the relative extremes in the data and the relative extremes implied by the curve.

The routine constructs only that portion of the trend curve common with the defined plotting area.

C. If two consecutive points lie either in a horizontal or vertical line, the routine connects these two points with a straight line.

Restrictions

A. Point Count: One point minimum.

B. Data Range: Range is limited only by the range of a floating-point number (single precision).

Usage

See the following individual subroutine writeups for:

A. FORTRAN call statement

B. Description of call list arguments

C. Routines Used

D. Example

Subroutine PFLILI

Purpose

To construct via a parabolic fairing technique a trend curve through a set of points in a rectangular coordinate system having linear functions scales for both axes.

Usage

A. FORTRAN call statement

CALL PFLILI (*X*,*Y*,*BUFX*,*BUFY*,*SPECS*)

B. Description of call list arguments:

X An array containing the X ordinates of data points to be processed. Values are floating-point.

Y An array containing the Y ordinates of data points to be processed. Values are floating-point.

BUFX An array of 500_{10} consecutive words provided as a work region for the routine to use to process the data.

BUFY An array of 500_{10} consecutive words provided as a work region for the routine to use to process the data.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

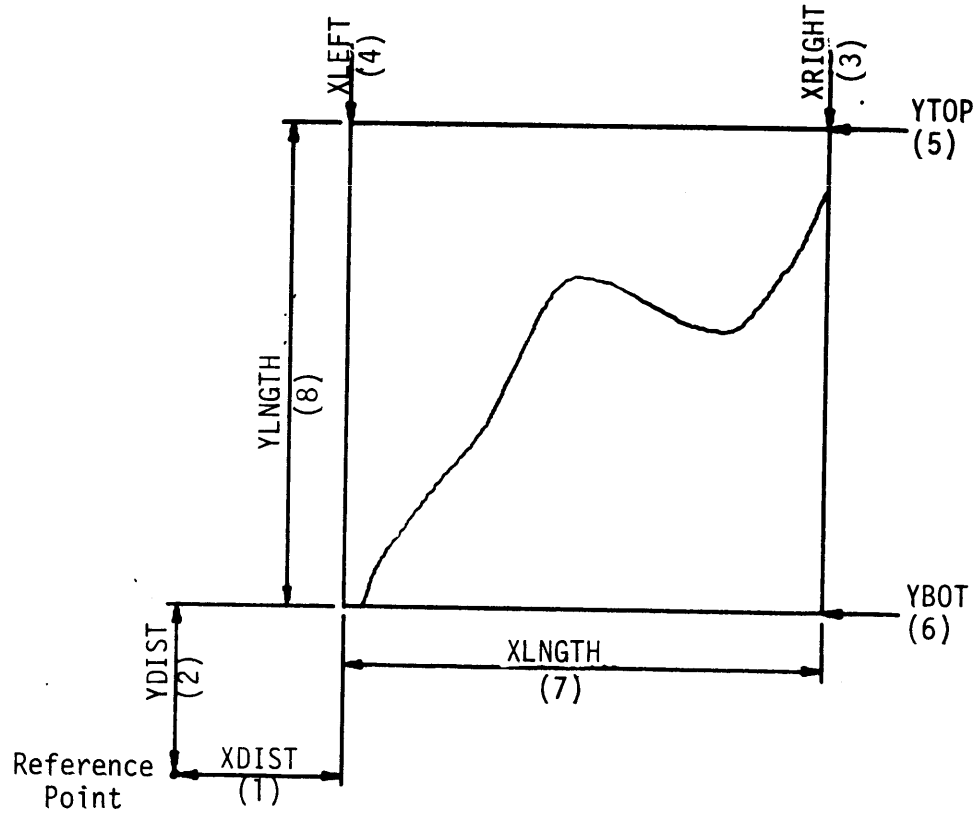
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(9)*		
SPECS(2)	*	YDIST	*SPECS(10)*		
SPECS(3)	*	XRIGHT	*SPECS(11)*	*	TOOL
SPECS(4)	*	XLEFT	*SPECS(12)*	*	CCTAPE
SPECS(5)	*	YTOP	*SPECS(13)*	*	POINTS
SPECS(6)	*	YBOT	*SPECS(14)*	*	XSKIPS
SPECS(7)	*	XLNGTH	*SPECS(15)*	*	YSKIPS
SPECS(8)	*	YLNGTH			

Data points are defined as follows: The first point is defined by the coordinate pair [X(1), Y(1)]; the next sequential point is [X(1+XSKIPS), Y(1+YSKIPS)], which is followed by [X(1+2XSKIPS), Y(1+2YSKIPS)], etc.

C. Routines Used: FAIR, SLLILI, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30), X(10), Y(10)
      DIMENSION BUFX(500), BUFY(500)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET XRIGHT, XLEFT
      SPECS(3) = 10.0
      SPECS(4) =  1.0
C     SET YTOP, YBOT
      SPECS(5) = 10.0
      SPECS(6) =  1.0
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY NUMBER OF POINTS
      SPECS(13) = 10.
C     SET XSKIPS, YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C     SET X,Y DATA ARRAYS
      X(1) = 0.1
      Y(1) = 0.1
      X(2) = 0.5
      Y(2) = 0.2
      X(3) = 1.2
      Y(3) = 0.7
      X(4) = 1.6
      Y(4) = 1.7
      X(5) = 2.5
      Y(5) = 3.0
      X(6) = 3.5
      Y(6) = 4.2
      X(7) = 5.3
      Y(7) = 7.2
      X(8) = 8.1
      Y(8) = 6.2
      X(9) = 9.2
      Y(9) = 7.3
      X(10) = 10.5
      Y(10) = 10.5
C     CALL PARABOLIC FAIRING ROUTINE
C     LINEAR X, LINEAR Y
      CALL PFLILI (X,Y,BUFX,BUFY,SPECS)
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine PFLILG

Purpose

To construct via a parabolic fairing technique a trend curve through a set of points in a rectangular coordinate system having a linear function scale for the X-axis and a logarithmic function scale for the Y-axis.

Usage

A. FORTRAN call statement:

CALL PFLILG (*X*,*Y*,*BUFX*,*BUFY*,*SPECS*)

B. Description of call list arguments:

X An array containing the X ordinates of data points to be processed. Values are floating-point.

Y An array containing the Y ordinates of data points to be processed. Y ordinate range must contain floating-point numbers (single precision) which are all positive, or all negative, and non-zero.

BUFX An array of 500_{10} consecutive words provided as a work region for the routine to use to process the data.

BUFY An array of 500_{10} consecutive words provided as a work region for the routine to use to process the data.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

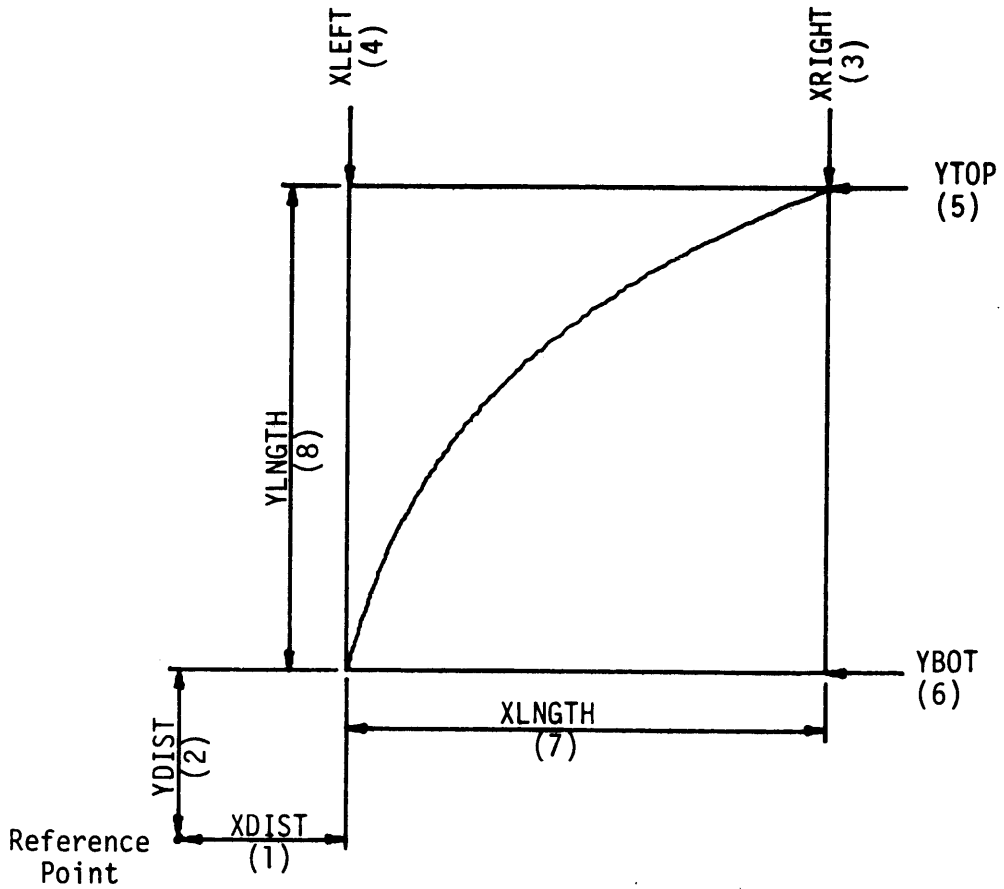
Parameter	Mnemonic	Parameter	Mnemonic
SPECS(1) *	XDIST	*SPECS(9)*	
SPECS(2) *	YDIST	*SPECS(10)*	
SPECS(3) *	XRIGHT	*SPECS(11)* *	TOOL
SPECS(4) *	XLEFT	*SPECS(12)* *	CCTAPE
SPECS(5) *	YTOP	*SPECS(13)* *	POINTS
SPECS(6) *	YBOT	*SPECS(14)* *	XSKIPS
SPECS(7) *	XLNGTH	*SPECS(15)* *	YSKIPS
SPECS(8) *	YLNGTH	⋮	
		SPECS(30) *	STAPE

Data points are defined as follows: The first point is defined by the coordinate pair [X(1), Y(1)]; the next sequential point is [X(1+XSKIPS), Y(1+YSKIPS)], which is followed by [X(1+2XSKIPS), Y(1+2YSKIPS)], etc.

C. Routines Used: NONLIN, PFLILI, FAIR, SLLILI, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30), X(10), Y(10)
      DIMENSION BUFX(500), BUFY (500)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT, XLEFT
      SPECS(3) = 10.0
      SPECS(4) =  1.0
C     SET YTOP, YBOT
      SPECS(5) = 10.0
      SPECS(6) =  1.0
C     SET XLNGTH, YLNGTH
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY NUMBER OF POINTS
      SPECS(13) = 10.
C     SET XSKIPS, YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C     SPECIFY SCRATCH FILE NUMBER
      SPECS(30) = 97.
C     SETUP SAMPLE X,Y, ARRAYS
      DO 10  I = 1,10
      X(I) = I
      Y(I) = I
 10   CONTINUE
C     CALL PARABOLIC FAIRING ROUTINE
C     LINEAR X, LOG Y
      CALL PFLILG (X,Y,BUFX,BUFY,SPECS)
          :
          :
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine PFLGLI

Purpose

To construct via a parabolic fairing technique a trend curve through a set of points in a rectangular coordinate system having a logarithmic function scale for the X-axis and a linear function scale for the Y-axis.

Usage

A. FORTRAN call statement:

CALL PFLGLI (*X*,*Y*,*BUFX*,*BUFY*,*SPECS*)

B. Description of call list arguments:

X An array containing the X ordinates of data points to be processed. X ordinate range must contain floating-point numbers (single precision) which are all positive, or all negative, and non-zero.

Y An array containing the Y ordinates of data points to be processed. Values are floating-point.

BUFX An array of 500_{10} consecutive words provided as a work region for the routine to use to process the data.

BUFY An array of 500_{10} consecutive words provided as a work region for the routine to use to process the data.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

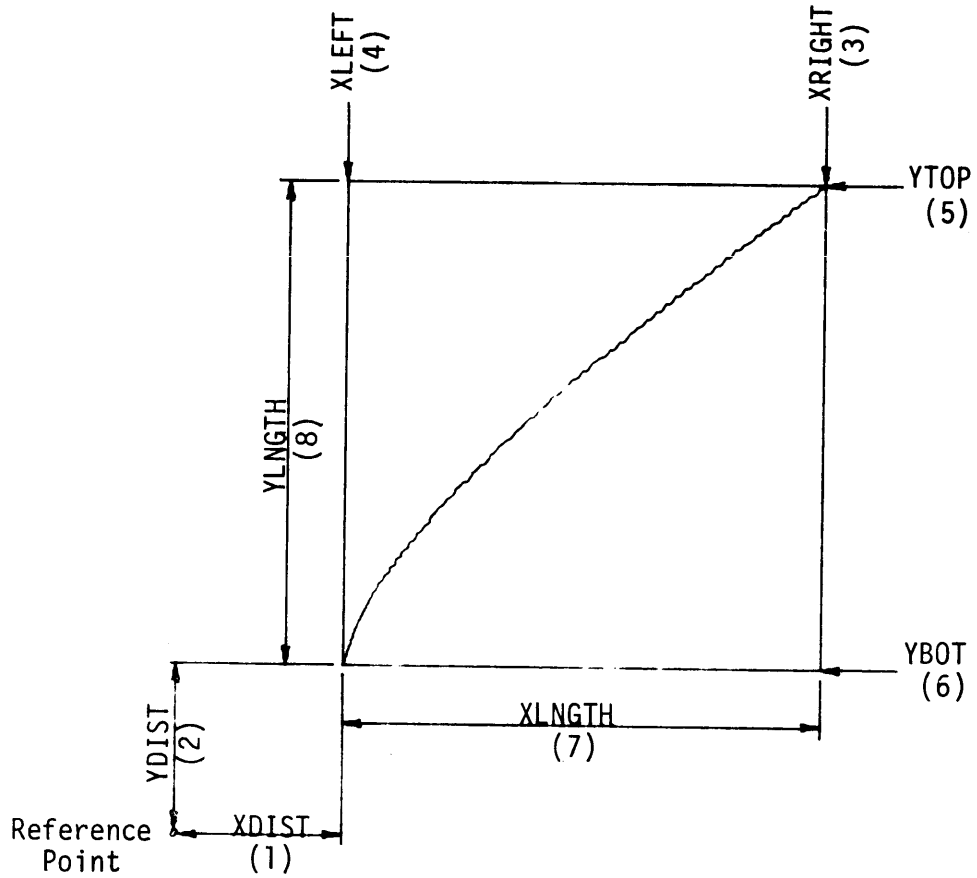
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(9)*		
SPECS(2)	*	YDIST	*SPECS(10)*		
SPECS(3)	*	XRIGHT	*SPECS(11)*	*	TOOL
SPECS(4)	*	XLEFT	*SPECS(12)*	*	CCTAPE
SPECS(5)	*	YTOP	*SPECS(13)*	*	POINTS
SPECS(6)	*	YBOT	*SPECS(14)*	*	XSKIPS
SPECS(7)	*	XLNGTH	*SPECS(15)*	*	YSKIPS
SPECS(8)	*	YLNGTH	⋮		
			SPECS(30)	*	STAPE

Data points are defined as follows: The first point is defined by the coordinate pair [X(1), Y(1)]; the next sequential point is [X(1+XSKIPS), Y(1+YSKIPS)], which is followed by [X(1+2XSKIPS), Y(1+2YSKIPS)], etc.

C. Routines Used: NONLIN, PFLILI, FAIR, SLLILI, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30), X(10), Y(10)
      DIMENSION BUFX(500), BUFY(500)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = .877
      SPECS(2) = .877
C     SET XRIGHT, XLEFT
      SPECS(3) = 5**5
      SPECS(4) = .5**.5
C     SET YTOP, YBOT
      SPECS(5) = 10.0
      SPECS(6) =  1.0
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = .25
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 97.
C     SPECIFY NUMBER OF POIkTS
      SPECS(13) = 10.
C     SET XSKIPS,YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C     SPECIFY SCRATCH FILE NUMBER
      SPECS(30) = 97.
C     SETUP SAMPLE X,Y ARRAYS
      DO 10  I=1,10
      RI = FLOAT(i)/2.
      X(I) = RI**RI
      Y(I) = I
 10   CONTINUE
C     CALL PARABOLIC FAIRING ROUTINE
C     LOG X, LINEAR Y
      CALL PFLGLI (X,Y,SPECS)
                .
                .
                .
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine PFLGLG

Purpose

To construct via a parabolic fairing technique a trend curve through a set of points in a rectangular coordinate system having logarithmic function scales for both axes.

Usage

A. FORTRAN call statement

CALL PFLGLG (*X*,*Y*,*BUFX*,*BUFY*,*SPECS*)

B. Description of call list arguments

X An array containing the X ordinates of data points to be processed. X ordinate range must contain floating-point numbers (single precision) which are all positive, or all negative, and non-zero.

Y An array containing the Y ordinates of data points to be processed. Y ordinate range must contain floating-point numbers (single precision) which are all positive, or all negative, and non-zero.

BUFX An array of 500_{10} consecutive words provided as a work region for the routine to use to process the data.

BUFY An array of 500_{10} consecutive words provided as a work region for the routine to use to process the data.

SPECS An array containing plot construction parameter. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

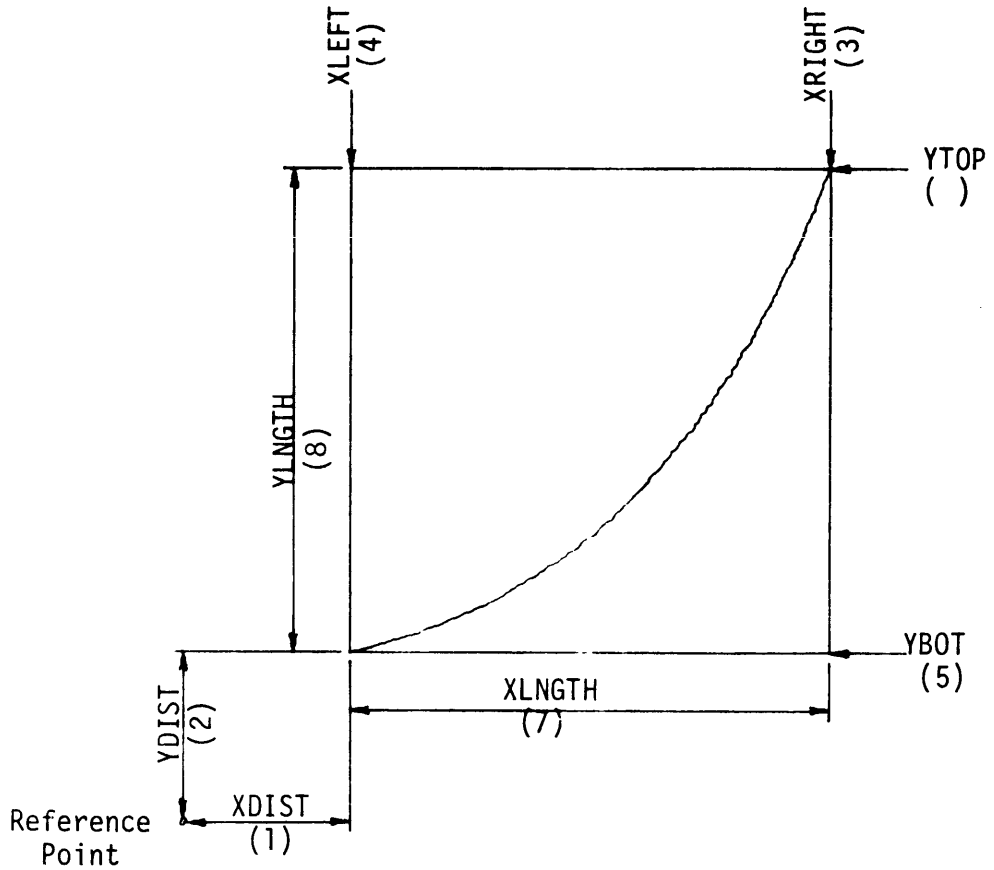
Parameter	Mnemonic	Parameter	Mnemonic
SPECS(1) *	XDIST	*SPECS(9)*	
SPECS(2) *	YDIST	*SPECS(10)*	
SPECS(3) *	XRIGHT	*SPECS(11)* *	TOOL
SPECS(4) *	XLEFT	*SPECS(12)* *	CCTAPE
SPECS(5) *	YTOP	*SPECS(13)* *	POINTS
SPECS(6) *	YBOT	*SPECS(14)* *	XSKIPS
SPECS(7) *	XLNGTH	*SPECS(15)* *	YSKIPS
SPECS(8) *	YLNGTH	:	
		SPECS(30) *	STAPE

Data points are defined as follows: The first point is defined by the coordinate pair [X(1), Y(1)]; the next sequential point is [X(1+XSKIPS), Y(1+YSKIPS)], which is followed by [X(1+2XSKIPS), Y(1+2YSKIPS)], etc.

C. Routines Used: NONLIN, PFLILI, FAIR, SLLILI, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30), X(10), Y(10)
      DIMENSION BUFX (500), BUFY(500)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT, XLEFT
      SPECS(3) = 100.
      SPECS(4) =   1.
C     SET YTOP,YBOT
      SPECS(5) = 2.**10
      SPECS(6) = 2.
C     SET XLNGTH,YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 25.
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY NUvBER OF POINTS
      SPECS(13) = 10.
C     SPECIFY XSKIPS, YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C     SPECIFY SCRATCH FILE NUMBER
      SPECS(30) = 97.
C     SETUP SAMPLE X,Y ARRAYS
      DO 10  I=1,10
      X(I) = I**2.
      Y(I) = 2.**I
 10   CONTINUE
C     CALL PARABOLIC FAIRING ROUTINE
C     LOG X, LOG Y
      CALL PFLGLG(X,Y,BUFX,BUFY,SPECS)
                  :
                  :
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine PFPOLR

Purpose

To construct a trend curve onto a rectangular coordinate system with data given in polar coordinates.†

Usage

A. FORTRAN call statement

CALL PFPOLR (*R*,*THETA*,*BUFX*,*BUFY*,*SPECS*)

B. Description of call list arguments

R An array containing the vector lengths of data points to be processed. Values are floating-point.

THETA An array containing the angles, given in radians of data points to be processed. Values are floating-point.

BUFX An array of 500_{10} consecutive words provided as a work region for the routine to use to process the data.

BUFY An array of 500_{10} consecutive words provided as a work region for the routine to use to process the data.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter	Mnemonic
SPECS(1)	*	XDIST	*SPECS(9)*	
SPECS(2)	*	YDIST	*SPECS(10)*	
SPECS(3)	*	XRIGHT	*SPECS(11)*	TOOL
SPECS(4)	*	XLEFT	*SPECS(12)*	CCTAPE
SPECS(5)	*	YTOP	*SPECS(13)*	POINTS
SPECS(6)	*	YBOT	*SPECS(14)*	XSKIPS
SPECS(7)	*	XLNGTH	*SPECS(15)*	YSKIPS
SPECS(8)	*	YLNGTH	and	
			SPECS(30)	STAPE

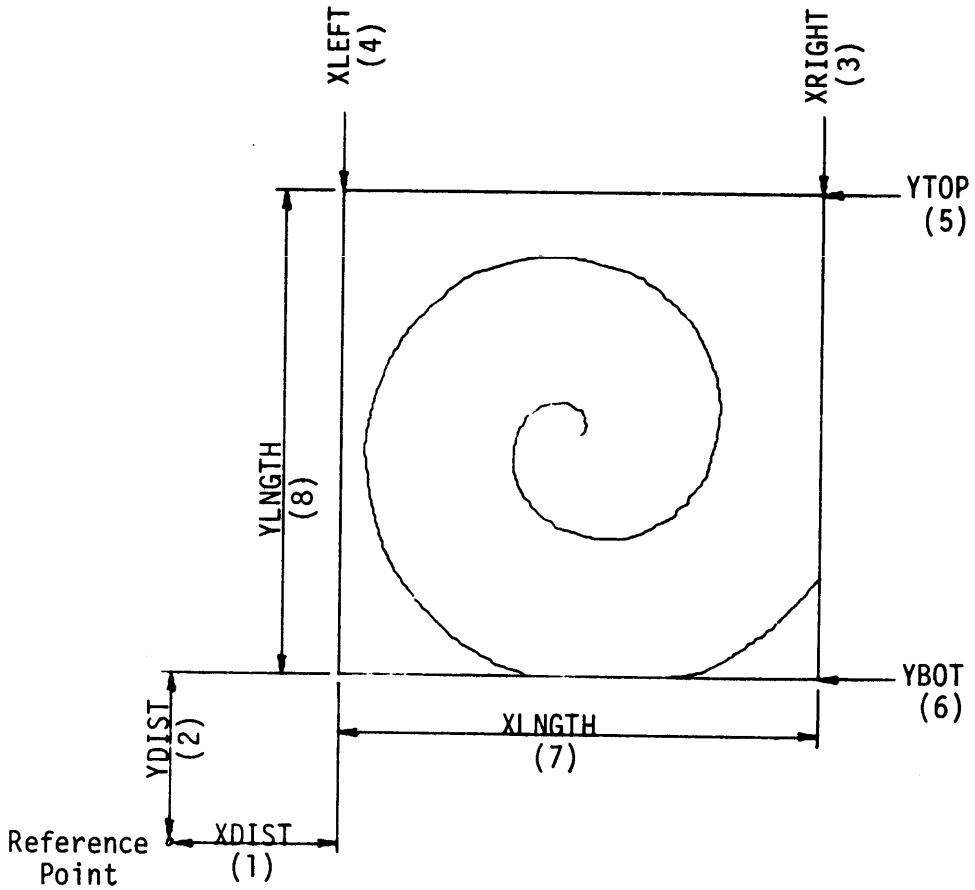
Data points are defined as follows: The first point is defined by the coordinate pair [R(1),THETA(1)]; the next sequential point is [R(1+XSKIPS),THETA(1+YSKIPS)], which is followed by [R(1+2XSKIPS), THETA(1+2YSKIPS)], etc.

C. Routines Used: POLAR, DATAPE, PFLILI, FAIR, SLLILI, SIGNON,WRGHT.

† The given polar coordinates are not changed.

D. Example

```
      DIMENSION SPECS(30),R(25),THETA(25)
      DIMENSION BUFX(500),BUFY(500)
C     SET XDIST,YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT,XLEFT
      SPECS(3) =  100.0
      SPECS(4) = -100.0
C     SET YTOP,YBOT
      SPECS(5) =  100.0
      SPECS(6) = -100.0
C     SET XLNGTH,YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY NUMBER OF POINTS
      SPECS(13) = 25.
C     SET XSKIPS,YSKIPS
      SPECS(14) = 1.
      SPECS(15) = 1.
C     SPECIFY SCRATCH FILE NUMBER
      SPECS(30) = 97.
C     SETUP SAMPLE R,THETA ARRAYS
      DO 10  I=1,25
      RI = I
      R(I) = 5*(I-1)
      THETA(I) = RI/2.
 10   CONTINUE
C     CALL POLAR PARABOLIC FAIRING
C     ROUTINE
      CALL PFPOLR(R,THETA,BUFX,
     BUFY,SPECS)
          :
          :
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Routines to Construct Numerical Values Along an Axis' Scale

Routine Names

NODLIB, NODLIT, NODLIL, NODLIR
NOSLIB, NOSLIT, NOSLIL, NOSLIR
NOLGB, NOLGT, NOLGL, NOLGR

Purpose

There are three sets of routines to construct numerical values along an axis' scale. Depending on the suffix letter of the routine (B,T,L,R) either the X or Y axis may be labeled. The suffix letters B and T denote X-axis annotation of scale values below or above the defined plotting area. For X-axis annotation the values are parallel to the X-axis. The suffix letters L and R denote Y-axis annotation of scale values to the left or to the right of the defined plotting area. The scale values may be either parallel or perpendicular to the Y-axis; *SPECS(20)*, ROTATE, specifies which layout is desired.

NODLI_ This set of routines constructs numerical scale values to a specified number of decimal places at regular intervals along a linear axis such that the values are oriented next to their respective scale lines (centered, if possible about their respective scale lines). If zero decimal places are specified, the decimal point is not drawn.

NOSLI_ This set of routines constructs numerical scale values to the number of significant figures specified at regular intervals along a linear axis such that the values are oriented next to their respective scale lines (centered, if possible).

NOLG_ This set of routines construct numerical scale values along a logarithmic (base 10) axis such that the values are oriented next to their respective scale lines (centered, if possible). The values will be in a form of scientific notation (i.e., 10^N).

Method

In general, all the above routines determine the largest subset of the given set of subdivisions such that any two scale values, when constructed, are separated by at least the width of a character. The distance between the closest point of the numerical value and the defined plotting area is the sum of the appropriate ZONE value and (0.25 * Character Size)†. After all construction has occurred, the appropriate ZONE value is incremented by the sum of "Greatest Distance used in constructing the values" plus (0.25 * Character Size)†. Specifically, the routines act as follows:

† Character Size is defined to be FONTH for the bottom and top (B,T) routines; FONTB for the left and right (L,R) routines.

NODLI_ Explained above.

NOSLI_ The maximum absolute value is rounded and expressed to the least significant digit. If necessary, the routine scales this value to establish the position of the decimal point. For example, if 123.43 is the maximum absolute vulue and four significant digits are desired, the routine constructs "123.4" and no scaling is required.

NOTE: If 50.27 is a scale value within the range, the routine constructs "50.3". This value is expressed to one decimal place to comply with the format established by the maximum absolute value.

If scaling is necessary, the routine constructs below the defined plotting area the phrase "X10^N" where N is the appropriate exponent, an integer.

NOLG_ The routine labels a logarithmic (base 10) scale such that any two scale values, when constructed, do not overlap. Although some scale values may be omitted to prevent overlap, but less than two scale values are constructed. The routine labels the logarithmic scale, subject to the above conditions, according to the following rules, given in their order of application.

A. If possible, each cycle value (that scale value which equals $\pm$ 10^N) is labeled "10^N" (or "-10^N") where the exponent N is an integer. Only the bottom-most cycle value of the scale range is constructed when the scale range contains two or more cycle values and no two of them can be labeled without overlap.

B. When all the cycle values of the scale range can be constructed or when no cycle value is contained within the range, labeling of subcycle values is then attempted. One of the following sets of values given in order of priority:

1. 2×10^N, 3×10^N, 4×10^N, 5×10^N, 6×10^N, 7×10^N, 8×10^N, 9×10^N
2. 2×10^N, 4×10^N, 6×10^N, 8×10^N
3. 2×10^N, 5×10^N

where N is an integer, will be used. If any of the above sets can be constructed without overlap, the form of the label will be that of the integers only with a negative sign if necessary. Otherwise no subcycle values are labeled.

C. If possible or if it is necessary, one or both of the minimum and maximum scale values are labeled. Each value is scaled to a X10^N, where $1 < |a| < 10$; only the a, rounded to two decimal places and including a negative sign if necessary, is constructed.

D. If a cycle is needed to define the scale range properly, the routine will append "X10^{N}" to either the leftmost or rightmost value constructed under rules B or C. (This rule covers exceptional cases.)

Restrictions

When using the NOLG_ routines the following restriction applies: Both the maximum and minimum values of the appropriate data range (X or Y) must agree in sign, be non-zero, and be unequal.

Usage

See the following individual subroutine writeups for:

A. FORTRAN call statement

B. Description of call list arguments

C. Routines Used

D. Example

Subroutine NODLIB

Purpose

To construct, to the number of decimal places specified, numerical scale values at regular intervals along a linear X-axis such that the values are oriented below and parallel to the bottom of the defined plotting area. If possible, the scale values are centered about their respective scale lines.

Usage

A. FORTRAN Call Statement

```
CALL NODLIB (SPECS)
```

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(15)*		
SPECS(2)	*	YDIST	*SPECS(16)*		
SPECS(3)	*	XRIGHT	*SPECS(17)*	*	FONTB
SPECS(4)	*	XLEFT	*SPECS(18)*	*	FONTH
SPECS(5)			*SPECS(19)*	*	SPACER
SPECS(6)			*SPECS(20)*		
SPECS(7)	*	XLNGTH	*SPECS(21)*	*	FONTNO
SPECS(8)			*SPECS(22)*		
SPECS(9)	*	XDIV	*SPECS(23)*		
SPECS(10)			*SPECS(24)*	*	ZONEB
SPECS(11)	*	TOOL	*SPECS(25)*		
SPECS(12)	*	CCTAPE	*SPECS(26)*		
SPECS(13)			*SPECS(27)*		
SPECS(14)			*SPECS(28)*	*	DECPLS

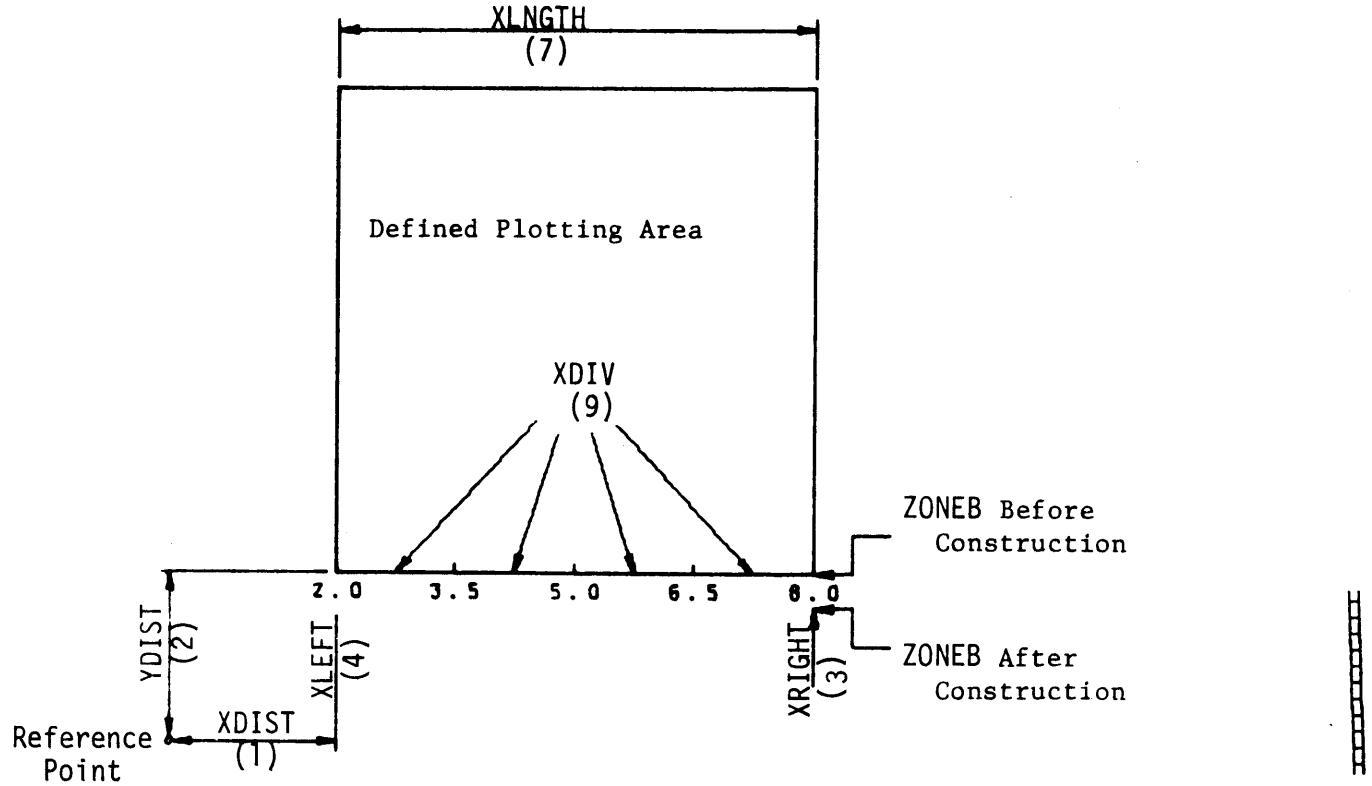
NOTE: After construction, *SPECS(24)* is incremented so that the next construction will not overlap with this construction.

C. Routines Used: LINLAB, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST,YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT,XLEFT
      SPECS(3) = 8.0
      SPECS(4) = 2.0
C     SET XLNGTH (IN INCHES)
      SPECS(7) = 2.5
C     SET XDIV
      SPECS(9) = 4.
C     SPECIFY TOOL (NORMALLY SET TO ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB, FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SET ZONEB (IN INCHES)
      SPECS(24) = 0.0
C     SPECIFY NUMBER OF DECIMAL PLACES
      SPECS(28) = 1.
      CALL NODLIB (SPECS)
          :
          :
```

NOTE: ZONEB is incremented after the construction so that the next construction below the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine NODLIT

Purpose

To construct, to the number of decimal places specified, numerical scale values at regular intervals along a linear X-axis such that the values are oriented above and parallel to the top of the defined plotting area. If possible, the scale values are centered about their respective scale lines.

Usage

A. FORTRAN call statement:

```
CALL NODLIT (SPECS)
```

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter	Mnemonic	Parameter	Mnemonic
SPECS(1) *	XDIST	*SPECS(15)*	
SPECS(2) *	YDIST	*SPECS(16)*	
SPECS(3) *	XRIGHT	*SPECS(17)* *	FONTB
SPECS(4) *	XLEFT	*SPECS(18)* *	FONTH
SPECS(5)		*SPECS(19)* *	SPACER
SPECS(6)		*SPECS(20)*	
SPECS(7) *	XLNGTH	*SPECS(21)* *	FONTNO
SPECS(8) *	YLNGTH	*SPECS(22)*	
SPECS(9) *	XDIV	*SPECS(23)*	
SPECS(10)		*SPECS(24)*	
*SPECS(11)**	TOOL	*SPECS(25)* *	ZONET
*SPECS(12)**	CCTAPE	*SPECS(26)*	
SPECS(13)		*SPECS(27)*	
SPECS(14)		*SPECS(28)* *	DECPLS

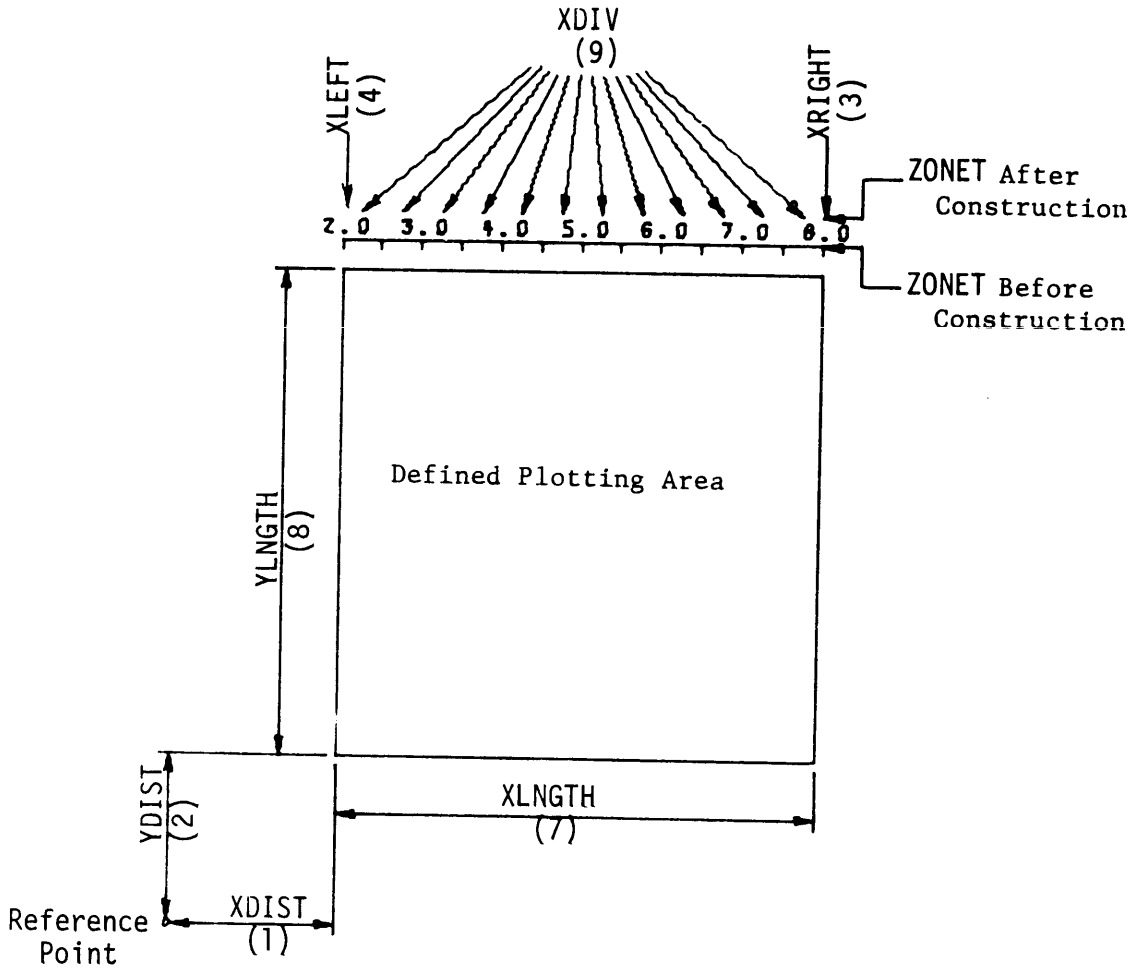
NOTE: After construction, *SPECS(25)* is incremented so that the next construction will not overlap with this construction.

C. Routines Used: LINLAB, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST,YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT,XLEFT
      SPECS(3) = 8.0
      SPECS(4) = 2.0
C     SET XLNGTH,YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SET XDIV
      SPECS(9) = 12.
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB,FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SET ZONET (IN INCHES)
      SPECS(25) = 0.15
C     SPECIFY NUMBER OF DECIMAL PLACES
      SPECS(28) = 1.
      CALL NODLIB (SPECS)
          :
          :
```

NOTE: ZONET is incremented after the construction so that the next construction above the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine NODLIL

Purpose

To construct, to the number of decimal points specified, numerical scale values at regular intervals along a linear Y-axis such that the values are oriented to the left of the defined plotting area. If possible, the scale values are centered about their respective scale lines.

Usage

A. FORTRAN call statement

```
CALL NODLIL (SPECS)
```

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	SPECS(15)		
SPECS(2)	*	YDIST	SPECS(16)		
SPECS(3)			SPECS(17)	*	FONTB
SPECS(4)			SPECS(18)	*	FONTH
SPECS(5)	*	YTOP	SPECS(19)	*	SPACER
SPECS(6)	*	YBOT	SPECS(20)	*	ROTATE
SPECS(7)			SPECS(21)	*	FONTNO
SPECS(8)	*	YLNGTH	SPECS(22)		
SPECS(9)			SPECS(23)		
SPECS(10)	*	YDIV	SPECS(24)		
SPECS(11)	*	TOOL	SPECS(25)		
SPECS(12)	*	CCTAPE	SPECS(26)	*	ZONEL
SPECS(13)			SPECS(27)		
SPECS(14)			SPECS(28)	*	DECPLS

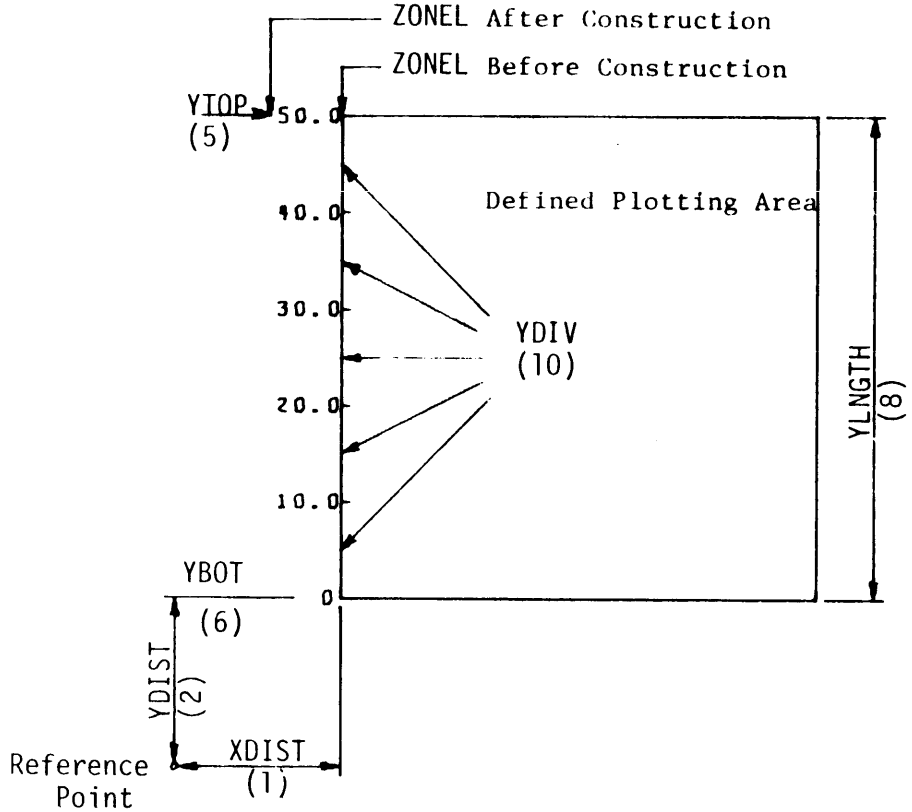
NOTE: After construction, *SPECS(26)* is incremented so that the next construction will not overlap with this construction.

C. Routines Used: LINLAB, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST,YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET YTOP, YBOT
      SPECS(5) = 50.0
      SPECS(6) =  0.0
C     SET YLNGTH (IN INCHES)
      SPECS(8) = 2.5
C     SET YDIV
      SPECS(10) = 5.
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB, FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY LABEL ORIENTATION
      SPECS(20) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SET ZONEL (IN INCHES)
      SPECS(26) = 0.0
C     SPECIFY NUMBER OF DECIMAL PLACES
      SPECS(28) = 1.
      CALL NODLIL (SPECS)
          :
          :
```

NOTE: ZONEL is incremented after the construction so that the next construction to the left of the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine NODLIR

Purpose

To construct, to the number of decimal places specified, numerical scale values at regular intervals along a linear Y-axis such that the values are oriented to the right of the defined plotting area. If possible, the scale values are centered about their respective scale lines.

Usage

A. FORTRAN call statement

CALL NODLIR (*SPECS*)

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(15)*		
SPECS(2)	*	YDIST	*SPECS(16)*		
SPECS(3)			*SPECS(17)*	*	FONTB
SPECS(4)			*SPECS(18)*	*	FONTH
SPECS(5)	*	YTOP	*SPECS(19)*	*	SPACER
SPECS(6)	*	YBOT	*SPECS(20)*	*	ROTATE
SPECS(7)	*	XLNGTH	*SPECS(21)*	*	FONTNO
SPECS(8)	*	YLNGTH	*SPECS(22)*		
SPECS(9)			*SPECS(23)*		
SPECS(10)	*	YDIV	*SPECS(24)*		
SPECS(11)	*	TOOL	*SPECS(25)*		
SPECS(12)	*	CCTAPE	*SPECS(26)*		
SPECS(13)			*SPECS(27)*	*	ZONER
SPECS(14)			*SPECS(28)*	*	DECPLS

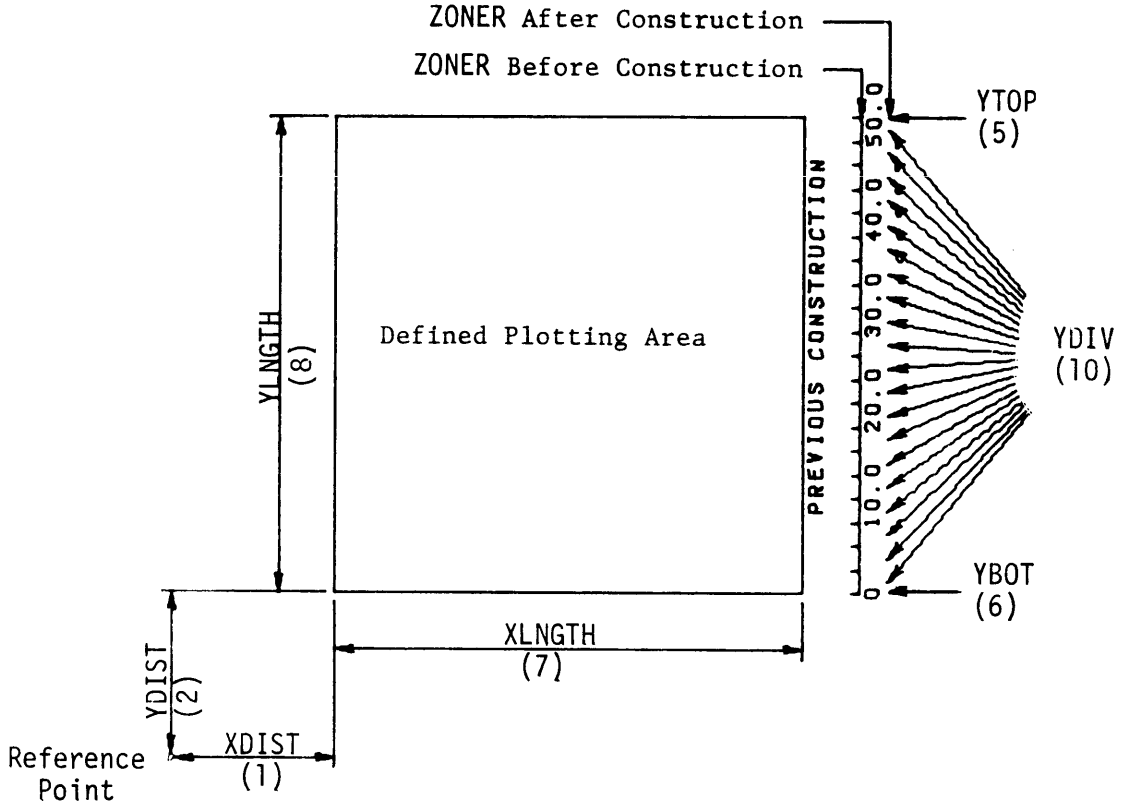
NOTE: After construction, *SPECS(27)* is incremented so that the next construction will not overlap with this construction.

C. Routines Used: LINLAB, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET YTOP, YBOT
      SPECS(5) = 50.0
      SPECS(6) =  0.0
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SET YDIV
      SPECS(10) = 20.
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB,FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY LABEL ORIENTATION
      SPECS(20) = 90.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS (21) = 1.
C     SET ZONER (IN INCHES)
      SPECS(27) = 0.3
C     SPECIFY NUMBER OF DECIMAL PLACES
      SPECS(28) = 1.
      CALL NODLIR (SPECS)
          .
          .
          .
```

NOTE: ZONER is incremented after the construction so that the next construction to the right of the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine NOSLIB

Purpose

To construct, to the number of significant figures specified, numerical scale values at regular intervals along a linear X-axis such that the values are oriented below and parallel to the bottom of the defined plotting area. If possible, the scale values are centered about their respective scale lines.

Usage

A. FORTRAN call statement

CALL NOSLIB (*SPECS*)

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(16)*		
SPECS(2)	*	YDIST	*SPECS(17)*	*	FONTB
SPECS(3)	*	XRIGHT	*SPECS(18)*	*	FONTH
SPECS(4)	*	XLEFT	*SPECS(19)*	*	SPACER
SPECS(5)			*SPECS(20)*		FONTNO
SPECS(6)			*SPECS(21)*	*	
SPECS(7)	*	XLNGTH	*SPECS(22)*		
SPECS(8)			*SPECS(23)*		
SPECS(9)	*	XDIV	*SPECS(24)*	*	ZONEB
SPECS(10)			*SPECS(25)*		
SPECS(11)	*	TOOL	*SPECS(26)*		
SPECS(12)	*	CCTAPE	*SPECS(27)*		
SPECS(13)			*SPECS(28)*		
SPECS(14)			*SPECS(29)*	*	SIGFIG
SPECS(15)					

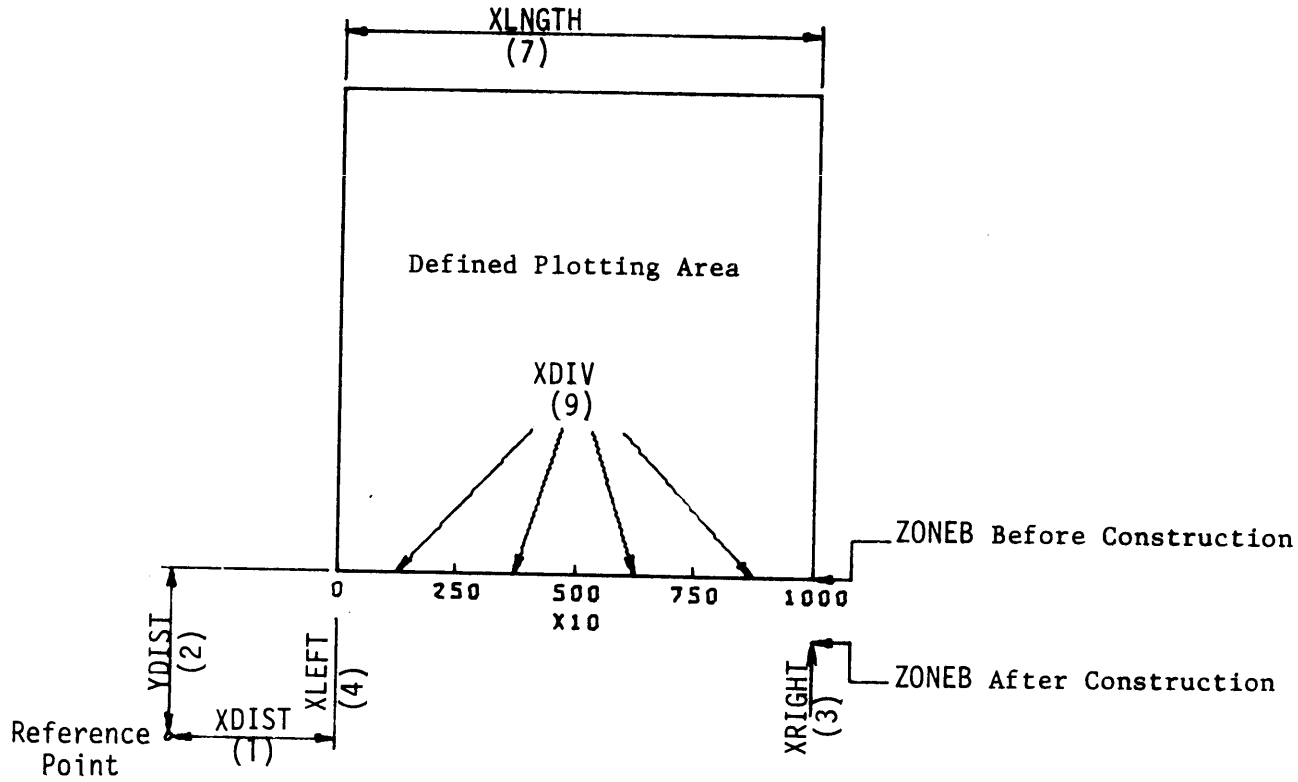
NOTE: After construction, *SPECS(24)* is incremented so that the next construction will not overlap with this construction.

C. Routines Used: SIGDEC, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT, XLEFT
      SPECS(3) = 10000.0
      SPECS(4) =    0.0
C     SET XLNGTH (IN INCHES)
      SPECS(7) = 2.5
C     SET XDIV
      SPECS(9) = 4.
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB,FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SET ZONEB (IN INCHES)
      SPECS(24) = 0.0
C     SPECIFY NUMBER OF SIGNIFICANT
C     FIGURES
      SPECS(29) = 4.
      CALL NOSLIB (SPECS)
          :
          :
```

NOTE: ZONEB is incremented after the construction so that the next construction below the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine NOSLIT

Purpose

To construct, to the number of significant figures specified, numerical scale values at regular intervals along a linear X-axis such that the values are oriented above and parallel to the top of the defined plotting area. If possible, the scale values are centered about their respective scale lines.

Usage

A. FORTRAN call statement

```
CALL NOSLIT (SPECS)
```

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(16)*		
SPECS(2)	*	YDIST	*SPECS(17)*	*	FONTB
SPECS(3)	*	XRIGHT	*SPECS(18)*	*	FONTH
SPECS(4)	*	XLEFT	*SPECS(19)*	*	SPACER
SPECS(5)			*SPECS(20)*		
SPECS(6)			*SPECS(21)*	*	FONTNO
SPECS(7)	*	XLNGTH	*SPECS(22)*		
SPECS(8)	*	YLNGTH	*SPECS(23)*		
SPECS(9)	*	XDIV	*SPECS(24)*		
SPECS(10)			*SPECS(25)*	*	ZONET
SPECS(11)	*	TOOL	*SPECS(26)*		
SPECS(12)	*	CCTAPE	*SPECS(27)*		
SPECS(13)			*SPECS(28)*		
SPECS(14)			*SPECS(29)*	*	SIGFIG
SPECS(15)					

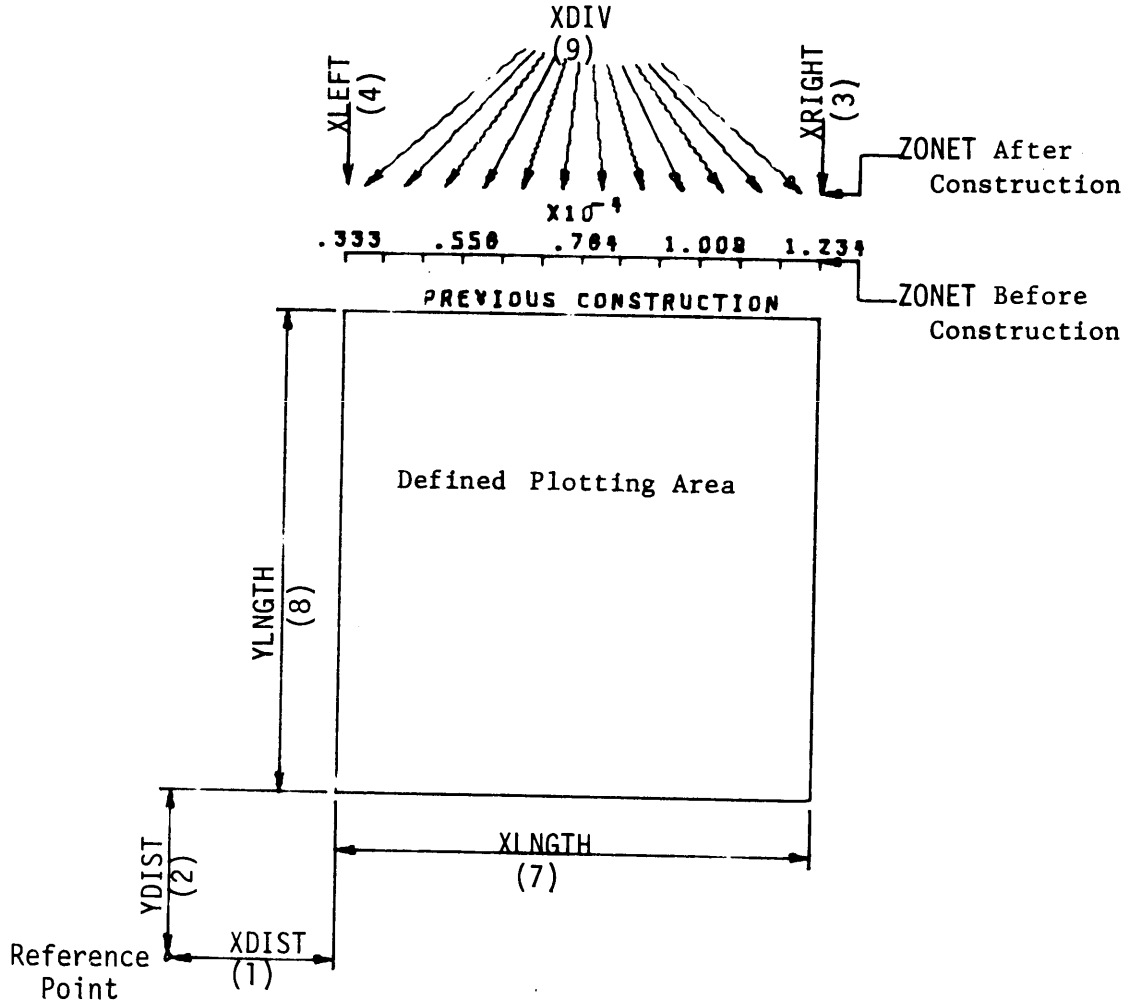
NOTE: After construction, *SPECS(25)* is incremented so that the next construction will not overlap with this construction.

C. Routines Used: SIGDEC, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT, XLEFT
      SPECS(3) = 0.0001234
      SPECS(4) = 0.0000333
C     SET XLNGTH (IN INCHES)
      SPECS(7) = 2.5
C     SET XDIV
      SPECS(9) = 12.
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB,FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SET ZONET (IN INCHES)
      SPECS(25) = 0.3
C     SPECIFY NUMBER OF SIGNIFICANT
C     FIGURES
      SPECS(29) = 4.
      CALL NOSLIT (SPECS)
          .
          .
          .
```

NOTE: ZONET is incremented after the construction so that the next construction above the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine NOSLIL

Purpose

To construct numerical scale values, to the number of significant figures specified, at regular intervals along a linear Y-axis such that the values are oriented to the left of the defined plotting area. If possible, scale values are centered about their respective scale lines.

Usage

A. FORTRAN call statement

```
CALL NOSLIL (SPECS)
```

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	SPECS(16)		
SPECS(2)	*	YDIST	SPECS(17)	*	FONTB
SPECS(3)			SPECS(18)	*	FONTH
SPECS(4)			SPECS(19)	*	SPACER
SPECS(5)	*	YTOP	SPECS(20)	*	ROTATE
SPECS(6)	*	YBOT	SPECS(21)	*	FONTNO
SPECS(7)			SPECS(22)		
SPECS(8)	*	YLNGTH	SPECS(23)		
SPECS(9)			SPECS(24)		
SPECS(10)	*	YDIV	SPECS(25)		
SPECS(11)	*	TOOL	SPECS(26)	*	ZONEL
SPECS(12)	*	CCTAPE	SPECS(27)		
SPECS(13)			SPECS(28)		
SPECS(14)			SPECS(29)	*	SIGFIG
SPECS(15)					

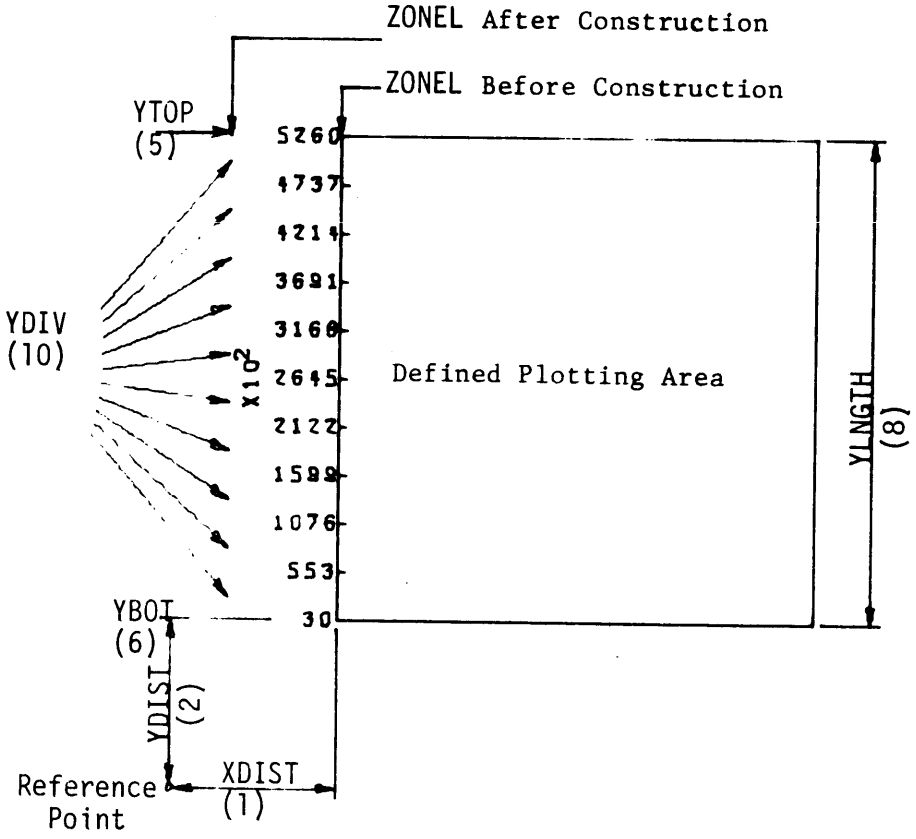
NOTE: After construction, *SPECS*(26) is incremented so that the next construction will not overlap with this construction.

C. Routines Used; SIGDEC, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET YTOP, YBOT
      SPECS(5) = 526000
      SPECS(6) =   3000
C     SET YDIV
      SPECS(10) = 10.
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB, FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY LABEL ORIENTATION
      SPECS(20) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SET ZONEL
      SPECS(26) = 0.0
C     SPECIFY NUMBER OF SIGNIFICANT
C     FIGURES
      SPECS(29) = 4.
      CALL NOSLIL (SPECS)
        :
```

NOTE: ZONEL is incremented after the construction so that the next construction to the left of the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine NOSLIR

Purpose

To construct, to the number of significant figures specified, numerical scale values at regular intervals along a linear Y-axis such that the values are oriented to the right of the defined plotting area. If possible, the scale values are centered about their respective lines.

Usage

A. FORTRAN call statement

```
CALL NOSLIR (SPECS)
```

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The SPECS Array" for definitions and array format. This routine requires the SPECS parameters indicated below by an * to be specified prior to calling it:

Parameter	Mnemonic	Parameter	Mnemonic
SPECS(1) *	XDIST	SPECS(16)	
SPECS(2) *	YDIST	SPECS(17) *	FONTB
SPECS(3)		SPECS(18) *	FONTH
SPECS(4)		SPECS(19) *	SPACER
SPECS(5) *	YTOP	SPECS(20) *	ROTATE
SPECS(6) *	YBOT	SPECS(21) *	FONTNO
SPECS(7) *	XLNGTH	SPECS(22)	
SPECS(8) *	YLNGTH	SPECS(23)	
SPECS(9)		SPECS(24)	
SPECS(10)*	YDIV	SPECS(25)	
SPECS(11)*	TOOL	SPECS(26)	
SPECS(12)*	CCTAPE	SPECS(27) *	ZONER
SPECS(13)		SPECS(28)	
SPECS(14)		SPECS(29) *	SIGFIG
SPECS(15)			

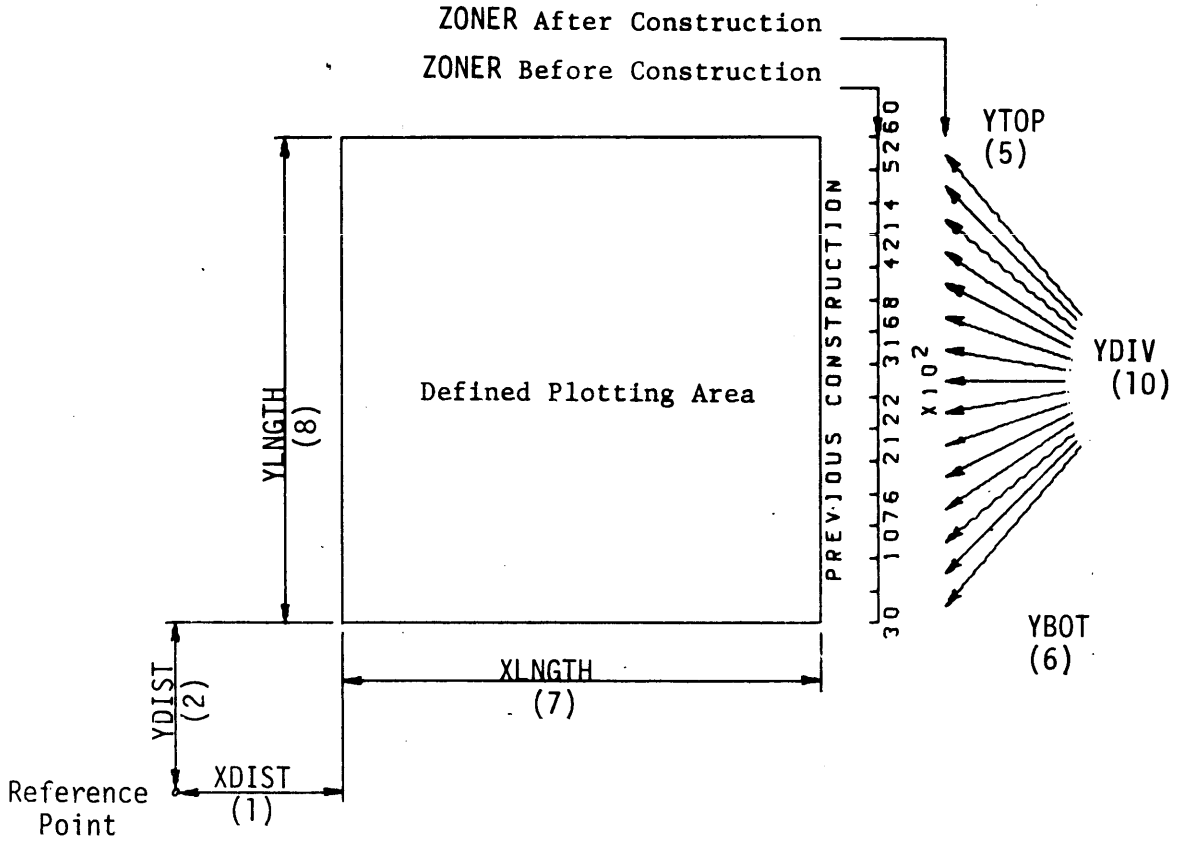
NOTE: After construction, SPECS(27) is incremented so that the next construction will not overlap with this construction.

C. Routines Used: SIGDEC, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET YTOP, YBOT
      SPECS(5) = 526000
      SPECS(6) =   3000
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SET YDIV
      SPECS(10) = 15.
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB, FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY LABEL ORIENTATION
      SPECS(20) = 90.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SET ZONER (IN INCHES)
      SPECS(27) = 0.3
C     SPECIFY NUMBER OF SIGNIFICANT FIGURES
      SPECS(29) = 4.
      CALL NOSLIR (SPECS)
        .
        .
        .
```

NOTE: ZONER is incremented after the construction so that the next construction to the right of the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine NOLGB

Purpose

To construct numerical scale values along a logarithmic (base 10) X-axis such that the values are oriented below and parallel to the bottom of the defined plotting area. The values will be in a form of scientific notation.

Usage

A. FORTRAN call statement

```
CALL NOLGB (SPECS)
```

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(13)*		
SPECS(2)	*	YDIST	*SPECS(14)*		
SPECS(3)	*	XRIGHT	*SPECS(15)*		
SPECS(4)	*	XLEFT	*SPECS(16)*		
SPECS(5)			*SPECS(17)*	*	FONTB
SPECS(6)			*SPECS(18)*	*	FONTH
SPECS(7)	*	XLNGTH	*SPECS(19)*	*	SPACER
SPECS(8)			*SPECS(20)*		
SPECS(9)			*SPECS(21)*	*	FONTNO
SPECS(10)			*SPECS(22)*		
SPECS(11)	*	TOOL	*SPECS(23)*		
SPECS(12)	*	CCTAPE	*SPECS(24)*	*	ZONEB

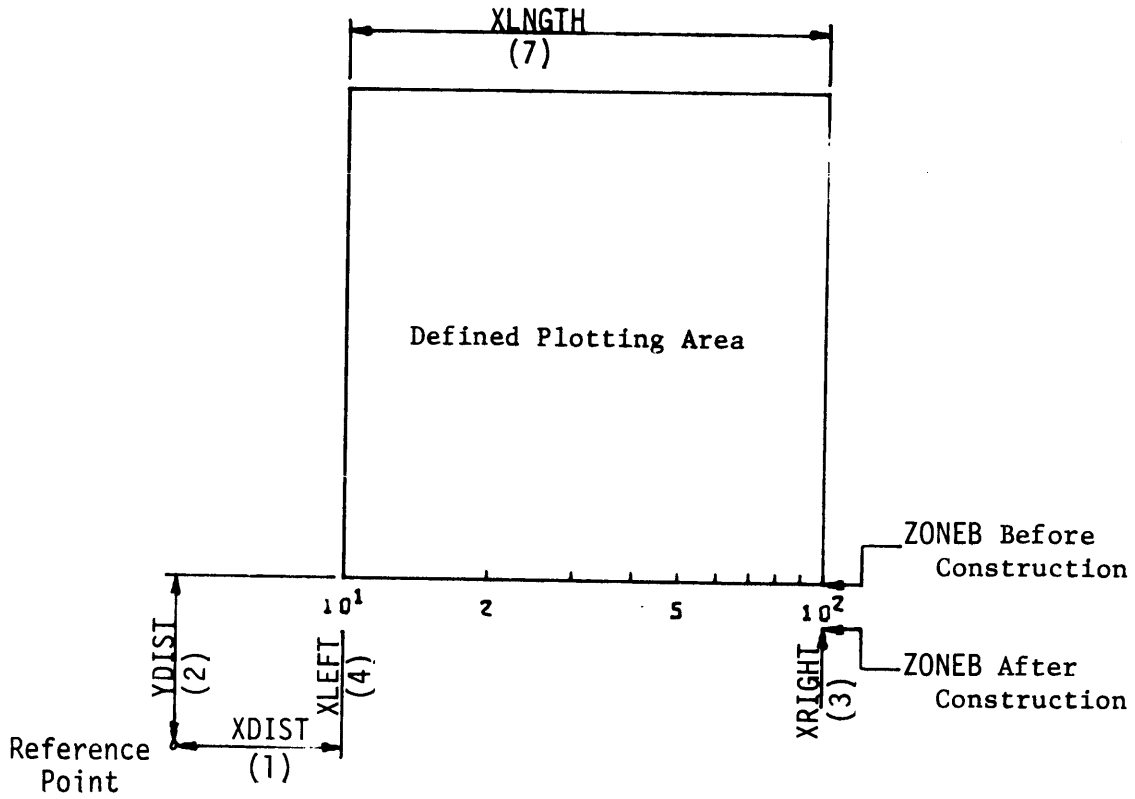
NOTE: After construction, *SPECS(24)* is incremented so that the next construction will not overlap with this construction.

C. Routines Used: LOGLAB, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT, XLEFT
      SPECS(3) = 100.0
      SPECS(4) =  10.0
C     SET XLNGTH
      SPECS(7) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB,FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SPECIFY ZONEB (IN INCHES)
      SPECS(24) = 0.0
      CALL NOLGB (SPECS)
          :
          :
```

NOTE: ZONEB is incremented after the construction so that the next construction below the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine NOLGT

Purpose

To construct numerical scale values along a logarithmic (base 10) X-axis such that the values are oriented above and parallel to the top of the defined plotting area. The values will be in a form of scientific notation.

Usage

A. FORTRAN call statement

```
CALL NOLGT (SPECS)
```

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(14)*		
SPECS(2)	*	YDIST	*SPECS(15)*		
SPECS(3)	*	XRIGHT	*SPECS(16)*		
SPECS(4)	*	XLEFT	*SPECS(17)*	*	FONTB
SPECS(5)			*SPECS(18)*	*	FONTH
SPECS(6)			*SPECS(19)*	*	SPACER
SPECS(7)	*	XLNGTH	*SPECS(20)*		
SPECS(8)	*	YLNGTH	*SPECS(21)*	*	FONTNO
SPECS(9)			*SPECS(22)*		
SPECS(10)			*SPECS(23)*		
SPECS(11)	*	TOOL	*SPECS(24)*		
SPECS(12)	*	CCTAPE	*SPECS(25)*	*	ZONET
SPECS(13)					

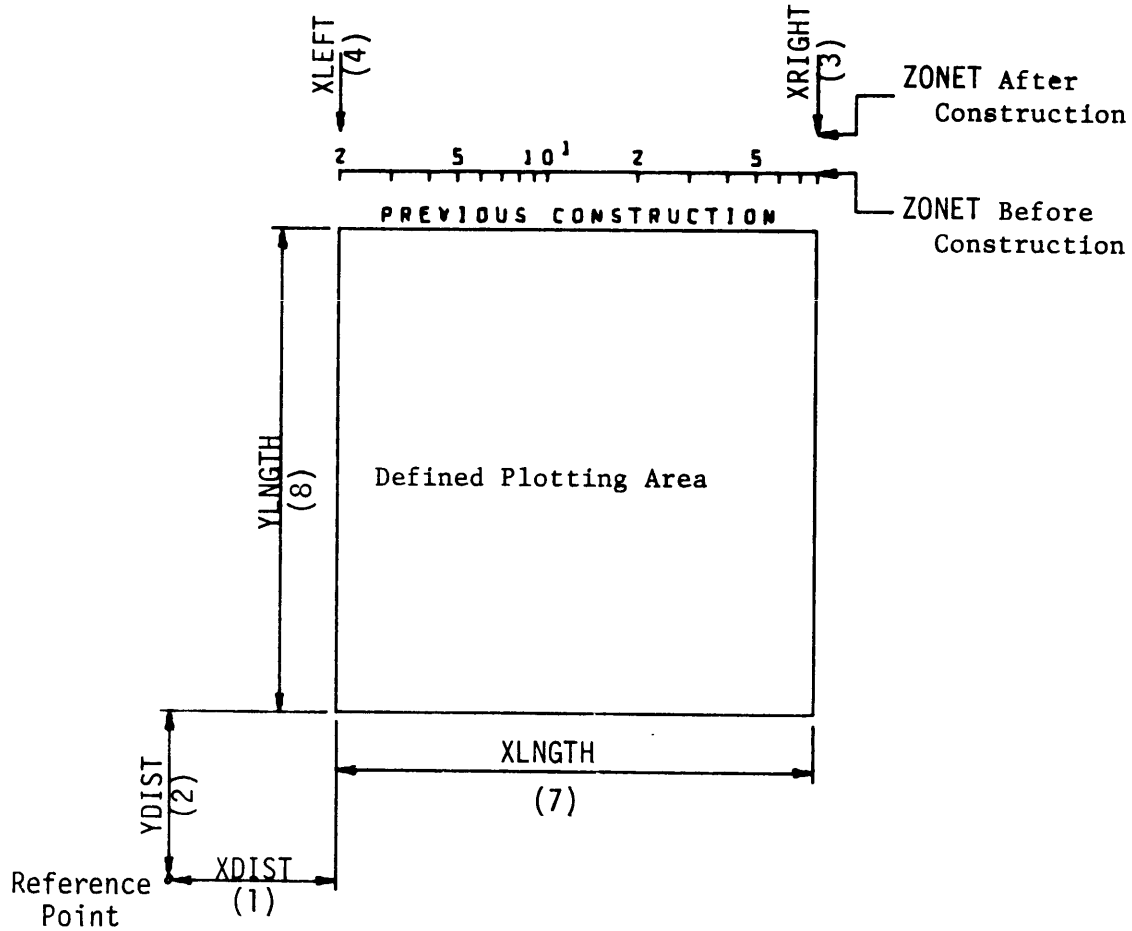
NOTE: After construction, *SPECS(25)* is incremented so that the next construction will not overlap with this construction.

C. Routines Used: LOGLAB, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XRIGHT, XLEFT
      SPECS(3) = 80.0
      SPECS(4) =  2.0
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB,FONTH (IN INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SPECIFY ZONET (IN INCHES)
      SPECS(25) = 0.3
      CALL NOLGT (SPECS)
          :
          :
```

NOTE: ZONET is incremented after the construction so that the next construction above the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine NOLGL

Purpose

To construct numerical scale values along a logarithmic (base 10) Y-axis such that the values are oriented to the left of the defined plotting area. The values will be in a form of scientific notation.

Usage

A. FORTRAN call statement

CALL NOLGL (*SPECS*)

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter	Mnemonic	Parameter	Mnemonic
SPECS(1) *	XDIST	*SPECS(14)*	
SPECS(2) *	YDIST	*SPECS(15)*	
SPECS(3)		*SPECS(16)*	
SPECS(4)		*SPECS(17)* *	FONTB
SPECS(5) *	YTOP	*SPECS(18)* *	FONTH
SPECS(6) *	YBOT	*SPECS(19)* *	SPACER
SPECS(7)		*SPECS(20)* *	ROTATE
SPECS(8) *	YLNGTH	*SPECS(21)* *	FONTNO
SPECS(9)		*SPECS(22)*	
SPECS(10)		*SPECS(23)*	
*SPECS(11)**	TOOL	*SPECS(24)*	
*SPECS(12)**	CCTAPE	*SPECS(25)*	
SPECS(13)		*SPECS(26)* *	ZONEL

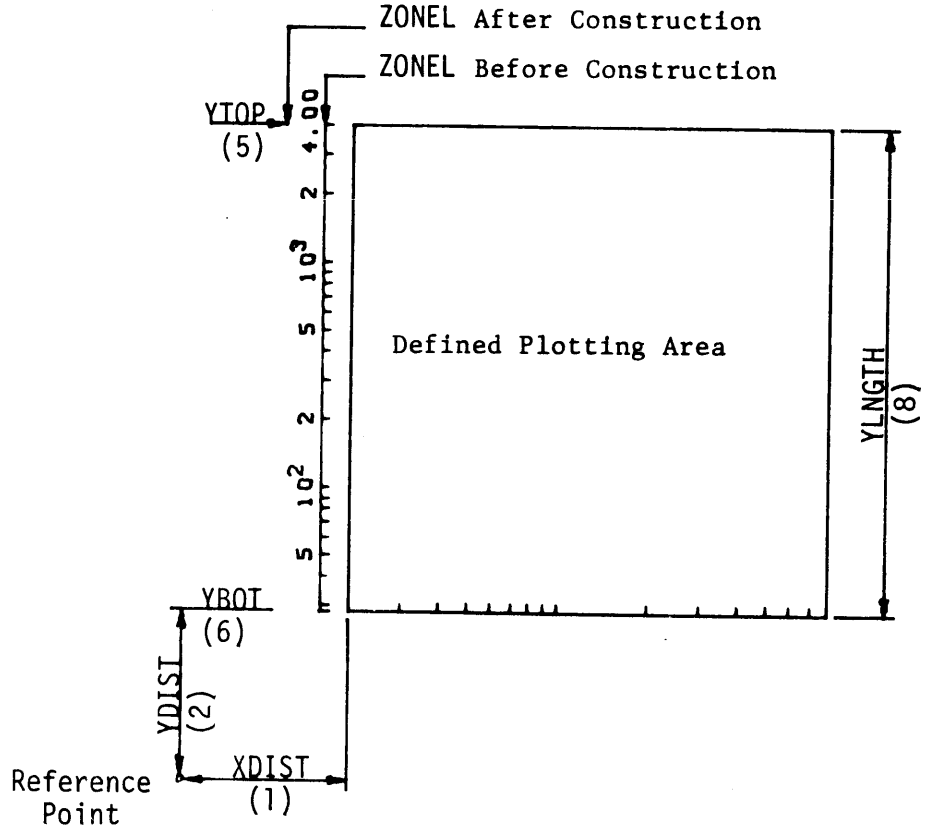
NOTE: After construction, *SPECS(26)* is incremented so that the next construction will not overlap with this construction.

C. Routines Used: LOGLAB, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(3)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET YTOP, YBOT
      SPECS(5) = 4000.0
      SPECS(6) =   28.0
C     SET YLNGTH (IN INCHES)
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB,FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY LABEL ORIENTATION
      SPECS(20) = 90.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
      CALL NOLGL (SPECS)
          :
          :
```

NOTE: ZONEL is incremented after the construction so that the next construction to the left of the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine NOLGR

Purpose

To construct numerical scale values along a logarithmic (base 10) Y-axis such that the values are oriented to the right of the defined plotting area. The values will be in a form of scientific notation.

Usage

A. FORTRAN call statement

```
CALL NOLGR (SPECS)
```

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	SPECS(15)		
SPECS(2)	*	YDIST	SPECS(16)		
SPECS(3)			SPECS(17)	*	FONTB
SPECS(4)			SPECS(18)	*	FONTH
SPECS(5)	*	YTOP	SPECS(19)	*	SPACER
SPECS(6)	*	YBOT	SPECS(20)	*	ROTATE
SPECS(7)	*	XLNGTH	SPECS(21)	*	FONTNO
SPECS(8)	*	YLNGTH	SPECS(22)		
SPECS(9)			SPECS(23)		
SPECS(10)			SPECS(24)		
SPECS(11)	*	TOOL	SPECS(25)		
SPECS(12)	*	CCTAPE	SPECS(26)		
SPECS(13)			SPECS(27)	*	ZONER
SPECS(14)					

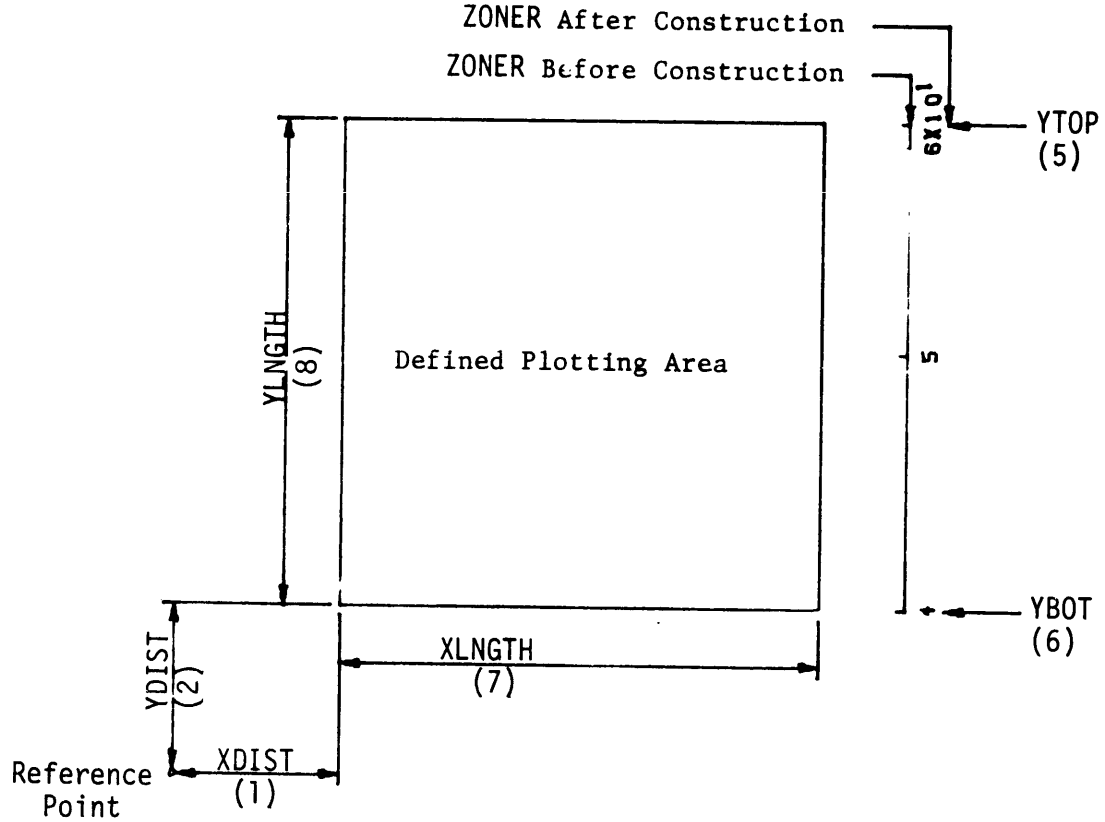
NOTE: After construction, *SPECS*(27) is incremented so that the next construction will not overlap with this construction.

C. Routines Used: LOGLAB, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET YTOP, YBOT
      SPECS(5) = 61.2
      SPECS(6) = 40.0
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB, FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY LABEL ORIENTATION
      SPECS(20) = 90.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SPECIFY ZONER (IN INCHES)
      SPECS(27) = 0.3
      CALL NOLGR (SPECS)
          :
          :
```

NOTE: ZONER is incremented after the construction so that the next construction to the right of the defined plotting area will not overlap.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Routines to Construct Centered Alphanumeric Titles

Routine Names

TITLEB, TITLET, TITLEL, TITLER

Purpose

To construct one or more lines of text to be centered below, above, to the left, or to the right of the defined plotting area depending on the suffix letter (B,T,L,R) of the subroutine called.

Method

For a single line of text:

To construct a single line of text, the routine computes the length of the line in order to center it about the defined plotting area. The line of text is spaced away from the defined plotting area by the distance which is computed by adding the appropriate ZONE value plus (0.25 * Character Size). Character Size is defined to be either FONTH or FONTB depending on whether a horizontal (TITLEB,TITLET) or a vertical (TITLEL, TITLER) titling routine is called. After the line of text has been constructed the appropriate ZONE value is incremented by (1.5 * Character Size).

For multiple lines of text:

To construct more than one line of text, the routines basically repeat the above computation for each additional line. Since the ZONE value is incremented, consecutive lines of text are separated by one-half the Character Size. TITLEB and TITLER process the consecutive lines of text in normal order given with the first line of text closest to the defined plotting area. TITLET and TITLEL process the consecutive lines of text in reverse order with the last line of text closest to the defined plotting area.

Restrictions

A. Text: A line of text cannot exceed 1060 alphanumeric characters.

B. Character Size: See appropriate post-processor writeup for any restrictions on the character size.

Usage

See the following individual subroutine writeups for:

A. FORTRAN call statement

B. Description of call list arguments

C. Routines Used

D. Example

Subroutine TITLEB

Purpose

To construct one or more lines of text to be centered below and parallel to the top of the defined plotting area.

Usage

A. FORTRAN call statement

For single line of text use

```
CALL TITLEB (#H...,SPECS)
        or
CALL TITLEB(LINE,SPECS)
```

For multiple lines of text use

```
CALL TITLEB (#H..,#H..,..,#H..,SPECS)
```

B. Description of call list arguments

#H... Array containing alphanumeric text. It is usually implied by Hollerith literals. The Hollerith literal may be simulated by using an array LINE with the following format:

Location	Contents
LINE(1)	THISbISbSA
LINE(2)	MPLEbLINEb
LINE(3)	OFbTEXT.bb
LINE(4)	0

where b represents a space. The last word (in this example LINE(4)) is set to an integer zero to signify the end of the line of text.

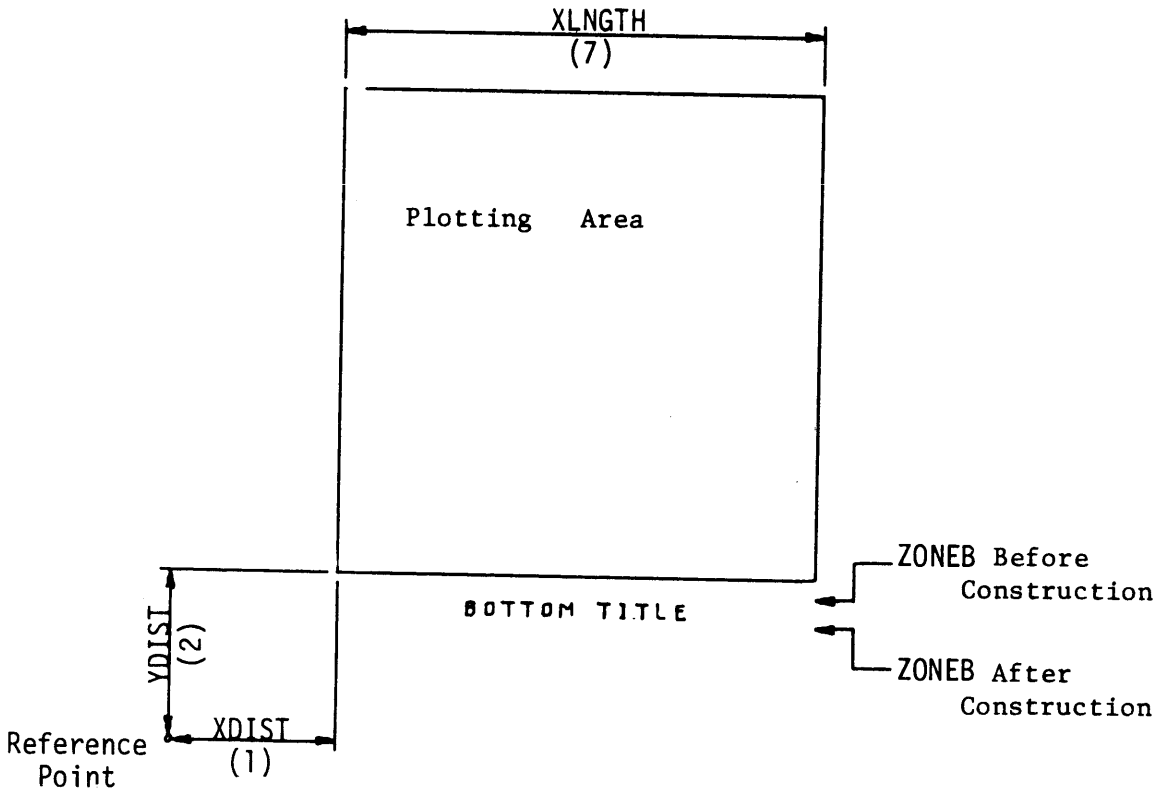
SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter	Mnemonic	Parameter	Mnemonic
SPECS(1) *	XDIST	*SPECS(13)*	
SPECS(2) *	YDIST	*SPECS(14)*	
SPECS(3)		*SPECS(15)*	
SPECS(4)		*SPECS(16)*	
SPECS(5)		*SPECS(17)* *	FONTB
SPECS(6)		*SPECS(18)* *	FONTH
SPECS(7) *	XLNGTH	*SPECS(19)* *	SPACER
SPECS(8)		*SPECS(20)*	
SPECS(9)		*SPECS(21)* *	FONTNO
SPECS(10)		*SPECS(22)*	
*SPECS(11)**	TOOL	*SPECS(23)*	
*SPECS(12)**	CCTAPE	*SPECS(24)* *	ZONEB

C. Routines Used: TITLED, TITLEG, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST,YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XLNGTH (IN INCHES)
      SPECS(7) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB,FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SET ZONEB (IN INCHES)
      SPECS(24) = 0.1
      CALL TITLEB(12HBOTTOM TITLE,SPECS)
            .
            .
            .
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine TITLET

Purpose

To construct one or more lines of text to be centered above and parallel to the top of the defined plotting area.

Usage

A. FORTRAN call statement

For a single line of text use

```
CALL TITLET (#H...,SPECS)
        or
CALL TITLET(LINE,SPECS)
```

For multiple lines of text use

```
CALL TITLET (#H..,#H..,..,#H..,SPECS)
```

B. Description of call list arguments

#H... Array containing alphanumeric text. It is usually implied by Hollerith literals. The Hollerith literal may be simulated by using an array LINE with the following format:

Location	Contents
LINE(1)	THISbISbSA
LINE(2)	MPLEbLINEb
LINE(3)	OFbTEXT.bb
LINE(4)	0

where b represents a space. The last word (in this example LINE(4)) is set to an integer zero to signify the end of the line of text.

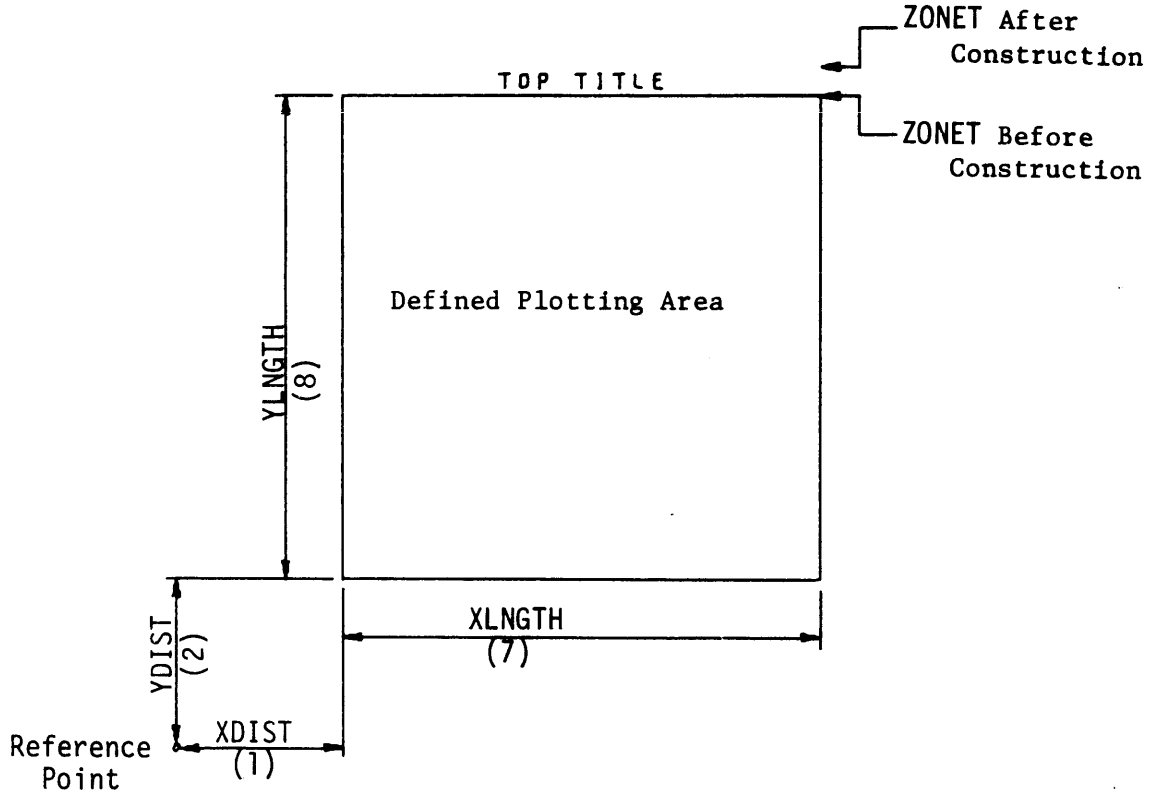
SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(14)*		
SPECS(2)	*	YDIST	*SPECS(15)*		
SPECS(3)			*SPECS(16)*		
SPECS(4)			*SPECS(17)*	*	FONTB
SPECS(5)			*SPECS(18)*	*	FONTH
SPECS(6)			*SPECS(19)*	*	SPACER
SPECS(7)	*	XLNGTH	*SPECS(20)*		
SPECS(8)	*	YLNGTH	*SPECS(21)*	*	FONTNO
SPECS(9)			*SPECS(22)*		
SPECS(10)			*SPECS(23)*		
SPECS(11)	*	TOOL	*SPECS(24)*		
SPECS(12)	*	CCTAPE	*SPECS(25)*	*	ZONET
SPECS(13)					

C. Routines Used: TITLED, TITLEG, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST,YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XLNGTH,YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB, FONTH(INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SET ZONET (IN INCHES)
      SPECS(25) = 0.0
      CALL TITLET(9HTOP TITLE,SPECS)
            .
            .
            .
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine TITLEL

Purpose

To construct one or more lines of text to be centered left of and parallel to the left side of the defined plotting area.

Usage

A. FORTRAN call statement

For a single line of text use

```
CALL TITLEL(#H...,SPECS)
```

or

```
CALL TITLEL(LINE,SPECS)
```

For multiple lines of text use

```
CALL TITLEL (#H..,#H..,..,#H..,SPECS)
```

B. Description of call list arguments

#H... Array containing alphanumeric text. It is usually implied by Hollerith literals. The Hollerith literal may be simulated by using an array LINE with the following format:

Location	Contents
LINE(1)	THISbISbSA
LINE(2)	MPLEbLINEb
LINE(3)	OFbTEXT.bb
LINE(4)	0

where b represents a space. The last word (in this example LINE(4)) is set to an integer zero to signify the end of the line of text.

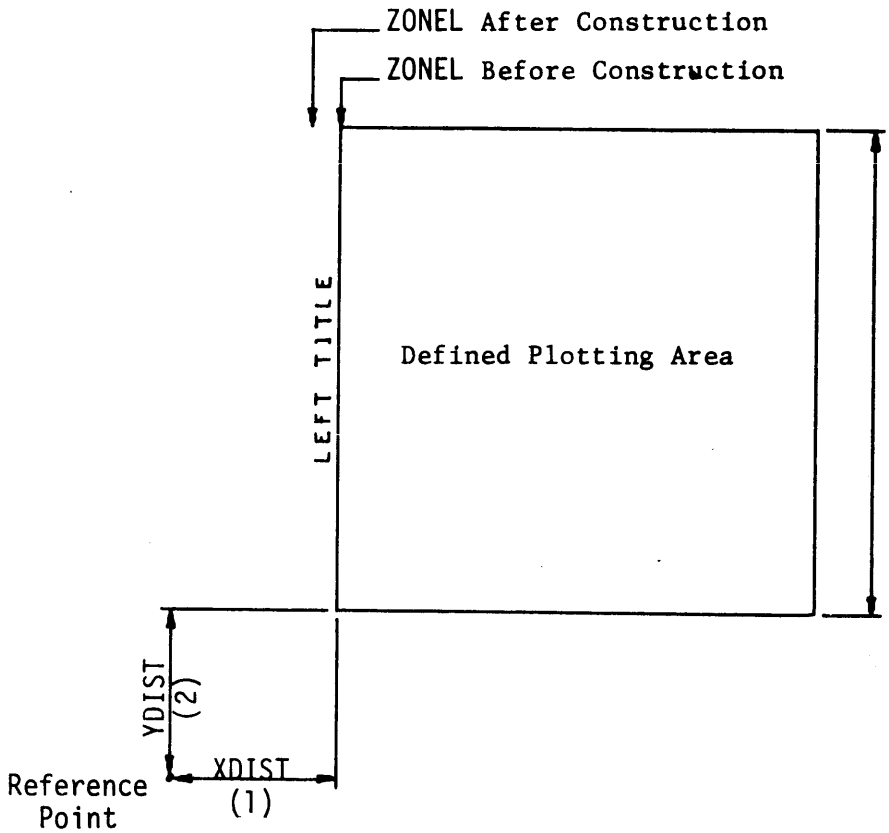
SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(14)*		
SPECS(2)	*	YDIST	*SPECS(15)*		
SPECS(3)			*SPECS(16)*		
SPECS(4)			*SPECS(17)*	*	FONTB
SPECS(5)			*SPECS(18)*	*	FONTH
SPECS(6)			*SPECS(19)*	*	SPACER
SPECS(7)			*SPECS(20)*		
SPECS(8)	*	YLNGTH	*SPECS(21)*	*	FONTNO
SPECS(9)			*SPECS(22)*		
SPECS(10)			*SPECS(23)*		
SPECS(11)	*	TOOL	*SPECS(24)*		
SPECS(12)	*	CCTAPE	*SPECS(25)*		
SPECS(13)			*SPECS(26)*	*	ZONEL

C. Routines Used: TITLED, TITLEG, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30), LINE(2)
      DATA LINE(1)/10HLEFT TITLE/0
      DATA LINE(2)/0/
C     SET XDIST,YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET YLNGTH (IN INCHES)
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB,FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SET ZONEL (IN INCHES)
      SPECS(26) = 0.0
      CALL TITLEL(LINE,SPECS)
            :
            :
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine TITLER

Purpose

To construct one or more lines of text to be centered to the right of and parallel to the right side of the defined plotting area.

Usage

A. FORTRAN call statement

For a single line of text use

CALL TITLER (#H...,*SPECS*)

or

CALL TITLER(*LINE*,*SPECS*)

For multiple lines of text use

CALL TITLER (#H..,#H..,..,#H..,*SPECS*)

B. Description of call list arguments

#H... Array containing alphanumeric text. It is usually implied by Hollerith literals. The Hollerith literal may be simulated by using an array LINE with the following format:

Location	Content
LINE(1)	THISbISbSA
LINE(2)	MPLEbLINEb
LINE(3)	OFbTEXT.bb
LINE(4)	0

where b represents a space. The last word (in this example LINE(4)) is set to an integer zero to signify the end of the line of text.

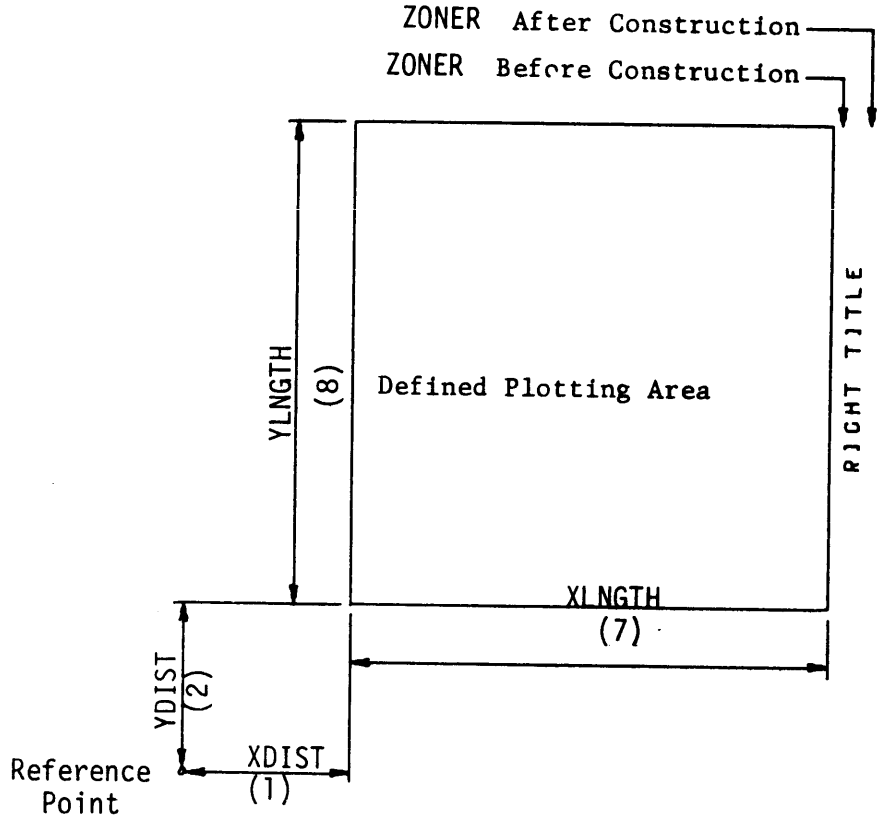
SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)	*	XDIST	*SPECS(15)*		
SPECS(2)	*	YDIST	*SPECS(16)*		
SPECS(3)			*SPECS(17)*	*	FONTB
SPECS(4)			*SPECS(18)*	*	FONTH
SPECS(5)			*SPECS(19)*	*	SPACER
SPECS(6)			*SPECS(20)*		
SPECS(7)	*	XLNGTH	*SPECS(21)*	*	FONTNO
SPECS(8)	*	YLNGTH	*SPECS(22)*		
SPECS(9)			*SPECS(23)*		
SPECS(10)			*SPECS(24)*		
SPECS(11)	*	TOOL	*SPECS(25)*		
SPECS(12)	*	CCTAPE	*SPECS(26)*		
SPECS(13)			*SPECS(27)*	*	ZONER
SPECS(14)					

C. Routines Used: TITLED, TITLEG, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SET XDIST, YDIST (IN INCHES)
      SPECS(1) = 0.877
      SPECS(2) = 0.877
C     SET XLNGTH, YLNGTH (IN INCHES)
      SPECS(7) = 2.5
      SPECS(8) = 2.5
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB, FONTH (INCHES)
      SPECS(17) = 0.1
      SPECS(18) = 0.1
C     SPECIFY SPACER (IN INCHES)
      SPECS(19) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SET ZONER (IN INCHES)
      SPECS(27) = 0.0
      CALL TITLER (11HRIGHT TITLE,SPECS)
            .
            .
            .
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine TITLEG

Purpose

To construct a single line of text above an arbitrary reference line.

Method

The routine constructs the line of text immediately above the reference line. The X ordinate (inches) of a specified point on the line (left, center, right) is *SPECS(22)*. The Y ordinate is *SPECS(23)*. *SPECS(20)* specifies the rotation angle (degrees) of the reference line with respect to the horizontal.

Restrictions

A line of text cannot exceed 1060 alphanumeric characters.

Usage

A. FORTRAN call statement

CALL TITLEG (*RULE*,#*H...*,*SPECS*)

or

CALL TITLEG(*RULE*,*LINE*,*SPECS*)

B. Description of call list arguments

RULE A floating-point variable which specifies which point on the line(s) of text is used with XSTART and YSTART for positioning the line. *RULE* may either be 1., 2., or 3. according to the following table:

RULE = 1. The specified point on the line of text is the lower left corner of the first character.

RULE = 2. The specified point is at the lower center of the line of text.

RULE = 3. The specified point on the line of text is the lower right corner of the last character.

#H... Array containing alphanumeric text. It is usually implied by Hollerith literals. The Hollerith literal may be simulated by using an array LINE with the following format:

Location	Contents
LINE(1)	THISbISbSA
LINE(2)	MPLEbLINEb
LINE(3)	OFbTEXT.bb
LINE(4)	0

where b represents a space. The last word (in this example LINE(4)) is set to an integer zero to signify the end of the line of text.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

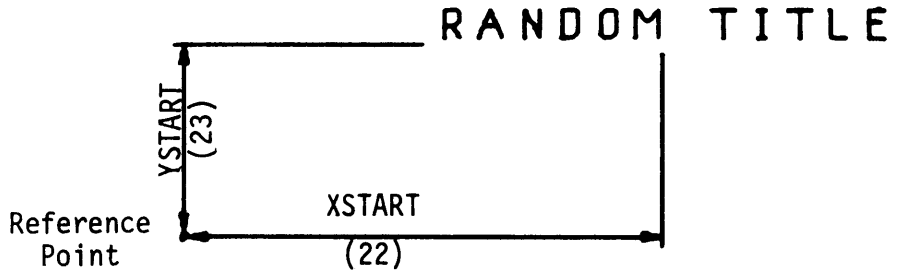
Parameter	Mnemonic	Parameter	Mnemonic
SPECS(1)		*SPECS(13)*	
SPECS(2)		*SPECS(14)*	
SPECS(3)		*SPECS(15)*	
SPECS(4)		*SPECS(16)*	
SPECS(5)		*SPECS(17)* *	FONTB
SPECS(6)		*SPECS(18)* *	FONTH
SPECS(7)		*SPECS(19)* *	SPACER
SPECS(8)		*SPECS(20)* *	ROTATE
SPECS(9)		*SPECS(21)* *	FONTNO
SPECS(10)		*SPECS(22)* *	XSTART
SPECS(11) *	TOOL	*SPECS(23)* *	YSTART
SPECS(12) *	CCTAPE		

NOTE: *SPECS(20)* is set relative to the horizontal. For example, *SPECS(20)* set equal to 0.0 specifies a horizontal line of text; *SPECS(20)* set equal to 90.0 specifies a vertical line of text. *SPECS(20)* may be set to <u>any</u> angle. See the appropriate post-processor writeup for any restrictions on line rotation.

C. Routines Used: SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30), LINE(3)
      DATA LINE (1)/10HRANDOM TIT/,LINE(2)/3HTLE/
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB,FONTH (INCHES)
      SPECS(17) = 0.2
      SPECS(18) = 0.2
C     SPECIFY SPACER (INCHES)
      SPECS(19) = 0.0
C     SPECIFY ROTATE (DEGREES)
      SPECS(20) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SPECIFY XSTART, YSTART (INCHES)
      SPECS(22) = 2.5
      SPECS(23) = 1.0
C     SPECIFY RULE FOR TITLE
C     CENTERED ABOUT XSTART
      RULE = 2.
C     SETUP ZERO WORD AFTER TITLE
      LINE(3) = 0
      CALL TITLEG (RULE,LINE,SPECS)
      .
      .
      .
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine SYMKEY

Purpose

To construct a plot symbol and a horizontal line of text.

Method

The routine constructs the plot symbol and begins the line of text immediately to the right of the plot symbol. The X ordinate (inches) of a specified point on the line (left, center, right) is *SPECS(22)*; the Y ordinate is *SPECS(23)*.

Restrictions

A line of text cannot exceed 1060 alphanumeric characters.

Usage

A. FORTRAN call statement

CALL SYMKEY (*RULE*,*#H...*,*SPECS*)

or

CALL SYMKEY(*RULE*,*LINE*,*SPECS*)

B. Description of call list arguments

RULE A floating-point variable which specifies which point on the line(s) of text is used with XSTART and YSTART for positioning the line. *RULE* may be either 1., 2., or 3. according to the following table:

RULE = 1. The specified point on the line of text is the lower left corner of the plot symbol.

RULE = 2. The specified point is at the lower center of the line of text, 2nd symbol.

RULE = 3. The specified point on the line of text is the lower right corner of the last character.

#H... Array containing alphanumeric text. It is usually implied by Hollerith literals. The Hollerith literal may be simulated by using an array LINE with the following format:

Location	Content
LINE(1)	THISbISbSA
LINE(2)	MPLEbLINEb
LINE(3)	OFbTEXT.bb
LINE(4)	0

where b represents a space. The last word (in this example LINE(4)) is set to an integer zero to signify the end of the line of text.

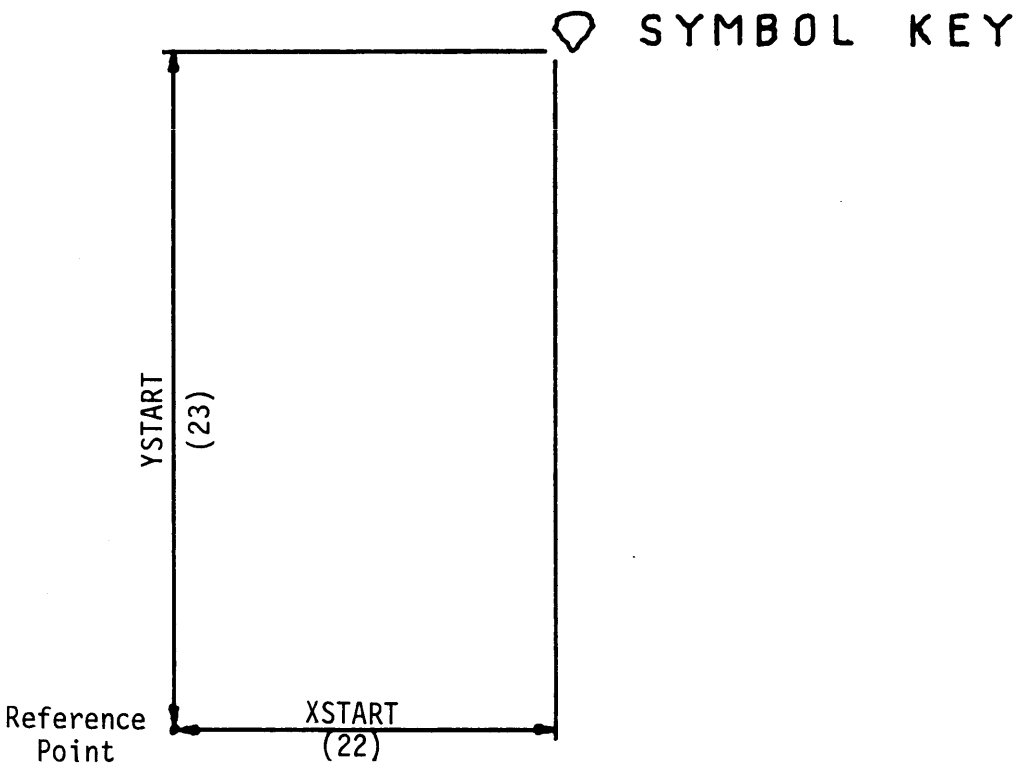
SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it;

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)			*SPECS(13)*		
SPECS(2)			*SPECS(14)*		
SPECS(3)			*SPECS(15)*		
SPECS(4)			*SPECS(16)*	*	SYMBOL
SPECS(5)			*SPECS(17)*	*	FONTB
SPECS(6)			*SPECS(18)*	*	FONTH
SPECS(7)			*SPECS(19)*	*	SPACER
SPECS(8)			*SPECS(20)*		
SPECS(9)			*SPECS(21)*	*	FONTNO
SPECS(10)			*SPECS(22)*	*	XSTART
SPECS(11)	*	TOOL	*SPECS(23)*	*	YSTART
SPECS(12)	*	CCTAPE			

C. Routines Used: TITLE,SIGNON,WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY SYMBOL TYPE (1-24)
      SPECS(16) = 6.
C     SPECIFY FONTB, FONTH (INCHES)
      SPECS(17) = 0.2
      SPECS(18) = 0.2
C     SPECIFY SPACER (INCHES)
      SPECS(19) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SPECIFY XSTART, YSTART (INCHES)
      SPECS(22) = 2.0
      SPECS(23) = 3.5
C     SPECIFY RULE SO XSTART REFERS
      TO LOWER LEFT-HAND CORNER OF SYMBOL KEY
      RULE = 1.
      CALL SYMKEY (RULE, 11H SYMBOL KEY, SPECS)
      :
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine DECVAL

Purpose

To construct, to the number of decimal places specified, a numerical value along an arbitrary reference line.

Method

The routine rounds and constructs the value to the specified number of decimal places immediately above the reference line. The X ordinate (inches) of a specified point on the line (left, center, right) is *SPECS(22)*. The Y ordinate is *SPECS(23)*. *SPECS(20)* specifies the rotation angle of the reference line with respect to the horizontal.

Usage

A. FORTRAN call statement

```
CALL DECVAL(RULE,VALUE,SPECS)
```

B. Description of call list arguments

RULE A floating-point variable which specifies which point on the line(s) of text is used with XSTART and YSTART for positioning the line. *RULE* may either be 1., 2., or 3. according to the following table:

RULE = 1. The specified point on the line of text is the lower left corner of the first character.

RULE = 2. The specified point is at the lower center of the line of text.

RULE = 3. The specified point on the line of text is the lower right corner of the last character.

#H... Array containing alphanumeric text. It is usually implied by Hollerith literals. The Hollerith literal may be simulated by using an array LINE with the following format:

Location	Contents
LINE(1)	THISbISbSA
LINE(2)	MPLEbLINEb
LINE(3)	OFbTEXT.bb
LINE(4)	0

where b represents a space. The last word (in this example LINE(4)) is set to an integer zero to signify the end of the line of text.

VALUE The floating-point value (single precision) to be constructed.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

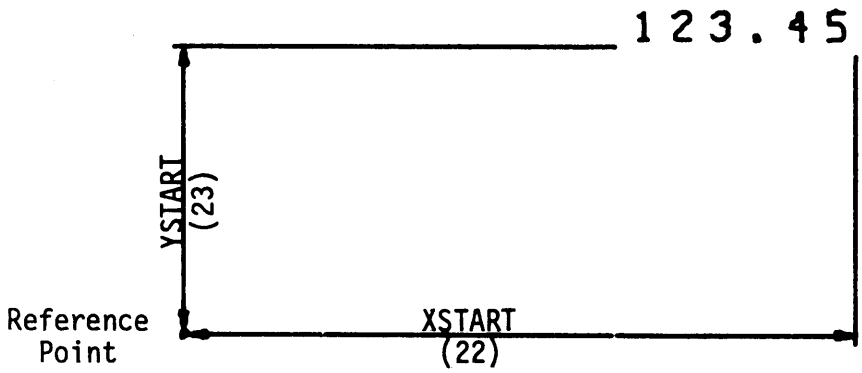
Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)			*SPECS(15)*		
SPECS(2)			*SPECS(16)*		
SPECS(3)			*SPECS(17)*	*	FONTB
SPECS(4)			*SPECS(18)*	*	FONTH
SPECS(5)			*SPECS(19)*	*	SPACER
SPECS(6)			*SPECS(20)*	*	ROTATE
SPECS(7)			*SPECS(21)*	*	FONTNO
SPECS(8)			*SPECS(22)*	*	XSTART
SPECS(9)			*SPECS(23)*	*	YSTART
SPECS(10)			*SPECS(24)*		
SPECS(11)	*	TOOL	*SPECS(25)*		
SPECS(12)	*	CCTAPE	*SPECS(26)*		
SPECS(13)			*SPECS(27)*		
SPECS(14)			*SPECS(28)*	*	DECPLS

NOTE: *SPECS(20)* is set relative to the horizontal. For example, *SPECS(20)* set equal to 0.0 specifies a horizontal line of text; *SPECS(20)* set equal to 90.0 specifies a vertical line of text. *SPECS(20)* may be set to <u>any</u> angle. See the appropriate post-processor writeup for any restrictions on line rotation.

C. Routines Used: DECBCD, TITLEG, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB, FONTH (INCHES)
      SPECS(17) = 0.2
      SOECS(18) = 0.2
C     SPECIFY SPACER (INCHES)
      SPECS(19) = 0.0
C     SPECIFY ANGLE OF LINE ROTATION (DEGREES)
      SPECS(20) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SPECIFY XSTART, YSTART (INCHES)
      SPECS(22) = 3.5
      SPECS(23) = 1.5
C     SPECIFY NUMBER OF DECIMAL PLACES
      SPECS(28) = 2.
C     SPECIFY RULE SO THAT XSTART, YSTART
      REFERS TO LOWER RIGHT CORNER OF LINE
      RULE = 3.
      VALUE = 123.45
      CALL DECVAL (RULE, VALUE, SPECS)
      .
      .
      .
```



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine SIGVAL

Purpose

To construct, to the number of significant figures specified, a numerical value along an arbitrary reference line.

Method

The routine rounds and constructs the value to the specified number of significant figures immediately above the reference line. The X ordinate (inches) of a specified point on the line (left, center, right) is *SPECS(22)*. The Y ordinate is *SPECS(23)*. *SPECS(20)* specifies the rotation angle of the reference line with respect to the horizontal.

The routine scales the constructed value (sets a value *IPOWER*) if the given value lies outside the range:

$$.1 \leq |VALUE| < 10^{(d+1)}$$

where d is the number of significant figures specified (*SPECS(29)*). Three examples best describe the scaling procedures.

Example 1 - if *VALUE* equals 0.0072546671 and four significant digits are specified, the routine would construct .7255 and set *IPOWER* equal to -2.

Example 2 - if *VALUE* equals 72.546671 and four significant digits are specified, the routine would construct 72.55 and set *IPOWER* equal to 0.

Example 3 - if *VALUE* equals 72546.671 and four significant digits are specified, the routine would construct 7255 and set *IPOWER* equal to 1.

Restrictions

None.

Usage

A. FORTRAN call statement

```
CALL SIGVAL (RULE,VALUE,IPOWER,SPECS)
```

B. Description of call list arguments

RULE A floating-point variable which specifies which point on the line(s) of text is used with XSTART and YSTART for positioning the line. *RULE* may either be 1., 2., or 3. according to the following table:

RULE = 1. The specified point on the line of text is the lower left corner of the first character.

RULE = 2. The specified point is at the lower center of the line of text.

RULE = 3. The specified point on the line of text is the lower right corner of the last character.

#H... Array containing alphanumeric text. It is usually implied by Hollerith literals. The Hollerith literal may be simulated by using an array LINE with the following format:

Location	Contents
LINE(1)	THISbISbSA
LINE(2)	MPLEbLINEb
LINE(3)	OFbTEXT.bb
LINE(4)	0

where b represents a space. The last word (in this example LINE(4)) is set to an integer zero to signify the end of the line of text.

VALUE The floating-point value (single precision) to be constructed.

IPOWER The scale factor determined by the routine to denote scaling of the constructed value. The value returned is a FORTRAN fixed-point integer.

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter		Mnemonic	Parameter		Mnemonic
SPECS(1)			*SPECS(16)*		
SPECS(2)			*SPECS(17)*	*	FONTB
SPECS(3)			*SPECS(18)*	*	FONTH
SPECS(4)			*SPECS(19)*	*	SPACER
SPECS(5)			*SPECS(20)*	*	ROTATE
SPECS(6)			*SPECS(21)*	*	FONTNO
SPECS(7)			*SPECS(22)*	*	XSTART
SPECS(8)			*SPECS(23)*	*	YSTART
SPECS(9)			*SPECS(24)*		
SPECS(10)			*SPECS(25)*		
SPECS(11)	*	TOOL	*SPECS(26)*		
SPECS(12)	*	CCTAPE	*SPECS(27)*		
SPECS(13)			*SPECS(28)*		
SPECS(14)			*SPECS(29)*	*	SIGFIG
SPECS(15)					

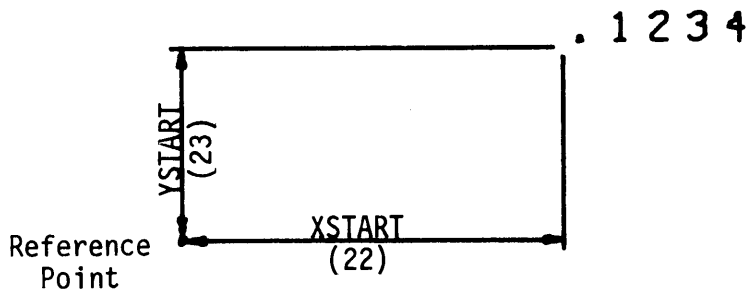
NOTE: *SPECS(20)* is set relative to the horizontal. For example, *SPECS(20)* set equal to 0.0 specifies a horizontal line of text; *SPECS(20)* set equal to 90.0 specifies a vertical line of text. *SPECS(20)* may be set to <u>any</u> angle. See the appropriate post-processor writeup for any restrictions on line rotation.

C. Routines Used: DECVAL, XSCAL2, DECBCD, TITLEG, SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
C     SPECIFY TOOL (NORMALLY ONE)
      SPECS(11) = 1.
C     SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
C     SPECIFY FONTB, FONTH (INCHES)
      SPECS(17) = 0.2
      SPECS(18) = 0.2
C     SPECIFY SPACER (INCHES)
      SPECS(19) = 0.0
C     SPECIFY ANGLE OF LINE ROTATION (DEGREES)
      SPECS(20) = 0.0
C     SPECIFY FONT TYPE (NORMALLY ONE)
      SPECS(21) = 1.
C     SPECIFY XSTART, YSTART (INCHES)
      SPECS(22) = 2.0
      SPECS(23) = 1.0
C     SPECIFY NUMBER OF SIGNIFICANT FIGURES
      SPECS(29) = 4.
C     SPECIFY RULE
      RULE = 1.
      VALUE = .001234
      CALL SIGVAL (RULE, VALUE, IPOWER, SPECS)
      :
      :
```

NOTE: IPOWER IS SET BY SIGVAL TO -2.



See the appropriate post-processor writeup for the definition of the reference point and the dimensions of the plotting area.

Subroutine GDSEND

Purpose

To terminate plotting.

Method

Places an end of plotting code on the intermediate file.

Restrictions

This must be the last plotting call. Any calls to the plot routine after this call will not be plotted.

Usage

A. FORTRAN call statement

```
CALL GDSEND (SPECS)
```

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter	Mnemonic	Parameter	Mnemonic
SPECS(1)		*SPECS(7)*	
SPECS(2)		*SPECS(8)*	
SPECS(3)		*SPECS(9)*	
SPECS(4)		*SPECS(10)*	
SPECS(5)		*SPECS(11)*	
SPECS(6)		*SPECS(12)*	* CCTAPE

C. Routines Used: SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
    C SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
      CALL GDSEND (SPECS)
         .
         .
         .
```

Subroutine NXTFRM

Purpose

To initiate a new frame.

Method

The method differs with each plotting device.

Restrictions

None

Usage

A. FORTRAN call statement

```
CALL NXTFRM (SPECS)
```

B. Description of call list arguments

SPECS An array containing plot construction parameters. Reference Table 1 under "The *SPECS* Array" for definitions and array format. This routine requires the *SPECS* parameters indicated below by an * to be specified prior to calling it:

Parameter	Mnemonic	Parameter	Mnemonic
SPECS(1)		*SPECS(7)*	
SPECS(2)		*SPECS(8)*	
SPECS(3)		*SPECS(9)*	
SPECS(4)		*SPECS(10)*	
SPECS(5)		*SPECS(11)*	
SPECS(6)		*SPECS(12)* *	CCTAPE

C. Routines Used: SIGNON, WRGHT

D. Example

```
      DIMENSION SPECS(30)
    C SPECIFY INTERMEDIATE FILE NUMBER
      SPECS(12) = 99.
      CALL NXTFRM (SPECS)
         .
         .
         .
```

E. Programming note: Be sure to reset the zone values (*SPECS(24)*, *SPECS(25)*, *SPECS(26)*, *SPECS(27)*).

Printer Postprocessor

Routine Name

PRINTR

Purpose

PRINTR processes an intermediate file, specified by *SPECS(12)*[1], and produces one or more plots on the line printer. Since it is inexpensive and provides speedy results, PRINTR is ideal for plot program debugging.

Plotting Device Description

A. Device type: Standard computer line printer with at least 130 printable characters and with the option to print 66 lines per page.

B. Accuracy: The line printer has fixed printing positions. There are 10 print positions per inch across the page (X direction) and 6 print positions per inch down the page (Y direction).

C. Plotting Area: The printer postprocessor has a maximum plotting area of 106 inches in the X direction by 11 inches in the Y direction. The first 6 inches (X direction) is right adjusted on the printer page and is printed only if used by the plot program. The rest of the plotting area is broken up into 13 inch sections (9 inches for the last section). Each section represents one computer page, and the program prints only up to the last page used by the plot program.

D. Reference Point: The Reference Point for PRINTR is located 6 inches from the left side of the maximum plotting area (X axis) and at the bottom of the maximum plotting area (Y axis). Therefore, from the reference point, the maximum plotting area is:

$$-6 \leq X \leq 100 \text{ inches}$$
$$0 \leq Y \leq 11 \text{ inches}$$

(This is the same definition as for the small Cal Comp plotter.)

---

[1] *SPECS(12)* = 99 for the current CDC 6400 implementation.

E. Plotter speed: Not applicable.

F. Options: None

Restrictions

1. Your program must not use file, TAPE99.

2. You must not use files TAPE18, TAPE19, TAPE20, TAPE21, TAPE22, or TAPE23 for anything other than scratch storage.

Method

A. PRINTR is an absolute overlay. It replaces and destroys the user's program.

B. In core page images: The -6 inch strip and a 2 page print image are held in core. When the program processes a point on a page not held in core, the 2 page print image is written on one of the intermediate files TAPE20, TAPE21, TAPE22, or TAPE23. The 2 page core image corresponding to the point being processed is then read into core. Therefore, plots requiring over 2 pages (26 inches, not including the -6 inch strip) will take much more time to process.

C. Order of processing: PRINTR processes points in the following order:

   1. Line segments - grids, axes and connected points
   2. Plot symbols
   3. All annotation

   Files TAPE18 and TAPE19 store the symbol and annotation information to defer their processing.

D. Annotation: Since the printer has only one size and spacing for characters the following conventions are used:

   1. Horizontal lines: The position of the first non-blank character is determined and then the rest of the characters are strung out in consecutive columns irrespective of the given character size.

   2. Vertical lines: The order of the characters are reversed (to make the line readable) and then the position of the first bottom nonblank character is determined and the rest of the characters are then strung out above each other irrespective of the given character size.

3. Imbedded blanks count as a space but do not overprint other plot lines.

4. Out of bounds error messages are not printed as long as at least one character is within the plot area.

Usage

A. PRINTR for debugging: To debug your plots with PRINTR, make your calls just like you would for CALCOMP and then execute PRINTR. Remember that the lettering may not correspond exactly to your CALCOMP plots. Dense grids tend to clutter the plot. See the example for control card usage.

B. PRINTR for final plots: PRINTR can be used to make very flexible and good looking printer plots. When using PRINTR for this purpose, note the following restrictions:

1. Set the character size to correspond to the print character size and insure proper spacing between lines

   a. for horizontal lines use:

   ```
   SPECS(17) = 0.1
   SPECS(18) = 0.111
   SPECS(19) = 0.0
   ```

   b. for vertical lines use:

   ```
   SPECS(17) = 0.166
   SPECS(18) = 0.133
   SPECS(19) = 0.0
   ```

2. Do not use dense grids (i.e., six grid lines per inch).

C. Symbols: According to the following table, a number of character is chosen to represent each symbol type.

Symbol	Character	Symbol	Character	Symbol	Character
0	.	9	9	17	G
1	1	10	0	18	H
2	2	11	A	19	I
3	3	12	B	20	J
4	4	13	C	21	K
5	5	14	D	22	L
6	6	15	E	23	M
7	7	16	F	24	N
8	8				

Calcomp Postprocessor

Routine Name

CALCOMP

Purpose

CALCOMP processes an intermediate file, specified by *SPECS(12)*[1], and produces a binary file suitable for the Calcomp plotter.

Plotting Device Description

A. Device type: The Model 663 Calcomp Plotter is an incremental pen and ink plotting device. The plotter may be driven on-line, as the IBM 360 does at the University, or it may be driven off-line via a magnetic tape unit. There are eight basic pen movements, shown in Figure 1 below, in addition to the ability of raising or lowering the pen. Figure 2 below shows an example of an approximation of an oblique straight line using the eight basic increments.

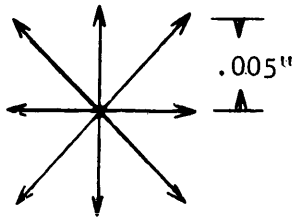


Figure 1
Eight Basic Pen Movements

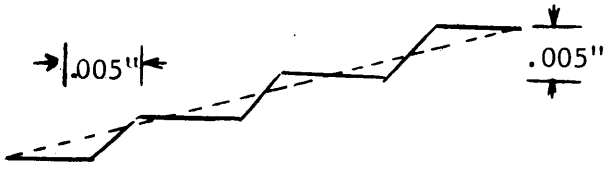


Figure 2
An approximation to the oblique straight line

B. Accuracy: The Model 565 Plotter is accurate to .0025 inch.

C. Plotting Area and Reference Point: The Calcomp postprocessor allows a maximum plotting area of 106 inches in the X direction by 11 inches in the Y direction for each frame. Care should be taken to allow a border of at least .1 inch at the extremeties of the Y direction as

---

[1] *SPECS(12)* = 99. for CDC 6400 implementation.

there are limits switches at each end which have a variation of about .1 inch. See Figure 3 below for a diagram of the plotting area and the definition of the reference point.

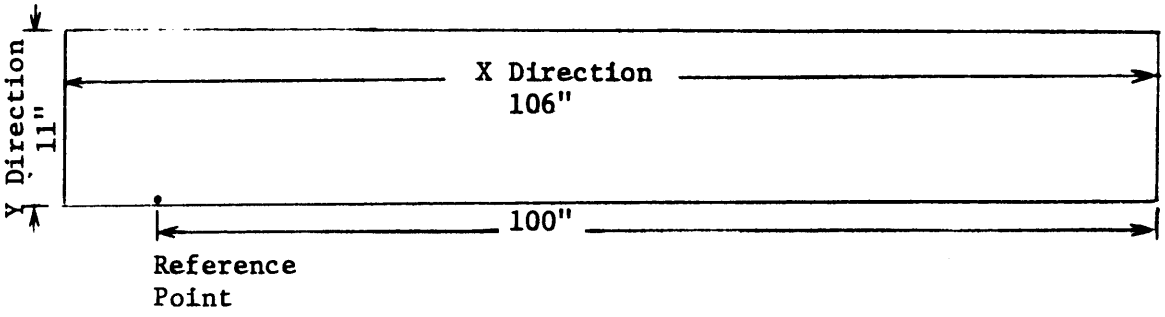


Figure 3

E. Plotter speed:

1. Pen speed: 2.2 inches per second.
2. Pen up or down: .01 second per pen up
   .05 second per pen down.

F. Options:

1. Pen colors: *SPECS(11)*, tool, specifies the pen color according to the following chart:

SPECS(11) tool	Pen color
1.	Black
2.	Red
3.	Blue
4.	Green

Restrictions

1. Your program must not use file TAPE99 (however you must specify it on your program card).
2. You must not use TAPE96. (You must NOT specify this file on your program card.)

Method

CALCOMP is an absolute overlay. It replaces and destroys the user's program.

Usage

A. Annotation: You have a choice of two different character fonts. *SPECS(21)* specifies which of these fonts is desired. See Figures 4 and 5 for an examples of these fonts.

For well formed letter using font 1 (with automatic spacing), you should use multiples of 0.1 inches for *SPECS(17)*, FONTB, and *SPECS(18)*, FONTH. *SPECS(19)*, SPACER, should be 0.0

Font 2 does not have automatic spacing. For well formed letters, use whole multiples of the following values for *SPECS(17)*, *SPECS(18)*, and *SPECS(19)* (FONTB, FONTH, and SPACER):

SPECS(17)	= 0.04	(FONTB)
SPECS(18)	= 0.06	(FONTH)
SPECS(19)	= 0.02	(SPACER)

B. Symbol: The symbol type is given by *SPECS(16)*. See Figure 6 for the symbols associated with each symbol type.

C. Binary Plot Deck: CALCOMP will produce an absolute binary† plot deck. Each frame will have binary sequencing in column 80 starting with one.

When you get your plot deck back (it will be filed with the regular punched output), do the following:

1. Take off the picture cards (cards with Job Number and Sequence Number).

2. Mark the deck with a felt pen to later identify it (i.e., the date run, number of frames, and any identification title). There is nothing more frsutrating and useless than a pile of unmarked binary plot decks which contains one deck you need to replot.

3. Add the following two cards to the deck:

   a. turn over a white job card and punch the following:

Col. 1	10	16	
./	PLOT	NNNN,NAME	(NNNN - Job Number)

   You may write any special requests on this card (i.e., Ball point). This card goes in front of your plot deck.

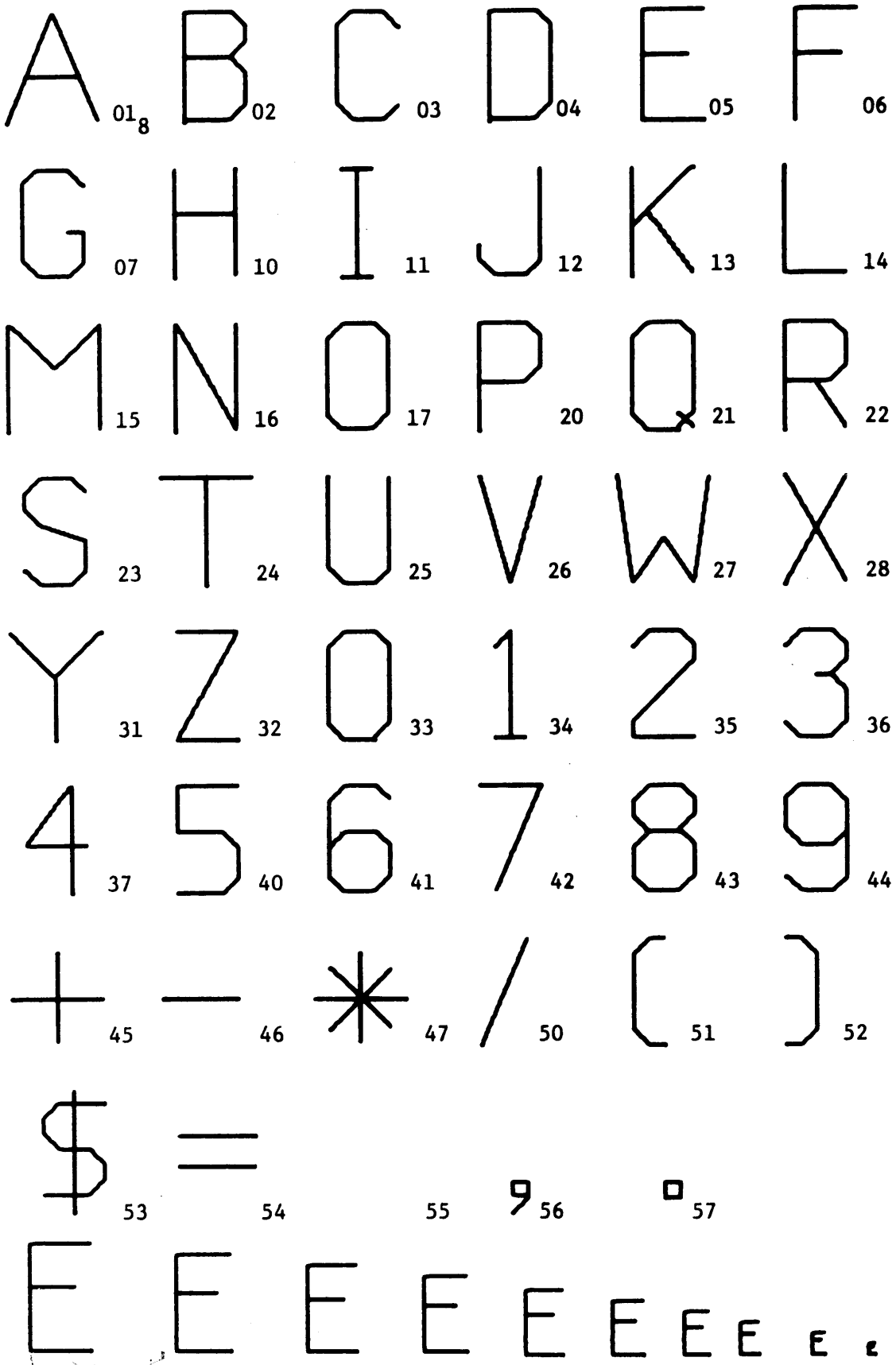
   b. Punch an "I" in column 80 (No, you don't punch the quotes.) This card goes in back of your plot deck.

4. Submit deck to IBM 360.

† Absolute binary cards are binary cards without a checksum and no mandatory 7-9 punch in column one.

Figure 4

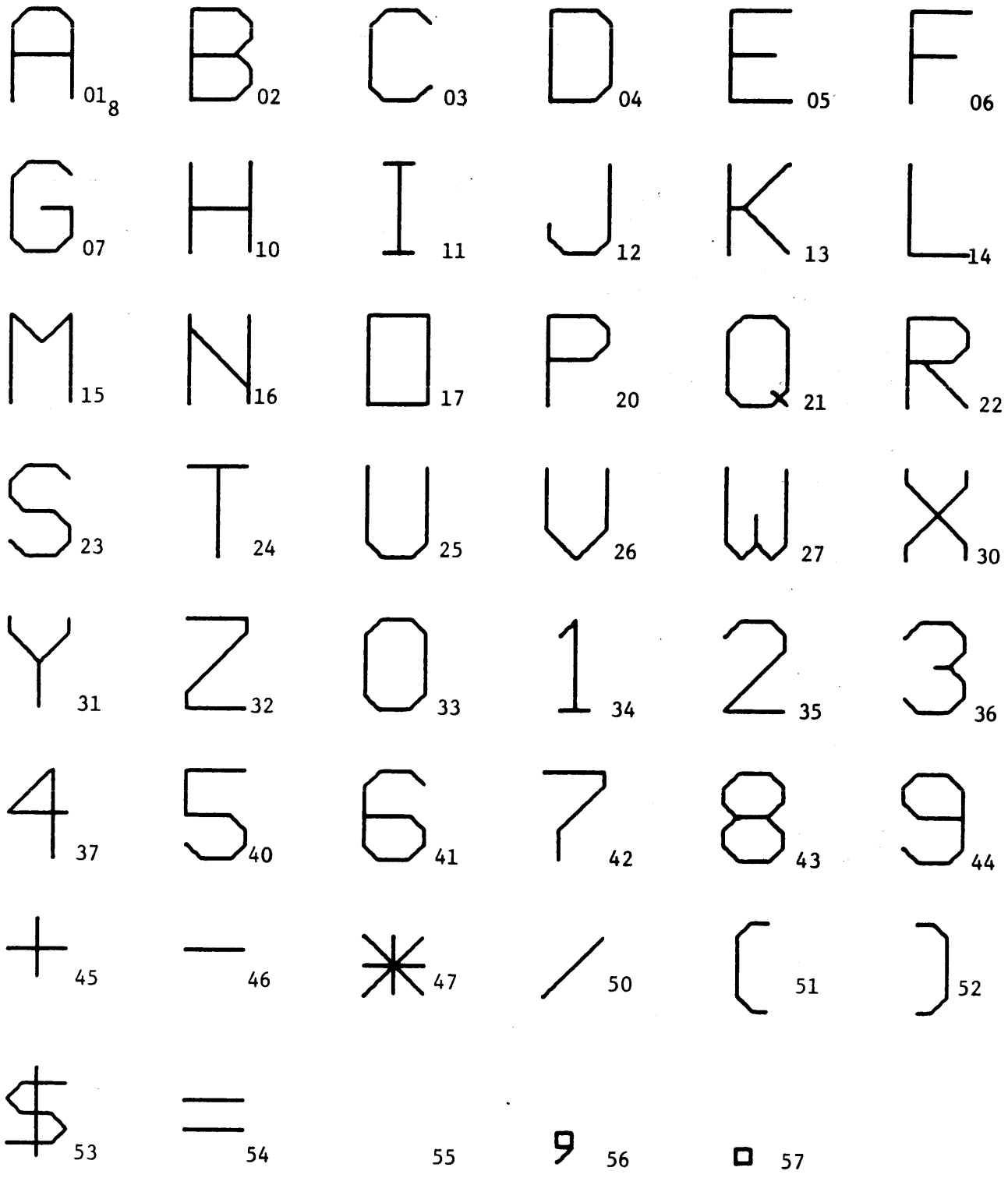
Font 1

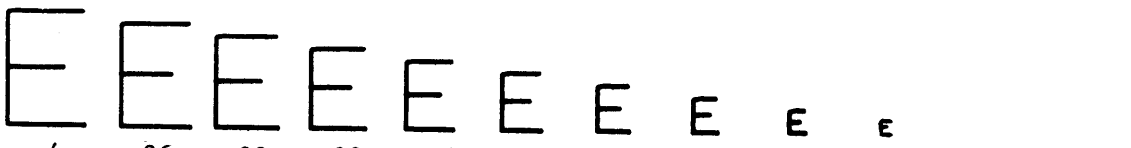


SPECS (17) = 1.0 .9 .8 .7 .6 .5 .4 .3 .2 .1
SPECS (18) = 0.0
SPECS (19) = 0.0

Figure 5

Font 2

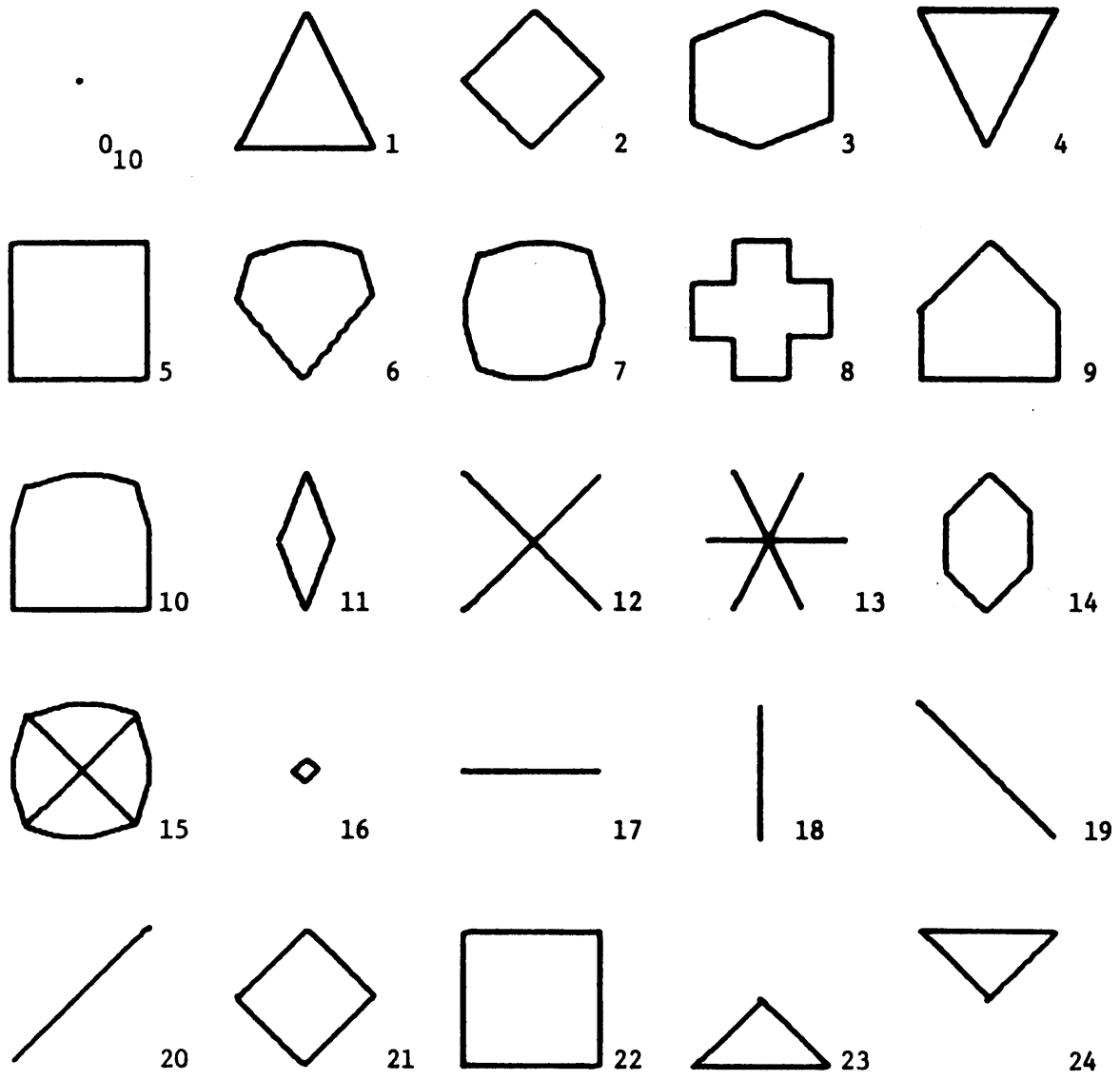




.4	.36	.32	.28	.24	.20	.16	.12	.08	.04	SPECS(17)
.6	.54	.48	.42	.36	.30	.24	.18	.12	.06	SPECS(18)
.2	.18	.16	.14	.12	.10	.08	.06	.04	.02	SPECS(19)

Figure 6

Symbol Table



Control Card Usage

A. CDC 6400 Implementation

The purpose of this section is to demonstrate how to use the Graphical Display system on the 6400 and possibly answer some of the questions which often arise.

1. Which of the three FORTRAN compilers, RUN, FUN, FTN, should be used to compile the main program?

   The Graphical Display System was compiled under the RUN compiler. Since RUN compiled and FUN compiled subroutines are interchangable, the main program may be compiled under either of these compilers. The Graphical Display System may not be used with programs compiled under FTN.

2. Where are the Graphical Display Subroutines found?

   The Graphical Display Routines are currently on the common file GDSLIB. The routine SELECT searches the LGO file and adds the needed plot routines to the end of the LGO file.

3. Where are the post-processing routines?

   Both post-processing routines are on the system library. They may be called in and executed with one control card. (See example below.)

4. Control Card Examples

   Example 1.

   To compile a plot program and execute both the PRINTR and CALCOMP post-processors.

```
"CDC 6400 Job Card"
RUN,S.                          Compile
CLDR,LIB=GDSLIB.
PRINTR.
CALCOMP.
7-8-9 card                      A card with 7,8, and 9 punched in
                                column one.
              PROGRAM PLOT(INPUT,OUTPUT,TAPE99)
                :
                :
7-8-9 card
              any data
6-7-8-9 card          A card with 6,7,8 and 9 punched in column one.
```

Example 2

To compile a plot program and execute only the PRINTR post-processor.

```
"Job Card"
RUN,S.
CLDR,LIB=GDSLIB.
PRINTR.
7-8-9 card

            PROGRAM PLOT(INPUT,OUTPUT,TAPE99)
              :
              :
7-8-9 card
            any data
6-7-8-9 card
```

Example 3

Load a binary deck and execute only the Calcomp post-processor.

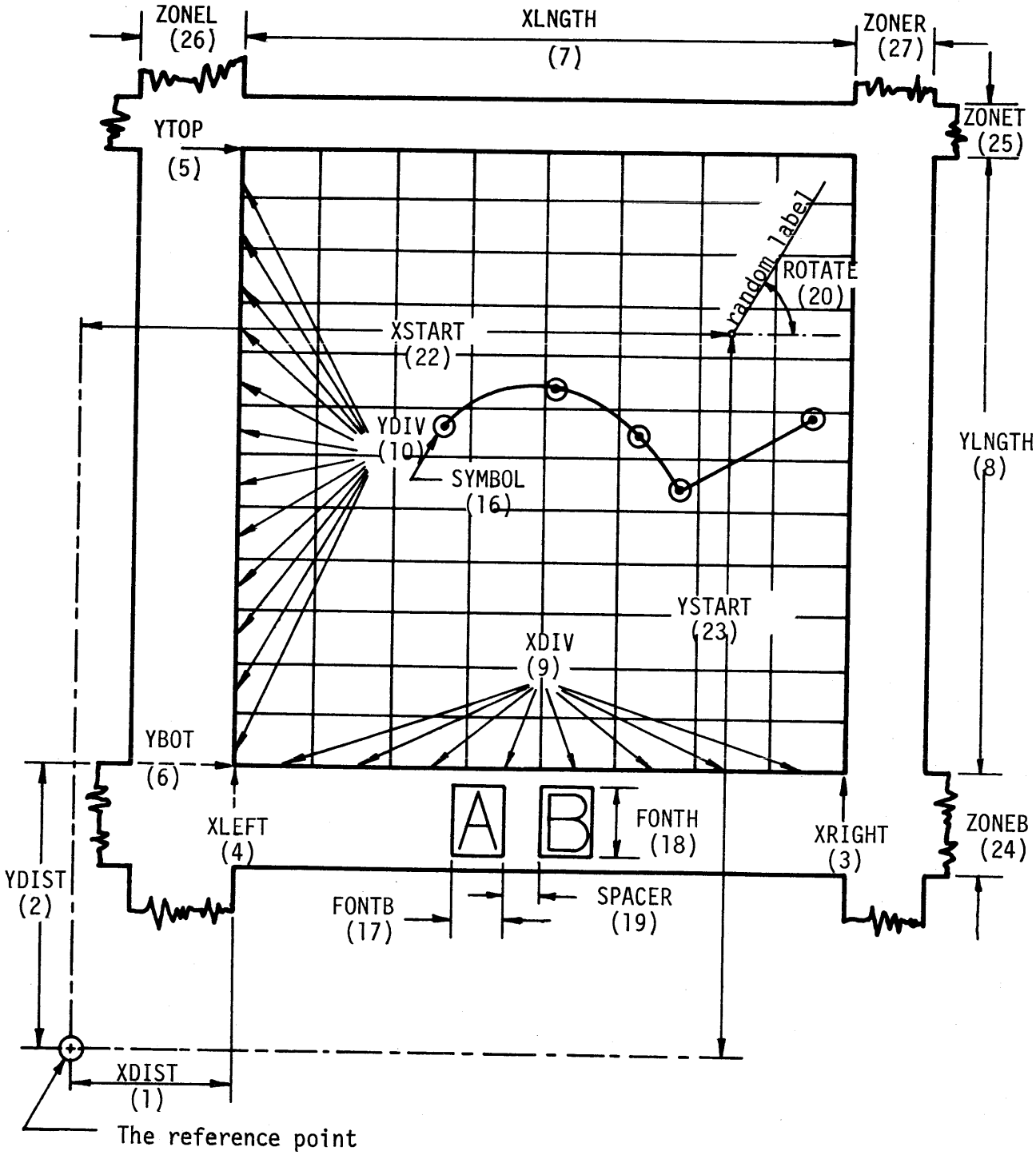
```
"Job Card"
CLDR,I,LIB=GDSLIB.
CALCOMP.
7-8-9 card

            (Binary Deck)
7-8-9 (17B)       A level 17 7-8-9 card.  A card with a 7-8-9
                  punched in column one, a 1 punched in column 2 and
                  a 7 punched in column  3, and a B in column 4.
            (any data)
6-7-8-9 card
```

Diagnostics

Currently, all diagnostics come from the post-processing routines. We are working on a general diagnostic routine which may be used in conjunction with the Graphical Display Subroutines. Until this routine is finished, and possibly even after that, the best diagnostics come from the plot itself. If the user runs into any trouble while using GDS, he should check the bulletin board outside the machine room for the schedule of the plotting consultant.

Good Luck!!!



See post-processor write-ups for the actual position of the Reference Point on any particular plotter.

Figure 1

A Sample Plot Illustrating *SPECS* Parameters