

Boys Town National Research Hospital

---

# **The PLT User's Guide**

Edward Carney

with additional material by Eric Javel, Stephen Neely and Theresa Langer

ÆCopyright 1991, Boys Town National Research Hospital  
555 North 30th Street, Omaha, Nebraska 68131

This publication describes Version 4.0 of the PLT program.

This publication was funded in part by the following grants  
from the National Institutes of Health:  
P50 DC00215  
P60 DC00982

## Table of Contents

### *Chapter*

1. Introduction.....	1-1
2. Tutorial .....	2-1
3. Functional Description.....	3-1
4. User Reference.....	4-1
4A. User Reference (Addendum) .....	4A-1
5. Error Messages.....	5-1
6. Examples.....	6-1

### *Appendices*

A. Complete TEST.PLT file from Chapter 2.....	A-1
B. Technical Data on Graphical Output and Printers.....	B-1
C. PLT External Fonts.....	C-1
D. PLT Command and Usage Summaries .....	D-1
1. PC.....	D-2
2. UNIX.....	D-9
E. Permuted Index of PLT Parameters .....	E-1

## **Chapter 1.**

# **Introduction**

The PLT program processes text files (ASCII) containing parameter values and commands and produces plots on several different graphics devices. The PLT files (often referred to as "pltfiles" in this document) may be created with any text editor or generated as the output of a data analysis program (like the BTNRH GREG program). The PLT program permits the user to preview a plot on the console screen and to generate output to various printers and pen-plotters. The laser printer output of PLT is considered to be of publication quality by several scientific journals.

The PLT program is currently used on four different computer systems at BTNRH: a DEC VAX-11/750 under BSD UNIX version 4.3, many IBM-PC/AT clones under MS-DOS 4.0, a SUN-386i under SunOS version 4.0 and a NeXT Computer under the NeXT version 2.0 of the UNIX operating system. The PLT program can generate several types of graphical output: to the computer screen, to printers and to plotters (See Appendix B, "Technical Data on Graphical Output and Printers").

This document describes the pltfile language, provides examples of its use and explains error messages that might be encountered. The PLT program also has several command line options which are not described in this document. Please refer to the system specific PLT manual page (DOS or UNIX) for command line options and other system dependent information. In addition, the version of PLT for the PC has a number of additional features, including the use of a cursor (with optional mouse) to permit the user to determine plot coordinates.

The PLT program was first developed by Drs. Eric Javel and Walt Jesteadt. Several programmers, including Roger Ferrel, Mike Collins and Matt Pruca, have been involved with translating and porting the program to various operating systems and computers. The program has evolved further in the past few years, primarily because of the ministrations and attentions of Dr. Stephen Neely, whose improvements are too numerous to mention.

## Chapter 2.

# Tutorial

In this chapter we will first describe how to create a PLT file on your computer. We will then take you step by step through the construction of a simple PLT file. The example used is a file called *test.plt*. The complete text of the *test.plt* file may be found in Appendix A. A copy of this file is also provided on the distribution disks, but we recommend that you construct the file in steps, just as you see in the tutorial.

### *Editing a PLT file*

The file that PLT reads must consist of ASCII character codes. It must not have any formatting information, such as is found in word processor files. If you are using an ASCII editor (such as Emacs, Vi, Brief or Ted), you only have to save the file and PLT will be able to read it. If you are using a word processor (such as Microsoft Word, WordPerfect, etc.), you must be sure to save the file as an ASCII text or "text only" file with line breaks. Consult your word processor manual for information on "text only" files.

When you type in the PLT parameters and keywords, keep the following items in mind:

1. Several parameters (but not keywords; see Chapter 3, "Functional Description") may be combined on a single line if each is separated from the other by a colon (:), surrounded by spaces. For example:

```
xlength=7.0  
xllc=2.0  
xmin=1.0  
xmax=21.0
```

and

```
xlength=7.0 : xllc=2.0 : xmin=1.0 : xmax=21.0
```

will produce the same results. The space before the colon is crucial; if it is not present, the results will be incorrect and might not be easily detectable on some plots;

2. Only the first four characters of a given parameter or command are significant; it is up to the user whether he or she wishes to use the remaining characters; for example:

```
xlength=7.0 and xlen=7.0
```

will produce the same results;

3. Parameters and commands may be spelled with upper- or lower case characters; for example:

```
XMIN=1.0 : XMAX=21.0 : XCYLE=0
```

and

```
xmin=1.0 : xmax=21.0 : xcycle=0
```

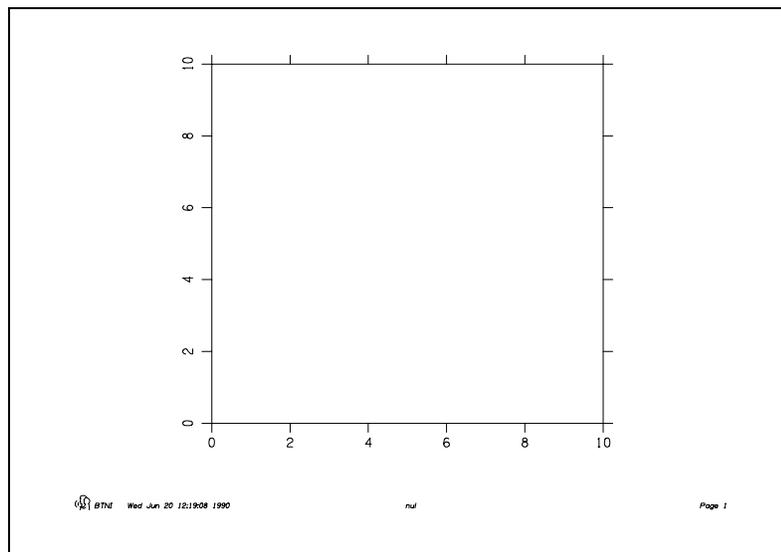
will produce the same results;

4. A line will have no effect if it is preceded by a semicolon (;). For example, the line, below, will not have its intended effect; the header will be plotted.

```
;header=no
```

## *Constructing a PLT file*

The simplest way to learn to use PLT is to begin with the simplest input we can think of. The empty file cannot be simplified, so, let's begin with that. Type **plt nul** at your keyboard (when you wish to display a file that is not empty, type **plt filename** at the DOS prompt). The file named *nul* is the "empty file" in DOS. Nothing comes out of it when you read from it and nothing remains when you write to it. You will see a plot that resembles Figure 2-1. (Just press the Enter key to return to DOS when you are finished viewing the display.)



*Figure 2-1.* The "NULL" plot.

This "NULL" plot tells us that the PLT program assumes certain things about the desired output. If the user doesn't provide a parameter value, that parameter keeps its default value. We will build on PLT's output from the NUL file and show you how to construct just the plot that you need. This requires that you edit a file of PLT parameters and commands using either an editor or a word processor as described above. The text will show you the result of every change that is made to an example file, but you should try this on your own. (Make sure that you have PLT Version 3.17 or later. To determine the version number, type **plt -v** at the DOS prompt.)

Let's assume that you have a set of data that you wish to plot. There are 12 data points in each of three conditions. You have decided that you wish to plot all three conditions in one graph, using both symbols and lines. You also have the idea that

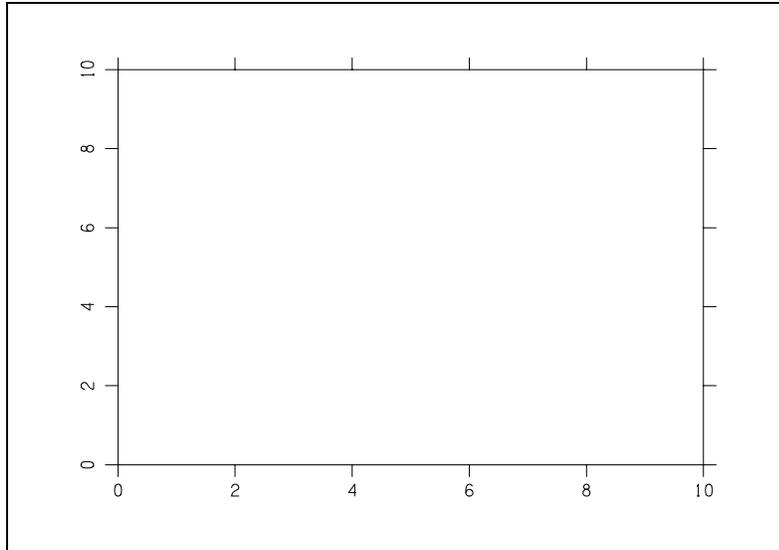
you want the graph to have what is termed "landscape" orientation (wider than it is tall). This is all that we need to start.

We will proceed as if you were plotting the graph by hand. We will start with the axes, axis labels and tick marks. We will progress to the data points and their connecting lines and, later, add a legend and some labels. We will also throw in some useful variations as we proceed.

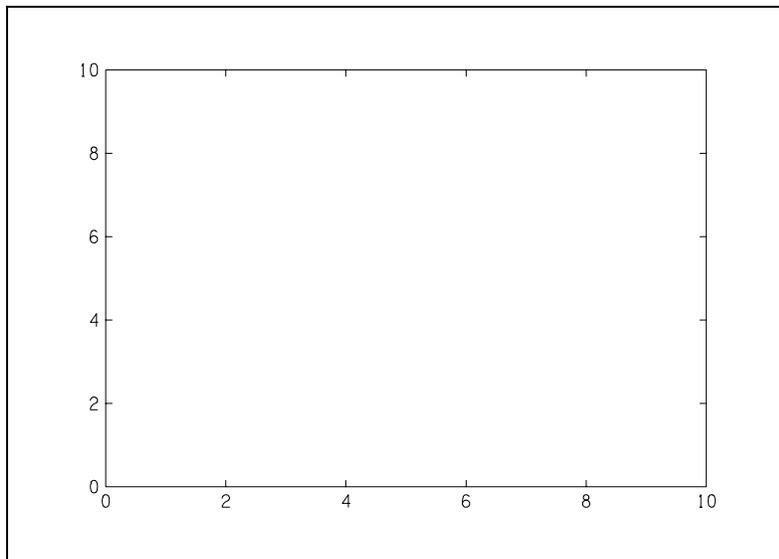
Many people begin hand plotting a graph by defining the space that the graph will occupy. The first step for us is to change the lengths of the x and y axes (each is 6 inches by default). Let's make our graph wider by making the x axis 7 inches wide. Just add the line *xlength=7.0* to your PLT file. (If you haven't already done so, create a file on your system and call it *test.plt*. If you are using a word processor make sure that the file is saved as "text only", without formatting.) Then, we will make the graph shorter by adding the line *ylength=5.0* to your PLT file.

The plot lies a little bit too far to the right on the page, now, so, let's move it to the left. The default of the lower left hand corner (*llc*) is 2.2 inches from the left and 1.5 inches from the bottom. We change the left-to-right position by adding *xllc=1.9* to the file. We don't really need to change the top-to-bottom position, so, we'll leave *yllc* set to its default value of 1.5. While we are about it, let's remove the "header" (actually a footer, of course) from the bottom of the page by adding the line *header=no*. Now, the plot looks like the one in Figure 2-2.

A couple of things to notice in Figure 2-2. The tick marks face outward. We can change the direction of the tick marks by using *ticdir=inward*. We can also change their size by typing a number which will change the tick mark size by some proportion. For example, if we want the tick marks to be twice as large we type *ticsiz=2.0*. Let's make them half as large by adding the line *ticsiz=0.5* to the PLT file. The second thing to notice about the figure is the direction of the annotations on the y axis. We can change that by adding the line *yhor=yes* to the file. We now have a plot as shown in Figure 2-3.



*Figure 2-2.* `xlength=7.0`, `ylength=5.0`,  
`xllc=1.9`, and `header=no`



*Figure 2-3.* `ticsiz=0.5`, `ticdir=inward` and `yhor=yes`

At this point, if you have not already done so, you might want to type in the file that we have created. The order of these parameters is not of great importance, but it should look similar to this:

```
header=no  
xllc=1.9  
xlength=7.0
```

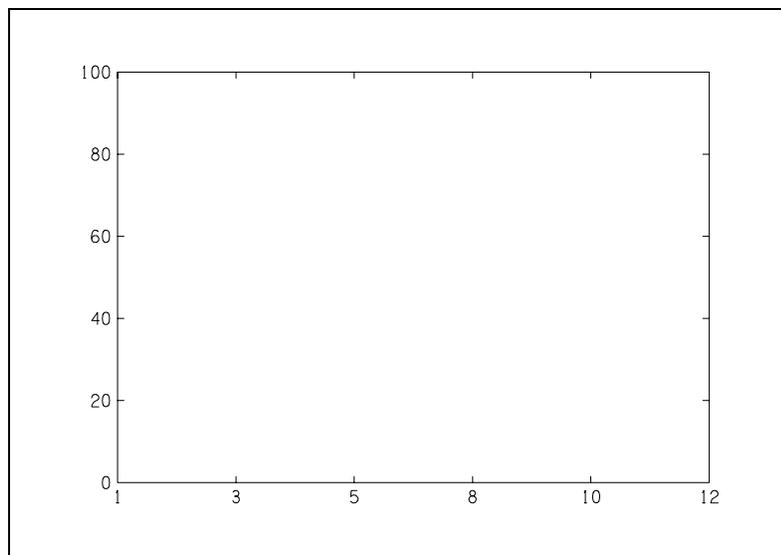
```
ylength=5.0
yhor=yes
ticdir=inward
ticsiz=0.5
```

Create the file *test.plt*, type in the lines and display the plot on the screen by typing **plt test** at the DOS prompt.

We will now set the stage for adding the data of interest. These values are found in Table 2-1. The data are average monthly temperature readings for each of the 12 months of 1973 in three cities in the U.S.: Eureka, CA, Lincoln, NE, and Newark, NJ. These data will have values of 1 through 12 on the x axis and values ranging from 21.8 to 79.6 on the y axis. We will set the x axis by placing the lines *xmin=1* and *xmax=12* in the file. This will create some difficulty at first, but we will remedy that shortly. The range for y values will be set from *ymin=0* to *ymax=100*. The resulting plot is seen in Figure 2-4.

*Table 2-1.* Average monthly temperature for three cities (1973).

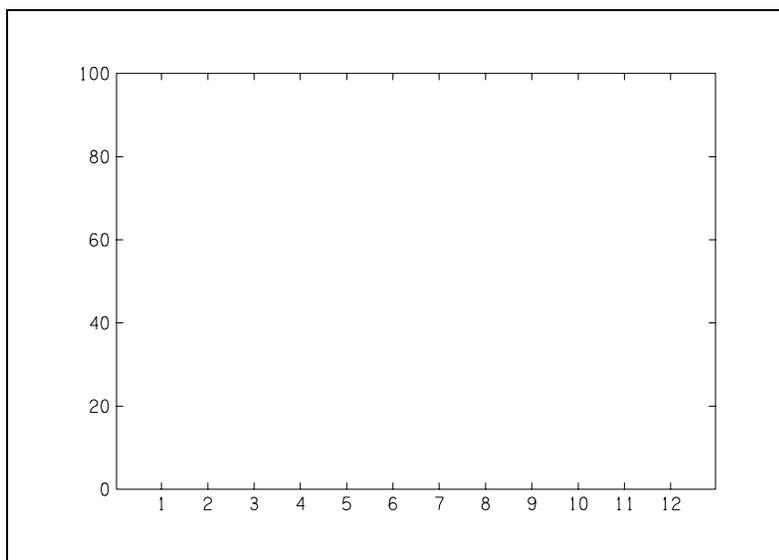
City	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Eureka, CA	47.3	50.7	47.4	50.1	52.0	55.1	55.6	54.8	57.1	52.1	51.4	51.2
Lincoln, NE	23.4	28.4	42.6	49.6	59.0	73.4	75.0	77.2	63.2	56.1	39.0	21.8
Newark, NJ	35.5	33.3	48.6	54.2	60.4	74.5	78.7	79.6	71.0	60.3	48.8	39.4



*Figure 2-4.* *xmin=1 : xmax=12* and *ymin=0 : ymax=100*

The y axis looks fine, but, if you look closely, you'll see that the x axis has some funny annotations (1, 3, 5, 8, 10 and 12). This occurs because the number of intervals is set to its default value of five. The PLT program then assigns annotations which are close to the values at the tick marks. Because we would like every month to be annotated, we will set the number of intervals to be 11 (*xint=11*). The number of intervals is one less than the number of annotations that we want to have plotted.

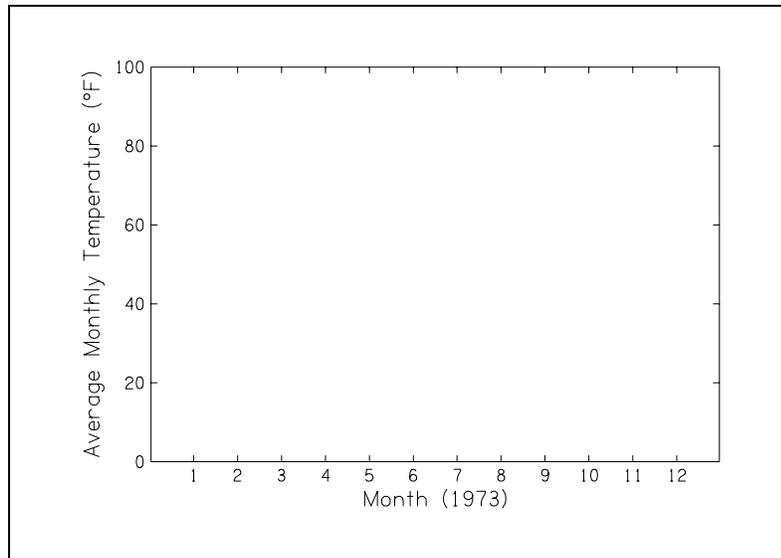
Another problem is now noticeable. The data for January and December (months 1 and 12) will now be plotted on the borders of the plot. This situation can be remedied by simply using less of the space available on the x axis. All we have to do is set *xpercent* to the percentage of the full axis that the data will occupy. The default is 100%. We will set the new value to 85% with *xpercent=85*. The results of this and the previous change are seen in Figure 2-5.



*Figure 2-5. xint=11 and xpercent=85*

We can now add labels to our axes. The label that we want on the x axis is "Month (1973)". This is set by adding the line *xlabel=Month (1973)* to the PLT file. We will create a label for the y axis with *ylabel=Average Monthly Temperature*. We would like to add a parenthetical (°F) to our plot, but we have no way of typing the degree symbol (°) on the keyboard. There is a way to plot this symbol in PLT without typing it.

Every symbol and letter may be represented as a number (see Figure 3-1 in Chapter 3). To represent a symbol in the standard font we type the number between two vertical bars (these bars will appear like this, |, in print, but on some keyboards they will appear with a separation). To represent a symbol in the alternate font we type the number between two vertical bars and then between two carets. The symbol number for the ° symbol is number 48 in the alternate font set, according to the table. We must therefore type ^ | 48 | ^ in order to obtain the symbol (another method is to type the corresponding character from the standard font set; so, typing ^0^ would do just as well). The y label is set by adding the line *ylabel=Average Monthly Temperature (^ | 48 | ^F)* to our file. The result is seen in Figure 2-6.



*Figure 2-6.* xlabel=Month (1973)  
ylabel=Average Monthly Temperature (^ | 48 | ^F)

Now, let's add the first set of data (from Eureka, CA). We must tell PLT what kind of graph to make. We have several to choose from: symbols only, lines only, both symbols and lines, histogram, rectangles, polygons, waves and contiguous rectangles. We want our plot to show both symbols and lines, so, we will set *pltype=both* (See **PLTYPE** in Chapter 4, "User Reference", for a complete description). We now have many choices of line types and symbol types. Each line and symbol type has a number and is set using parameters *symbol* and *lintype*. We will use unfilled squares for Eureka, CA (*symbol=0*), unfilled circles for Lincoln, NE (*symbol=1*) and unfilled triangles for Newark, NJ (*symbol=2*). The lines will be the

same for all three cities—a solid line (*lintype=0*) (See Figure 4-3 in Chapter 4, for a complete list of symbols and line types).

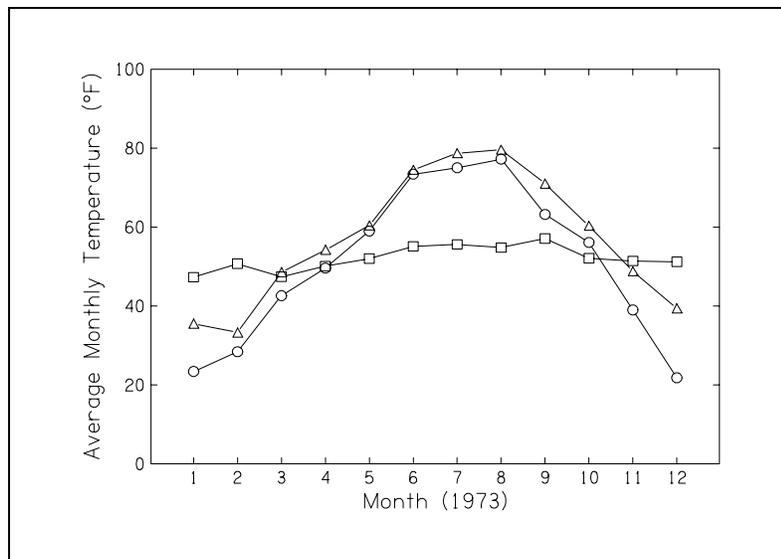
The data may appear anywhere, but, for convenience, we will place the data immediately after the symbol setting for each city. The data are typed in by pairs, first the x datum and then the y, separated by spaces, a tab or a comma. The data must be followed by the command *plot* to force PLT to plot the data before we change the symbol type for the next city. Our file will now contain the following additional lines for the Eureka, CA, data:

```

pltype=both
lintype=0      ; this line is optional (default is 0)
symbol=0
data          ; this line is optional
1 47.3
2 50.7
[other data]; rest of data go here
12 51.2
plot         ; indicates the end of this set of data

```

We'll add the other data at the same time. For each new set, we need only change the *symbol* parameter to its appropriate value (*symbol=1* for Lincoln, NE; *symbol=2* for Newark, NJ). The *pltype* and *lintype* parameters remain unchanged. We end up with a plot that looks like the one in Figure 2-7.



*Figure 2-7.* pltype=both : symbol=0 : lintype=0  
all data values plotted

Now we will add a legend so that readers will know what they are looking at. The legend is plotted as a "message" on the plot. Messages are simply clumps of text which are plotted anywhere you choose--they are not tied to an axis like the axis label and the annotations. There are two ways to place messages on the plot. You may plot the message at specific distances in inches from the x and y axes or at a location corresponding to the x-y coordinates of a data value. The best place for the legend in our plot is in the upper left corner. Let us first do this by distance.

The x distance is set by the parameter *mxllc* (message x lower left corner) and the y distance by the parameter *myllc*. We will add the lines *mxllc=0.5* and *myllc=4.5* to our file. We will make the message a little smaller using *msgsiz=0.8*, so it doesn't overwhelm the space it occupies. The message follows these lines and is enclosed in double quotes ("). The message may extend over several lines and it may contain symbols like the one in the *ylabel*, above. In addition we can add special symbols to the message which correspond to the symbol and line type used in our data plot. In the case of the Eureka data we would use "|0| |\_0| |0| Eureka, CA". Each |0| corresponds to the unfilled square; the |\_0| corresponds to line type 0.

The following will display a legend for our data as in Figure 2-8:

```
mxllc=0.5
myllc=4.5
msgsiz=0.8
"|0| |_0| |0| Eureka, CA
|1| |_0| |1| Lincoln, NE
|2| |_0| |2| Newark, NJ"
```

The x-y location and the message may be combined on the same line as follows:

```
0.5,4.5,"|0| |_0| |0| Eureka, CA
|1| |_0| |1| Lincoln, NE
|2| |_0| |2| Newark, NJ"
```

You may also place the message at the same point in the plot by using data values rather than inches. Simply set *msgdat=yes*. The lines are as follows:

```
msgdat=y
1,90,"|0| |_0| |0| Eureka, CA
|1| |_0| |1| Lincoln, NE
|2| |_0| |2| Newark, NJ"
```

Now, the x-y location is expressed in data values (1,90), not in inches (0.5,4.5). This corresponds to month 1 (January) and a temperature of 90°F. The first line of the message will start at this position.

The next item we might like to have is a title across the top of our plot. We will plot this with a message just as we plotted the legend. We will use *msgdat=no* this time and enlarge the message slightly with *msgsiz=1.5*. We will be sure that the message will be centered by changing the horizontal alignment of the message to center by using *mhalign=center*. All we have to do then is set the location to the number of inches from the x and y axes where we want the message centered and PLT takes care of the centering for us. The lines to add to the file are:

```
msgdat=no
msgsiz=1.5
mhalign=center
3.5,5.75,"Average Monthly Temperature for Three Cities"
```

We end up with the plot in Figure 2-8.

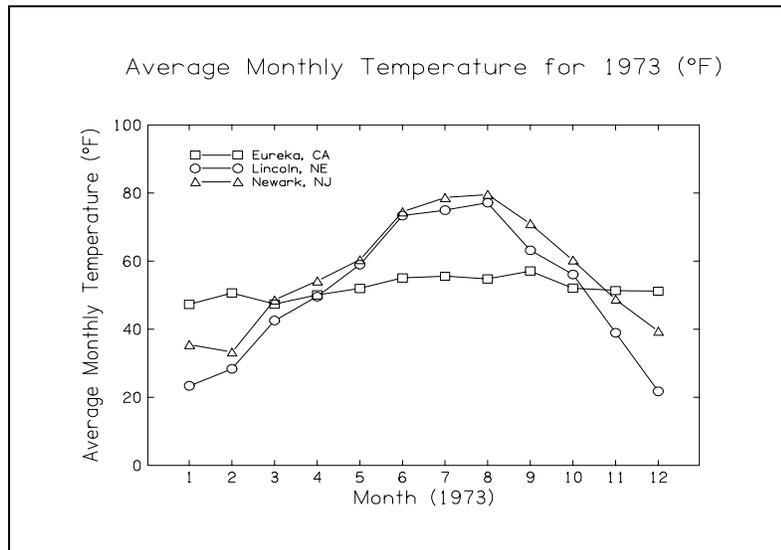


Figure 2-8. message for main title

There is just one more thing that is troublesome, and that is the use of 1, 2, 3, etc. to designate the months. This is easily fixed using the *xannot* parameter. This is a string substitution parameter for the annotations on the x axis (there is a corresponding parameter, *yannot*, for the y axis). For each data value we wish to

have labelled, we enter the datum and the desired string of characters. In our example we set it as follows:

```
xannot= 1,J 2,F 3,M 4,A 5,M 6,J 7,J 8,A 9,S 10,O 11,N 12,D
```

This tells PLT to substitute the string to the right of the comma for each of the data values to the left. So, where '1' would appear a 'J' shows up, for a '2' an 'F', etc. (We could just as easily use the first three letters of each month. Such a line would read **xannot= 1,Jan 2,Feb** etc.) We will add this parameter in the section with the other x-axis parameters (make sure it comes before the first PLOT keyword). Our final version of the plot appears in Figure 2-9 on the next page.

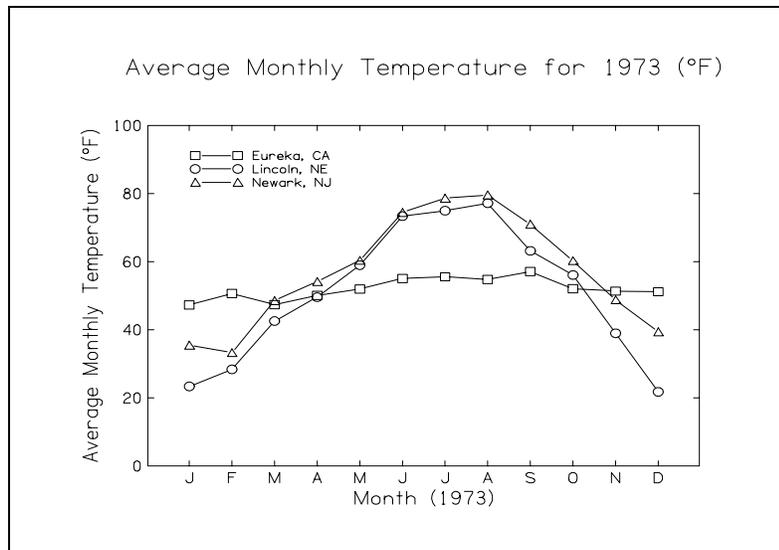


Figure 2-9. xannot= 1,J 2,F ... 12,D

## Chapter 3.

# Functional Description

PLT may be considered from two viewpoints: 1) the text file from which the plot is made and 2) the plot which is produced. If we take the textfile as our object of interest, we find that it consists of seven types of lines:

- |                   |                               |
|-------------------|-------------------------------|
| 1) comments       | 5) keywords                   |
| 2) messages       | 6) data value representations |
| 3) parameters     | 7) macros                     |
| 4) user variables | 8) auxiliary file commands    |

If, on the other hand, we consider the plot itself, we will find that there are five main sets of elements in a plot:

- 1) axes (including tick marks, annotations and labels)
- 2) data representations
- 3) messages
- 4) grids, header, and other miscellaneous elements
- 5) colors

We will briefly consider the components of the PLT text file (called a pltfile) and, then, move on to describe how the various parts of the pltfile combine to produce the plot elements mentioned above. This chapter will provide you with an overview of the parameters and keywords. For detailed descriptions, consult Chapter 4, "User Reference."

## **A. Pltfile Elements**

### ***1. Comments***

Text following a semicolon (;) or an exclamation point (!) are internal and external comment lines, respectively. These are not executed by PLT. Rather, they serve only as references for the user. Internal comments, i.e., those following a semicolon, are for organizational use. They allow the creator to keep track of the file during editing, and they are only printed (along with everything else) when echoing is desired. The semicolon may appear anywhere on the line, and anything following it on that line is treated as an internal comment.

External comments, i.e., those following an exclamation point, instruct PLT to print that comment regardless of echo status. If the console is being used for graphic output, the comments will be written to the log file (plt.log); for other devices the external comments appear on the console. Again, the exclamation point may appear anywhere in the line, and the external comment is the text following it. External comments are useful when execution of the plot requires user intervention, for example when pens or paper need to be changed.

Since semicolons and exclamation points denote comments, they cannot be used as characters in axis labels or messages unless they are bounded by special characters (see the next section).

### ***2. Messages***

Users can place text messages anywhere on the page, whether inside or outside plot frames. Features of messages include the ability to (a) define and plot multiple messages without intervening PLT commands, (b) plot superscripted and subscripted text, (c) insert symbols in message strings, and (d) enter messages from the keyboard at run time.

Message strings are delimited by double-quote marks, such as "texttexttext". The message is the text inside the quotation marks and it may occupy several lines. The quotation marks will not appear on the plot. The message position may be specified by explicitly setting the MXLLC and MYLLC parameters (see below) or by

placing the X and Y values on the same line as (and immediately preceding) the opening double-quote mark. This provides an alternative (and, often, more convenient) way of setting the MXLLC and MYLLC parameters. Messages are plotted as they are encountered in the PLT file. They are always positioned relative to the lower-left corner of the current frame (either in inches or in data units).

Several characters are treated as special characters when they appear in the message text. A dollar sign, \$, may be used as a space-holder, and it is converted into a blank before plotting the message. A caret, ^, is used to switch between the standard and alternate fonts (see Figure 4-1). Putting an asterisk, \*, as the first character after the leading quotation mark instructs PLT to accept a message string from the user's keyboard, provided that the console screen is not the graphical output device.

Symbols in message strings are defined by enclosing the number of the desired symbol in vertical bars like this: |number|. An asterisk in place of a symbol number instructs PLT to use the value of the SYMBOL parameter to determine the symbol number. Symbols in message lines are plotted at the same size as the text characters in the message, unless modified by the MVKEY parameter. If the number within the vertical bars is preceded by an underscore character like |\_number|, then a short line is inserted into the message instead of a symbol. The line type is determined by the number; the length of the line is controlled by the parameter MHKEY. If the number within the vertical bars is preceded by a number character like|#number|, then a small box is inserted into the message which is filled with a pattern determined by the value of the number. The height and width of this "shade box" are controlled by the parameters MVKEY and MHKEY.

The user can define superscripted text by bounding the text string with { and }. In a similar fashion, subscripted text is obtained by bounding the text with [ and ]. The brackets do not appear in the message. The user can force a literal interpretation of special characters such as {, }, [, ], ", \$, \*, ^, ;, and ! by typing them with a preceding backslash or by using their equivalent standard font numbers. For example, both \\* and |42| will produce an asterisk in the message text.

Message control will be discussed in Section B, Plot Elements, below.

### ***3. Parameters***

Parameter manipulations control the features of the plot. It is by altering these that the user obtains his or her desired graph. There are four general classes of parameters and these correspond to the four sets of elements seen in a plot. These are (1) axis specifications and annotation control, (2) data input control and data representation control, (3) message control specifications, and (4) miscellaneous parameters. The structure of a parameter manipulation consists of a word, followed by an equals sign, followed by the desired parameter value. PLT examines only the first four characters of the parameter, even though its full name may be longer. More than one parameter may be placed on a line of text, if each is separated from its neighbors by a colon (:) preceded by a space (the space is critical).

The value specified for a parameter may be an integer (1, 2, 73, etc.) or a floating-point number (4.2, 3.14159, etc.). Exponential notation may be used for floating-point numbers (42.0e-1, .314159E1, etc.). The parameter value may also be indicated as an arithmetic expression including operators for addition, subtraction, multiplication, division and exponentiation (+, -, \*, /, and ^). The arithmetic expression may include reference to PLT parameters or user variables (see Section A.4, below). Parentheses may be used freely to group these arithmetic expressions.

Parameters and their control of axes, annotations, data modifications, grids and other miscellaneous elements will be discussed in Section B, "Plot Elements", below.

#### ***4. User Variables***

Users may specify variables for storage of values and for use in expressions. These variables may be named with the letters of the alphabet (a-z only; case is ignored). The variables are set by typing the name followed by an equal sign, followed by an expression (e.g. a=2.3). The expression itself may contain values of numerical parameters (e.g. l=xlen, b=(xmax-xmin)). The user variables retain their values until they are reset; they are unaffected by the PLOT keyword.

#### ***5. Keywords***

The keywords used in PLT may be considered the "verbs" of the pltfile; an action is associated with each. They control aspects of plotting that do not relate to the physical dimensions of the plot. Included in this category are commands for plotting the accumulated data (PLOT and KEEP), printing out lists of variable,

parameter and data values (VARLIST, PRMLIST and DATLIST), starting a new page (NEWPAGE), etc. None of the keywords has a value associated with it.

**PLOT and KEEP**

Instructs PLT to plot a frame (set of axes) based on the current values of all of the parameters and existing data.

**DATLIST**

Causes PLT to output the contents of the data buffer into a log file.

**PRMLIST**

Instructs PLT to display the current parameter values on the console or output them to a log file.

**VARLIST**

Instructs PLT to display the current user variable values on the console or output them to a log file.

**DATA**

This keyword indicates that the lines following the DATA keyword are the beginning of a new set of data.

**NEWRANGE**

This keyword instructs PLT to begin a new data range within the current frame (axis set).

**NEWFRAME**

Instructs PLT to begin a new plot frame (axis set) on the current page.

**NEWPAGE**

Instructs PLT to advance to a new page.

**FINISH**

Causes PLT to stop taking its input from the current input file.

**RESET**

This keyword causes all parameter values and internal status flags to be reset to their default values, i.e., their initial values when the PLT program begins execution.

**PALETTE**

Enables the user to make modifications in the colors of the basic PLT palette.

**6. Data Value Representations**

Data values may be put into PLT's data buffer in one of four ways: The most common way is to enter X,Y,Z values directly into the pltfile and have PLT interpret them. These values are interpreted such that the first two values correspond to the X- and Y-coordinates of the datum, and the third or Z value, if it exists, corresponds to an error measure for Y that is plotted as an error bar. If *pltype=histogram*, the PLT program interprets the X value as bin height and the Y value as the size of the error bar.

Another common method is to store the data in a file which was created by a program. The data may either be in text form or in binary form. A command is placed in the pltfile which tells PLT to read in the file of data at this point in the plot interpretation and production. (See also Section 7 on "Auxiliary File Commands", below.)

Data may also be entered from the keyboard in one of two ways. If PLT encounters a single asterisk as the only input on a pltfile line, then PLT asks the user to input a single X,Y,Z data value. If the user's terminal is the output device, then the request for input is ignored. A double asterisk (\*\*) causes PLT to request multiple data values, with the user terminating input by entering a blank line, i.e., by typing ENTER or RETURN with no preceding characters.

Valid data values are interpreted automatically by PLT. The user need not precede data values with a DATA keyword. Consider the following 4 data lists:

(1)	(2)	(3)	(4)
DATA	DATA	0	10,10
10,10	*	4	20,20
20,20,3	*	10	**
30	*	5	

Sequence (1) causes three values to be read and put at the end of the data buffer, after its current contents. The first value has an X-coordinate of 10 and a Y-coordinate of 10, with no error bar. The second value has X- and Y-coordinates of 20, with an error bar three units long on either side. The third value has an X-axis

coordinate of 30. The Y-coordinate, since it is unspecified, is set to 0. Sequence (2) causes PLT to accept three data values from the user's keyboard. Sequence (3) causes PLT to read 4 values into the data buffer, assigning them to consecutive X-axis values. Each Y-value is zero. Valid histogram bin values often appear in this form. Sequence (4) reads two data values from the pltfile, then asks the user to supply the remainder.

Data values may also be written as arithmetic expressions including operators for addition, subtraction, multiplication, division and exponentiation (+, -, \*, /, and ^), bearing in mind that no data line may begin with an alphabetic character. The arithmetic expressions may reference parameter values or user variables. Parentheses may be used to group arithmetic expressions and may be required in some cases for proper interpretation by the PLT program; they may also be used to surround initial alphabetic characters in data lines (See Example IV in Chapter 6).

The data may be changed globally by use of the XDATA, YDATA and ZDATA parameters. Data are usually stored in the data buffer with no modification. If, however, the user wishes to offset the data in the Y dimension by some constant amount (1.3 times user variable "a", for example), this can be done by setting **YDATA=\$2 + (1.3\*a)**. The second datum read in from each data line (\$2) is first incremented by 1.3 times the current value of user variable "a" before it is stored in the data buffer.

The original reason for the XDATA, YDATA and ZDATA parameters was to allow alternate column assignments. For example, one might wish to plot an existing data file in which the X and Y data of interest appear in the 5th and 2nd data positions, respectively, on each line of data. This may be done by setting **XDATA=\$5** and **YDATA=\$2**. To avoid having PLT interpret the third column as an error value (which it will do by default), you must set **ZDATA=0**. The PLT program can refer to the input from up to nine columns in this way.

Random jitter may also be added by use of a PLT function, "\$random," which will insert a newly generated random number between 0.0 and 1.0 into the expression. For example, if the user would like the X dimension values of the data to vary randomly within 0.005 data units of the original value, XDATA would be set as follows:

**XDATA=\$1+(\$random-0.50)\*0.01.**

A random number between -0.005 (-0.5 \* 0.01) and 0.005 (+0.5 \* 0.01) is added to

each X dimension value read into the data buffer from the first datum on each data line (\$1).

Data are added to the buffer as they occur in the plfile. It is only when a PLOT or FINISH keyword (see above) is encountered that the data are plotted and the buffer is cleared. If a KEEP keyword is encountered, the data are plotted and retained in the buffer.

## **7. Macros**

Macros are blocks of text which are inserted into the input of PLT program, just as if the user had typed them in. First, the macro is defined, then, it is "invoked" by the user as often as it is needed. There are three predefined macros in the PLT program:

<b>%define</b>	enables the user to define a macro
<b>%undefine</b>	permits the user to remove a previous macro definition
<b>%repeat</b>	enables the user to repeat lines a specific number of times

Macros are very flexible. (1) The invocation may specify parameters which are then substituted in the input by referring to the cardinal number (\$1 for the first, \$2 for the second, etc.) and (2) The macro may call another macro as part of its invocation (it may even call itself).

A macro is defined by delimiting the desired text by, first, a line which begins with a percent sign (%) followed by the word "define" (a predefined macro) plus the macro name and, second, a line which begins with two percent signs (%%). The text between these lines will be processed by PLT each time the macro is invoked. If a third argument appears on the same line as %define, it will be taken as the complete macro definition and the %% symbol will not be necessary. Invoking a macro is done by typing a line which begins with a percent sign (%) following by the name of the macro.

For example, suppose that we want to insert a message at a specific location on our plot. We first define our macro and give it a name:

```
%define DISPMSG ;optional comment <x loc> <y loc> <msg>
$1,$2,"$3"
%%
```

Then, we invoke it as many times as we like (the quotation marks permit us to have a message with spaces but which takes up only one parameter):

```
%DISPMSG 1.3 2.8 "Milly was here."  
%DISPMSG 3.7 4.2 "Sammy was there."
```

This is, of course, a simple example but the macro can be made much more complex.

Sections of PLT text may be repeated several times, without resorting to multiple invocations of a macro, by use of %repeat. This structure is used by bracketing the desired text between a line which indicates the repeat and the number of times (e.g. %repeat 5) and a line with two percent signs (%%), just as in the definition of a macro, above.

For example, the following mini-PLT file will set a user variable to 1 and then increment the variable each time the repeated text (which just plots the symbol at location [a,a]) is run.

```
a=1  
pltype=symbol  
symbol=9  
%repeat 8  
    0+a 0+a  
    plot  
    a=a+1  
%%
```

An alternative to the use of the user variable "a" is to use the parameter \$0, which is set to the current repeat count; 1 for the first, 2 for the second time through, and so forth. The following will produce an identical plot to the section above:

```
pltype=symbol  
symbol=9  
%repeat 8  
    $0 $0  
    plot  
%%
```

The use of macros and repeats will make the function of pltfiles more obvious and the files themselves will look neater. Examples of their use may be found in Chapter 5, "Examples."

## ***8 Auxiliary File Commands***

These commands instruct PLT to transfer input control to another file. Files may be pltfootiles or files containing data stored in binary format (readable by a program, but not by a person).

**CALL filespec****INCLUDE filespec**

Cause PLT to open the pltfootile with the file specification "filespec" and take subsequent input from that file until that pltfootile is exhausted or until a FINISH command is encountered.

**BDATA datafile**

Causes PLT to read data from a binary data file, "datafile."

**STDFNT fontfile****ALTFNT fontfile****SYMFNT fontfile**

Redefines the standard character font, the alternate character font, and the symbol font, respectively, to a font found in the file "fontfile."

## **B. Plot Elements**

The four groups of plot elements will be discussed separately, but some common threads should be noted. There are often parallel parameters which control similar aspects of the different plot elements. For example, the line weight of axes, annotations, data representations, grids, messages and shading will all be controlled by parameters which end in *lwt*, such as *axlwt*, *pltlwt*, *msglwt*, etc. The relative size of plot elements will be controlled by parameters which end in *siz*, such as *labsiz*, *symsiz*, *msgsiz*, etc. The type of data representation or line will be controlled by parameters which end in *type*, such as *lintype*, *pltype*, *grdtype*, etc. There are others, but these are the ones which users will encounter most often.

The default value is shown in brackets next to the name of each parameter.

### ***1. Axis and Annotation Control Parameters***

#### **A. Axis Control**

This group of parameters controls the features of four axis lines, i.e., the X or bottom horizontal axis, the Y or left vertical axis, the R or right vertical axis and the T or top horizontal axis. The parameters are as follows:

**XLEN [6.0]**

This parameter controls the length, in inches, of the horizontal or x axis.

**YLEN [6.0]**

This parameter controls the length, in inches, of the vertical or y axis.

**XLLC [2.2]**

This parameter controls the position of the lower left corner of the plot frame on the plotted page.

**YLLC [1.5]**

This parameter controls the position of the lower left corner of the plot frame on the plotted page.

**XMIN [0] and XMAX [10]**

**YMIN [0] and YMAX [10]**

**RMIN [0] and RMAX [0]**

**TMIN [0] and TMAX [0]**

These parameters denote the minimum and maximum values of the various axes.

**XCYCLE, YCYCLE, RCYCLE and TCYCLE [0.0]**

These parameters cause the specified axis to appear as a logarithmic axis.

**XINT [5], YINT [5], RINT [0], and TINT [0]**

These parameters control the way tick marks defining intervals on the specified (linear) axes will be drawn.

**XPERCENT, YPERCENT, RPERCENT, and TPERCENT [100]**

These parameters control the percentage of the respective axis to be used in plotting the values between MIN and MAX.

**XSHIFT, YSHIFT, RSHIFT, and TSHIFT [0]**

The shift parameters are used to move the range which has been restricted by the "PERCENT" parameter, above, toward one side or the other.

**CORNER [1]**

This parameter controls the printing of the axis corners when the "PERCENT" parameters, above, are set to values less than 100.

### **XBREAK, YBREAK, RBREAK and TBREAK [0]**

The "BREAK" parameters are used to permit discontinuous ranges of data values to be represented on the same axis.

### **TICDIR [0] and TICSIZ [1]**

TICDIR controls the direction that tick marks will be plotted. TICSIZ controls the length of the tick marks.

### **XTICK, YTICK, RTICK and TTICK [1]**

The values of these parameters control tick mark lengths for the specified axis.

### **OPENAX [0]**

This parameter causes PLT to draw only the bottom and left axes, leaving the plot "open".

### **AXLWT [1]**

This parameter controls the line weight (thickness) of axes.

## **B. Annotation and Label Control**

Annotations are the values printed next to each major interval tick mark. PLT adjusts annotations such that they are centered next to tick marks. The default condition for all axes is for annotations to run parallel to the axis. Labels are strings of characters which are automatically placed by PLT next to the desired axis. Parameters in this class are as follows:

### **XANSKP, YANSKP, RANSKP and TANSKP [0]**

These parameters control the manner in which major intervals are annotated (or skipped).

### **ANNSIZ [1]**

ANNSIZ controls the size of the characters used in annotations.

### **XFMT, YFMT, RFMT and TFMT [I]**

These are text string parameters that define annotation formats for the X, Y, R and T axes.

### **XANNOT, YANNOT, RANNOT, and TANNOT [none]**

These parameters are text string parameters which permit the user to substitute strings as annotation for specific values of data.

**YHOR [0]**

Controls the orientation of the Y- and R-axis annotations.

**ANLWT [1]**

Controls the line weight (thickness) of annotations.

**XLABEL, YLABEL, RLABEL and TLABEL [no label]**

Permit the user to set the labels to be plotted next to the specified axis.

**LABSIZ [1]**

Controls the height of the characters which appear in the axis labels.

**LABLWT [1]**

Controls line weight (thickness) of axis labels.

***2. Data Representation Control Parameters***

**A. Data Input Control**

This group of parameters determines how the numbers in the pltfile are interpreted as data values.

**XGAIN, YGAIN and ZGAIN [1]**

**XOFST, YOFST and ZOFST [0.0]**

Enable the user to scale data values as they appear in the plot without changing data values in the pltfile or the source file.

**XDATA [(datum \* XGAIN) + XOFST]**

**YDATA [(datum \* YGAIN) + YOFST]**

**ZDATA [(datum \* ZGAIN) + ZOFST]**

Permit the user to modify the data values in the plot in a variety of ways.

**B. Data Output Control**

This group of parameters determines how the specified data are to be plotted.

**PLTYPE [both symbols and lines]**

Defines the plot type. Eight types of plot exist, namely:

- |                          |                         |
|--------------------------|-------------------------|
| 1 SYMBOLS ONLY           | 5 RECTANGLES            |
| 2 LINES ONLY             | 6 POLYGON               |
| 3 BOTH SYMBOLS AND LINES | 7 WAVES                 |
| 4 HISTOGRAM              | 8 CONTIGUOUS RECTANGLES |

**LINTYPE [0]**

Defines the type of line, dashed or solid, to use in plotting the data (when lines are specified).

**PLTLWT [1]**

Controls the line weight (thickness) of lines which connect data values.

**SYMBOL [1]**

Defines the symbol number to use in plotting data (when use of symbols is specified).

**SOLID [0]**

Controls the filling of symbols.

**SHADE [0]**

Enables the user to create histogram bars, rectangles and polygons that are shaded or filled.

**SHDLWT [1]**

Sets the line weight for the hatch lines in the various shade patterns.

**SHDGRAY [1]**

Sets the background within shaded areas.

**SYMSIZ [1]**

Controls the size of symbols in a plot.

**SYMANG [1]**

Controls the angle for the symbol in degrees counterclockwise.

**EBNUM [2] and EBWIDTH [1]**

Control the presence and dimensions of error bars in a plot.

**GAPSIZ [1]**

Controls size of gap between line segments and symbols when plotting both symbols and lines.

### **3. Message Control Parameters**

Messages provide a means of placing additional text anywhere on the plot as discussed above. Messages are plotted as they are encountered, regardless of whether or not any data values have been plotted. It is for this reason that it is recommended that users define the "frame" parameters (xllc, yllc, xlength and ylength) before plotting any messages; otherwise, the message might be sized and positioned differently from the frame itself.

#### **MXLLC and MYLLC [0]**

Define the X and Y positions of the lower left corner of the first character of the message.

#### **MSGDAT, MXDAT and MYDAT [0]**

Enables the user to place messages according to the values of the data, instead of in inches from the lower left-hand corner.

#### **MSGANGLE [0]**

Controls the angle at which a message is to be plotted.

#### **MSGSIZ [1]**

Controls the size of characters in messages.

#### **MSLANT [0]**

Specifies the italic slant of characters in messages in degrees clockwise from upright.

#### **MHXP [1]**

Controls the width of characters in messages relative to their height (aspect ratio).

#### **MHSP [0] and MVSP [1.64]**

Control the amount of horizontal space inserted between characters and the amount of vertical space inserted between lines, respectively.

#### **MHALIGN [LEFT] and MVALIGN [BASE]**

Determines how the message text will be placed at the desired location (centered, up to the left, down to the right, etc.).

#### **MHKEY [3] and MVKEY [1]**

Control the size of symbols, shade boxes and line samples which have been inserted into messages.

#### ***4. Miscellaneous Parameters***

##### **SIZFAC [0]**

Allows the user to control the "standard" sizes of tick marks, axis labels, annotations, messages, symbols, etc., as well as line weights.

##### **ECHO [no]**

Controls the echoing of each input line as it is read by the PLT program.

##### **GRID [no]**

##### **GRDLWT [1] GRDTYPE [1] GRDSIZ [1]**

Control the presence and appearance of grid lines on a plot.

##### **SMOOTH [no]**

Permits a user to smooth data using a standard algorithm.

##### **SPLINE [no]**

Control the plotting of lines that connect data points as a B-spline curve.

##### **SORT [no]**

Causes the data points to be sorted in order of increasing x value.

##### **CLIP [no]**

Eliminates any lines between data points and any symbols at data points that extend beyond the frame (axis) boundary.

##### **HEADER [yes]**

Controls the plotting of a status line at the bottom of each page indicating the date of the pltfile, the pltfile name and the page number.

##### **MAGNIFY [1]**

Scales the entire plot page by the ratio specified.

##### **ROTATE [no]**

Permits the user to rotate the entire plot 90 degrees.

##### **VXLLC and VYLLC [0]**

Allow the entire plot to be shifted relative to the lower left corner of the page.

**WXLEN [10.5] and WYLEN [8]**

Determine the page boundary window at which clipping occurs.

**5. Color Parameters**

Colors may be used to display plots on color monitors (PC or NeXT only). Color output may also be obtained in the output to files in CGM (Metafile) or PostScript file format. For pen plotters (Tektronix or HPGL output) the color index will select the pen number.

**FGCOL and BGCOL [-1]**

Set the foreground and background colors, respectively, for the plot.

**PLTCOL, AXCOL, ANNCOL, LABCOL, GRDCOL, SHDCOL and MSGCOL [-1]**

Set the colors of plotted data, axes, annotation, label, grid lines, shading and messages, respectively.

**FNCOL [-1]**

Sets the color of the upper part of the background of the plot which will shade gradually and evenly into the color specified in BGCOL.

## **Chapter 4.**

# **User Reference**

This section of the User's Guide provides an alphabetical listing of all the PLT parameters and keywords with a detailed description of each. If you are unfamiliar with PLT parameters, you should read Chapter 2, "Tutorial," and Chapter 3 "Functional Description," first. Also included in this chapter are figures describing various display elements:

- 4-1. Standard and Alternate Fonts
- 4-2. Line Weight Values
- 4-3. Symbols and Line Types
- 4.4. Area Fill Patterns
- 4.5. Message Alignments

These are collected at the end of the chapter beginning on page 4-26.

**ALTFNT** fontfile

See **STDFNT**.

**ANNCOL [-1]**

See **PLTCOL**.

**ANNLWT [1]**

This parameter controls the line weight (thickness) of annotations by modifying the default value of one. Values between 0 and 1 make the line weight lighter than the default; values greater than 1 make the line weight heavier. Values less than zero set the line weight to a fixed size, corresponding to the weights shown in Figure 4-2, page 4-26. Setting ANNLWT to zero will completely eliminate annotations.

**ANNSIZ [1]**

ANNSIZ controls the size of the characters used in annotations. If it is set to 0 or to a positive value, it acts as a multiplier that operates on the internally-computed annotation character size. The default value is 1. Values greater than one produce larger-than-normal annotations, and values less than one produce proportionally smaller annotations. Using a negative value for ANNSIZ signals the PLT program to override internal computation of annotation character size and to use the absolute value of ANNSIZ to set the height of the annotation characters in inches. For example, setting ANNSIZ=-0.75 will cause annotation characters to be plotted 0.75 inches high.

**AXCOL [-1]**

See **PLTCOL**.

**AXLWT [1]**

This parameter controls the line weight (thickness) of axes by modifying the default value of 1. Values between 0 and 1 make the line weight lighter than the default; values greater than 1 make the line weight heavier. Values less than zero set the line weight to a fixed size, corresponding to the weights shown in Figure 4-2, page 4-26. Setting AXLWT=0 will completely eliminate axes.

**BDATA** datafile

PLT can also read binary data files using the BDATA command. The "datafile" contains integer data values preceded by a short header. The header contains four, two-byte integers (least significant byte first). (1) A "magic number" ( $12345_{10} = 3039_{16} = 30071_8$ ) for identification as a PLT binary data file, (2) the number of bytes per integer in the data which follow (must be 2 bytes/integer), (3) the number of coordinate values (x,y,z) specified for each data point (must be 1, 2, or 3 and all data points must have the same number of coordinate values), and (4) the number of data points in the file. The default filename extension is ".plt".

**BGCOL [-1]**

See **FGCOL**.

**CALL** filespec

**INCLUDE** filespec

These two commands are functionally equivalent and may be used interchangeably. They cause PLT to open the pltfile with the file specification "filespec" and take subsequent input from it, continuing until that pltfile is exhausted or until a FINISH command is encountered. When this happens, the CALLEd pltfile is closed, and control reverts back to the original pltfile at the line following the CALL command. INCLUDED or CALLEd files may be nested to a depth of 15. Because the default filename extension is .PLT, a CALLEd file should have an extension. If it does not, the "filespec" must end with a period to indicate that there is no extension.

**CLIP [no]**

Setting CLIP to YES or 1 causes the lines between data points and the symbols at data points to be clipped at the frame (axis) boundary. The default action is to clip at the page boundary.

**CORNER [1]**

This parameter controls the printing of the axis corners when the XPERCENT, YPERCENT, RPERCENT or TPERCENT parameters are set to values less than 100. This will produce an "open-cornered" plot when both XPERCENT and YPERCENT are set less than 100 and CORNER is set to 0 or to "NO".

**DATA**

This keyword indicates that the lines following the DATA keyword are the beginning of a new set of data. This line is not needed by the PLT program, because lines which can be interpreted as data values are read in as such. The DATA keyword will clear any previously accumulated data from the data buffer.

**DATLIST**

This keyword causes PLT to output the contents of the data buffer. If the user's terminal is the output device, the listing of data values is stored in the log file (*plt.log*). If output is generated for another output device and redirected, the data list will be displayed on the user's console. Because the data buffer is cleared as soon as keyword PLOT is encountered, entering DATLIST after PLOT will produce a "\*\*\* Data Buffer Empty \*\*\*" message.

**%define** macroname

This enables the user to define a macro. The macro is terminated by a line with the characters "%%". The PLT keywords and parameters between the %define and the terminator will be

processed each time the macro is invoked. The macro is invoked by a line such as "%macroName". For a fuller explanation see the section on macros in Chapter 3, "Functional Description".

### **EBNUM [2]**

EBNUM, when set to zero, will suppress the plotting of error bars. A value of 1 produces only the upper portion of the error bar. The default value of EBNUM=2 produces both the upper and lower portion of the error bar. EBNUM=-1 produces only the lower portion of the error bar.

### **EBWIDTH [1]**

EBWIDTH modifies the default error bar line widths as a scale factor. When EBWIDTH=0, no caps appear on the error bars. When EBWIDTH=2, the caps will be twice as wide. When EBWIDTH is set to values less than zero, an absolute line width is chosen. The default value is 1. See **EBNUM**.

### **ECHO [no]**

Setting this parameter to a non-zero value (or to Yes, Y or y) will cause PLT to echo each input line from the user's file as it is read. If the console is being used for graphical display, then the echoed output will be written to the log file (*plt.log*).

### **FGCOL and BGCOL [-1]**

Sets the foreground and background colors, respectively, for the plot. Even though the screen may not be in color, any output to files will contain the correct color information. Negative values indicate the default colors for the device (either black on white or white on black). The default color palette is:

0	Black	8	Dark Gray
1	Blue	9	Light Blue
2	Green	10	Light Green
3	Cyan	11	Light Cyan
4	Red	12	Light Red
5	Magenta	13	Light Magenta
6	Brown	14	Yellow
7	Light Gray	15	White

These colors are the same as the standard EGA colors on a PC. The color palette may be changed by using the PALETTE command.

### **FINISH**

PLT treats the FINISH keyword the same as an end-of-file condition. If a FINISH keyword is encountered in a pltfile that has been CALLED or INCLUDED from the main pltfile, then transfer returns to the calling pltfile, and the CALLED pltfile is closed. If a FINISH keyword is encountered in the main pltfile the program will plot any accumulated data before terminating execution.

FINISH may appear anywhere in the plfile that the user wishes input processing to be halted. Its presence, however, is not required.

### **FNCOL [-1]**

Sets the colors of the top color for the background of the plot. This color will shade gradually and evenly into the color specified in BGCOL. (This is termed a "fountain" in the printing trade; hence, the name FNCOL.) Even though the screen may not be in color, any output to files in CGM format will contain the correct color information for printing, making slides, etc. A negative value indicates "no fountain." (See **FGCOL** for information on the colors in the palette.)

### **GAPSIZ [1]**

This parameter controls size of gap between line segments for plotting symbols when PLTYPE=BOTH SYMBOLS AND LINES. The default value is 1, which leaves a gap equal to the height or diameter of the symbol. Negative values are interpreted to generate gaps corresponding to the absolute value of GAPSIZ in inches.

### **GRDCOL [-1]**

See **PLTCOL**.

### **GRDLWT [1]**

See **GRID**.

### **GRDTYPE [1]**

See **GRID**.

### **GRDSIZ [1]**

See **GRID**.

### **GRID [no]**

#### **GRDLWT [1] GRDTYPE [1] GRDSIZ [1]**

Setting this parameter to a non-zero value (or to Yes, Y or y) will cause PLT to plot grid lines at each major interval. If a right axis has been specified that differs from the left axis, or if OPENAX has been set to a non-zero value, then the grid will not appear. The default condition is GRID=0 (or NO, N, no or n). The parameter GRDLWT [1] controls the thickness of the line used in plotting the grid. It takes on values just as PLTLWT does. GRDTYPE [1] is used as LINTYPE is used and permits choosing a line which is solid (0) or dotted (1) or broken (>1). GRDSIZ [1] permits the user to control the appearance of broken lines used in plotting the grid, just as in the case of LINSIZ.

**HEADER [yes]**

If HEADER=1 or HEADER=YES, a status line will be plotted at the bottom of each page indicating the date of the pltfile, the pltfile name and the page number. If the pltfile date is unavailable, the current date and time will be substituted.

**INCLUDE filespec**

See **CALL**.

**KEEP**

See **PLOT**.

**LABCOL [-1]**

See **PLTCOL**.

**LABLWT [1]**

This parameter controls line weight (thickness) of axis labels by modifying the default value of one (see Figure 4-2, page 4-26). Values between 0 and 1 make the line weight lighter than the default; values greater than 1 make the line weight heavier. Values less than zero set the line weight to a fixed size, corresponding to the weights shown in Figure 4-2, page 4-26. Setting LABLWT to zero will completely eliminate labels.

**LABSIZ [1]**

The default label character height is computed internally by PLT, and is a function of the average length of the X and Y axes. The user may modify this by adjusting LABSIZ. When LABSIZ is positive or zero, it acts as a multiplier that operates on the default label size. A zero value produces no axis labels, and values greater or less than one produce proportionally larger-than-normal or smaller-than-normal character heights for axis labels. When LABSIZ is negative, the internal computation of label height is overridden, and the label character height used (in inches) is the absolute value of LABSIZ. For example, if the user desires labels that are 1/2 inch tall, then the setting should be LABSIZ=-0.5 .

**LINSIZ [1]**

This parameter permits dashed or broken lines to be "magnified" by the ratio specified. Thus, the line will contain the same relative amount of black and white, but the segments will be longer and spaced further apart. Used in conjunction with **LINTYPE**.

**LINTYPE [0]**

This parameter defines the type of line, dashed or solid, to use in plotting the data (when lines are specified). Values greater than 1 designate dashed lines and determine the length of the dashes, with larger values generating longer dashes. A value of 1 produces dotted lines. Solid lines (the default)

are designated by setting LINTYPE equal to 0 (see Figure 4-3, page 4-26). For dashed lines, as a rule of thumb, when the value of LINTYPE is greater than 1, the ratio of black to white is approximately equal to the value of (LINTYPE - 1). See also **LINSIZ**.

### **MAGNIFY [1]**

Scales the entire plot page by the ratio specified. The default value is 1.

### **MHALIGN [LEFT] and MVALIGN [BASE]**

Determines the location of MXLLC and MYLLC relative to the message text. The user may specify the message's location by a position other than the lower left corner.

Possible values include the following (numerical equivalents in parentheses):

**MHALIGN = LEFT (1), CENTER (2), or RIGHT (3).**

**MVALIGN = TOP (1), CAP (2), HALF (3), BASE (4), or BOTTOM (5).**

The 15 possible message alignments are illustrated in Figure 4-5, page 4-26. The default alignment is MHALIGN=LEFT and MVALIGN=BASE. Using numerical values of 0 will cause the alignments to be set to their defaults.

### **MHKEY [3] and MVKEY [1]**

These parameters control the size of symbols, shade boxes and line samples which have been inserted into messages and axis labels. Positive values indicate a multiple of the character height and negative values indicate the absolute size in inches. MHKEY controls the horizontal size of shade boxes and the length of line samples. Its default value is MHKEY=3 giving shade boxes and line samples which are three times as wide as the character height. MVKEY controls the vertical size of shade boxes and the diameter of symbols. Its default value is MVKEY=1.

### **MHSP [0] and MVSP [1.64]**

MHSP controls the amount of extra space inserted between characters, and MVSP controls the amount of vertical space between lines in messages and axis labels. Both parameters are multiplied by the current character height to determine the actual amount of space inserted. The default value of MHSP is 0. A value of MHSP=0.5 will leave extra space between characters equal to one-half the character height. The default value of MVSP is 1.64. A value of MVSP=2 will cause the vertical space between successive message lines to be twice the character height.

### **MHXP [1]**

This parameter controls the width of characters in messages relative to their height (aspect ratio). The default value is 1. Values larger than 1 result in wide characters and values smaller than 1 result in narrow characters.

### **MSGANGLE [0]**

This parameter controls the angle at which a message is to be plotted. This angle is measured in degrees that increase in a counterclockwise direction from the X axis (0°). Thus, if MSGANGLE=90, the message is oriented upward and if MSGANGLE=180, the message is upside-down. The default value is 0.

### **MSGCOL [-1]**

See **PLTCOL**.

### **MSGDAT, MXDAT and MYDAT [0]**

Setting this parameter to a non-zero value, or to YES, yes, or y, instructs PLT to interpret MXLLC and MYLLC as an (X,Y) coordinate relative to the axis values. For example, if the X and Y axes are both 6 inches long and both go from 0 to 100, and if MSGDAT is set to YES, then a MXLLC/MYLLC combination of 50 and 50 will cause the lower left corner of the message to begin at the point where a data value of (50,50) would be positioned, i.e., three inches up from the X-axis line and three inches to the right of the Y-axis line. MSGDAT has a default value of 0 or NO. MXDAT and MYDAT work the same way as MSGDAT except that they apply to the X and Y coordinates independently.

### **MSGSIZ [1]**

MSGSIZ is a multiplier that controls the size of characters in messages. This size is determined relative to a "standard" message character size generated internally by PLT. Values less than 1 produce smaller-than-normal messages and, values greater than 1 produce larger-than-normal messages. The default value for MSGSIZ is 1. When MSGSIZ is negative, internal computation of annotation character size is overridden and the height of the annotation characters (in inches) is the absolute value of MSGSIZ.

### **MSLANT [0]**

MSLANT specifies the italic slant of characters in messages in degrees (clockwise) from upright. The default value is 0.

### **MVALIGN [BASE]**

See **MHALIGN**.

### **MVKEY [1]**

See **MHKEY**.

### **MVSP [1.64]**

See **MHSP**.

### **MXDAT [0]**

See **MSGDAT**.

### **MXLLC and MYLLC [0]**

There are two ways to place messages on the plot. You may plot the message at specific distances in inches from the x and y axes (**MSGDAT=0**) or at a location corresponding to the x-y coordinates of a data value (**MSGDAT=1**). The values of **MXLLC** and **MYLLC** are in inches relative to the lower left corner of the axes, that is, from the point where the X-axis line meets the Y-axis line as defined by **XLLC** and **YLLC**. These values, by default, define the X and Y positions of the lower left corner of the first character of the message, but the alignment of the message relative to the **MXLLC**/**MYLLC** position may be modified using **MHALIGN** and **MVALIGN**. Messages may be positioned anywhere, even outside the plot boundaries.

The values of **MXLLC** and **MYLLC** may be set explicitly or they may be set by placing these values immediately in front of the opening quotation mark of the message. For example, the following line

```
0.5,0.8,"This is a message."
```

sets **MXLLC=0.5** and **MYLLC=0.8**, prior to plotting the message.

### **MYDAT [0]**

See **MSGDAT**.

### **MYLLC [0]**

See **MXLLC**.

### **NEWFRAME**

This keyword instructs PLT to begin a new plot frame on the current page. The plot frame is the term used for a set of axes (X, Y, right and top). A plot may consist of a number of sets of axes. **NEWFRAME** is the command given when the user wants to end one plot frame and put another plot frame on the same page. This enables the user to create plots with multiple components. See Example II in Chapter 6.

### **NEWPAGE**

**NEWPAGE** instructs PLT to advance to a new page. This permits a user to combine several plots into one PLT text file. If the console screen is being used for output, an audible signal sounds as each page is finished. When this signal is heard, the user must type a carriage return in order to have PLT erase the screen and continue.

**NEWRANGE**

This keyword instructs PLT to begin a new data range within the current frame. **NEWRANGE** is the command given when the user wants to plot more than one data range on the same axis. PLT will not accept new values for the MIN and MAX parameters of the X and Y axes unless this keyword appears. See Example VI in Chapter 6.

**OPENAX [0]**

The default plot has four sides to it. Tick marks are placed on all four axes, and in the case where only X and Y axes are defined, the tick marks on the right (R) and top (T) axes are duplications of those on the bottom (X) and left (Y) axes. In some cases, however, you might wish to draw only the bottom and left axes, leaving the plot "open". Setting **OPENAX** to a non-zero value (or to YES, yes or y) allows the user to do this. The user may still specify a right axis, even though **OPENAX** is non-zero, and the right axis will appear in its usual place, leaving only the top of the plot open.

**PALETTE**

This keyword causes all x, y, z values which have been buffered to be interpreted as the red, green and blue (RGB) components of a new color palette (that is, the selection of colors to be used from the range available on the display in use). The value of each component should be in the range 0 to 1. For example, in order to change a palette position to pure red, the user would type a line consisting of the numbers 1, 0, 0. The palette position would have 100% red, 0% green and 0% blue.

In order to change the first three palette positions to green, blue and red, in that order, the user would merely have to enter the following:

```
data
0 1 0
0 0 1
1 0 0
palette
```

**PLOT and KEEP**

This instructs PLT to plot a frame (set of axes) based on the current values of all of the parameters and existing data. After a **PLOT** command has been given, the data buffer is cleared. **KEEP** is the same as **PLOT** except that the data are retained for subsequent plotting.

**PLTCOL, AXCOL, ANNOL, LABCOL, GRDCOL, SHDCOL and MSGCOL [-1]**

Sets the colors of plotted data, axes, annotation, label, grid lines, shading and messages, respectively. Even though the screen may not be in color, any output to files will contain the correct color information. A negative value indicates that the foreground color (**FGCOL**) should be used. (See **FGCOL** for information on the colors in the palette).

**PLTLWT [1]**

This parameter controls the weight (thickness) of lines which connect data values by modifying the default value of 1 (See Figure 4-2, page 4-26). It also controls the outlines of rectangles and polygons as well as the width of the line which draws the PLT symbols. Values between 0 and 1 make the line weight lighter than the default; values greater than 1 make the line weight heavier. Values less than zero set the line weight to a fixed size, corresponding to the weights shown in Figure 4-2, page 4-26. If PLTLWT is set to zero, the narrowest possible line will be used for plotting.

**PLTYPE [both symbols and lines]**

This parameter defines the plot type. Eight types of plot exist, namely:

- |                          |                         |
|--------------------------|-------------------------|
| 1 SYMBOLS ONLY           | 5 RECTANGLES            |
| 2 LINES ONLY             | 6 POLYGON               |
| 3 BOTH SYMBOLS AND LINES | 7 WAVES                 |
| 4 HISTOGRAM              | 8 CONTIGUOUS RECTANGLES |

The default plot type is Type 3 (both symbols and lines). The user may define PLTYPE with words or a numeral. For example, the commands

```
PLTYPE=symbols
```

```
PLTYPE=SYMBOL
```

```
PLTYPE=s
```

```
PLTYPE=1
```

all produce the same result. In order to use a text argument, the user need only provide the first letter, either upper or lower case, with S designating "symbols only", L designating "lines only", B designating "both symbols and lines", H designating "histogram", R designating "rectangles", P designating "polygons", W designating "waves" and C designating "contiguous rectangles".

For PLTYPE=1, 2, and 3, data values are read in line by line. For each line the first column corresponds to the X dimension, the second column to the Y dimension, and the third column to the length of an error bar in units of Y. (This ordering can be modified. See Section A.5, Chapter 3, "Functional Description" for more information on data entry. See also **XCYCLE** for information on special problems presented by logarithmic axes.)

A "histogram" plot will cause the data to be plotted in bar-graph form along the X axis, using as many bars as there are data values. (Note that empty histogram bins must be accounted for.) The second data value on each line, if there is one, defines an error value to be indicated by an error bar which will appear on the upper part of each bar of the histogram.

The "rectangles" plot type takes two pairs (one pair on each of two lines) of x,y data values as the corners of a rectangle. The order doesn't matter. The upper left may be first, then, the lower right; lower left, upper right; etc.

The "polygon" data type takes all x,y pairs as corners of a polygon. This permits the user to construct some rather complex structures.

The "waves" data type permits the user to use Y dimension values only. The PLT program will automatically index these values in the X dimension by dividing the X axis interval into a number of

equal distances according to the number of Y values. The X dimension data will be stored and plotted automatically.

The "contiguous rectangles" data type is very similar to the rectangle data type. The difference between the two is that adjoining rectangles are plotted by adding a data line which corresponds to the corresponding corner of the next rectangle. For example, if the user were to type the following:

```
0 0
1 2
1.5 1.5
2.3 .8
```

the PLT program would plot three adjoining rectangles. The first would be plotted just as if the PLTYPE were set to "rectangle" and the first two data lines were encountered. The next rectangle will be plotted as if PLTYPE were set to "rectangle" and the corners were set to (1,0) and (1.5,1.5). The third would be plotted in the same way with its corners at (1.5,0) and (2.3,.8). This data type just makes it a little easier for a user to create variable width histograms. It could be used to plot a histogram in which the width of each bin were proportional to some controlling parameter.

### **PRMLIST**

This keyword instructs PLT to display the current parameter values. If the user's terminal is the output device, the listing of data values is stored in the log file (*plt.log*). If output is generated for another output device and redirected, the data list will be displayed on the user's console.

### **RANNOT [none]**

See **XANNOT**.

### **RANSKP [0]**

See **XANSKP**.

### **RBREAK [0]**

See **XBREAK**.

### **RCYCLE [0.0]**

See **XCYCLE**.

### **%repeat** number

This macro enables the user to repeat lines a specific number of times. The %repeat macro, like the %define macro, is terminated by a line with the characters "%%". The PLT keywords and parameters between the %repeat line and the terminator will be processed exactly the number of times specified by "number". The "number" may also be indicated by a parameter value or by the value of a user variable. (See also %define.) For a fuller explanation see the section on macros in Chapter 3, "Functional Description".

## **RESET**

This keyword causes all parameter values and internal status flags to be reset to their default values, i.e., their initial values when the PLT program begins execution.

## **RFMT [I]**

See **XFMT**.

## **RINT [0]**

See **XINT**.

## **RLABEL [no label]**

See **XLABEL**.

## **RMAX [0]**

See **XMIN**.

## **RMIN [0]**

See **XMIN**.

## **ROTATE [no]**

Setting **ROTATE** to **YES** or 1 rotates the plot by 90 degrees so that the x dimension is oriented along the short edge of the page and the y dimension is oriented along the long edge. Rotate may also be set to 2 or 3 for 180 and 270 degree rotation. If **ROTATE**=1 or to 3, the user must set the parameter **WYLEN**=10.5 in order to use the full height of the rotated page.

## **RPERCENT [100]**

See **XPERCENT**.

## **RTICK [1]**

See **XTICK**.

## **SHADE [0]**

This parameter, when non-zero or **YES**, will cause histogram bars, rectangles and polygons to be shaded or filled. A value of one specifies solid (black) shading. Values from 2 - 39 specify hatch patterns (See Figure 4-4, page 4-26). Not apparent in Figure 4-4 is the fact that **SHADE**=20 causes the background (set by **SHDGRAY**) to be filled prior to drawing the outline (on devices that support such filling). In the range of values 0 - 39, only **SHADE**=20 has a filled background. **SHADE**=40 will plot a **SHDGRAY** background with no outline. Values in the range 41 - 79 will give the same

patterns as 1 - 39, but the background will be filled with a gray level corresponding to **SHDGRAY** before drawing the pattern.

### **SHDCOL [-1]**

See **PLTCOL**.

### **SHDGRAY [1]**

This parameter sets the background within shaded areas to a contrast value ranging from 0 (black) to 1 (white) through the values of gray in between. This parameter applies when areas are filled before shading (SHADE=40 through SHADE=79).

### **SHDLWT [1]**

This parameter sets the line weight for the hatch lines in the various shade patterns (See **SHADE**). Values between 0 and 1 make the line weight lighter than the default; values greater than 1 make the line weight heavier. Values less than zero set the line weight to a fixed size, corresponding to the weights shown in Figure 4-2, page 4-26. If SHDLWT is set to zero, the narrowest possible line will be used for plotting.

### **SIZFAC [0]**

Internally, PLT computes "standard" character sizes, line weights, symbol heights, etc., on the basis of the average axis length, i.e., half the sum of XLEN and YLEN. This is done so that plots retain the same aspect ratios as they differ in size.

SIZFAC allows the user to control the "standard" sizes of tick marks, axis labels, annotations, messages, symbols, etc., as well as line weights. It is used, primarily, to standardize aspects of multiple-panel plots in situations where the user desires consistency in the way individual panels are plotted, even though they may be of different size.

SIZFAC may take on three types of values. A negative value instructs PLT to use, for the current frame (set of axes), the standard sizes computed for the previous plot frame, regardless of the current plot frame's size. Thus, setting SIZFAC to, say, -1, will cause the annotation height, tick mark length, axis label height, etc., from the previous frame to be retained for the current frame.

Setting SIZFAC to a positive value instructs PLT to compute the various sizes, line weights, etc., as if the average axis length were equal to SIZFAC. Thus, if SIZFAC=7, then the various sizes, weights and lengths are determined as though the average axis length were 7, as it would be for a plot that was 6x8 or 5x9, even though the actual average axis length is something else.

The default value, SIZFAC=0 or NO, causes PLT to re-compute the various lengths and heights based on the current average axis length.

### **SMOOTH [no]**

Smoothing of data using the ACM 3-point smoothing algorithm. Three-point weighting is (1/3, 1/3, 1/3) except at the end point where it is (5/6, 1/3, -1/6). The default is NO.

**SOLID [0]**

This parameter controls the filling of symbols. If SOLID is non-zero or YES and the symbol is "open", PLT will choose the appropriate filled symbol to use, if one exists. If SOLID is set to -1, the open symbols will not be filled in with white. This permits the user to display a symbol over previously plotted material and have the old material show through the new symbol. The default value is SOLID=0 or NO.

**SORT [no]**

Setting SORT to YES or 1 causes the data points to be sorted in order of increasing x value. The default action is not to sort the data points.

**SPLINE [no]**

When this parameter is nonzero and PLTYPE=2 or 3, a B-spline curve\* will be drawn instead of connecting the data points. At least 4 data points are required to compute the spline curve. The curve will approximate all data points except the end points (first and last). The value of this parameter determines the number of points (and, consequently, the smoothness) of the spline curve. If SPLINE has a positive value, then a corresponding number of positions will be computed on the spline curve for each interval between data points. If SPLINE has a negative value, then it determines the absolute (total) number of positions used to represent the spline.

\*Foley, J. D. and VanDam, A. *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, 1982, p. 521

**STDFNT** fontfile

**ALTFNT** fontfile

**SYMFNT** fontfile

PLT can use external files to redefine the normal text characters and symbols. STDFNT redefines the standard character font, ALTFNT redefines the alternate character font, and SYMFNT redefines the symbol font. The fontfiles are created and modified by a separate program called PLTFNT. If PLT encounters one of these keywords without a fontfile argument, then the corresponding font reverts to its internal definition. The default filename extension is ".pft". See Appendix C for more information on using external font files in PLT.

**SYMANG [1]**

SYMANG specifies a rotation angle for the symbol in degrees (counter-clockwise) relative to normal orientation.

**SYMBOL [1]**

This parameter defines the symbol number to use in plotting data (when PLTYPE=1 or 3). There are 30 symbols available, of which 10 are filled and 10 are open. The codes for the symbols are listed in the table below (See also Figure 4-3, page 4-26).

Note that the numbers for filled symbols 10-15 are equal to their unfilled values plus ten, and the numbers for filled symbols 16-19 are equal to their unfilled values minus ten. The default symbol number is 1 (open circle).

Symbol	Shape	Symbol	Shape	Symbol	Shape
0	Open square	10	Filled square	20	Ray-box
1	Open circle	11	Filled circle	21	Circle-dot
2	Open erect triangle	12	Filled erect triangle	22	Happy Face
3	Open inverted triangle	13	Filled inv. triangle	23	Sad Face
4	Open diamond	14	Filled diamond	24	Asterisk-6
5	Hourglass	15	Filled hourglass	25	Dot
6	Asterisk-8	16	Filled pentagon	26	Open pentagon
7	Plus sign	17	Filled hexagon	27	Open hexagon
8	"Picnic table"	18	Filled Star-5	28	Open Star-5
9	Up-pointing arrow	19	Filled Star-6	29	Open Star-6

The symbols which are termed "open" in the above table (0-5 & 26-29) are actually filled with white on most output devices to improve the appearance of overlapping symbols (See **SOLID**).

**SYMFNT** fontfile

See **STDFNT**.

**SYMSIZ** [1]

This is a multiplier for symbol size. A default symbol size is determined internally, and it varies with the size of the plot or with the value of SIZFAC. Half-normal-size symbols may be produced by setting SYMSIZ to 0.5, and double-size symbols may be produced by setting SYMSIZ to 2. When SYMSIZ is negative, the internal computation of symbol size is overridden and the symbol height (in inches) is the absolute value of SYMSIZ.

**TANNOT** [none]

See **XANNOT**.

**TANSKP** [0]

See **XANSKP**.

**TBREAK** [0]

See **XBREAK**.

**TCYCLE** [0.0]

See **XCYCLE**.

**TFMT [I]**

See **XFMT**.

**TICDIR [0]**

TICDIR controls the direction that tick marks will be plotted, i.e., inward from the axis frame (set of axes) or outward from the frame. Setting TICDIR to 0 (the default value) indicates outward ticks, and setting it to a non-zero value indicates inward ticks. Inward ticks are 1.5 times longer than outward ticks. For convenience to users, PLT recognizes INWARD and OUTWARD as valid entries for TICDIR. Thus, these lines all have the same result:

```
TICDIR=0 .  
TICDIR=out  
TICDIR=OUTWARD
```

**TICSIZ [1]**

TICSIZ controls the length of the tick marks as a scale factor. Its default value is 1. Negative values of TICSIZ set the absolute length of major tick marks in inches. Minor tick marks are always half the length of major tick marks. The TICSIZ parameter also controls the size of the break symbol (see **XBREAK**).

**TINT [0]**

See **XINT**.

**TLABEL [no label]**

See **XLABEL**.

**TMAX [0]**

See **XMIN**.

**TMIN [0]**

See **XMIN**.

**TPERCENT [100]**

See **XPERCENT**.

**TTICK [1]**

See **XTICK**.

**%undefine** macroname

This enables the user to remove a previous macro definition. (See also **%define**.) For a fuller explanation see the section on macros in Chapter 3, "Functional Description".

**VARLIST**

This keyword instructs PLT to display the current user variable values. If the user's terminal is the output device, the listing of data values is stored in the log file (*plt.log*). If output is generated for another output device and redirected, the data list will be displayed on the user's console.

**VXLLC and VYLLC [0]**

Allows the entire plot to be shifted relative to the lower left corner of the page. Indicates the offset (in inches) of the PLT page relative to the physical page. Default values are 0.

**VYLLC [0]**

See **VXLLC**.

**WXLEN [10.5] and WYLEN [8]**

Determines the page boundary window at which clipping occurs. Default values are WXLEN=10.5 and WYLEN=8 (inches). It may be desirable to change these values when the plot is rotated.

**WYLEN [8]**

See **WXLEN**.

**XANNOT, YANNOT, RANNOT, and TANNOT [none]**

These parameters are text strings which list the annotations desired, together with the data value for which the substitution is to be made. The parameter consists of a string with a set of annotation substitutions separated by spaces. Each substitution consists of a data value (which may be an expression) and the string to be substituted for it. For example, XANNOT=1,Su 2,Mo 3,Tu 4,We 5,Th 6,Fr 7,Sa will place the abbreviation for each day of the week at the position on the X axis corresponding to the data value to its left.

The substitution string is plotted exactly as it appears in the list. The substitution strings may contain newline characters (`\n`) as well as spaces (use a `$` to force a space); they may also be eliminated as in this line:

```
XANNOT=1 2 3 4 5
```

The PLT program will replace the X data value listed with the number itself. This is useful in logarithmic plots when a user wishes to have the minor interval tick marks annotated. It is also useful when a user wishes to have a mixture of formats in the annotation. For example, there is no

method other than the use of the XANNOT parameter to have PLT produce a mixture of decimal and whole number annotations as in the following:

```
XANNOT=1 1.5 2 2.5 3 3.5 4 4.5 5
```

These parameters are also useful in cases when multiple ranges are plotted on a single axis (See Example VI in Chapter 6). When they are present, annotation strings completely override the usual automatic generation of annotations.

### **XANSKP, YANSKP, RANSKP and TANSKP [0]**

These parameters control the manner in which major intervals are annotated. Each is a skip factor whose values range from zero upward. A value of zero indicates that no annotations are to be skipped, that is, that all annotations are to be labeled. A value of one indicates that every other annotation is to be labeled, and so on. A value less than zero indicates that no annotations are to be produced, i.e., the axis has tick marks, but has no annotation values adjacent to the tick marks.

The annotation skip parameters may also be specified in the form MM.NN, where MM indicates the number skipped between annotations and NN represents the number of annotations skipped at the beginning of the axis. For example, XANSKP=1.1 causes the first annotation on the X-axis to be skipped and the second, fourth, sixth, etc. to be plotted.

### **XBREAK, YBREAK, RBREAK and TBREAK [0]**

These break parameters are used when two discontinuous ranges of data values are being represented on the same axis. The axis line is broken with two short diagonal strokes at the place which is the percentage distance along the axis indicated by the value of the parameter. For example a value of YBREAK=75 will position the break three quarters of the distance up the Y axis. The default value of 0 indicates no break in the axis. No more than one axis break is permitted on each axis. The plotted size of the break symbol is controlled by the TICSIZ parameter. (If a user wishes to change the size of the X axis tick marks, without affecting the size of the break symbol, modifying the XTICK parameter would permit it.) The BREAK parameters may be used in conjunction with the NEWRANGE keyword. See Example VI in Chapter 6.

### **XCYCLE, YCYCLE, RCYCLE and TCYCLE [0.0]**

These parameters, when non-zero, denote that the indicated axis is to be logarithmic and that it is to have that many cycles. The defaults are 0.0 for all three, which means that all four plottable axes are linear.

When a logarithmic axis is to be specified, the user need only set the number (whole or fractional) of cycles desired and either the minimum or maximum axis value. The user doesn't have to compute the number of cycles when both the minimum and maximum axis values are specified; the user need only make the number of cycles non-zero. For example, the following sequences

(1)	(2)	(3)	(4)
XMIN=2	XMIN=2	XMIN=0	XMIN=2
XMAX=200	XMAX=0	XMAX=200	XMAX=200
XCYCLE=2	XCYCLE=2	XCYCLE=2	XCYCLE=1

all have the same result. Sequence (1) defines all three parameters in a consistent manner. Sequences (2) and (3) have enough information to generate the logarithmic axis. In these cases the missing parameter is calculated and used internally. Sequence (4) has an inconsistency--XCYCLE does not agree with the axis limits. In this case, the limits XMIN and XMAX take precedence and XCYCLE will be corrected internally to the proper value, which is 2.

There are two things the user should remember about logarithmic axes. First, when defining only two logarithmic-axis parameters, the third parameter must be 0. If the "missing" parameter is non-zero, perhaps having been set in processing a previous plot frame (set of axes) or having a non-zero default value (XMAX), then incorrect and possibly disastrous results will occur. Second, it is valid to set the number of cycles to a fractional amount such as 2.50 or 3.10. In the case where only XMIN or XMAX is defined, PLT will generate the non-integral number of cycles specified.

Normally, each cycle of a log axis represents a multiple of 10. As a special case, the log axis can be made to have cycles which are multiples of 2 by setting the corresponding interval parameter (XINT, YINT, RINT, or TINT) equal to -2. Any other value for the interval parameter causes the log cycles to be multiples of 10.

The method by which error bar lengths are calculated on a logarithmic Y axis is different from that which is used in a linear axis. In the linear case, the Y position of the top of the error bar is calculated by simply adding the amount of error to the Y value of the datum (the bottom is found by subtraction). In the logarithmic case, the Y position of the top of the error bar is calculated by multiplying the amount of the error by the Y position of the datum (the bottom is found by division).

This difference might cause problems. For example, if you have a set of data which includes "4 10 2" as a line of data, the linear plot will display the error as an error of 2 in whatever units your Y axis is measured. The top of the error bar will be found at 12 and the bottom at 8 as you might expect. The logarithmic plot, on the other hand, will display the error bar with its top at 20 (a ratio of 2) and its bottom at 5 (a ratio of 0.5).

In order to have this error bar come out correctly on the plot, you must convert it to a ratio of the datum value. This may be done by hand, but a much simpler way is to use the ZDATA parameter and the expression evaluation capability of the PLT program. In our example above, we would add a line similar to the following:

```
ZDATA=( $2+$3 ) / $2
```

This line will convert the error value to a ratio of the Y component of each datum read in. The error bars will appear at the (nearly) correct data locations of 12 and 8.33, from our example above. If you wish the PLT program to ignore the third datum in each line, set to ZDATA=0 and it will replace each of these data values with zero.

**XDATA [(datum \* XGAIN) + XOFST]**

**YDATA [(datum \* YGAIN) + YOFST]**

**ZDATA [(datum \* ZGAIN) + ZOFST]**

These parameters permit the user to modify the data in any number of ways. Various PLT functions may be applied, user variables may be used to modify the data and the data may even be ignored completely. The expression in each of these parameters will be applied to the corresponding

input data values as they are processed into the data buffer. For an example of the use of this parameter, see Section A.5 on Data Value Representations in Chapter 3, "Functional Description". The default transformation may be obtained by setting the parameter to a "null" value, such as "XDATA=".

### **XFMT, YFMT, RFMT and TFMT [I]**

These are text strings that define annotation formats for the X, Y, R and T axes. The permitted formats are I (for integer) and F (for floating-point). The form of the string is Im or Fm.n, where I specifies integer annotations, i.e., whole numbers, and F specifies floating-point annotations, i.e., values with a decimal point. In keeping with standard FORTRAN format specifications, m defines the number of characters to use; however, PLT internally determines how many characters exist in each annotation and adjusts accordingly, so that specifying m is actually unnecessary. n specifies the number of digits after the decimal point in floating-point annotations, and it must be specified. The default formats are I for all four axes.

### **XGAIN, YGAIN and ZGAIN [1]**

### **XOFST, YOFST and ZOFST [0.0]**

As each data line is read by the PLT program, the x,y,z values are multiplied by XGAIN, YGAIN, ZGAIN (default value for each is 1.0) and then added to XOFST, YOFST, ZOFST (default value 0.0) prior to being placed in the data buffer. This gives the user the opportunity to scale data values on the plot without changing data values in the plfile or the source file.

### **XINT [5], YINT [5], RINT [0], and TINT [0]**

These parameters control the way tick marks defining intervals on the respective (linear) axes will be drawn. This parameter's value takes the form MM.N. It defines two types of intervals, "major" and "minor". MM defines the number of major intervals (and produces MM+1 tick marks) along the specified axis. Only major intervals will be annotated, and only major intervals will generate grid lines (when a grid is desired). N defines the number of minor intervals per major interval (and produces N-1 tick marks per minor interval); N is limited to a maximum of 9. A minor interval is not annotated and carries no grid lines. Major intervals possess tick marks that are twice as long as those denoting minor intervals.

For example, the command line XINT=5.2 will cause the X axis to be plotted as 5 major intervals equally spaced between XMIN and XMAX. In addition to this, each major interval is subdivided into 2 minor intervals, making a total of 10 intervals overall. If XMIN=0 and XMAX=100, then each major interval is 20 units wide and each minor interval is 10 units wide. Annotations will be plotted at the tick marks corresponding to 0, 20, 40, 60, 80 and 100 (the major intervals). Small tick marks and no annotations will appear at the points on the X axis corresponding to data values of 10, 30, 50, 70 and 90.

If the axis is logarithmic, then the number of ticks is defined by the number of cycles, and the number of intervals is ignored, unless it has the value -2. This special value causes the log axis to have cycles which are multiples of 2 instead of the normal cycles which are multiples of 10. The

default value of this parameter is 5 for the X and Y axes and 0 for the R and T axes. (See also **XCYCLE**.)

### **XLABEL, YLABEL, RLABEL and TLABEL [no label]**

These parameters accept text input, and they define the labels to be given to the respective axis. Axis labels may be up to 250 characters long and may have superscripted and subscripted text as well as symbols in them. The special characters, described in the section on "Messages" in Chapter 3, "Functional Description", may be used in labels as well. The major differences between axis labels and messages are that (a) an axis label must occupy only one line and (b) axis labels are not bounded by leading and trailing quotation marks.

Axis labels are centered along the desired axis. The X-axis label is plotted below the X axis; the Y-axis label, along with its annotations, is plotted to the left of the Y axis reading upward at a 90-degree angle (see **YHOR**); the R-axis label is plotted to the right of the R axis (also at a 90-degree angle); and the T-axis label is plotted above the T axis.

Multi-line axis labels may be obtained by placing the newline sequence (\n) at the point in the LABEL parameter string at which the second line is to begin. The following will generate a two-line label for the X axis:

```
XLABEL=Category of Manipulation\n(Classified by Group)
```

The succeeding lines will be placed at a position farther from the corresponding axis (See **MVSP**). Do not place the second part of the label on a separate line in the PLT file itself as in the following:

```
XLABEL=Category of Manipulation  
(Classified by Group)
```

This will cause the PLT program to generate an error message.

### **XLEN [6.0]**

This parameter controls the length, in inches, of the horizontal or x axis. For an 8.5" x 11" page, this is the 11" dimension (See also **ROTATE** and **WXLEN**). This length does not include space for annotations or axis labels. Although the user may set XLEN to any value, the length of the plottable area on the page is 10.5 inches. The default X-axis length is 6.0 inches.

### **XLLC [2.2]**

This parameter controls the horizontal position, in inches, of the lower left corner of the plot frame (set of axes) on the plotted page; that is, the location of the beginning of the X axis and the left-right position of the Y axis. The value of XLLC does not take into account the position of the annotations and axis labels; rather, it positions the plot frame in the left-to-right dimension. The default value is 2.2 inches from the left-hand edge of the page.

### **XMAX [10]**

See **XMIN**.

**XMIN [0] and XMAX [10]****YMIN [0] and YMAX [10]****RMIN [0] and RMAX [0]****TMIN [0] and TMAX [0]**

These parameters denote the minimum and maximum values of the various axes. That is, the annotations for each axis will begin at \*MIN and extend to \*MAX (where \* indicates one of the axes, X, Y, R or T). The default values are XMIN=0, XMAX=10; YMIN=0, YMAX=10; RMIN=0, RMAX=0; TMIN=0, TMAX=0.

Data X and Y values, when they are plotted, are never plotted according to the limit values for the top (T) or right (R) axes; they are always plotted according to the bottom-axis (X) or left-axis (Y) limits. The top and right axes exist only for reference, and it is the user's responsibility to set them up accordingly. There is an indirect way to refer X-values to the top axis, which is to (a) set up and plot one set of axes, (b) suppress X-axis annotations, (c) set XMIN and XMAX to the top axis values, and (d) enter and plot the data. This may also be done with Y-values.

Whenever MIN and MAX for the reference axes (R and T) are *both* zero (the default values) the tick marks for these axes will be the same as those on the corresponding data axes (Y and X).

**XOFST [0.0]**

See **XGAIN**.

**XPERCENT, YPERCENT, RPERCENT, and TPERCENT [100]**

These parameters control the percentage of the respective axis to be used in spanning the values between MIN and MAX. Its value may vary continuously from 0 to 100, with the default being 100 in all cases.

For example, a designation of XPERCENT=50 with XLENGTH=6.0 will cause PLT to use only half of the axis length to plot the intervals between XMIN and XMAX (the center 3 inches). The tick mark denoting XMIN will be positioned 1.5 inches from the left end of the X axis; the tick mark for XMAX will appear 1.5 inches from the right end. In the default condition (XPERCENT=100), the entire length of the axis is used, and XMIN appears at its left edge.

**XSHIFT, YSHIFT, RSHIFT, and TSHIFT [0]**

If the range of data values has been restricted to occupy less than the full length of the axis (using the PERCENT parameters), this restricted range is normally centered within the axis length. The shift parameters may be used to shift the restricted range toward one side or the other. The value of these parameters is taken as a percentage of the full axis length by which the restricted range should be shifted in the positive direction. So, taking our example from XPERCENT, above, if XPERCENT=50, XLENGTH=6.0 and XSHIFT=20, the tick mark for XMIN will appear 2.7 inches from the left and the tick mark for XMAX will appear 0.3 inches from the right. The axis range is shifted 20 percent of 6 inches (1.2 inches) from the position it would have occupied for a total of 2.7 inches. The default value of all shift parameters is 0.

**XTICK, YTICK, RTICK and TTICK [1]**

The values of these parameters serve as multipliers of the tick mark lengths for the specified axis. This enables the user to control the presence and size of tick marks for each axis independently. This would be of use in a case in which tick marks make little sense, such as in a plot of a histogram. You could eliminate the tick marks on the top and bottom of the plot by setting XTICK and TTICK to 0 and retain the tick marks on the left and right for reader reference. These parameters also permit a user to change the size of the axis tick marks, without affecting the size of the break symbol (See **TICSIZ** and **XBREAK**).

**YANNOT [none]**

See **XANNOT**.

**YANSKP [0]**

See **XANSKP**.

**YBREAK [0]**

See **XBREAK**.

**YCYCLE [0.0]**

See **XCYCLE**.

**YDATA [(datum \* YGAIN) + YOFST]**

See **XDATA**.

**YFMT [I]**

See **XFMT**.

**YGAIN [1]**

See **XGAIN**.

**YHOR [0]**

This parameter, when non-zero or YES, instructs PLT to plot the Y- and R-axis annotations horizontally, i.e., in the same direction as the X-axis annotations. This allows annotations to be read somewhat more easily. When horizontal Y- and R-axis annotations are used, the axis labels are moved away from the axes by the correct distance, depending on the annotation values and formats. When employing horizontal annotations, users should allow enough space on the left and right sides of the plot for both the annotations and the axis labels. The default condition is vertical orientation of Y- and R-axis annotations (YHOR=0).

**YINT [5]**

See **XINT**.

**YLABEL [no label]**

See **XLABEL**.

**YLEN [6.0]**

This parameter controls the length, in inches, of the vertical or y axis. For an 8.5" x 11" page, this is the 8.5" dimension (See also **ROTATE** and **WYLEN**). It operates in the same manner as **XLEN**, but the plottable area is limited to 8 inches. The default Y-axis length is 6.0 inches.

**YLLC [1.5]**

This parameter controls the vertical position, in inches, of the lower left corner of the plot frame (set of axes) on the plotted page, i.e., the location of the beginning of the Y axis and the up-down position of the X axis. The operation of **YLLC** is the same as for **XLLC**. The default value is 1.5 inches from the bottom of the page.

**YMAX [10]**

See **XMIN**.

**YMIN [0]**

See **XMIN**.

**YOFST [0.0]**

See **XGAIN**.

**YPERCENT [100]**

See **XPERCENT**.

**YTICK [1]**

See **XTICK**.

**ZDATA [(datum \* ZGAIN) + ZOFST]**

See **XDATA**.

**ZGAIN [1]**

See **XGAIN**.

**ZOFST [0.0]**

See **XGAIN**.

## Chapter 4A.

# User Reference (Addendum)

### Keywords **STAT**

This keyword causes PLT to compute a set of values for the following internal variables:

<code>\$sum_x</code> .....	sum of x data
<code>\$sum_y</code> .....	sum of y data
<code>\$sum_xx</code> .....	sum of square of x data
<code>\$sum_yy</code> .....	sum of square of y data
<code>\$sum_xy</code> .....	sum of product of x data times y data
<code>\$lr_a</code> .....	linear regression intercept "a" such that: $y = a + bx$
<code>\$lr_b</code> .....	linear regression slope "b" such that: $y = a + bx$
<code>\$lr_c</code> .....	linear regression intercept "c" such that: $x = c + dy$
<code>\$lr_d</code> .....	linear regression slope "d" such that: $x = c + dy$
<code>\$corr_xy</code> .....	product-moment correlation between x and y data
<code>\$x_min</code> .....	minimum value of x data
<code>\$y_min</code> .....	minimum value of y data
<code>\$x_max</code> .....	maximum value of x data
<code>\$y_max</code> .....	maximum value of y data
<code>\$x_mean</code> .....	average (mean) value of x data
<code>\$y_mean</code> .....	average (mean) value of y data
<code>\$x_sd</code> .....	sample standard deviation of x data
<code>\$y_sd</code> .....	sample standard deviation of y data

The STAT keyword should appear after all the data have been read into the data buffers and before the PLOT keyword, which clears those buffers. The values in the internal variables will be retained until the next occurrence of the STAT keyword.

The names of the internal variables may be used in arithmetic expressions. For example, to obtain a regression equation and plot the regression line from a set of data, do the following:

```
; read in the data and use the stat keyword to compute
; values for the internal variables
pltype=symbols
1 5
2 4
3 3
4 2
5 1
stat
plot
; to plot the regression line, read in new data. use
; $x_min and $x_max internal variables as the X data values
; and set the YDATA parameter to the regression equation
; (using the $lr_a and $lr_b internal variables) to
; obtain the Y data values
pltype=lines
ydata = $lr_a + $lr_b * $1
zdata=0
($x_min)
($x_max)
plot
```

It is also possible to place the numerical values of the internal variables in messages anywhere on the plot. See the %msg macro.

### **%macrosmg x-position y-position format-string variable\_1 variable\_2 ...**

This macro enables the user to plot the numerical values of PLT internal variables in messages anywhere on the plot. The format string is a text string with floating point formats embedded as done in the C language. For example, to print the correlation between the X and Y data (obtained by use of the STAT keyword), use a statement like the following:

```
%msg 0 ylen+1 "correlation = %.2f" $corr_xy
```

See any reference manual on the C language for further details. Two well-known ones are:

Kernighan, B. W. and Ritchie, D. M. (1988). *The C Programming Language (2<sup>d</sup> Edition)*. (Englewood Cliffs, N.J.: Prentice-Hall, Inc.).

Harbison, S. P. & Steele, G. L., Jr. (1991). *C: A Reference Manual (3<sup>d</sup> Edition)*. (Englewood Cliffs, N.J.: Prentice-Hall, Inc.).

**SIN, COS, TAN, LN, LOG, ABS, MIN, MAX FUNCTIONS**

Users may now use the following common functions

**sin(x)** - sine of x (x in radians)

**cos(x)** - cosine of x (x in radians)

**tan(x)** - tangent of x (x in radians)

**atan(x)** - arctan of x

**ln(x)** - natural logarithm of x (base e=2.718281828...)

**log(x)** - logarithm of x (base 10)

**abs(x)** - takes absolute value of x

**min(x<sub>1</sub>,x<sub>2</sub>,...,x<sub>8</sub>)** - takes minimum value of all (up to 8) arguments

**max(x<sub>1</sub>,x<sub>2</sub>,...,x<sub>8</sub>)** - takes maximum value of all (up to 8) arguments

**XDAT, YDAT, ZDAT FUNCTIONS**

Users may access the data buffer values using the following functions:

**xdat(i), ydat(i), zdat(i)** - *i*-th value in corresponding data buffer ( $1 \leq i \leq n$ )

These functions return the corresponding data buffer value if the argument is between 1 and the number of data points. Otherwise, the function returns a value of zero.

**ZDEV, ZTOP FUNCTIONS**

The **zdev()** function takes as input a number (proportion) in the range from 0 to 1 and returns the normalized *z*-value. You can use this in the PLT program through the *xdata* and *ydata* parameters.

Data are plotted as follows:

```
xdata=zdev($1)
```

```
ydata=zdev($2)
```

or, if you have percentiles:

```
xdata=zdev($1/100)
```

```
ydata=zdev($2/100)
```

This will make it easier to plot ROC curves on *z*-score coordinates. Another use for this function is aid in plotting *theoretical quantile-quantile* plots. The latter subject is covered in Chambers, J.M., Cleveland, W.S., Kleiner, B., & Tukey, P.A. (1983). *Graphical Methods for Data Analysis*. (Belmont, CA: Wadsworth International Group). This plot permits the user to determine how close the distribution of data is to a normal distribution. Deviations from a straight line indicate variations such as skewness and lightness/heaviness in the tails. An example of each of these kinds of plots is included at the end of this addendum.

The **ztop()** (pronounced "z to p") function takes a *z*-value as argument and returns the proportion of area under the normal curve to the left of that value. It is the inverse of the **zdev()** function.

## LIMIT FUNCTION

The function **limit(x)** is used to restrict data to a specified range. It's like a combination of the **min()** and **max()** functions. For example:

```
xdata=limit(0.01, $1, 0.99)
```

will set xdata to the following values:

\$1	if \$1 is in range	$(0.01 \leq \$1 \leq 0.99)$
0.01	if \$1 is out of range, low	$(\$1 < 0.01)$
0.99	if \$1 is out of range, high	$(\$1 > 0.99)$

## IFELSE FUNCTION

The **ifelse()** function permits the user to make tests of values. The format is **ifelse (test,a,b)**. If the **test** is true (non-zero), the value **a** is returned; otherwise **b** is returned. For example, the PLT file below will plot what resembles a step function. If the values generated by xdata are less than 0, the y-values will be .75; otherwise they will be .25. The **ifelse()** function may be nested, so that an interval may be broken up further, as in the commented line immediately below the first ydata line. If this line is substituted, the negative region of x values will be separated further into values above and below -1.5.

```
annlwt=0 : header=no : yhor=yes
ymin=0 : ymax=1.0 : xmin=-3 : xmax=3.0
pltype=symbols
symsiz=.5 : symbol=1
ndata=100
xdata=-3+($0*.06)
ydata=ifelse($x<0,.75,.25)
; ydata=ifelse($x<0,ifelse($x<-1.5,.75,.50),.25)
0 0
plot
```

## USER-DEFINED FUNCTIONS

Users may define their own functions and use them to set parameters such as ydata to calculate values and simplify the typing. The definition has a format similar to a macro definition: **%define fnname <definition>**. The arguments are simply set out in the definition as \$1, \$2, etc. For example, the function **norm**, below, will generate values from the normal distribution using four parameters. (The double quotes are needed because of the internal spaces. Note, also, the use of the **\$x** substitution in the function definition; this is covered on page 4)

```
head=0 : annlwt=0 : ndata=100 : e=2.718281828
%define norm "$3+ $4 * e^(-((($x-$1)/$2)^2)"
s=(xmax/ndata)
xdata=$0*s
ydata=norm($1,$2,$3,$4)
zdata=0
pltype=lines
4 2 3 5 ; <mean> <s.d.> <baseline> <height>
```

```
plot
6 2 3 5      ; <mean> <s.d.> <baseline> <height>
plot
```

### **\$x** SUBSTITUTION

The user may now use the current x-value (after it has been processed by the xdata setting) as a value to be substituted in some other location. For example, the PLT file below will generate z-values between -3 and +3, and use the current x-data value as the argument to the **ztop()** function. The plot is the distribution function of the normal distribution (a familiar sigmoid curve).

```
ymin=0 : ymax=1.0 : yint=5 : yfmt=f3.1
xmin=-3 : xmax=3.0 : xint=6 : xfmt=f3.1
pltype=symbols
symbol=11
header=no
yhor=yes
ndata=100
xdata=-3+($0*.06)
ydata=ztop($x)
0 0
plot
```

### **X**AXIS, **Y**AXIS, **T**AXIS, **R**AXIS [1]

The values of these parameters permit users to include or eliminate axes as they choose. For example, setting **taxis=0** and **raxis=0** is equivalent to setting the parameter **openax=y**.

### **EXIT**

This keyword enables the user to immediately halt the processing of all PLT files, even from a nested included file. The EXIT keyword also prevents the PLT program from plotting to the screen, as seen in the example for PLAY, below.

### **PLAY**

This keyword enables the user to play sounds through the internal speaker of the PC. For example, the following PLT file plays 10 randomly generated sounds. The x-value is the frequency of the sound in Hertz and the y-value is the duration in seconds.

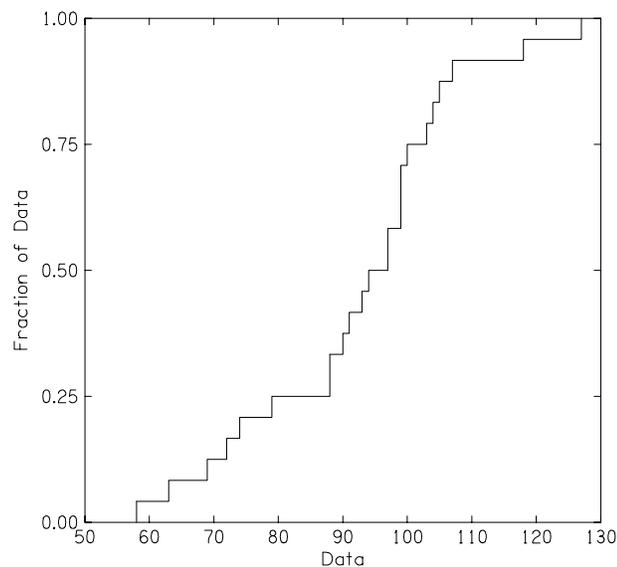
```
header=0
ndata=10
xdata=1000+500*$random
ydata=0.3+0.1*($random+$random)
0 0
play
exit
```

**PLTYPE=CUMULATIVE DISTRIBUTION**

A new value has been added to PLTYPE. The "cumulative distribution" (type 9) is an *empirical distribution function* which permits the user to plot each datum against its corresponding proportion value. A quantile is defined to be the data value at or below which some proportion of the data falls. So, for 5 sorted data points, the first datum is the .20 quantile, the second datum is the .40 quantile, etc. (It is analogous to a percentile except that a percentile refers to a percent of the data and a quantile refers to a fraction of the data.) The values plotted on the x axis are the data values themselves (quantiles) and the values on the y axis are the corresponding proportions.

PLT will sort the data, calculate the values for x and y and plot the function. The curve will be plotted as a connected series of steps. The vertical portion of each step will be drawn at the data value (x) and the horizontal portion will be drawn at the fraction value (y) as seen in the example below. Several equal data values will produce a longer vertical line. Use of a plot like this enables the user to make some observations about the median, quartiles, interquartile range, as well as the asymmetry or skewness of a data set.

In the example shown below, the median (the .50 quantile) may be seen to be between 94 and 97 and the interquartile range a little under 20 (the actual values are median = 95.5 and the IQR = 18). It can also be seen that the distribution appears to be positively skewed, the upper two values (118 and 127) being out of line with their nearest neighbors (shifted to the right).



```
header=no : yhor=y : ticsiz=.5 : ticdir=inward
ymin=0 : ymax=1.0 : yfmt=f5.2 : yint=4
xmin=0 : xmax=250 : xint=5
xlabel=Data Values
ylabel=Fraction of Data
pltype=cumulative
include datafile.dat
plot
```

## CHARACTERS AS SYMBOLS

Previously, it was necessary to use message text, setting the horizontal and vertical alignment to "center" and "half," in order to place characters as symbols. It is now possible to use any character from the standard and alternate character sets as a symbol in a plot. For example, the plot below will place the Greek letters alpha and beta in the data space, each corresponding to one of the members of a set of randomly generated data. The symbol parameter is set to the symbol number (Figure 4-1 in the PLT User's Guide) for the standard character set and preceded with a minus sign to obtain the symbol from the alternate character set. (Another way to obtain the alternate character set is to set the standard font to `altfnt` using `stdfnt altfnt` and set `symbol` to the number from the alternate character set table without a minus sign.)

```
annlwt=0 : yhor=yes : header=no
ticsiz=.5 : ticdir=inward
pltype=symbol
symbol=-97
xdata=5*$random+1
ydata=5*$random+1
ndata=5
0 0
plot
symbol=-98
0 0
plot
```

## CALCULATED CHARACTERS IN MESSAGES

It is also possible to set a character in a message to a calculated value. This is useful when passing parameters to a macro that will (in turn) print a message with that character. The following example illustrates this idea. The file below will add a legend to the PLT file above by passing each symbol as a parameter to a macro called "msgout."

```
%define msgout      ; <x loc> <y loc> <char> <string>
msgdat=y
$1,$2,"|($3)| $4"
%%
; example from "Characters As Symbols," above, goes here
msgout 8 9 -97 Preliminary$values
msgout 8 8.5 -98 Final$values
```

## COMMAND LINE PARAMETER SETTING

PLT now allows you to set MAGNIFY (or any other parameter) on the command line. For example:

```
plt -m5 magn=.5 rota=y file.plt > file.eps
```

will plot *file.plt* in PostScript format, rotated at 1/2 size, and store it in file *file.eps*.

### **PLT\_BELL ENVIRONMENT VARIABLE**

The prompting sound at the end of each plotted page can be turned off by setting the environment variable PLT\_BELL. Typing

```
set plt_bell=n
```

will turn off the sound. The sound can also be programmed using the syntax described in the "sound" section of the GRLIB tech memo. The default sound is `plt_bell=t900o7eb`, which sets the tempo to 900 beats/minute, the octave range to 7, and plays the two notes e and b.

### **PLT\_DEVICE ENVIRONMENT VARIABLE**

The environment variable PLT\_DEVICE enables the user to set the decimal number of a specific VGA or SVGA graphics mode. For users with VESA compatible BIOS extension for their graphics card, PLT will support 640 x 480 resolution with 256 colors. Just set the **plt\_device=vesa**. If you have a VGA adapter and would like to use PLT in 256 color mode, you can set

```
set plt_device=19
```

to use the 320x200x256 VGA color mode.

If your video adapter also has super VGA capability and supports the VESA standard bios extensions (usually provided as a TSR, "terminate and stay resident" program, with most SVGA cards) then you can

```
set plt_device=256, for 640x400x256 colors  
set plt_device=257, for 640x480x256 colors  
set plt_device=259, for 800x600x256 colors  
set plt_device=259, for 1024x768x256 colors
```

This feature has been tested at resolutions up to 640x480x256. It is not known if PLT will work at higher resolutions.

An example of a 256 color pltfile may be found at the end of this addendum.

```
; ROC CURVES
; PRODUCE TWO PLOTS
; UPPER - ROC curve plotted in percentile coordinates
; LOWER - ROC curve plotted in z-score coordinates
; uses zdev(), ztop(), limit() functions
;
; the dp macro will plot a d-prime curve for percentile plots
; and a straight line (subject to constraints set by limit() fn)
; for z-score coordinates.
;
%define dp      ; <xdprime> <yprime> <l=percentile,0=z-score> <ndata>
s=6/$4
xdata=ifelse($3,ztop($1+((($0-1)*s))*100,limit($1,$1+((($0-1)*s),-$2))
ydata=ifelse($3,ztop($2+((($0-1)*s))*100,limit($2,$2+((($0-1)*s),-$1))
ndata=$4
0 0
plot
ndata=1
%%
header=no : yhor=yes
ticdir=inward : ticsiz=.5
xlen=3.0 : ylen=3.0 : xllc=1.7 : yllc=4.5
xmin=0 : xmax=100 : xper=85 : xint=10
ymin=0 : ymax=100 : yper=85 : yint=10
ylabel=HIT RATE (%)
xlabel=FALSE ALARM (%)
msgdat=yes : msgsiz=1.25
65, 15 "Plotted
on Percentile
Coordinates"
;
; first, plot the ROC curves
;
pltype=both
lintype=0 : symbol=0
0,3
0,16
0,48
2,59
5,69
10,75
19,85
24,92
31,97
38,98
50,100
62,100
79,100
88,100
90,100
95,100
100,100
plot
;
; generate d-prime curves
;
pltype=2 : lintyp=1
%dp -3 -3 1 20 ; dprime=0
%dp -3 -2 1 20 ; dprime=1
%dp -3 -1 1 20 ; dprime=2
%dp -3 0 1 20 ; dprime=3
;
; generate d-prime labels
;
```

```
mhalign=left : msgangle = 45 : msgsiz=.85
46,49,"d' = 0.0"
27,67,"d' = 1.0"
12,82,"d' = 2.0"
3,91,"d' = 3.0"
plot
```

```
newframe
;
; now, in z-score coordinates
; data are same as above, except any values in which either x or y
; is 0 or 100 have been removed
;
xmin=-3 : xmax=3 : xint=6
ymin=-3 : ymax=3 : yint=6 : yllc=.8
ylabel=HIT RATE (Z SCORES)
xlabel=FALSE ALARM (Z SCORES)
msgangle=0 : msgsiz=1.25
1.2,-2.3,"Plotted
on Z-Score
Coordinates"
pltype=both
xdata=zdev($1/100)
ydata=zdev($2/100)
lintype=0 : symbol=0
2,59
5,69
10,75
19,85
24,92
31,97
38,98
plot
```

```
;
; plot the d-prime lines in z-score coordinates
;
pltype=lines : lintype=1
xdata=$1
ydata=$2
-3 -3
3 3
plot
%dp -3 -2 0 20 ; dprime=1
%dp -3 -1 0 20 ; dprime=2
%dp -3 0 0 20 ; dprime=3
;
; generate d-prime labels
;
mhalign=center : msgangle=45 : msgsiz=.85
zdev(.47),zdev(.49),"d' = 0.0"
zdev(.27),zdev(.67),"d' = 1.0"
zdev(.0025),zdev(.28),"d' = 2.0"
zdev(.03),zdev(.91),"d' = 3.0"
plot
finish
```

```
; THEORETICAL QUANTILE-QUANTILE PLOT (PROBABILITY PLOT)
; (Created by Edward Carney)
;
; REFERENCE:
; Chambers, J. M. et al., Graphical Methods for Data Analysis.
; Monterey, CA: Wadsworth International Group, 1983.
;
; This plot is equivalent to Figure 6.6, p. 204.
;
header=no : yhor=y
ticdir=inward : ticsiz=0.5

xlen=5 : ylen=5 : xper=90 : yper=90
yint=6 : ymin=0.5 : ymax=3.5 : yfmt=f3.1
xint=6 : xmin=-3 : xmax=3 : xfmt=f3.1 : yhor=y
xlabel=Normal Quantiles
ylabel=Log Seeded Rainfall

msgdat=y : mhalgn=center : msgsiz=1.1
0,(ymax+.3),"Normal Quantile Plot for Log Seeded Rainfall"

;
;
; the data are ordered from smallest to largest (order statistic, x(i)).
; the proportion of the data values to the left of the i-th value
;  $p(i) = (i-.5)/n$ 
;
; set n
n=26
;
; user defined function to calculate the p value
;
%define pval ($1-.5)/n
; produce a quantile-plot using calculated normal quantiles
xdata=zdev(pval($0))

; normal distribution
;ydata=$1

; log normal distribution      log(y)
; (See Chambers et al.)
; adjust ymin/ymax/ylabel above as needed
ydata=log($1)

; power transformed variable has a
; power normal distribution    y^theta (See Chambers et al.)
; adjust ymin/ymax/ylabel above as needed
;ydata=$1^.12
```

```
; plot the data

pltype=symbols
symbol=1 : symsiz=.7
solid=-1
data

; seeded rainfall data from Chambers et al. Appendix 5, p 351.
; (sorted in ascending order)

4.1
7.7
17.5
31.4
32.7
40.6
92.4
114.3
118.3
119
129.6
198.6
200.7
242.5
255
274.7
274.7
302.8
334.1
430.0
489.1
703.4
978
1656
1697.8
2745.6
plot
finish
```

```
; PLOT TO USE 256 COLORS (Created by Steven Neely)

; BE SURE TO SET plt_device=19 FOR VGA
; OR TO
;     set plt_device=19 for 640X480X256 colors (VGA)
;     set plt_device=256, for 640x400x256 colors (SVGA)
;     set plt_device=257, for 640x480x256 colors (SVGA)
;     set plt_device=259, for 800x600x256 colors (SVGA)
;     set plt_device=259, for 1024x768x256 colors (SVGA)
; Note the use of nested %repeat macros.
; Each successive level of nesting requires double the number
; of $ to access the correct parameter.
; So, inside the innermost %repeat, $0 will access the outermost
; counter, $$0 will access the counter for the %repeat in which
; the outer one is nested, the next needs $$$$0, and so forth.
; Another way of putting this is that $0 will change most slowly,
; and $$$$$$$$0 will change most rapidly.
; For the %% symbols, you need only add 1 % for each successive
; level of nesting.

;
header=0 : axlwt=0 : annlwt=0
xmin=0 : ymin=0 : xmax=8 : ymax=8 : xllc=0 : yllc=0 : xlen=8 : ylen=8
plot

n=4 : b=(1/((n-1)*(n+1))) : a=(b*n)
%define RGBW "((($1-1)*a+($4-1)*b) ((($2-1)*a+($4-1)*b) ((($3-1)*a+($4-1)*b)"
data
%repeat n
    %repeat n
        %repeat n
            %repeat n
                %RGBW $0 $0 $$$$0 $$$$$$$$0
                %%%
            %%%
        %%%
    %%%
%%
palette

%define box ; <xllc> <yllc> <color_index>
xofst=($1)*xgain : yofst=($2)*ygain : pltype=6 : shade=21 : shdcol=$3
0 0
0 1
1 1
1 0
plot
%%
```

```
xgain=(10.5/(n*n)) : ygain=(8/(n*n)) : c=0
%repeat n
  %repeat n
    %repeat n
      %repeat n
        x=($0-1)*n + ($$$$$$0-1)
        y=($0-1)*n + ($$$0-1)
        c=(((($0-1)*n+($0-1))*n+($$$0-1))*n+($$$$$$0-1)
        %box x ((n+1)*(n-1)-y) c
          %%%
        %%%
      %%%
    %%%
  %%%
%%
```

## Chapter 5.

# Error Messages

The possible error messages generated by the PLT program are listed below. If the console device is being used for graphical output, when an error is encountered, the PLT program will attempt to open a log file named "plt.log". Subsequent error messages will be written to the log file.

### Fatal Errors

The errors in this section are the most severe and will cause the PLT program to terminate.

#### **ERROR: Can't allocate memory for data arrays.**

An error was encountered when trying to allocate memory for the X,Y,Z data arrays. The program probably needs more memory to run successfully.

#### **plt: can't open file ...**

The PLT file specified on the command line was not found or could not be opened for reading. If no extension was specified in the filename, the PLT program will first look for a file with no extension and then for a file with a ".plt" extension before printing this error message.

#### **plt: can't open device #...**

An error was encountered opening the PLT output device. The number indicates which device was being opened. See the PLT manual page for device information.

## Warnings

The errors in this section are less severe and the PLT program will continue to process the remainder of the PLT text file.

### **... is not a valid keyword**

A line with no '=' was encountered in the PLT file; it begins with a word (...) that is not recognized as a valid keyword.

### **\*\*\* BAD header in BDATA file: ...**

There is an error in the header of the binary data file (...).

### **\*\*\* can't open BDATA file: ...**

Either the binary data file (...) wasn't found or it couldn't be opened for reading.

### **\*\*\* can't open FONT file: ...**

An error was encountered trying to open a font file (...). Either the file does not exist or an error was made specifying the file name.

### **\*\*\* can't open INCLUDE or CALL file: ...**

Either the INCLUDE file (...) wasn't found or it couldn't be opened for reading.

### **\*\*\* FONT file has incorrect format: ...**

The PLT program has found invalid data in the font file (...) header. The file is not a valid font file.

### **\*\*\* not enough memory to load FONT file: ...**

The specified font file (...) cannot be loaded because available memory is insufficient. An internal font definition will be used.

### **cnglun: tried to close invalid file**

This indicates a programming error and should never be encountered in normal use.

### **Data Buffer Empty**

A DATLIST command was encountered, but there's no data to list.

### **error: keyboard DATA input requested**

The PLT facility for inputting data from the console as the pltfile is being processed should not be used when the graphics output of the PLT program is also directed to the console screen.

**error in data value: ...**

Syntax error encountered while trying to parse a line (...) that seemed to be a data value line.

**error: too many data points for spline smoothing**

The spline computation needs temporary space in the data buffer and the size of the data buffer is inadequate. The value of the spline parameter should be decreased.

**Illegal CALL/INCLUDE nesting**

INCLUDE files are nested too deep. The maximum is 15 levels of nested INCLUDE files.

**Invalid Parameter: ...**

A line with an '=' was encountered in the PLT file; the line begins with a word (...) which was not recognized as a valid parameter.

**Unknown command: ...**

PLT encountered a line (...) in the pltfile which was not recognized as a valid command line. This line probably begins with a character which is not a number or a letter.

**WARNING: error reading binary data**

An error was encountered while reading from a binary data (BDAT) file.

**WARNING: Number of data points exceeds buffer size.**

The data buffer will expand as needed to accommodate the data. If additional memory is unavailable, then this warning is issued and a plot command is automatically performed to flush the buffer.

## **Chapter 6.**

# **Examples**

This section contains six examples of PLT text files which demonstrate some of the features of the pltfile language and illustrate the type of plots that the PLT program was designed to produce. Each example begins with a short descriptive paragraph followed by a complete listing of the pltfile and the plot produced by the pltfile.

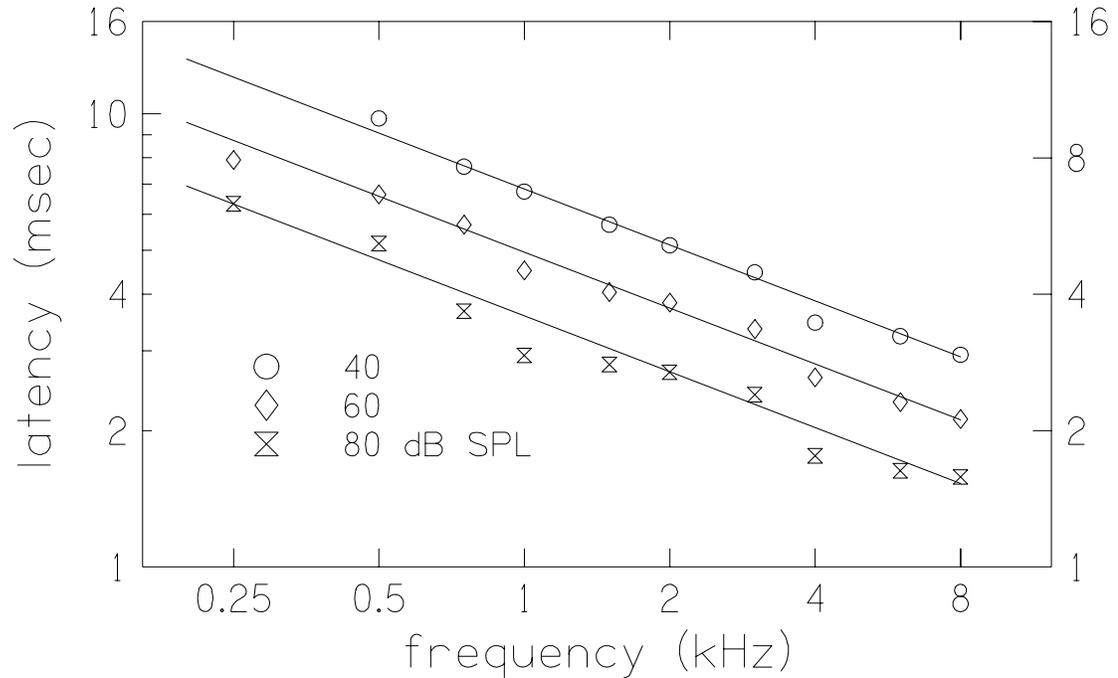
**Example I**

Figure 6-1. Example I

The first example is similar in format to the sample plot in the tutorial section. Note that the straight lines which represent a fit to the data points are plotted before the data points. The reason for this order is that the hollow symbols are filled with white and will thus appear (see Figure 6-1) to be on top of the lines. This pltfile uses XANNOT and YANNOT to specify axis annotations which differ from the default annotations. The right axis uses a log axis with cycles which are multiples of two instead of the usual cycles of 10 which appear on the left axis. More than one parameter value can be specified on a single line separated by " : ". (The space before the colon is required.)

```
; fig6-1.plt - Example I
;
rotate=yes : wylen=10.5 : head=0 : mhal=center
1.8 -0.75 "Figure 6-1. Example I"
wylen=8 : mhal=0
;
; ABR Wave V latency - 5 msec vs. frequency
;
wxlen=8 : clip=yes : head=no
msgsiz=-0.13 : labsiz=-0.18 : annsiz=-0.15 : symsiz=1
xllc=1.75 : xcyc=2 : xmin=0.25 : xmax=8 : xint=-2 : xlen=5 :
xper=80
yllc=2.75 : ycyc=1 : ymin=1 : ymax=16 : yint=0 : ylen=3 :
yper=100
xlabel = frequency (kHz)
ylabel = latency (msec)
yhor=yes : ticdir=in
rmin=1 : rmax=16 : rcyc=4 : rint=-2
xgain=0.001
xannot= 0.25 0.5 1 2 4 8
yannot= 1 2 4 10 16

pltype=lines : lintype=0
; latency = 13 * 500**(-i/100.) * f**(-0.41), f = 0.20 to 8 (kHz)
  200 13.211
 8000 2.911
plot
  200 9.575
 8000 2.110
plot
  200 6.940
 8000 1.529
plot
pltype=1 : lintype=4
symbol = 1 ; 40 dB
  500 9.77
  750 7.65
 1000 6.74
 1500 5.69
 2000 5.13
 3000 4.47
 4000 3.46
 6000 3.23
 8000 2.94
plot
symbol = 4 ; 60 dB
  250 7.91
  500 6.64
  750 5.69
 1000 4.51
 1500 4.04
 2000 3.83
 3000 3.35
 4000 2.62
 6000 2.31
 8000 2.12
```

```
plot
symbol = 5 ; 80 dB
 250 6.33
 500 5.17
 750 3.67
1000 2.93
1500 2.80
2000 2.69
3000 2.40
4000 1.76
6000 1.63
8000 1.58
plot
msgsiz=-0.13
0.6 1.25 "
|1| 40
|4| 60
|5| 80 dB SPL"
```

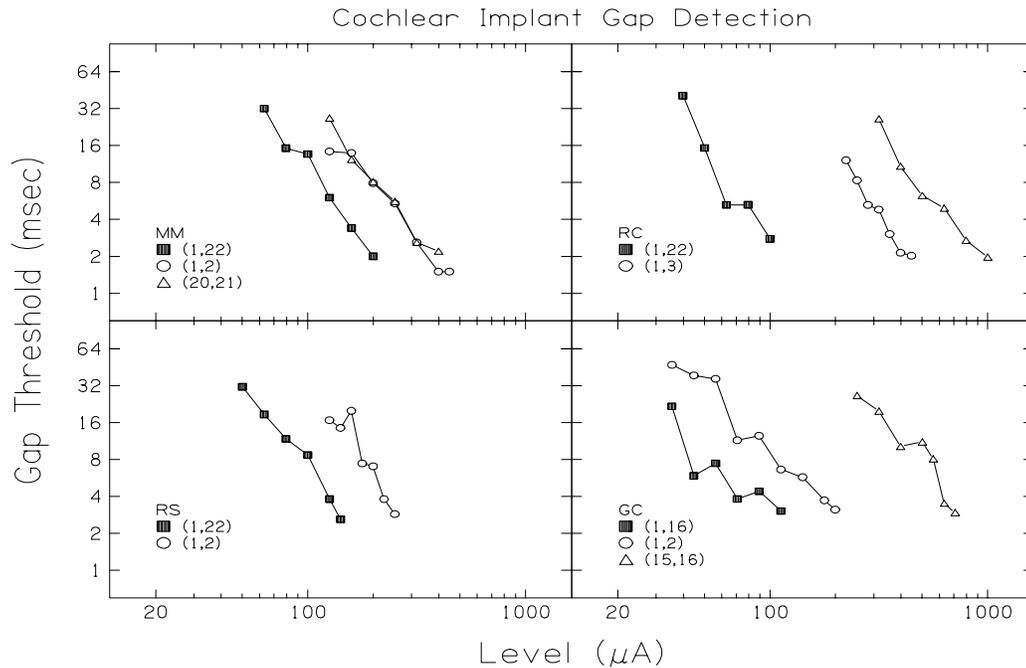
**Example II**

Figure 6-2. Example II

The four-panel plot in Figure 6-2 is created using four frames in the pltfile. Note the use of XANSKP and YANSKP in the pltfile to suppress annotations where they would be inappropriate. The axis labels must be done with messages to position them properly. Note the use of user defined variables (a, b, c, and d) in the pltfile to control the positioning of the frames and the positioning of the message inside each frame. Also, note the shift to the alternate font in the second message to obtain the Greek letter mu.

```

; fig6-2.plt - Example II
;
rotate=yes : wylen=10.5 : head=0 : mhal=center
1.8 -0.75 "Figure 6-2. Example II"
vylloc=1 : wylen=9.5 : magnify=0.6 : mhal=0
;
;   Gap detection for four subjects for apical-basal
;   and mono-bipolar comparisons
;
a=1.7 : b=1.2 : c=0.4 : d=0.9
xllc=a : xlen=4 : xmin=20 : xmax=1000 : xcyc=3 : xper=80
yllc=b : ylen=3 : ymin=1 : ymax=64 : ycyc=6 : yper=80 : yint=-
2
head=0 : ticdir=inward : annsiz = 1.5 : yhor=yes
msgsiz=2 : mhal align=center
xlen, 6.2, "Cochlear Implant Gap Detection"

```

```
msgsiz=2.5
xlen, -0.7, "Level (^m^A)"
msgangle=90
-0.6, ylen, "Gap Threshold (msec)"
msgsiz=1.25 : msgangle=0 : mhalign=left
pltype=both : lintype=0 : symsiz=1 : axlwt=1.5

symbol = 10 ; RS 1000 pps 100 us/phase (1,22)
    50.12    31.3
    63.1     18.6
    79.43    11.8
    100      8.7
    125.89   3.8
    141.25   2.6

plot
symbol = 1 ; RS (1,2) 1000 Hz 100 us/phase
    125.89   16.71
    141.25   14.47
    158.49   19.91
    177.83    7.43
    199.53    7.03
    223.87    3.81
    251.19    2.87

plot
c, d, "RS
|10| (1,22)
|1| (1,2)"
newframe
xllc=a : xanskp = -1
yllc=b+ylen
symbol = 10 ; MM 1000 Hz 100 us/phase (1,22)
    63.1     32.0
    79.43    15.2
    100      13.6
    125.89   6.0
    158.49   3.4
    199.53   2.0

plot
symbol = 1 ; MM 1000 Hz 100 us/phase (1,2)
    125.89   14.3
    158.49   13.9
    199.53    7.9
    251.19    5.4
    316.23    2.6
    398.11    1.5
    446.68    1.5

plot
symbol = 2 ; MM 1000 Hz 100 us/phase (20,21)
    125.89   26.8
    158.49   12.3
    199.53    8.1
    251.19    5.6
    316.23    2.6
    398.11    2.2

plot
c, d, "MM
|10| (1,22)
|1| (1,2)"
```

```

|2| (20,21)"

newframe
xllc = a+xlen : xanskp = 1
yllc = b      : yanskp =-1
symbol = 1          ; GC 1000 Hz 100 us/phase (1,2)
  35.48    47.2
  44.67    38.7
  56.23    36.34
  70.79    11.47
  89.13    12.5
 112.2     6.62
 141.25    5.73
 177.83    3.72
 199.53    3.12

plot
symbol = 2          ; GC 1000 Hz 100 us/phase (15,16)
 251.19    26.55
 316.23    19.81
 398.11    10.22
 501.19    11.11
 562.34     8.1
 630.96     3.51
 707.95     2.95

plot
symbol = 10         ; GC 1000 Hz 100 us/phase (1,16)
  35.48    21.65
  44.67     5.88
  56.23     7.43
  70.79     3.82
  89.13     4.41
 112.2     3.04

plot
c, d, "GC
|10| (1,16)
|1| (1,2)
|2| (15,16)"

newframe
xllc=a+xlen : xanskp = -1
yllc=b+ylen : yanskp = -1
symbol = 10          ; BC (1,22) 1000 pps 100 us/phase
  39.81    40.77
  50.12    15.29
  63.1     5.24
  79.43    5.2625
 100      2.78

plot
symbol = 2          ; BC (10,11) 1000 pps 100 us/phase
 316.23    26.34
 398.11    10.83
 501.19     6.25
 630.96     4.94
 794.33     2.7
 1000      1.97

plot
symbol = 1          ; BC (1,3) 1000 pps 100 us/phase
 223.87    12.11

```

251.19	8.33
281.84	5.24
316.23	4.81
354.81	3.04
398.11	2.14
446.68	2.02

```
plot  
c, d, "RC  
|10| (1,22)  
|1| (1,3)"
```

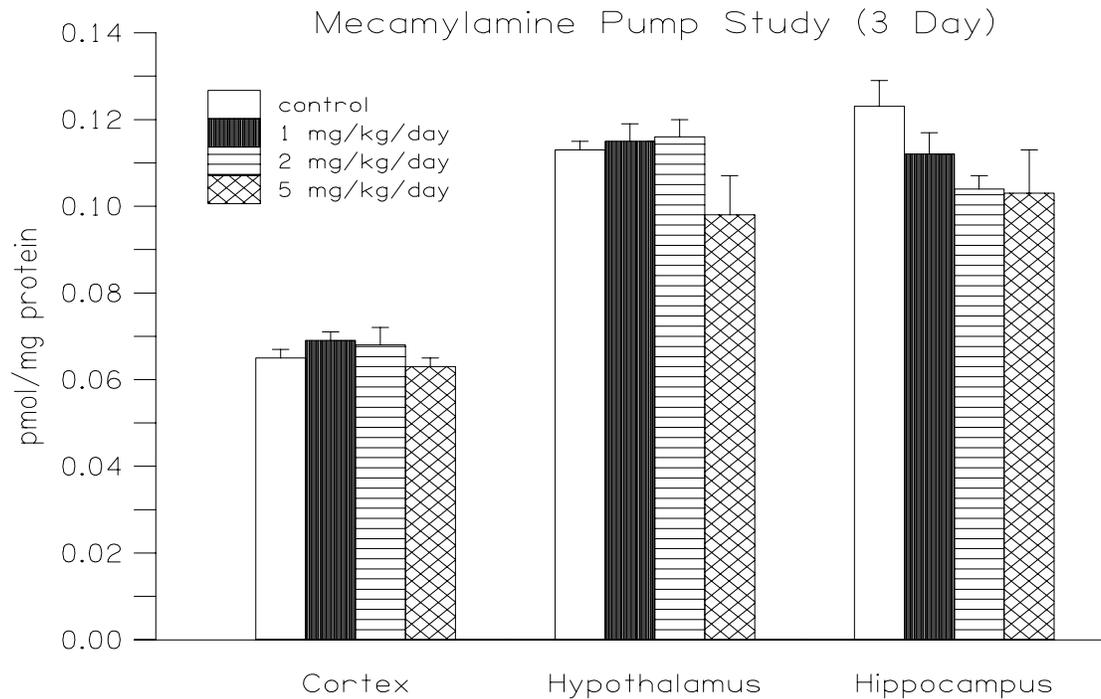
**Example III**

Figure 6-3. Example III

This example uses the histogram style plot (PLTYPE=4) to create a vertical bar chart (see Figure 6-3). Four separate histograms are plotted, each with a different SHADE value. Note the use of shade boxes in the message to provide a key to the meaning of each SHADE value. Also, note the use of EBNUM=1 to obtain only the upper error bars at the top of the histogram bins.

```

; fig6-3.plt - Example III
;
head=0 : mhal=center
3.2 -.7 "Figure 6-3. Example III"
vyl1c=.8 : wylen=9.5 : magnify=0.8 : mhal=0

```

```
;
; Mecamylamine Pump Study (3 Day)
;
head=no : openax=yes : ticdir=out : yhor=yes
xlen=8 : xllc=2.9 : xmin=2 : xmax=18 : xper=80
ylen=6 : yllc=1.4 : ymin=0 : ymax=0.14
xint=0 : xfmt=i3 : xanskp=-1
yint=14 : yfmt=f6.2 : yanskp=1
ylabel=pmol/mg protein

%define bars ; <order> <shade> <data1> <data2> <data3>
plttype=histo : ebnun=1
shade=$2 ; control
%repeat ($1-1) 0
$3
%repeat 5 0
$4
%repeat 5 0
$5
%repeat (4-$1) 0
plot
%%

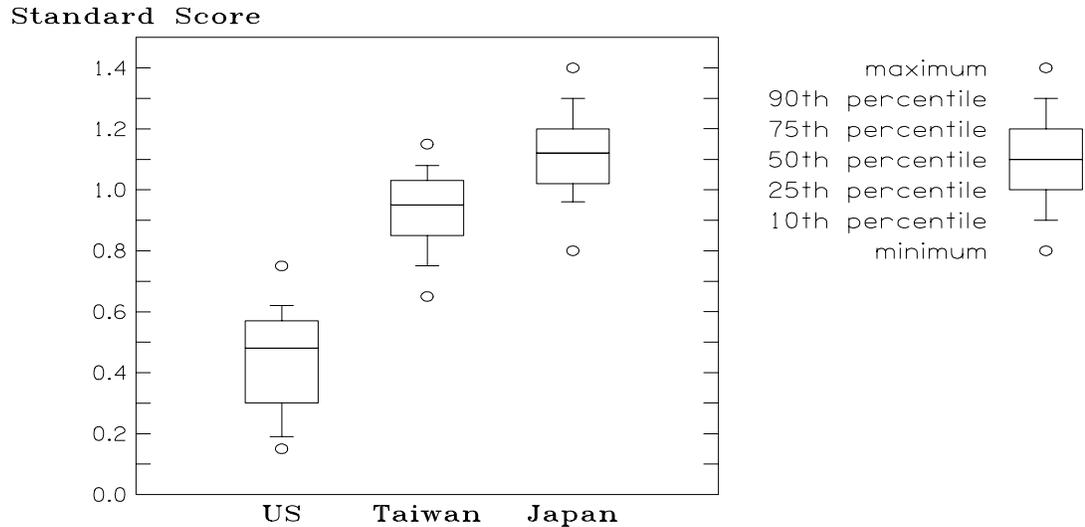
; Cortex Hypothalamus Hippocampus
%bars 1 20 0.065,0.002 0.113,0.002 0.123,0.006 ; control
%bars 2 21 0.069,0.002 0.115,0.004 0.112,0.005 ; 1
mg/kg/day
%bars 3 38 0.068,0.004 0.116,0.004 0.104,0.003 ; 2
mg/kg/day
%bars 4 32 0.063,0.002 0.098,0.009 0.103,0.010 ; 5
mg/kg/day

msgsiz=1 : mhalgn=c : mvalgn=h : msgdat=y
4. -0.01 "Cortex"
10 -0.01 "Hypothalamus"
16 -0.01 "Hippocampus"
msgsiz=0.8 : mhalgn=l : mvalgn=h : mhkey=3 : mvkey=2 : mvsp=2
1 0.13 "
| #20 | control
| #21 | 1 mg/kg/day
| #38 | 2 mg/kg/day
| #32 | 5 mg/kg/day"
msgsiz=1.2 : mvalgn=b : mhalgn=c
10 0.14 "Mecamylamine Pump Study (3 Day)"
finish
```

**Example IV**

## Japanese 5th Graders Excel in Math

### Achievement Test Scores for 20 Classrooms



Redrawn from H. Stevenson, *et al.* (1986). *Science* 231, 693-699.  
 Figure 8 from "Ten Graphs (and How to Use Them)" *PC World*, September, 1988, page 216

Figure 6-4. Example IV

This example (Figure 6-4) illustrates the use of rectangles (PLTYPE=5) with error bars and the use of external font files to obtain different character styles. Note the use of an italic font as the alternate font in the credit line at the bottom of the plot. Note, also, that the error bar will appear on the same side of the rectangle as the corner on whose line its datum appears.

```

; fig6-4.plt - Example IV
;
;rotate=yes : wlen=10.5
head=0 : mhal=center
2.8 -0.75 "Figure 6-4. Example IV"
vyl1c=1 : wxlen=11 : magnify=0.8 : mhal=0
;
; Figure 8 from "Ten Graphs (and How to Use Them)"
;           PC World, September, 1988, page 216
;
head=0
xllc=2.75 : xlen=5 : xmin=0 : xmax=4 : xint=0
yllc=1.125 : ylen=5 : ymin=0 : ymax=1.5 : yint=15
yhor=yes : yanskp=1 : yfmt=f.1 : ticdir=in : annsiz=1.2

%define box      ; <xloc> <min> <10%> <25%> <50%> <75%> <90%>
<max>

```

```

pltype=symb : symbol=1
($1) $2 ; min
($1) $8 ; max
plot
pltype=rect : ebnun=1 : ebwidth=2
w=0.25 ; half-width of box
($1-w) $4 ($4-$3) ; 10 & 25 percentile
($1+w) $5 ; 50 percentile
($1-w) $6 ($7-$6) ; 75 & 90 percentile
($1+w) $5 ; 50 percentile
plot
%%

u=1 : t=2 : j=3 : k=6.25
xannot= u,US t,TAIWAN j,JAPAN

; min 10% 25% 50% 75% 90% max
%box u 0.15 0.19 0.30 0.48 0.57 0.62 0.75 ; US
%box t 0.65 0.75 0.85 0.95 1.03 1.08 1.15 ; TAIWAN
%box j 0.80 0.96 1.02 1.12 1.20 1.30 1.40 ; JAPAN
%box k 0.80 0.90 1.00 1.10 1.20 1.30 1.40 ; KEY

m=k-0.4 : p=-1/72 ; 1 point = 1/72 inch
msgdat=yes : mhalgn=right : mvalgn=half : msgsiz=12*p
(m) 1.4 "maximum"
(m) 1.3 "90th percentile"
(m) 1.2 "75th percentile"
(m) 1.1 "50th percentile"
(m) 1.0 "25th percentile"
(m) 0.9 "10th percentile"
(m) 0.8 "minimum"

stdfnt trprom
altfnt trpita
msgdat=no : mhalgn=center : mvalgn=bottom : msgsiz=12*p : m=0.1
0 ylen+m "Standard Score"
mxdat=no : mhalgn=center : mvalgn=base : msgsiz=20*p
3.5 6.250 "Japanese 5th Graders Excel in Math"
msgsiz=12*p
3.5 5.875 "Achievement Test Scores for 20 Classrooms"
mhalgn=right : mvalgn=bottom : msgsiz=8*p : m=0.1-ylrc
7.5 m "Redrawn from H. Stevenson, ^et al.^ (1986). ^Science^
231, 693-699."

```

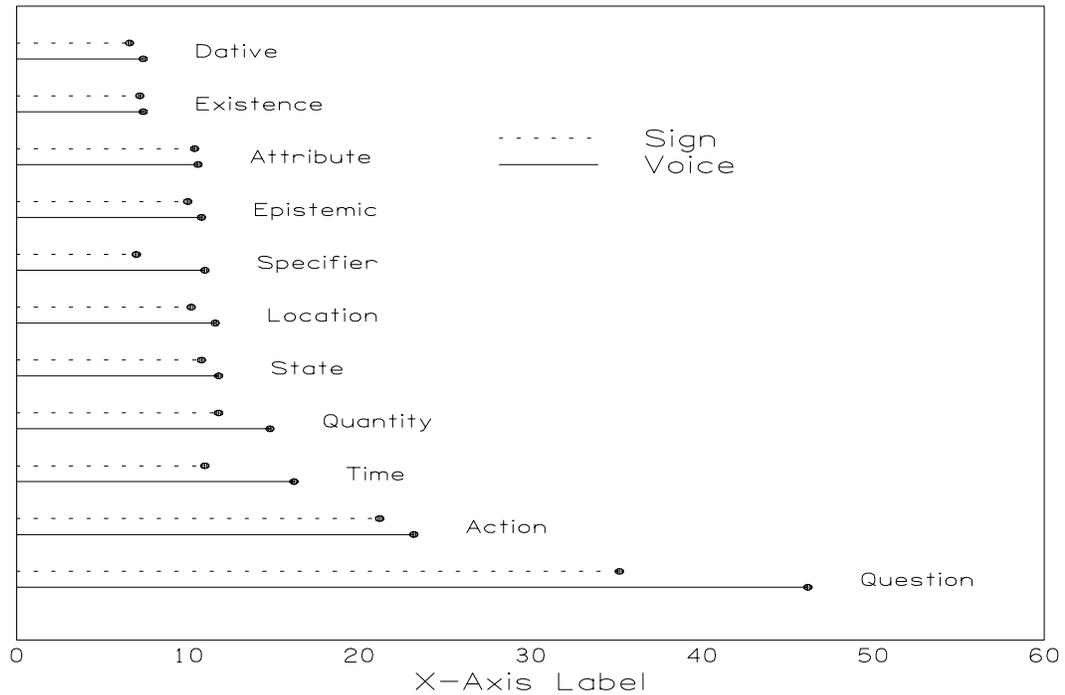
**Example V**

Figure 6-5. Example V

This example (Figure 6-5) illustrates the use of macros in PLT as well as the creation of a somewhat unusual style of plot. The parameters passed when the macro is invoked consist of the Y location of the lower member of each pair of lines, the X dimension for each line and the text string to be printed to the right of each pair (a total of four parameters). First, the lower (solid) line is drawn, then, the upper (broken) line. Next, the right end of each line is capped with a filled circle. Finally, the text string is printed as a message at the right end of the lower line. Note, also, that the Figure 6-5 message has been centered on the X axis by first changing the horizontal alignment (MHALIGN=CENTER) and then setting the MXLLC location using an expression (XLEN/2; that is, half the length of the X axis).

```
; fig6-5.plt - Example V
;
;
; macro definitions
;
%define DATDISP      ; <y-loc> <voice-val> <sign-val> <label>
pltype=lines : lintype=0
0   $1
$2  $1
plot
lintype=2
0   $1+0.3
$3  $1+0.3
plot
pltype=symbols
$2  $1
$3  $1+0.3
plot;
$2+4,$1+0.15,"$4"
%%
;
; FIRST, THE LABEL & ANNOTATION SIZES; NO HEADER
;
mhalign=left : mvalign=half
labsiz=.70 : axlwt=.4 : annsiz=0.8
yanskp=-1 : header=no
;
; ROTATE IT
;
rotate=yes
;
; CHANGE CLIPPING BOUNDARIES
;
wylen=10.5 : wxlen=8.0
;
; SET UP X AXIS
;
xlength=5.1 : xllc=1.5 : xmin=0.0
xmax=60.0 : xcyc=0 : xint = 6
xlabel=X-Axis Label
;
; NOW, THE Y AXIS
;
ylength=6.8 : yllc=2.5 : ymin=0.0
ymax=22.0 : ycyc=0 : yhor=yes
;
; NOW THE FIGURE NUMBER; CENTERED WITH MHALIGN & (XLEN/2)
;
mhalign=center
(xlen/2) -1.75 "Figure 6-5.  Example V"
;
; TICK MARKS, NONE
;
ticsiz=0
;
; DATA VALUES ARE THREE-PART: SOLID LINE, BROKEN LINE, DOTS
;
pltlwt=.5 : symbol=11 : symsiz=0.4 ; THESE WILL NOT CHANGE
```

```
mhalign=left : msgdat=yes : msgsiz=0.5 ;
;
%DATDISP 1 46.2 35.2 Question
%DATDISP 2 23.2 21.2 Action
%DATDISP 3 16.2 11.0 Time
%DATDISP 4 14.8 11.8 Quantity
%DATDISP 5 11.8 10.8 State
%DATDISP 6 11.6 10.2 Location
%DATDISP 7 11.0 7.0 Specifier
%DATDISP 8 10.8 10.0 Epistemic
%DATDISP 9 10.6 10.4 Attribute
%DATDISP 10 7.4 7.2 Existence
%DATDISP 11 7.4 6.6 Dative
%DATDISP 12 5.4 5.2 Denial
%DATDISP 13 4.4 4.2 Rejection
%DATDISP 14 4.4 3.9 Recurrence
%DATDISP 15 3.8 2.0 Possession
%DATDISP 16 2.0 1.6 Communication
%DATDISP 17 1.2 1.0 Additive
%DATDISP 18 1.0 1.0 Nonexistence
%DATDISP 19 0.2 0.2 Adversative
%DATDISP 20 0.0 0.0 Causality
;
; LEGEND
;
pltype=lines : lintype=0 : msgsiz=0.75 : mhkey=4
28 19.0 " |_0| Voice"
28 19.8 " |_2| Sign"
finish
```

**Example VI**

Average Areas of Graphs for 50 Articles in Selected Journals which Print in One-Column Format

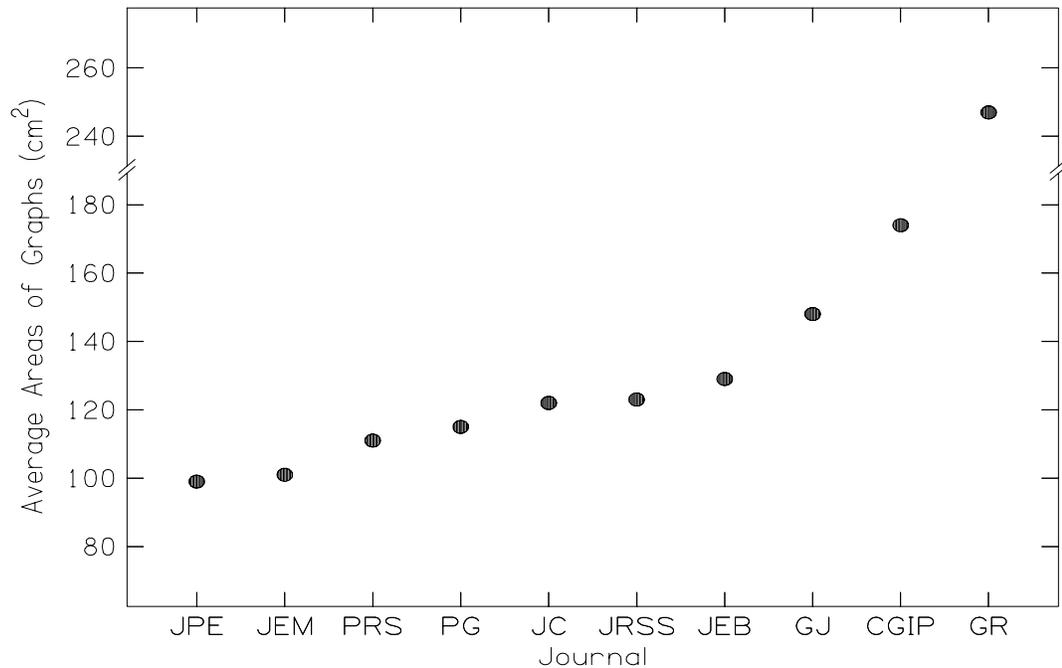


Figure 6-6. Example VI

This example (Figure 6-6) illustrates the use of the YBREAK and NEWRANGE parameters in PLT. First, the lower portion of the Y axis is plotted, along with the data. Then, the NEWRANGE keyword is invoked to permit changing the Y axis to the upper portion of the range. The YANNOT parameter is used to place annotations only at the desired data locations. Note, also, the use of expressions to place the main label in relative, not absolute, positions in the plot space.

```
; fig6-6.plt - Example VI
;
;
; YBREAK ILLUSTRATION
;
; PARAMETERS ON THE SAME LINE ARE SEPARATED BY SPACE-COLON-SPACE
;
xlength=6.5
xmin=1 : xmax=10 : xint=9 : xpercent=85
xlabel=Journal
xannot=1,JPE 2,JEM 3,PRS 4,PG 5,JC 6,JRSS 7,JEB 8,GJ 9,CGIP 10,GR
;
ymin=80 : ymax=220 : yint=7 : ypercent=80
ylabel=Average Areas of Graphs (cm{2})
yannot=80 100 120 140 160 180
ylen=5 : yhor=y
ybreak=73
ticdir=in : ticsiz=.8
;
; FIRST, THE DATA BELOW THE BREAK
;
pltype=symbols
symbol=11
symsiz=.9
data
1 99
2 101
3 111
4 115
5 122
6 123
7 129
8 148
9 174
plot
;
; NOW, THE DATA ABOVE THE BREAK
;
; NOTE THE USE OF THE NEWRANGE PARAMETER
;
newrange
ymin=120
ymax=260
yannot=240 260
10 247
plot
;
; MAIN LABEL
;
; NOTE THE USE OF EXPRESSIONS TO PUT MESSAGE IN THE CORRECT
; LOCATION IN THE PLOT SPACE; THE LENGTH OF THE X AND Y AXES
; MAY BE ADJUSTED WITHOUT AFFECTING THE MAIN LABEL'S PLACEMENT.
```

```
;
; XLEN/2    MEANS HALFWAY ALONG THE X AXIS
;
; YLEN+0.6 MEANS 0.6 INCHES ABOVE THE TOP END OF THE Y AXIS
;
; {2} IS THE SEQUENCE FOR SUPERSCRIPED TEXT IN A MESSAGE
;
malign=center
msgsiz=1.3
(xlen/2),(ylen+0.6),"Average Areas of Graphs for 50 Articles
in Selected Journals Printed in One-Column Format"
finish
```

## Appendix A:

# The Complete TEST.PLT File

A great many of the parameters have been combined into single lines by use of the space+colon (:) as a separator. The comment lines have a semicolon in them; usually in the first column. The order of PLT parameters is of little consequence; however, parameter settings will have their intended effect only if they are placed before the next occurrence of the PLOT (or KEEP) keyword. See the section, "Editing a PLT File," in Chapter 2 for other information.

```
;  
; test.plt  
;  
header=no  
xllc=1.9  
;  
; x axis values  
;  
xlength=7.0 : xmin=1 : xmax=12 : xint=11 : xper=85  
xlabel=Month (1973)  
xannot=1,J 2,F 3,M 4,A 5,M 6,J 7,J 8,A 9,S 10,O 11,N 12,D  
; y axis values  
;  
ylength=5.0 : ymin=0 : ymax=100 : yhor=yes  
ylabel=Average Monthly Temperature (^|48|^F)  
;  
; tick marks  
;  
ticdir=inward : ticsiz=0.5  
;  
; Eureka, CA data  
;  
pltype=both  
lintype=0 ; this line is optional (default is 0)  
symbol=0  
1 47.3  
2 50.7  
3 47.4  
4 50.1  
5 52.0  
6 55.1  
7 55.6  
8 54.8  
9 57.1  
10 52.1  
11 51.4  
12 51.2  
plot
```

```
;
; Lincoln, NE data
;
symbol=1
data                ; this line is optional
1 23.4
2 28.4
3 42.6
4 49.6
5 59.0
6 73.4
7 75.0
8 77.2
9 63.2
10 56.1
11 39.0
12 21.8
plot
;
; Newark, NJ data
;
symbol=2
1 35.5
2 33.3
3 48.6
4 54.2
5 60.4
6 74.5
7 78.7
8 79.6
9 71.0
10 60.3
11 48.8
12 39.4
plot
;
; legend
;
msgsiz=0.80 : msgdata=yes
1,90,"|0||_0||0| Eureka, CA
|1||_0||1| Lincoln, NE
|2||_0||2| Newark, NJ"
;
; main title
;
msgdata=no : msgsiz=1.5 : mhalign=center
3.5,5.75,"Average Monthly Temperature for 1973 (^|48|^F)"
;
finish                ; this line is optional
```

## Appendix B:

# Technical Data on Graphical Output and Printers

Direct output supported:

1. Screen output on the Sun, IBM-PC and NeXT computers.
2. The Adobe PostScript printer language output has been tested on the NeXT, the HP PostScript Cartridge for the LaserJet IID and the Panasonic KX-4455. It also works with *GoScript* and *ghostscript* on the IBM-PC. Color PostScript has been tested with *ghostscript* on EGA and VGA color screens, but not on a PostScript color printer.
3. Imagen Impress printer language.
4. Tektronix 4010 graphics terminal commands.
5. Tektronix 4663 pen-plotter language.
6. Hewlett-Packard HPGL pen-plotter language (this format has been imported into Microsoft Word Version 5.0).
7. CGM MetaFile format (this format has been successfully imported into Aldus PageMaker, Microsoft Word for Windows, Corel Systems CorelDRAW and the MirusImage software for the Mirus FilmPrinter slide imaging system).

Indirect output supported (a filter program is required to translate from Imagen Impress printer language):

1. Hewlett-Packard LaserJet printers (output has been printed on the LaserJet Series II and the LaserJet IID models).
2. Epson FX-compatible printers.

## Appendix C:

### External Font Files in PLT

The PLT program has two internal fonts which are referred to as the "standard font" and the "alternate font". Within any message text or label text the "^" character causes the PLT program to switch between these two fonts.

The PLT program can replace these internal fonts with external fonts which are defined in a special binary font file. For example, the PLT command "stdfnt filename" replaces the standard font with an external font stored in the file "filename.pft"; similarly, the PLT command "altfnt filename" replaces the alternate font. The external font files must be located in the current directory or in a standard directory (`c:\fonts` on a PC and `/usr/local/lib/pltfnt` on UNIX systems).

The PLTFNT program can be used to create the .PFT font files and to view the characters in each font file. PLTFNT is not a font editor; it cannot modify font characters as such. But PLTFNT can translate the binary font files to a text file which contains a set of descriptors for each character in the font. Each character's descriptor set consists of a series of *move* and *draw* "commands" which describe how each stroke in the character is to be drawn. A character's descriptor set can be edited with any text editor. When the text file has been edited, PLTFNT can translate the modified text file back into the binary .PFT form.

The PLTFNT program is not documented. A summary of command line options can be obtained by typing PLTFNT with no arguments.

Here's an example of how to use PLTFNT to modify a character. Suppose you wanted to put an accent stroke over the lowercase "a" in the external font called "smprom.pft".

First, with smprom.pft in the current working directory, create a text version of the font by typing the command "`pltfnt -t smprom`". This creates a new file called "smprom.txt". You can then view the entire font by typing "`pltfnt -s`

`smprom`" and determine that the lowercase "a" has the (ASCII) character code #97. To get a magnified view of this character, type `pltfmt -s97 smprom`.

With a text editor (such as  $\mu$ EMACS or EMACS), locate the definition of character #97 in the file "smprom.txt" by searching for a line that begins with "{ 97" (that is, "97" preceded by { and four spaces). Following this line are the *move* and *draw* commands which define the character. From the fourth line in the file (the line beginning with "fval"), it is possible to determine that the vertical coordinate of the "cap" position is 12 (the third number) and the "half" position is 0 (the fourth number). Look at the *moves* and *draws* for character 97 and notice that the horizontal extent of this character is from -6 to 6. Add a stroke above the character by adding one *move* and one *draw* command to the end of the character definition: "m -6 8" and "d 6 11".

Check the appearance of this new stroke by viewing character #97 with the command `pltfmt -s97 smprom`. When you're finished modifying characters, translate the text file back into the .PFT file with the command `pltfmt -p smprom`.

This procedure can be very tedious, even for simple character modifications, but it's the only method available for now.

## Appendix D:

# PLT Command and Usage Summaries

This appendix contains the PLT command and usage summaries as stored for on-line reference on UNIX machines and IBM-PC compatibles. The UNIX pages are specifically written for the version of PLT which runs on the NeXT machine, but the differences between the versions on UNIX machines are minimal.

**NAME**

plt - draw a graph

**SYNOPSIS**

plt [ options ] pltfile ...

**DESCRIPTION**

Plt causes the named pltfiles to be plotted on the indicated output device. The options are:

- e** Causes the input file to be echoed on the error output.
- cm** Use centimeters instead of inches as the basic unit.
- mm** Use millimeters instead of inches as the basic unit.
- m0** Output to stdout in Tektronix 4663 plotter language.
- m1** Output to stdout for Tektronix 4010 terminal.
- m2** Output to stdout in Impress language.
- m3** Output to stdout in HPGL graphics language.
- m4** Output directly to a PC graphics adapter (HGC, CGA, EGA, VGA), choice influenced by environment variable `PLT_DEVICE`.
- m5** Output encapsulated PostScript to stdout.
- m6** Output CGM (computer graphics metafile) to stdout.
- nq** Don't wait for user input between pages.
- s** "Slow mode" allows wide lines to be drawn.
- tc** Use device alphanumeric mode for text. (This is the default for -m0.)
- te** Speeds plotting by replacing all text strings with rectangles which represent the text extent.
- ts** Draw text in graphics mode. (This is the default for most devices.)
- v** Print PLT program version.

If the environment variable `PLT_DEVICE` is set to PC, then the plt program will try to determine whether the graphics adapter is HGC, CGA, EGA or VGA. To override this choice, `PLT_DEVICE` can be set explicitly to HGC, CGA, EGA or VGA.

Plt files are ASCII files with either a keyword, a transfer of control, a numeric parameter, a string parameter, a message, a macro or data values on each line. Only the first four characters are necessary to specify a keyword or parameter name.

## KEYWORDS

Keywords have no argument or value associated with them, but cause an action or change of mode.

DATA	Indicates the following lines are data points.
DATLIST	Prints contents of data buffer.
FINISH	Terminates and closes plot file.
KEEP	Plots data and keeps data for replotting.
NEWFRAME	Begin a new plot frame on the current page.
NEWPAGE	Advances to a new page.
NEWRANGE	Begin a new range of X and/or Y values in the current frame.
PALETTE	Uses data to set RGB color palette.
PLOT	Plots data and frame if necessary.
PRMLIST	Prints out the current parameter values.
RESET	Sets all parameters to their default values.
VARLIST	Prints out the current user variable values.

## AUXILIARY FILES

These commands have a file name as an argument. If "." does not appear in the filename then a default extension is added.

BDATA	Includes specified file as a binary data file.
CALL	Transfers control to another file.
INCLUDE	Includes specified file as a ".plt" file.
STDFNT	Load external font file as the standard font.
ALTFNT	Load external font file as the alternate font.
SYMFNT	Load external font file as the symbol font.

## NUMERIC PARAMETERS

Numeric parameters are followed by "=" and either a number or a word. Default values are given below after each parameter name. The numbers in brackets refer to Notes at the end of this section.

ANNCOL	-1	Annotation color. [6]
ANNLWT	1	Annotation line weight. [1]
ANNSIZ	1	Annotation size. [2]
AXCOL	-1	Axes color. [6]
AXLWT	1	Axes line weight. [1]
BGCOL	-1	Background color. [6]
CLIP	0	Clip lines at frame boundary. [3]
CORNER	1	Draw axis frame corners when xpercent or ypercent are less than 100. [3]
EBNUM	2	Number of error bars: 0 = none, 1 = upper bar only, 2 = both upper and lower, -1 = lower bar only.
EBWID	1	Width of error bars scale factor. [2]
ECHO	0	Echo each input line read. [3]

FGCOL	-1	Foreground color. [6]
FNCOL	-1	Fountain color. Specifies a different background color at the top of the page which gradually changes to BGCOL toward the bottom of the page. [6]
GAPSIZE	1	Controls size of gap between line segments for plotting symbols. [2]
GRDCOL	-1	Grid color. [6]
GRDLWT	1	Grid line weight. [1]
GRDSIZ	1	Grid line pattern size. [7]
GRDTYPE	1	Grid line type: 0 = solid lines, 1 = dotted lines, >1 = dashed lines.
GRID	0	Plot grid lines at major intervals. [3]
HEADER	1	Date, time, filename and page number printed at bottom of plot. [3]
LABCOL	-1	Label color. [6]
LABLWT	1	Label line weight. [1]
LABSIZ	1	Label size. [2]
LINSIZ	1	Plot line pattern size. [7]
LINTYPE	0	Plot line type: 0 = solid lines, 1 = dotted lines, >1 = dashed lines.
MAGNIFY	1	Magnify entire plot by this scale factor.
MHALIGN	0	Interpret MXLLC as a position for alternate message string horizontal alignment: 0 = Normal, 1 = Left, 2 = Center, 3 = Right.
MHKEY	3	Controls horizontal size of shade boxes and lines in messages and axis labels. [2]
MHSP	0	Controls horizontal spacing between characters in messages and axis labels.
MHXP	1	Controls horizontal expansion of characters in messages and axis labels relative to their height.
MSGANGLE	0	Message angle (degrees counterclockwise from x-axis).
MSGCOL	-1	Message color. [6]
MSGDAT	0	Interpret MXLLC and MYLLC in coordinate system of the data instead of inches relative to XLLC and YLLC. [3]
MSGLWT	1	Message line weight. [1]
MSGSIZ	1	Message size. [2]
MSLANT	0	Italic slant of characters in messages in degrees clockwise from the upright.
MVALIGN	0	Interpret MYLLC as a position for alternate message string vertical alignment: 0 = Normal, 1 = Top, 2 = Cap, 3 = Half, 4 = Base, 5 = Bottom.
MVKEY	1	Controls vertical size of shade boxes and symbols in messages and axis labels. [2]
MVSP	1.64	Controls vertical spacing between lines in messages and axis labels.
MXDAT	0	Interpret MXLLC in coordinate system of the data instead of inches relative to XLLC. [3]
MXLLC	0	Message location in the x direction (inches).

MYDAT	0	Interpret MYLLC in coordinate system of the data instead of inches relative to YLLC. [3]
MYLLC	0	Message location in the y direction (inches).
OPENAX	0	Leave the top and right side of axes open. [3]
PLTCOL	-1	Plot color. [6]
PLTLWT	1	Plot line weight. [1]
PLTYPE	3	Plot type: 1 = symbols only, 2 = lines only, 3 = both symbols and lines, 4 = histogram, 5 = rectangles, 6 = polygon, 7 = wave, and 8 = contiguous rectangles.
RANSKP	0	Right annotation skip. [5]
RBREAK	0	Right-axis break location (percent).
RCYCLE	0	Number of logarithmic cycles in the right axis.
RINT	5	Right intervals for tick marks. [4]
RMAX	0	Right axis maximum value.
RMIN	0	Right axis minimum value.
ROTATE	0	Rotate plot; 1 = 90 degrees, 2 = 180 degrees, and 3 = 270 degrees. [3]
RPERCENT	100	Percent of right axis used to position the limits.
RSHIFT	0	Right-axis shift of tick marks and annotations (percent).
RTICK	1	Controls length of the tick marks on the right axis. [3]
SHADE	0	Shade in the histogram bars: 0 = Hollow, 1 = Solid, (2-39) = Hatch. SHADE value 40 fills the area with a gray level between black and white (see SHDGRAY). SHADE values 41-79 are the same as 1-39 except that the interior is filled with gray before the hatch pattern and outline are drawn.
SHDCOL	-1	Shade color. [6]
SHDGRAY	1	Gray area fill for SHADE values 41-79. The gray values range from 0 (black) to 1 (white).
SHDLWT	1	Shade hatch line line weight. [1]
SIZFAC	0	Size factor. Controls size of everything. For negative values PLT does not change the sizes from frame to frame. Positive values instruct PLT to compute sizes as if average axis length was equal to SIZFAC.
SMOOTH	0	Smooth the data. [3]
SOLID	0	Solid symbols are produced if possible. [3]
SORT	0	Sort data in order of increasing x values. [3]
SPLINE	0	Draw a B-spline instead of connecting points for PLTYPE=2 or 3.
SYMANG	0	Rotation angle of symbol (degrees counter-clockwise) relative to normal orientation.
SYMBOL	1	Symbol represents the number of symbol to plot (0-29).
SYMSIZ	1	Symbol size. [2]
TANSKP	0	Top annotation skip. [5]
TBREAK	0	Top-axis break location (percent).
TCYCLE	0	Number of logarithmic cycles in the top axis.
TICDIR	0	Tick direction: value or "inward" produce tick marks inward otherwise they are outward.
TICSIZ	1	Controls length of the tick marks on all axes. [2]

TINT	5	Top intervals for tick marks. [4]
TMAX	0	Top axis maximum value.
TMIN	0	Top axis minimum value.
TPERCENT	100	Percent of top axis used to position the limits.
TSHIFT	0	Top-axis shift of tick marks and annotations (percent).
TTICK	1	Controls length of the tick marks on the top axis. [3]
VXLLC	0	X dimension virtual page offset (inches).
VYLLC	0	Y dimension virtual page offset (inches).
WXLEN	10.5	X dimension virtual page length (inches).
WYLEN	8	Y dimension virtual page length (inches).
XANSKP	0	X-axis annotation skip. [5]
XBREAK	0	X-axis break location (percent).
XCYCLE	0	Number of logarithmic cycles in the x axis.
XGAIN	1	Scale factor applied to X data before plotting.
XINT	5	X axis intervals. [4]
XLEN	6	X axis length (inches).
XLLC	2.2	Location of lower-left corner of axes in x direction (inches).
XMAX	10	X axis maximum value.
XMIN	0	X axis minimum value.
XOFST	0	Offset applied to x data before plotting.
XPERCENT	100	Percent of x axis used to position the limits.
XSHIFT	0	X-axis shift of tick marks and annotations (percent).
XTICK	1	Controls length of the tick marks on the bottom axis. [3]
YANSKP	0	Y-axis annotation skip. [5]
YBREAK	0	Y-axis break location (percent).
YCYCLE	0	Number of logarithmic cycles in the y axis.
YGAIN	1	Scale factor applied to Y data before plotting.
YHOR	0	Place the labels on the y axis horizontally. [3]
YINT	5	Y axis intervals. [4]
YLEN	6	Y axis length (inches).
YLLC	1.5	Location of lower-left corner of axes in y direction (inches).
YMAX	10	Y axis maximum value.
YMIN	0	Y axis minimum value.
YOFST	0	Offset applied to Y data before plotting.
YPERCENT	100	Percent of y axis used to position the limits.
YSHIFT	0	Y-axis shift of tick marks and annotations (percent).
YTICK	1	Controls length of the tick marks on the left axis. [3]
ZGAIN	1	Scale factor applied to Z data before plotting.
ZOFST	0	Offset applied to Z data before plotting.

- 
- Note 1: Negative values select absolute line weights, positive values multiply the default line weight, and zero eliminates the lines.
- Note 2: Positive values are used as multipliers of default size. Negative values are used as absolute size in inches.
- Note 3: YES = Y = 1 and NO = N = 0.

- Note 4: Intervals for tick marks are of the form MM.N where MM and N are the number of major and minor intervals respectively. Zero value indicates axis line with no tick marks.
- Note 5: If set to zero no annotations are to be skipped. Positive values indicate the number of annotations to be skipped. The first digit after the decimal point indicates an offset to the first annotation not skipped. Negative values produce no annotations.
- Note 6: The interpretation of color is device dependent. A small positive value selects a pen number (1-9) on pen plotter devices (HPGL and Tektronix) and selects a predefined color index (0-15) on other devices that support color output (PC, PostScript and CGM). A negative value for fgcol, bgcol and fncol indicates that device default colors should be used. A negative value for all other color parameters indicates that the color should be set to the same value as fgcol.
- Note 7: The line pattern size controls the length of the dash plus space for line types greater than (or equal to) 1. Negative values select absolute pattern size in inches, positive values multiply the default pattern size.

### USER VARIABLES

Users may specify variables for storage of values and for use in expressions. These variables may be named with the letters of the alphabet (a-z, only; case is ignored). The variables are set by typing the name followed by an equal sign, followed by an expression (e.g. a=2.3). The expression itself may contain values of numerical parameters (e.g. l=xlen, b=(xmax-xmin)). The user variables retain their values until they are reset; they are unaffected by the PLOT keyword.

### STRING PARAMETERS

String parameters are followed by an equals sign ("=") and a string.

RANNOT	no substitution	Right axis annotation.
RFMT	IRight axis annotation format.	
RLABEL	noneRight axis label.	
TANNOT	no substitution	Top axis annotation.
TFMT	ITop axis annotation format.	
TLABEL	noneTop axis label.	
XANNOT	no substitution	X axis annotation.
XFMT	IX axis annotation format.	
XLABEL	noneX axis label.	
YANNOT	no substitution	Y axis annotation.
YFMT	IY axis annotation format.	
YLABEL	noneY axis label.	
XDATA	(datum * XGAIN) + XOFST	X data definition.
YDATA	(datum * YGAIN) + YOFST	Y data definition.
ZDATA	(datum * YGAIN) + ZOFST	Z data definition.

**MESSAGES**

Messages are quoted strings which may be several lines long. The position of the message may be specified immediately preceding the open quote on the first line of the message or by setting the parameters MXLLC and MYLLC.

**MACROS**

Macros are invoked by a line that begins with "%macroname" followed by up to 9 arguments (delimited by spaces). There are two internal macros %define and %repeat. New macros may be defined using the %define macro. The first argument of %define is the new macro name (without the % character) and the second argument may contain the body of the new macro. If the %define macro has no second argument, the body of the new macro will be obtained from the succeeding lines of the pltfile up to a line that begins with "%%". Occurrences of \$1, \$2, ... \$9 within the body of the macro are replaced by arguments when the new macro is invoked. The %repeat macro is similar to the %define macro except that the first argument is a repeat count instead of a new macro name. The argument \$0 may be used in %repeat as the index of the current iteration.

**EXTERNAL FONTS**

The internal definitions that plt program normally uses for symbols and text characters can be replaced by loading external definitions from plt font files. Use SYMFNT to redefine symbols, STDFNT to redefine the standard character set and ALTFNT to replace the alternate character set (usually Greek letters). The plt font files are created and modified by a separate program (pltfont).

**BINARY DATA**

Plt can read binary data files using the BDATA command. The binary data file must have a short header followed by integer data values. All integer values are assumed to be stored in the file with the least-significant byte first. This is the standard integer storage on the VAX and PC, but not on the NeXT. The header contains four two-byte integers: (1) a magic number for identification = 12345 decimal, (2) the number of bytes per integer value in the data (must be 2), (3) the number of coordinate values (x, y, z) to be specified for each data point (must be 1, 2 or 3), and (4) the number of data points in the file.

**SPECIAL CHARACTERS**

Some characters have special meaning in messages or labels.

[str]	Subscript <i>str</i> .
{str}	Superscript <i>str</i> .
^str^	Shift to alternate font for <i>str</i> .
str	If <i>str</i> is null or "*", then the current symbol is inserted. If <i>str</i> is a number (0-31), then the symbol with the corresponding number is inserted. If the number is preceded by an "_", then a line with corresponding line type is inserted. If the number is preceded by a "#", then a box with corresponding shade type is inserted. Otherwise, <i>str</i> and the surrounding " " are taken literally.
\$	Replace "\$" in string with space.
;	Begin comment.
!	Begin comment and write comment to error output.
\c	Insert character <i>c</i> without special meaning indicated above, where <i>c</i> is [ ] { } ^ \$   ; ! " or \. A special case is \n which forces a new line in messages and axis labels.
*	obtain datum from keyboard, if the console screen is not the output device.

**CURSOR MODE**

If the plt program is writing directly to a PC graphics adaptor, then the user can enter a special cursor mode to identify the x,y coordinates of any position on the screen. The cursor mode is activated by pressing '+' (instead of [Enter]) when the 'beep' sounds at the end of any plotted page. A cursor should appear at the position 1 inch from the bottom and 1 inch from the left side. The cursor can be moved in 0.01 inch increments using the cursor arrow keys. Holding the shift key increases the increment to 0.5 inch. The coordinates are displayed (in inches) in the upper-right corner of the screen. The reference point can be changed to a new position by pressing [Ins] or reset to the lower-left corner of the page by pressing [Del]. Exit cursor mode by pressing '+' or [Esc].

**PLOTTER SUPPORT**

The plt program can directly drive an HPGL pen plotter (such as the HP-7475) attached to a serial port on the PC. Make sure that the serial cable includes the wires needed for hardware handshaking and that the DOS "mode" command has been used to indicate that DOS should continue to retry output on time-out errors. For example, if the plotter is on COM1: and its switches are set to run at 9600 baud, use the following mode command:

```
mode com1:9600,N,8,1,p
```

Then redirect the output of the plt program to COM1: using the HPGL output mode.

```
plt -m3 foo > com1:
```

The procedure should be similar (although not tested at BTNRH) to output directly to a Tektronix 4663 pen plotter (-m0) or an Imagen laser printer (-m2).

**PRINTER SUPPORT**

Two printers are supported indirectly as output devices by using additional programs to translate the IMPRESS output of the plt program: the EPSON FX-80 and the HP Laserjet II. Several other printer models are compatible with these printers and can also be used. The printers are assumed to be attached to LPT1:.

The Laserjet must have sufficient memory (at least 1 Megabyte) to hold a full page of graphics. The output of the plt program (-m2) is translated for the Laserjet by a program called "imp2lj". The batch files pltlj.bat (for 75 dpi) and pltljq.bat (for 300 dpi) can be used to automate the use of imp2lj.

```
pltljq foo
```

The other supported printer, the Epson FX, uses a translating program called "imp2fx". The batch files pltfx (single density) and pltfxq (quad density) are provided to automate the use of imp2fx.

**NAME**

plt - draw a graph

**SYNOPSIS**

plt [ options ] pltfile ...

**DESCRIPTION**

Plt causes the named pltfiles to be plotted on the indicated output device. The options are:

- cm**            Use centimeters instead of inches as the basic unit.
- cgm**           Writes CGM (computer graphics metafile) to stdout.
- e**             Causes the input file to be echoed on the error output.
- eps**           Writes encapsulated PostScript to stdout.
- i**             Specifies the Imagen printer as the output device.
- mm**           Use millimeters instead of inches as the basic unit.
- n**             Send the output to /dev/null instead of plotting.
- nw**           Specifies the NeXT display as the output device. This is also the default output device, if plt is executed from the console.
- o file**        Put the tek plotter language output in a file instead of plotting.
- p**             Specifies the NeXT printer as the output device.
- t**             Specifies /dev/tty as the output device and assumes a Tektronix graphics terminal.
- te**            Speeds plotting by replacing all text strings with rectangles which represent the text extent.
- v**             Print PLT program version.

Plt files are ASCII files with either a keyword, a transfer of control, a numeric parameter, a string parameter, a message, a macro or data values on each line. Only the first four characters are necessary to specify a keyword or parameter name.

**KEYWORDS**

Keywords have no argument or value associated with them but cause an action or change of mode.

DATA	Indicates the following lines are data points.
DATLIST	Prints contents of data buffer.
FINISH	Terminates and closes plot file.
KEEP	Plots data and keeps data for replotting.
NEWFRAME	Begin a new plot frame on the current page.
NEWPAGE	Advances to a new page.
NEWRANGE	Begin a new range of X and/or Y values in the current frame.
PALETTE	Uses data to set RGB color palette.
PLOT	Plots data and frame if necessary.
PRMLIST	Prints out the current parameter values.
RESET	Sets all parameters to their default values.
VARLIST	Prints out the current user variable values.

**AUXILIARY FILES**

These commands have a file name as an argument. If "." does not appear in the filename then a default extension is added.

BDATA	Includes specified file as a binary data file.
CALL	Transfers control to another file.
INCLUDE	Includes specified file as a "plt" file.
STDFNT	Load external font file as the standard font.
ALTFNT	Load external font file as the alternate font.
SYMFNT	Load external font file as the symbol font.

**NUMERIC PARAMETERS**

Numeric parameters are followed by "=" and either a number or a word. Default values are given below after each parameter name. The numbers in brackets refer to Notes at the end of this section.

ANNCOL	-1	Annotation color. [6]
ANNLWT	1	Annotation line weight. [1]
ANNSIZ	1	Annotation size. [2]
AXCOL	-1	Axes color. [6]
AXLWT	1	Axes line weight. [1]
BGCOL	-1	Background color. [6]
CLIP	0	Clip lines at frame boundary. [3]
CORNER	1	Draw axis frame corners when xpercent or ypercent are less than 100. [3]
EBNUM	2	Number of error bars: 0 = none, 1 = upper bar only, 2 = both upper and lower, -1 = lower bar only.
EBWID	1	Width of error bars scale factor. [2]
ECHO	0	Echo each input line read. [3]

PLT(1)		UNIX Programmers' Manual	PLT(1)
FGCOL	-1	Foreground color. [6]	
FNCOL	-1	Fountain color. Specifies a different background color at the top of the page which gradually changes to BGCOL toward the bottom of the page. [6]	
GAPSIZE	1	Controls size of gap between line segments for plotting symbols. [2]	
GRDCOL	-1	Grid color. [6]	
GRDLWT	1	Grid line weight. [1]	
GRDSIZ	1	Grid line pattern size. [7]	
GRDTYPE	1	Grid line type: 0 = solid lines, 1 = dotted lines, >1 = dashed lines.	
GRID	0	Plot grid lines at major intervals. [3]	
HEADER	1	Date, time, filename and page number printed at bottom of plot. [3]	
LABCOL	-1	Label color. [6]	
LABLWT	1	Label line weight. [1]	
LABSIZ	1	Label size. [2]	
LINSIZ	1	Plot line pattern size. [7]	
LINTYPE	0	Plot line type: 0 = solid lines, 1 = dotted lines, >1 = dashed lines.	
MAGNIFY	1	Magnify entire plot by this scale factor.	
MHALIGN	0	Interpret MXLLC as a position for alternate message string horizontal alignment: 0=Normal, 1=Left, 2=Center, 3=Right.	
MHKEY	3	Controls horizontal size of shade boxes and lines in messages and axis labels. [2]	
MHSP	0	Controls horizontal spacing between characters in messages and axis labels.	
MHXP	1	Controls horizontal expansion of characters in messages and axis labels relative to their height.	
MSGANGLE	0	Message angle (degrees counterclockwise from x-axis).	
MSGCOL	-1	Message color. [6]	
MSGDAT	0	Interpret MXLLC and MYLLC in coordinate system of the data instead of inches relative to XLLC and YLLC. [3]	
MSGLWT	1	Message line weight. [1]	
MSGSIZ	1	Message size. [2]	
MSLANT	0	Italic slant of characters in messages in degrees clockwise from the upright.	
MVALIGN	0	Interpret MYLLC as a position for alternate message string vertical alignment: 0 = Normal, 1 = Top, 2 = Cap, 3 = Half, 4 = Base, 5 = Bottom.	
MVKEY	1	Controls vertical size of shade boxes and symbols in messages and axis labels. [2]	

PLT(1)	UNIX Programmers' Manual	PLT(1)
MVSP	1.64	Controls vertical spacing between lines in messages and axis labels.
MXDAT	0	Interpret MXLLC in coordinate system of the data instead of inches relative to XLLC. [3]
MXLLC	0	Message location in the x direction (inches).
MYDAT	0	Interpret MYLLC in coordinate system of the data instead of inches relative to YLLC. [3]
MYLLC	0	Message location in the y direction (inches).
OPENAX	0	Leave the top and right side of axes open. [3]
PLTCOL	-1	Plot color. [6]
PLTLWT	1	Plot line weight. [1]
PLTYPE	3	Plot type: 1 = symbols only, 2 = lines only, 3 = both symbols and lines, 4 = histogram, 5 = rectangles, 6 = polygon, 7 = wave, and 8 = contiguous rectangles.
RANSKP	0	Right annotation skip. [5]
RBREAK	0	Right-axis break location (percent).
RCYCLE	0	Number of logarithmic cycles in the right axis.
RINT	5	Right intervals for tick marks. [4]
RMAX	0	Right axis maximum value.
RMIN	0	Right axis minimum value.
ROTATE	0	Rotate plot; 1 = 90 degrees, 2 = 180 degrees, and 3 = 270 degrees. [3]
RPERCENT	100	Percent of right axis used to position the limits.
RSHIFT	0	Right-axis shift of tick marks and annotations (percent).
RTICK	1	Controls length of the tick marks on the right axis. [3]
SHADE	0	Shade in the histogram bars: 0 = Hollow, 1 = Solid, (2-39) = Hatch. SHADE value 40 fills the area with a gray level between black and white (see SHDGRAY). SHADE values 41-79 are the same as 1-39 except that the interior is filled with gray before the hatch pattern and outline are drawn.
SHDCOL	-1	Shade color. [6]
SHDGRAY	1	Gray area fill for SHADE values 41-79. The gray values range from 0 (black) to 1 (white).
SHDLWT	1	Shade hatch line line weight. [1]
SIZFAC	0	Size factor. Controls size of everything. For negative values PLT does not change the sizes from frame to frame. Positive values instruct PLT to compute sizes as if average axis length was equal to SIZFAC.
SMOOTH	0	Smooth the data. [3]
SOLID	0	Solid symbols are produced if possible. [3]
SORT	0	Sort data in order of increasing x values. [3]
SPLINE	0	Draw a B-spline instead of connecting points for PLTYPE=2 or 3.

PLT(1)	UNIX Programmers' Manual	PLT(1)
SYMANG	0	Rotation angle of symbol (degrees counter-clockwise) relative to normal orientation.
SYMBOL	1	Symbol represents the number of symbol to plot (0-29).
SYMSIZ	1	Symbol size. [2]
TANSKP	0	Top annotation skip. [5]
TBREAK	0	Top-axis break location (percent).
TCYCLE	0	Number of logarithmic cycles in the top axis.
TICDIR	0	Tick direction: value or "inward" produce tick marks inward; otherwise, they are outward.
TICSIZ	1	Controls length of the tick marks on all axes. [2]
TINT	5	Top intervals for tick marks. [4]
TMAX	0	Top axis maximum value.
TMIN	0	Top axis minimum value.
TPERCENT	100	Percent of top axis used to position the limits.
TSHIFT	0	Top-axis shift of tick marks and annotations (percent).
TTICK	1	Controls length of the tick marks on the top axis. [3]
VXLLC	0	X dimension virtual page offset (inches).
VYLLC	0	Y dimension virtual page offset (inches).
WXLEN	10.5	X dimension virtual page length (inches).
WYLEN	8	Y dimension virtual page length (inches).
XANSKP	0	X-axis annotation skip. [5]
XBREAK	0	X-axis break location (percent).
XCYCLE	0	Number of logarithmic cycles in the x axis.
XGAIN	1	Scale factor applied to X data before plotting.
XINT	5	X axis intervals. [4]
XLEN	6	X axis length (inches).
XLLC	2.2	Location of lower-left corner of axes in x direction (inches).
XMAX	10	X axis maximum value.
XMIN	0	X axis minimum value.
XOFST	0	Offset applied to X data before plotting.
XPERCENT	100	Percent of x axis used to position the limits.
XSHIFT	0	X-axis shift of tick marks and annotations (percent).
XTICK	1	Controls length of the tick marks on the bottom axis. [3]
YANSKP	0	Y-axis annotation skip. [5]
YBREAK	0	Y-axis break location (percent).
YCYCLE	0	Number of logarithmic cycles in the y axis.
YGAIN	1	Scale factor applied to Y data before plotting.
YHOR	0	Place the labels on the y axis horizontally. [3]
YINT	5	Y axis intervals. [4]
YLEN	6	Y axis length (inches).
YLLC	1.5	Location of lower-left corner of axes in y direction (inches).

YMAX	10	Y axis maximum value.
YMIN	0	Y axis minimum value.
YOFST	0	Offset applied to Y data before plotting.
YPERCENT	100	Percent of y axis used to position the limits.
YSHIFT	0	Y-axis shift of tick marks and annotations (percent).
YTICK	1	Controls length of the tick marks on the left axis. [3]
ZGAIN	1	Scale factor applied to Z data before plotting.
ZOFST	0	Offset applied to Z data before plotting.

- 
- Note 1: Negative values select absolute line weights, positive values multiply the default line weight and zero eliminates the lines.
- Note 2: Positive values are used as multipliers of default size. Negative values are used as absolute size in inches.
- Note 3: YES = Y = 1 and NO = N = 0.
- Note 4: Intervals for tick marks are of the form MM.N where MM and N are the number of major and minor intervals respectively. Zero value indicates axis line with no tick marks.
- Note 5: If set to zero no annotations are to be skipped. Positive values indicate the number of annotations to be skipped. The first digit after the decimal point indicates an offset to the first annotation not skipped. Negative values produce no annotations.
- Note 6: The interpretation of color is device dependent. A small positive value selects a pen number (1-9) on pen plotter devices (HPGL and Tektronix) and selects a predefined color index (0-15) on other devices that support color output (PC, PostScript and CGM). A negative value for fgcol, bgcol and fncol indicates that device default colors should be used. A negative value for all other color parameters indicates that the color should be set to the same value as fgcol.
- Note 7: The line pattern size controls the length of the dash plus space for line types greater than (or equal to) 1. Negative values select absolute pattern size in inches, positive values multiply the default pattern size.

## USER VARIABLES

Users may specify variables for storage of values and for use in expressions. These variables may be named with the letters of the alphabet (a-z, only; case is ignored). The variables are set by typing the name followed by an equal sign, followed by an expression (e.g. a=2.3). The expression itself may contain values of numerical parameters (e.g. l=xlen, b=(xmax-xmin)). The user variables retain their values until they are reset; they are unaffected by the PLOT keyword.

**STRING PARAMETERS**

String parameters are followed by an equals sign ("=") and a string.

RANNOT	no substitution	Right axis annotation.
RFMT		IRight axis annotation format.
RLABEL	none	Right axis label.
TANNOT	no substitution	Top axis annotation.
TFMT		ITop axis annotation format.
TLABEL	none	Top axis label.
XANNOT	no substitution	X axis annotation.
XFMT		IX axis annotation format.
XLABEL	none	X axis label.
YANNOT	no substitution	Y axis annotation.
YFMT		IY axis annotation format.
YLABEL	none	Y axis label.
XDATA	(datum * XGAIN) + XOFST	X data definition.
YDATA	(datum * YGAIN) + YOFST	Y data definition.
ZDATA	(datum * YGAIN) + ZOFST	Z data definition.

**MESSAGES**

Messages are quoted strings which may be several lines long. The position of the message may be specified immediately preceding the open quote on the first line of the message or by setting the parameters MXLLC and MYLLC.

**MACROS**

Macros are invoked by a line that begins with "%macroname" followed by up to 9 arguments (delimited by spaces). There are two internal macros %define and %repeat. New macros may be defined using the %define macro. The first argument of %define is the new macro name (without the % character) and the second argument may contain the body of the new macro. If the %define macro has no second argument, the body of the new macro will be obtained from the succeeding lines of the pltfile up to a line that begins with "%%". Occurrences of \$1, \$2, ... \$9 within the body of the macro are replaced by arguments when the new macro is invoked. The %repeat macro is similar to the %define macro except that the first argument is a repeat count instead of a new macro name. The argument \$0 may be used in %repeat as the index of the current iteration.

**EXTERNAL FONTS**

The internal definitions that `plt` program normally uses for symbols and text characters can be replaced by loading external definitions from `plt` font files. Use `SYMFNT` to redefine symbols, `STDFNT` to redefine the standard character set and `ALTFNT` to replace the alternate character set (usually Greek letters). The `plt` font files are created and modified by a separate program (`pltfont`).

**BINARY DATA**

`Plt` can read binary data files using the `BDATA` command. The binary data file must have a short header followed by integer data values. All integer values are assumed to be stored in the file with the least-significant byte first. This is the standard integer storage on the VAX and PC, but not on the NeXT. The header contains four two-byte integers: (1) a magic number for identification = 12345 decimal, (2) the number of bytes per integer value in the data (must be 2), (3) the number of coordinate values (x, y, z) to be specified for each data point (must be 1, 2 or 3), and (4) the number of data points in the file.

**SPECIAL CHARACTERS**

Some characters have special meaning in messages or labels.

<code>[str]</code>	Subscript <i>str</i> .
<code>{str}</code>	Superscript <i>str</i> .
<code>^str^</code>	Shift to alternate font for <i>str</i> .
<code> str </code>	If <i>str</i> is null or "*", then the current symbol is inserted. If <i>str</i> is a number (0-31), then the symbol with the corresponding number is inserted. If the number is preceded by an "_" then a line with corresponding line type is inserted. If the number is preceded by a "#" then a box with corresponding shade type is inserted. Otherwise, <i>str</i> and the surrounding " " are taken literally.
<code>\$</code>	Replace "\$" in string with space.
<code>;</code>	Begin comment.
<code>!</code>	Begin comment and write comment to error output.
<code>\c</code>	Insert character <i>c</i> without special meaning indicated above, where <i>c</i> is [ ] { } ^ \$   ; ! " or \. A special case is <code>\n</code> which forces a new line in messages and axis labels.
<code>*</code>	obtain datum from keyboard, if the console screen is not the output device.

**FILES**

/usr/local/bin/plt	shell script to direct output to the appropriate device.
/usr/local/bin/tplt	translates plt language files into device specific output.
/usr/local/lib/pltfont	external plt font files

**BUGS**

The interrupt signal (generated by ^C) is not always passed to the appropriate routine responsible for removing the plt window from the screen. It may sometimes be necessary to log out to remove the plt window.

## Appendix E:

### Permuted Index of PLT Parameters

The index below is designed to assist you in finding the parameter which controls a desired function. It consists of a simple one-line description for each PLT parameter. The difference between this and an ordinary index is that the line is repeated for each important word in the description. These important words are then alphabetized. Suppose you wished to find which parameter enabled you to set the horizontal alignment of a message. You could look it up in this index under "horiz" or "message". You could also look under "alignment" or "string".

size factor	affects all plot elements.	SIZFAC
message string horiz	alignment.	MHALIGN
message string vert	alignment.	MVALIGN
length of tick marks	(all axes).	TICSIZ
message	angle in deg cnterclockwise fr X axis.	MSGANGLE
symbol	angle in deg cnterclockwise fr normal.	SYMANG
	annotation color.	ANNCOL
	annotation line weight.	ANNLWT
	annotation size.	ANNSIZ
right (R)	annotation skip.	RANSKP
top (T) axis	annotation skip.	TANSKP
X axis	annotation skip.	XANSKP
Y axis	annotation skip.	YANSKP
right (R) axis shift of tick marks and	annotations.	RSHIFT
top (T) axis shift of tick marks and	annotations.	TSHIFT
X axis shift of tick marks and	annotations.	XSHIFT
Y axis shift of tick marks and	annotations.	YSHIFT
gray	area fill.	SHDGRAY
length of tick marks on all	axes.	TICSIZ
	axes color.	AXCOL
lower left corner of	axes in X direction.	XLLC
lower left corner of	axes in Y direction.	YLLC
	axes line weight.	AXLWT
	axes open.	OPENAX
leave the top and right side of	axis.	MSGANGLE
msg angle in deg cnterclockwise fr X	axis.	RCYCLE
number of log cycles in the right (R)	axis.	RTICK
length of tick marks on right (R)	axis.	TCYCLE
number of log cycles in the top (T)	axis.	TTICK
length of tick marks on top (T)	axis.	

number of log cycles in the X	axis.	XCYCLE
length of ticks on the bottom (X)	axis.	XTICK
number of log cycles in the Y	axis.	YCYCLE
length of tick marks on the left (Y)	axis.	YTICK
right (R)	axis annotation skip.	RANSKP
top (T)	axis annotation skip.	TANSKP
X	axis annotation skip.	XANSKP
Y	axis annotation skip.	YANSKP
right (R)	axis break location.	RBREAK
top (T)	axis break location.	TBREAK
X	axis break location (percent).	XBREAK
Y	axis break location (percent).	YBREAK
draw	axis frame corners.	CORNER
X	axis intervals.	XINT
Y	axis intervals.	YINT
right (R)	axis intervals for tick marks.	RINT
top (T)	axis intervals for tick marks.	TINT
Y	axis labels placed horizontally.	YHOR
X	axis length.	XLEN
Y	axis length.	YLEN
right (R)	axis maximum value.	RMAX
top (T)	axis maximum value.	TMAX
X	axis maximum value.	XMAX
Y	axis maximum value.	YMAX
right (R)	axis minimum value.	RMIN
top (T)	axis minimum value.	TMIN
X	axis minimum value.	XMIN
Y	axis minimum value.	YMIN
right (R)	axis shift of tick marks and annotations.	RSHIFT
top (T)	axis shift of tick marks and annotations.	TSHIFT
X	axis shift of tick marks and annotations.	XSHIFT
Y	axis shift of tick marks and annotations.	YSHIFT
percent of right (R)	axis used to position limits.	RPERCENT
percent of top (T)	axis used to position the limits.	TPERCENT
percent of X	axis used to position the limits.	XPERCENT
percent of Y	axis used to position the limits.	YPERCENT
	background color.	BGCOL
number of error	bars.	EBNUM
width of error	bars scale factor.	EBWID
shade in the histogram	bars/rectangles/polygons.	SHADE
length of tick marks on the	bottom (X) axis.	XTICK
date, time, filename, etc. printed at	bottom of plot.	HEADER
clip lines at frame	boundary.	CLIP
horiz size of shade	boxes and lines in messages.	MHKEY
vert size of shade	boxes and symbols in messages.	MVKEY
right-axis	break location.	RBREAK
top (T) axis	break location.	TBREAK
Y axis	break location.	YBREAK
X axis	break location (percent).	XBREAK
italic slant of	characters in messages.	MSLANT
horiz spacing between	chars in messages.	MHSP
horiz expansion of	chars in messages.	MHXP
	clip lines at frame boundary.	CLIP
message angle in deg	counterclockwise from X axis).	MSGANGLE

symbol angle in deg	cntrclckwise fr normal.	SYMANG
annotation	color.	ANNCOL
axes	color.	AXCOL
background	color.	BGCOL
foreground	color.	FGCOL
fountain	color.	FNCOL
grid	color.	GRDCOL
label	color.	LABCOL
message	color.	MSGCOL
plot	color.	PLTCOL
shade	color.	SHDCOL
spline (B-spline) curve for	connecting points.	SPLINE
interpret mxllc/myllc in	coord system of data.	MSGDAT
interpret mxllc in	coord system of data.	MXDAT
interpret myllc in	coord system of data.	MYDAT
lower left	corner of axes in X direction.	XLLC
lower left	corner of axes in Y direction.	YLLC
draw axis frame	corners.	CORNER
spline (B-spline)	curve for connecting points.	SPLINE
number of log	cycles in the X axis.	XCYCLE
number of log	cycles in the Y axis.	YCYCLE
number of log	cycles in the right (R) axis.	RCYCLE
number of log	cycles in the top (T) axis.	TCYCLE
interp mx/myllc in coord system of	data.	MSGDAT
interpret mxllc in coord system of	data.	MXDAT
interpret myllc in coord system of	data.	MYDAT
smooth the	data.	SMOOTH
scale factor applied to X	data before plotting.	XGAIN
offset applied to X	data before plotting.	XOFST
scale factor applied to Y	data before plotting.	YGAIN
offset applied to Y	data before plotting.	YOFST
scale factor applied to Z	data before plotting.	ZGAIN
offset applied to Z	data before plotting.	ZOFST
sort	data in order of increasing X values.	SORT
printed at bottom of plot.	date, time, filename, etc.	HEADER
message angle in	deg cntrclckwise fr X axis).	MSGANGLE
symbol angle in	deg cntrclckwise fr normal.	SYMANG
rotate plot by 90	degrees.	ROTATE
X	dimension virtual page length.	WXLEN
Y	dimension virtual page length.	WYLEN
X	dimension virtual page offset.	VXLLC
Y	dimension virtual page offset.	VYLLC
message location in the X	direction.	MXLLC
message location in the Y	direction.	MYLLC
tick	direction.	TICDIR
lower left corner of axes in X	direction.	XLLC
lower left corner of axes in Y	direction.	YLLC
	draw axis frame corners.	CORNER
	echo each input line read.	ECHO
size factor affects all plot	elements.	SIZFAC
magnify	entire plot by this scale factor.	MAGNIFY
number of	error bars.	EBNUM
width of	error bars scale factor.	EBWID
horiz	expansion of chars in messages.	MHXP

date, time, gray area	filename, etc. printed at bottom of plot. fill.	HEADER SHDGRAY
	foreground color.	FGCOL
	fountain color.	FNCOL
message angle in deg cntclockwise	fr X axis.	MSGANGLE
symbol angle in deg cntclockwise	fr normal.	SYMANG
clip lines at	frame boundary.	CLIP
draw axis	frame corners.	CORNER
size of	gap between plot symbol line segments.	GAPSIZE
	gray area fill.	SHDGRAY
	grid color.	GRDCOL
	grid line pattern size.	GRDSIZ
	grid line type.	GRDTYPE
	grid line weight.	GRDLWT
plot	grid lines at major intervals.	GRID
shade	hatch line line weight.	SHDLWT
shade in the	histogram.	SHADE
message string	horiz alignment.	MHALIGN
	horiz expansion of chars in messages.	MHXP
	horiz size of shade boxes and lines in msgs.	MHKEY
	horiz spacing between chars in messages.	MHSP
Y axis labels placed	horizontally.	YHOR
	interpret mxllc in coord system of data.	MXDAT
	interpret mxllc/myllc in coord syst of data.	MSGDAT
	interpret myllc in coord system of data.	MYDAT
plot grid lines at major	intervals.	GRID
X axis	intervals.	XINT
Y axis	intervals.	YINT
right(R) axis	intervals for tick marks.	RINT
top (T) axis	intervals for tick marks.	TINT
	italic slant of characters in messages.	MSLANT
	label color.	LBCOL
	label line weight.	LABLWT
	label size.	LABSIZ
Y axis	labels placed horizontally.	YHOR
	leave the top and right side of axes open.	OPENAX
length of tick marks on the	left (Y) axis.	YTICK
lower	left corner of axes in X direction.	XLLC
lower	left corner of axes in Y direction.	YLLC
X dimension virtual page	length.	WXLEN
Y dimension virtual page	length.	WYLEN
X axis	length.	XLEN
Y axis	length.	YLEN
	length of tick marks (all axes).	TICSIZ
	length of tick marks on right (R) axis.	RTICK
	length of tick marks on top (T) axis.	TTICK
	length of tick marks on the bottom (X) axis.	XTICK
	length of tick marks on the left (Y) axis.	YTICK
shade hatch	line line weight.	SHDLWT
grid	line pattern size.	GRDSIZ
plot	line pattern size.	LINSIZ
echo each input	line read.	ECHO
size of gap between plot symbol	line segments.	GAPSIZE
grid	line type.	GRDTYPE

plot	line type.	LINTYPE
annotation	line weight.	ANNLWT
axes	line weight.	AXLWT
grid	line weight.	GRDLWT
label	line weight.	LABLWT
message	line weight.	MSGLWT
plot	line weight.	PLTLWT
shade hatch line	line weight.	SHDLWT
clip	lines at frame boundary.	CLIP
plot grid	lines at major intervals.	GRID
horiz size of shade boxes and	lines in messages.	MHKEY
vert spacing between	lines in messages.	MVSP
right (R) axis	location.	RBREAK
top (T) axis break	location.	TBREAK
X axis break	location (percent).	XBREAK
Y axis break	location (percent).	YBREAK
message	location in the X direction.	MXLLC
message	location in the Y direction.	MYLLC
number of	log cycles in the X axis.	XCYCLE
number of	log cycles in the Y axis.	YCYCLE
number of	log cycles in the right (R) axis.	RCYCLE
number of	log cycles in the top (T) axis.	TCYCLE
	lower left corner of axes in X direction.	XLLC
	lower left corner of axes in Y direction.	YLLC
	magnify entire plot by this scale factor.	MAGNIFY
plot grid lines at	major intervals.	GRID
right intervals for tick	marks.	RINT
top (T) axis intervals for tick	marks.	TINT
length of tick	marks (all axes).	TICSIZ
right-axis shift of tick	marks and annotations.	RSHIFT
top (T) axis shift of tick	marks and annotations.	TSHIFT
X axis shift of tick	marks and annotations.	XSHIFT
Y axis shift of tick	marks and annotations.	YSHIFT
length of tick	marks on right (R) axis.	RTICK
length of tick	marks on top (T) axis.	TTICK
length of tick	marks on the bottom (X) axis.	XTICK
length of tick	marks on the left (Y) axis.	YTICK
right (R) axis	maximum value.	RMAX
top (T) axis	maximum value.	TMAX
X axis	maximum value.	XMAX
Y axis	maximum value.	YMAX
	message angle in deg ctrclckwise fr X axis.	MSGANGLE
	message color.	MSGCOL
	message line weight.	MSGLWT
	message location in the X direction.	MXLLC
	message location in the Y direction.	MYLLC
	message size.	MSGSIZ
	message string horiz alignment.	MHALIGN
	message string vert alignment.	MVALIGN
horiz size of shade boxes and lines in	messages.	MHKEY
horiz spacing between chars in	messages.	MHSP
horiz expansion of chars in	messages.	MHXP
italic slant of characters in	messages.	MSLANT
vert size shade boxes and symbols in	messages.	MVKEY

vert spacing between lines in right (R) axis top (T) axis X axis Y axis interpret interpret interpret rotate plot by symbol angle in deg cntreclkwise fr	messages. minimum value. minimum value. minimum value. minimum value. mxllc in coord system of data. mxllc/myllc in coord system of data. myllc in coord system of data. 90 degrees. normal. number of error bars. number of log cycles in the X axis. number of log cycles in the Y axis. number of log cycles in the right (R) axis. number of log cycles in the top (T) axis. number of symbol to plot. offset. offset. offset applied to X data before plotting. offset applied to Y data before plotting. offset applied to Z data before plotting. open. order of increasing X values. page length. page length. page offset. page offset. pattern size. pattern size. (percent). percent X axis used to position the limits. percent Y axis used to position the limits. percent right (R) axis used to posit limits. percent top (T) axis used to position limits. plot color. plot grid lines at major intervals. plot line pattern size. plot line type. plot line weight. plot type. position the limits. position the limits. position the limits. position the limits. printed at bottom of plot. produced if possible. (R) axis. (R) axis. (R) axis used to position limits. right-axis break location. right-axis shift of tick marks and annotations. rotate plot by 90 degrees. scale factor. scale factor.	MVSP RMIN TMIN XMIN YMIN MXDAT MSGDAT MYDAT ROTATE SYMANG EBNUM XCYCLE YCYCLE RCYCLE TCYCLE SYMBOL VXLLC VYLLC XOFST YOFST ZOFST OPENAX SORT WXLEN WYLEN VXLLC VYLLC GRDSIZ LINSIZ XBREAK XPERCENT YPERCENT RPERCENT TPERCENT PLTCOL GRID LINSIZ LINTYPE PLTLWT PLTYPE RPERCENT TPERCENT XPERCENT YPERCENT HEADER SOLID RCYCLE RTICK RPERCENT RBREAK RSHIFT ROTATE EBWID MAGNIFY
X dimension virtual page Y dimension virtual page		
leave the top and right side of axes sort data in X dimension virtual Y dimension virtual X dimension virtual Y dimension virtual grid line plot line X axis break location		
percent of right (R) axis used to percent of top (T) axis used to percent of X axis used to percent of Y axis used to date, time, filename, etc. solid symbols are number of log cycles in the right length of tick marks on right percent of right		
width of error bars magnify entire plot by this		

	scale factor applied to X data before plotting.	XGAIN
	scale factor applied to Y data before plotting.	YGAIN
	scale factor applied to Z data before plotting.	ZGAIN
size of gap between plot symbol line	segments.	GAPSIZE
horiz	size of shade boxes and lines in messages.	MHKEY
vert	size of shade boxes and symbols in messages.	MVKEY
	shade color.	SHDCOL
	shade hatch line line weight.	SHDLWT
	shade in the histogram bars/rects/polygons	SHADE
right (R) axis	shift of tick marks and annots.	RSHIFT
top (T) axis	shift of tick marks and annots.	TSHIFT
X axis	shift of tick marks and annotations.	XSHIFT
Y axis	shift of tick marks and annotations.	YSHIFT
leave the top and right	side of axes open.	OPENAX
annotation	size.	ANNSIZ
grid line pattern	size.	GRDSIZ
label	size.	LABSIZ
plot line pattern	size.	LINSIZ
message	size.	MSGSIZ
symbol	size.	SYMSIZ
	size factor affects all plot elements.	SIZFAC
	size of gap between plot symbol line segs.	GAPSIZE
horiz	size of shade boxes and lines in messages.	MHKEY
vert	size of shade boxes and symbols in messages	MVKEY
right annotation	skip.	RANSKP
top (T) axis annotation	skip.	TANSKP
X axis annotation	skip.	XANSKP
Y axis annotation	skip.	YANSKP
italic	slant of characters in messages.	MSLANT
	smooth the data.	SMOOTH
	solid symbols are produced if possible.	SOLID
	sort data in order of increasing X values.	SORT
horiz	spacing between chars in messages.	MHSP
vert	spacing between lines in messages.	MVSP
	spline (B-spline) curve for connecting points.	SPLINE
message	string horiz alignment.	MHALIGN
message	string vert alignment.	MVALIGN
	symbol angle in deg cntrclockwise fr normal.	SYMANG
size of gap between plot	symbol line segments.	GAPSIZE
	symbol size.	SYMSIZ
number of	symbol to plot.	SYMBOL
solid	symbols are produced if possible.	SOLID
vert size of shade boxes and	symbols in messages.	MVKEY
interpret mxllc/myllc in coord	system of data.	MSGDAT
interpret mxllc in coord	system of data.	MXDAT
interpret myllc in coord	system of data.	MYDAT
length of tick marks on top	(T) axis.	TTICK
number of log cycles in the top	(T) axis.	TCYCLE
right	(R) axis annotation skip.	RANSKP
right	(R) axis break location.	RBREAK
right	(R) axis intervals for tick marks.	RINT
right	(R) axis maximum value.	RMAX
right	(R) axis minimum value.	RMIN
right	(R) axis shift of tick marks and annotations.	RSHIFT

top	(T) axis annotation skip.	TANSKP
top	(T) axis break location.	TBREAK
top	(T) axis intervals for tick marks.	TINT
top	(T) axis maximum value.	TMAX
top	(T) axis minimum value.	TMIN
top	(T) axis shift of tick marks and annotations.	TSHIFT
percent of top	(T) axis used to position the limits.	TPERCENT
	tick direction.	TICDIR
right (R) intervals for	tick marks.	RINT
top (T) axis intervals for	tick marks.	TINT
length of	tick marks (all axes).	TICSIZ
right (R) axis shift of	tick marks and annotations.	RSHIFT
top (T) axis shift of	tick marks and annotations.	TSHIFT
X axis shift of	tick marks and annotations.	XSHIFT
Y axis shift of	tick marks and annotations.	YSHIFT
length of	tick marks on right (R) axis.	RTICK
length of	tick marks on right (T) axis.	TTICK
length of	tick marks on the bottom (X) axis.	XTICK
length of	tick marks on the left (Y) axis.	YTICK
date,	time, filename, etc. printed at bottom of plot.	HEADER
number of log cycles in the	top (T) axis.	TCYCLE
	right (R) axis annotation skip.	RANSKP
	right (R) axis break location.	RBREAK
	right (R) axis intervals for tick marks.	RINT
	right (R) axis maximum value.	RMAX
	right (R) axis minimum value.	RMIN
	right (R) axis shift of tick marks and annots.	RSHIFT
	top (T) axis annotation skip.	TANSKP
	top (T) axis break location.	TBREAK
	top (T) axis intervals for tick marks.	TINT
	top (T) axis maximum value.	TMAX
	top (T) axis minimum value.	TMIN
	top (T) axis shift of tick marks and annots.	TSHIFT
	top (T) axis used to position the limits.	TPERCENT
percent of	top (T) axis used to position the limits.	TPERCENT
leave the	top and right side of axes open.	OPENAX
grid line	type.	GRDTYPE
plot line	type.	LINTYPE
plot	type.	PLTYPE
message string	vert alignment.	MVALIGN
	vert size of shade boxes/symbols in msg.	MVKEY
	vert spacing between lines in messages.	MVSP
X dimension	virtual page length.	WXLEN
Y dimension	virtual page length.	WYLEN
X dimension	virtual page offset.	VXLLC
Y dimension	virtual page offset.	VYLLC
annotation line	weight.	ANNLWT
axes line	weight.	AXLWT
grid line	weight.	GRDLWT
label line	weight.	LABLWT
message line	weight.	MSGLWT
plot line	weight.	PLTLWT
shade hatch line line	weight.	SHDLWT
	width of error bars scale factor.	EBWID
message angle in deg cntcrckwise fr	X axis.	MSGANGLE

number of log cycles in the length of tick marks on the bottom	X axis.	XCYCLE
	(X) axis.	XTICK
	X axis annotation skip.	XANSKP
	X axis break location (percent).	XBREAK
	X axis intervals.	XINT
	X axis length.	XLEN
	X axis maximum value.	XMAX
	X axis minimum value.	XMIN
	X axis shift of tick marks and annotations.	XSHIFT
percent of scale factor applied to offset applied to	X axis used to position the limits.	XPERCENT
	X data before plotting.	XGAIN
	X data before plotting.	XOFST
	X dimension virtual page length.	WXLEN
	X dimension virtual page offset.	VXLLC
message location in the lower left corner of axes in sort data in order of increasing number of log cycles in the length of tick marks on the left	X direction.	MXLLC
	X direction.	XLLC
	X values.	SORT
	Y axis.	YCYCLE
	(Y) axis.	YTICK
	Y axis annotation skip.	YANSKP
	Y axis break location.	YBREAK
	Y axis intervals.	YINT
	Y axis labels placed horizontally.	YHOR
	Y axis length.	YLEN
	Y axis maximum value.	YMAX
	Y axis minimum value.	YMIN
	Y axis shift of tick marks and annotations.	YSHIFT
percent of scale factor applied to offset applied to	Y axis used to position the limits.	YPERCENT
	Y data before plotting.	YGAIN
	Y data before plotting.	YOFST
	Y dimension virtual page length.	WYLEN
	Y dimension virtual page offset.	VYLLC
message location in the lower left corner of axes in scale factor applied to offset applied to	Y direction.	MYLLC
	Y direction.	YLLC
	Z data before plotting.	ZGAIN
	Z data before plotting.	ZOFST

## INDEX (not including Chapter 4A)

- Axis
  - logarithmic 28
  - maximum 28
  - minimum 28
- Alternate font 10, 27
- ANNCOL [-1] 34
- ANNLWT [1] 29
- Annotations 29
  - character size 29
  - formats 29
  - line weight 29
  - manner 29
  - orientation 29
  - substitutions 29
- ANNSIZ [1] 29
- Arithmetic expressions 20
- ASCII 1, 102
- Asterisk (\*) 19, 22
- Auxiliary File Commands 26, 95, 103
- AXCOL [-1] 34
- Axis
  - corners 28
  - discontinuous 28
  - labels 29
    - height 29
    - line weight 30
  - length 27
  - line weight 29
  - open 29
  - percentage 28
  - range shift 28
- AXLWT [1] 29
- BGCOL [-1] 34
- Brackets ([]) 19
- CGM 34, 90, 102
- Characters
  - special 19, 100, 108
- CLIP [no] 33
- Clipping 33
- Clipping window 34
- Colors 21
  - background 34
  - foreground 34
  - fountain 34
  - graded background 34
  - setting 34
- Command summary 93
  - PC 94
  - UNIX 102
- Comments 18
- Computer systems 1
- CoreDRAW 90
- CORNER [1] 28
- Cursor mode 101
- Data
  - smoothing 33
  - sort 33
  - spline curve 33
- Data entry 11
  - arithmetic expressions 23
  - binary file 22
  - column assignments 23
  - global changes 23
  - keyboard 22
  - random number 23
  - text file 22
- Data file
  - binary 27, 100, 108
- Data input 20
  - modify 30
  - scale 30
- Data output
  - error bars 31
  - filling of symbols 31
  - line weight 30
  - line-symbol gap 31
  - plot type 30
  - shade 31
    - background 31
    - line weight 31
  - symbol
    - angle 31
    - size 31
  - symbol number 30
  - type of line 30
- Data range, new 21
- Data Value Representations 22
- Data values 30
- Data values, reset 21
- %define 25
- Double-quote marks (") 18
- EBWIDTH [1] 31
- ECHO [no] 33
- Epson FX-compatible printers 90
- Error bars 46
- Error messages 66
  - Fatal 66
  - Warnings 67
- Examples 70
  - alternate font 74, 80
  - EBNUM 78
  - error bars 78, 80
  - histogram plot 78
  - italic font 80
  - MHALIGN 82
  - multi-panel plot 74
  - MXLLC 82
  - NEWRANGE 85
  - rectangles plot 80
  - SHADE 78
  - simple plot 71
  - suppress annotations 74

- use of expressions 82, 85
- use of macros 82
- user variables 74
- YANNOT 85
- YBREAK 85
- Exclamation point 18
- External comments 18
- External font files 91
- External fonts 91
- FGCOL [-1] 34, 45
- Figures
  - fill patterns 64
  - fonts 61
  - line weights 62
  - message alignments 65
  - symbols and line types 63
- File commands
  - ALTFNT fontfile 27, 50
  - BDATA datafile 27, 37
  - CALL filespec 26, 38
  - INCLUDE filespec 26, 38
  - STDFNT fontfile 27, 50
  - SYMFNT fontfile 27, 50
- Floating-point numbers 20
- FNCOL [-1] 34
- Fonts
  - external 91, 100, 108
  - files
    - description 91
    - editing 91
    - PLTFNT program 91
  - internal 91
- Frame 21, 31
  - lower left corner 27, 28
- GAPSIZ [1] 31
- Graphical Output 90
- GRDCOL [-1] 34
- GRDLWT [1] 33
- GRDSIZ [1] 33
- GRDTYPE [1] 33
- GRID [no] 33
- Grid lines 33
- HEADER [yes] 6, 33
- Header line 33
- Hewlett-Packard LaserJet 90
- Histogram 46
- HPGL 34, 90
- Imagen 90, 102
- Index by function 110
- Index, permuted 110
- Internal comments 18
- Internal fonts 91
- Keywords 20, 95, 102
  - DATA 21, 38
  - DATLIST 21, 38
  - FINISH 21, 39
  - KEEP 21, 45
  - NEWFRAME 21, 44
  - NEWPAGE 21, 44
  - NEWRANGE 21, 45
  - PALETTE 21, 45
  - PLOT 45
  - PRMLIST 21, 47
  - RESET 21, 48
  - VARLIST 21, 53
- LABCOL [-1] 34
- LABLWT [1] 30
- LABSIZ [1] 29
- Lines 46
- LINTYPE [0] 11, 30
- Macros 25, 100, 108
  - %define 25, 38
  - example 25
  - invocation 25
  - remove 25
  - %repeat 25, 47
  - repeat lines 25
  - %undefine 52
- MAGNIFY [1] 33
- Messages 12, 18, 99, 108
  - angle 32
  - character
    - size 32
    - slant 32
    - spacing 32
    - width 32
  - place as data 32
  - placement 32
  - position 31
  - size of line samples 32
  - size of shade boxes 32
  - size of symbols 32
  - vertical spacing 32
- Metafile 34, 90
- MHALIGN [LEFT] 14, 32
- MHKEY [3] 19 32
- MHSP [0] 32
- MHXP [1] 32
- Mirus FilmPrinter 90
- MSGANGLE [0] 32
- MSGCOL [-1] 34
- MSGDAT [0] 13, 32
- MSGSIZ [1] 32
- MSLANT [0] 32
- MVALIGN [BASE] 32
- MVKEY [1] 19 32
- MVSP [1.64] 32
- MXDAT [0] 32
- MXLLC [0] 13, 18, 31
- MYDAT [0] 32
- MYLLC [0] 13, 18, 31
- NeXT computer 34, 90, 93, 102
- "NULL" plot 5
- Number character (#) 19
- OPENAX [0] 29
- Page, new 21

PageMaker (Aldus) 90  
Parameters  
  annotation 27  
  annotation control 20  
  axis 27  
  axis specification 20  
  classes 19  
  color 34  
  data representation 20, 30  
  FINISH 23  
  KEEP 24  
  message 31  
  message control 20  
  miscellaneous 20, 33  
  numeric 95, 103  
  PLOT 23  
  string 99, 107  
  XDATA 23  
  YDATA 23  
  ZDATA 23

Parameters (alphabetical)

ANNCOL [-1] 45  
ANNLWT [1] 37  
ANNSIZ [1] 37  
AXCOL [-1] 45  
AXLWT [1] 37  
BGCOL [-1] 39  
CLIP [no] 38  
CORNER [1] 38  
EBNUM [2] 39  
EBWIDTH [1] 39  
ECHO [no] 39  
FGCOL [-1] 39  
FNCOL [-1] 40  
GAPSIZ [1] 40  
GRDCOL [-1] 45  
GRDLWT [1] 40  
GRDSIZ [1] 40  
GRDTYPE [1] 40  
GRID [no] 40  
HEADER [yes] 41  
LABCOL [-1] 45  
LABLWT [1] 41  
LABSIZ [1] 41  
LINSIZ [1] 41  
LINTYPE [0] 41  
MAGNIFY [1] 42  
MHALIGN [LEFT] 42  
MHKEY [3] 42  
MHSP [0] 42  
MHXP [1] 42  
MSGANGLE [0] 43  
MSGCOL [-1] 45  
MSGDAT [0] 43  
MSGSIZ [1] 43  
MSLANT [0] 43  
MVALIGN [BASE] 42  
MVKEY [1] 42

MVSP [1.64] 42  
MXDAT [0] 43  
MXLLC [0] 44  
MYDAT [0] 43  
MYLLC [0] 44  
OPENAX [0] 45  
PLTCOL [-1] 45  
PLTLWT [1] 46  
PLTYPE [both] 46  
RANNOT [none] 53  
RANSKP [0] 54  
RBREAK [0] 54  
RCYCLE [0.0] 54  
RFMT [I] 56  
RINT [0] 56  
RLABEL [no label] 57  
RMAX [0] 58  
RMIN [0] 58  
ROTATE [no] 48  
RPERCENT [100] 58  
RSHIFT [0] 58  
RTICK [1] 59  
SHADE [0] 48  
SHDCOL [-1] 45  
SHDGRAY [1] 49  
SHDLWT [1] 49  
SIZFAC [0] 49  
SMOOTH [no] 49  
SOLID [0] 50  
SORT [no] 50  
SPLINE [no] 50  
SYMANG [0] 50  
SYMBOL [1] 50  
SYMSIZ [1] 51  
TANNOT [none] 53  
TANSKP [0] 54  
TBREAK [0] 54  
TCYCLE [0.0] 54  
TFMT [I] 56  
TICDIR [0] 52  
TICSIZ [1] 52  
TINT [0] 56  
TLABEL [no label] 57  
TMAX [0] 58  
TMIN [0] 58  
TPERCENT [100] 58  
TSHIFT [0] 58  
TTICK [1] 59  
VXLLC [0] 53  
VYLLC [0] 53  
WXLEN [10.5] 53  
WYLEN [8] 53  
XANNOT [none] 53  
XANSKP [0] 54  
XBREAK [0] 54  
XCYCLE [0.0] 54  
XDATA [(datum \* XGAIN) + XOFST]

XFMT [I] 56  
 XGAIN [1] 56  
 XINT [5] 56  
 XLABEL [no label] 57  
 XLEN [6.0] 57  
 XLLC [2.2] 57  
 XMAX [10] 58  
 XMIN [0] 58  
 XOFST [0.0] 56  
 XPERCENT [100] 58  
 XSHIFT [0] 58  
 XTICK [1] 59  
 YANNOT [none] 53  
 YANSKP [0] 54  
 YBREAK [0] 54  
 YCYCLE [0.0] 54  
 YDATA [(datum \* YGAIN) + YOFST]  
     55  
 YFMT [I] 56  
 YGAIN [1] 56  
 YHOR [0] 59  
 YINT [5] 56  
 YLABEL [no label] 57  
 YLEN [6.0] 60  
 YLLC [1.5] 60  
 YMAX [10] 58  
 YMIN [0] 58  
 YOFST [0.0] 56  
 YPERCENT [100] 58  
 YSHIFT [0] 58  
 YTICK [1] 59  
 ZDATA [(datum \* ZGAIN) + ZOFST]  
     55  
 ZGAIN [1] 56  
 ZOFST [0.0] 56  
 Parentheses 20  
 PC 34, 90  
 Plot Elements 27  
 Plot frame, new 21  
 Plotter support 101  
 PLTCOL [-1] 34  
 Pltfiles 1  
 PLTFNT program 91  
 PLTLWT [1] 30  
 PLTYPE [both symbols and lines] 11, 30  
 Polygon 46  
 PostScript 34, 90, 102  
 Print values  
     current parameter 21  
     data buffer 21  
     user variable 21  
 Printer support 101  
 Printers 90  
 RANNOT [none] 29  
 RANSKP [0] 29  
 RBREAK [0] 28  
 RCYCLE [0.0] 28  
 Rectangles 46

Rectangles, contiguous 47  
 %repeat 25  
 RFMT [I] 29  
 RINT [0] 28  
 RLABEL [no label] 29  
 RMAX [0] 28  
 RMIN [0] 28  
 ROTATE [no] 33  
 Rotate plot 33  
 RPERCENT [100] 28  
 RSHIFT [0] 28  
 RTICK [1] 28  
 Scale plot 33  
 Semicolon (;) 18  
 SHADE 49  
 SHADE [0] 31  
 SHDCOL [-1] 34  
 SHDGRAY 48  
 SHDGRAY [1] 31  
 SHDLWT [1] 31  
 Shift plot 33  
 SIZFAC [0] 33  
 SMOOTH [no] 33  
 SOLID [0] 31  
 SORT [no] 33  
 Special characters 19, 100, 108  
 SPLINE [no] 33  
 Standard font 10, 27  
 Subscripted text 19, 100, 108  
 Sun computer 90  
 Superscripted text 19, 100, 108  
 SYMANG [1] 31  
 Symbol 11, 19  
 SYMBOL [1] 30  
 Symbol font 27  
 Symbols 46  
 SYMSIZ [1] 31  
 TANNOT [none] 29  
 TANSKP [0] 29  
 TBREAK [0] 28  
 TCYCLE [0.0] 28  
 Tektronix 34, 90, 102  
 Test.plt file 3, 88  
 Text editor 3, 92  
 Text files 1  
 TFMT [I] 29  
 TICDIR [0] 6, 28  
 Tick marks  
     direction 28  
     intervals 28  
     lengths 28  
 TICSIZ [1] 6, 28, 59  
 TINT [0] 28  
 TLABEL [no label] 29  
 TMAX [0] 28  
 TMIN [0] 28  
 TPERCENT [100] 28  
 TSHIFT [0] 28

TTICK [1] [1] 28  
%undefine 25  
Underscore ( \_ ) 19  
UNIX operating system 93  
Usage summary 93  
User Variables 20, 26, 99, 107  
Vertical bars ( | ) 19  
VXLLC [0] 33  
VYLLC [0] 33  
Waves 46  
Word for Windows (Microsoft) 90  
Word processor 3  
WXLEN [10.5] 34  
WYLEN [8] 34  
X axis 27  
XANNOT [none] 14, 29, 71  
XANSKP [0] 29, 74  
XBREAK [0] 28, 59  
XCYCLE [0.0] 28, 46  
XDATA [(datum \* XGAIN) + XOFST] 30  
XFMT [I] 29  
XGAIN [1] 30  
XINT [5] 9, 28  
XLABEL [no label] 10, 29  
XLEN [6.0] 6, 27, 58  
XLLC [2.2] 6, 27  
XMAX [10] 8, 28

XMIN [0] 8, 28  
XOFST [0.0] 30  
XPERCENT [100] 9, 28, 58  
XSHIFT [0] 28  
XTICK [1] 28  
Y axis 27  
YANNOT [none] 14, 29, 71  
YANSKP [0] 29, 74  
YBREAK [0] 28  
YCYCLE [0.0] 28  
YDATA [(datum \* YGAIN) + YOFST] 30  
YFMT [I] 29  
YGAIN [1] 30  
YHOR [0] 6, 29  
YINT [5] 28  
YLABEL [no label] 10, 29  
YLEN [6.0] 6, 27  
YLLC [1.5] 6, 28  
YMAX [10] 8, 28  
YMIN [0] 8, 28  
YOFST [0.0] 30  
YPERCENT [100] 28  
YSHIFT [0] 28  
YTICK [1] 28  
ZDATA [(datum \* ZGAIN) + ZOFST] 30  
ZGAIN [1] 30  
ZOFST [0.0] 30