NASA Contractor Report 189553

TADPLOT Program - Version 2.0 User's Guide

Dana P. Hammond

Computer Sciences Corporation
Hampton, Virginia

Prepared For
Langley Research Center
under Contract NAS1-19038
November 1991

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

Preface

This document describes how to use the TADPLOT Program (version 2.00). All users should be familiar with their host computer operating system and the graphics terminal(s) used.

This manual has been developed as a User's Guide and as a reference tool for researchers using the TADPLOT Program. The manual is organized into sections and appendices.

The user must read sections one through four to understand the fundamental concepts and nomenclature necessary to interface with the Plot Program to generate plots.

Section 5 discusses more "advanced operations" needed to generate plots with multiple curves, and/or manipulate the layout of a simple plot.

Sections 6 and 7 discuss all of the Plot Program's directives. Section 6 provides a syntactical summary of all the directives. Section 7 provides a detailed description of each directive, along with its purpose, syntax, and related notes.

If the user has any suggestions regarding this program, contact:

Dr. Neal T. Frink
Mail Stop 361
Transonic Aerodynamics Branch
Applied Aerodynamics Division
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665

This program was developed by Computer Sciences Corporation, at NASA Langley Research Center. If the user has any requests for documentation, or questions about code implementation and support, contact:

Mr. Dana P. Hammond
Mail Stop 157D
Computer Sciences Corporation
Applied Technology Division
3217 North Armistead Avenue
Hampton, VA 23666-1379

Acknowledgements

The following people have contributed significantly in the development, implementation, verification, and installation of TADPLOT: Dr. John E. Lamar, Bradford D. Bingel, Judith M. Patterson, Michael L. Nelson, Keith D. Hoffler, and Nancy F. Schott.

This page is intentionally left blank.

TABLE OF CONTENTS

TABLE OF CONTENTS

Section 1

Overview

Background

The TADPLOT Program was developed to give the researcher a general purpose plotting package that is easy to use, can quickly create XY plots, and yet is powerful enough to create full-feature publication quality plots. Such a package gives the researcher control over the content and layout of each image, and delivers publication quality plots quickly. Integrated with an applications software network, researchers can view experimental (or theoretical) results immediately after collection; an advantage when working at a scheduled facility or while on a short duration project. The underlying graphics package enables the user to selectively save plot information which can subsequently be displayed on several different devices including "hard copy" plotters. Additionally, postprocessing features are available which enable the user to manipulate and combine plots.

Design Features

The TADPLOT Program is unique because it takes many of the best features from other software products and combines them into a single package.

The primary features are:

- conforms to ANSI FORTRAN 77 standards;
- contains a full interactive HELP subsystem;
- accepts abbreviated directives with abbreviated or omitted arguments;
- supports comment statements, and continued command lines;
- accepts directive sets from files;
- maintains a log file;
- automatically determines the input raw data file's format;
- automatically scales and determines increments for the plot's x-axis and y-axis;
- draws multiple curves per plot;
- allows random text positioning;
- performs both normal and parametric curve-fits; and

- allows control of the symbol codes, line patterns, legend contents, and text attributes (font, height, width, etc.).

- enables the selection of frames for postprocessing using metafiles (see Appendix F). Specifically, metafiles are a means by which "hardcopies" and/or multiple charts on a page can be obtained.

Currently, TADPLOT doesn't enable the user to generate composite plots (i.e. multiple charts per single frame), but postprocessing capabilities are available to merge saved plots into composite plots (see Appendix F).

Section 2

Introduction to Program Use

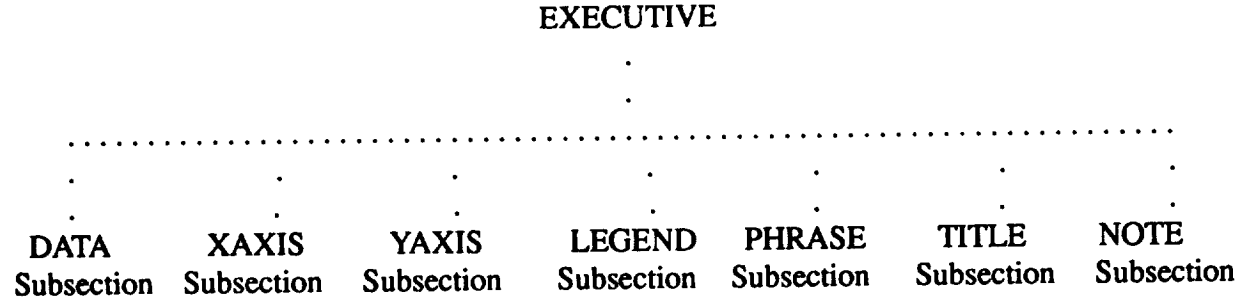
Introduction

The TADPLOT Program is a command-driven, interactive plotting package designed to aid researchers in the generation of publication standard XY plots from data in, at least, SIF or TOAD formats.

Conceptually, the program is divided into a set of interrelated modules each consisting of commands (or directives). Each module performs a specific interrelated activity (i.e. the XAXIS module allows the user to modify or design an axis). Each module is comprised of commands (directives) which allow the user to perform operations specific to that activity. For example, in the XAXIS module, the directive LENGTH, changes the axis length.

Subsections

The program is structured into an overhead EXECUTIVE module and a series of task-oriented sub-executive modules, hereafter referred to as "subsections". The following diagram shows the primary structure of the Plot Program.



The user always has access to the EXECUTIVE module, and therefore its commands and directives. The EXECUTIVE issues the prompts, and processes the user's commands. The other subsections, being subordinate to the EXECUTIVE, must be invoked one at a time, to enable the user to utilize that module's commands and directives.

Thus the EXECUTIVE calls on the subsections to actually perform the request. A brief description of each subsection's function follows:

DATA

- gathers the information necessary to select raw data from a disk file.
- manipulates the data by allowing the user to apply "offsets" and "gains".
- sets the symbol, line pattern, and curve fit method (if any) for each set of raw data.
- provides information to the LEGEND subsection for generating the legend keys.

XAXIS/YAXIS

- sets the axis length, axis minimum and maximum, and the frequency of the major and minor tick marks.
- determines the content and format of the numeric values at the tick marks and the overall axis label.

LEGEND

- gathers and organizes the information necessary to generate the plot's legend.

PHRASE

- sets up simple or complex text definitions for use in all subsections generating graphics text.

TITLE

- gathers and organizes the information necessary to generate the plot's title.

NOTE

- gathers and organizes the information necessary to generate random text.

When the program begins, the user is initially "attached" to the DATA subsection module. From here any EXECUTIVE or DATA directive may be specified, but not any other subsection directives. The user may enter any of the other subsections by keying its name. For example, the directive

XAXIS

attaches the user to the XAXIS subsection. All of the EXECUTIVE directives are still available, along with all of the XAXIS subsection directives. The user may leave this subsection and enter any other subsection by entering its name (the user does not have to return to the EXECUTIVE module in order to change subsections). For example, the directive

LEGEND

attaches the user to the LEGEND subsection. Again, all of the EXECUTIVE directives are still available, along with all of the LEGEND directives.

The subsection the user is attached to is hereafter called the "controlling" subsection. All of the directives the user enters must be appropriate for either the EXECUTIVE module or for the controlling subsection. If the user enters an inappropriate directive, it will be misinterpreted or ignored.

The directive

WHEREAMI

tells the user which subsection is currently attached. Directives for all subsections are fully detailed in Section 7.

Help Facility

While relatively few commands are required to meet most plotting needs, there are many directives, directive variations, and specialized mnemonics distributed throughout all of the subsections. To make the program easier to use and easier to learn, there is a complete interactive HELP facility which is always available. When the user enters

HELP

all EXECUTIVE directives will be listed in alphabetical order. If the user is in a subordinate subsection, then that subsection's directives will also be displayed in alphabetical order.

If the user needs more information about a particular directive, enter

HELP keyword

and a detailed description of the directive, associated by its corresponding keyword, is displayed. In general, each explanation briefly describes the function of the directive, its syntax (all variations), and defines any arguments.

For example, after entering the DATA subsection

DATA

the user may wish to ask for more information about the SORT directive

HELP SORT

and the program would respond with

SORT TURN THE SORT OPTION ON OR OFF.

SYNTAX: [NO] SOR[T]

Because of time and memory restrictions, the interactive HELP facility only provides basic information about any particular directive. If the user needs more information than the HELP facility provides, please refer to Section 7 of this document.

This page is intentionally left blank.

Section 3

General Command Syntax

Commands and Directives

A "command" is made up of one or more directives. A "directive" is a singular keyword (possibly with arguments) which asks for a singular action to be taken. For example, the directive PLOT asks the program to create a plot. The set of usable directives is called the plot program's vocabulary. Directives and arguments are analogous to verbs and nouns, with a simple sentence structure of

verb [noun] [noun]...

Since commands may be composed of more than one directive, they may be compared to compound sentences (i.e. with multiple verb constructs).

Arguments

Some directives call for a specific action, while other directives only provide information. For example, the directive

FILE TEST119

indicates that TEST119 is the name of the file containing the raw input data. The file name TEST119 following the directive FILE, is called an argument.

There are two kinds of arguments: numeric and text. Numeric arguments are numbers, like 3, -12.5, or 75. All numeric arguments may be entered with or without a decimal point. A whole number is always acceptable as a numeric argument. However, if a whole number is expected, and a fractional number is entered, it is rounded to the nearest whole number. Text arguments are usually names or descriptions, like "TEST119", "ALPHA", or "Cp UPPER". If the text contains embedded blanks (like "Cp UPPER") it must be enclosed within single or double quotation marks. For example, in the command

FILE TEST119 X 'Cp UPPER'

"Cp UPPER" must be enclosed in quotes, to ensure recognition of the embedded blank.

Directives and their arguments are separated from each other with a blank or a comma. Thus the directive

FILE TEST119

and the directive

FILE,TEST119

are equivalent.

Some directives require more than one argument. Depending on the directive, the arguments may be all numeric, all text, or a mixture of both. For example, the directive

HELP SHOW PAGESIZE PLOT

is a "HELP" directive with the arguments (in this case keywords) "SHOW", "PAGESIZE", and "PLOT". This is a request for additional information about the SHOW, PAGESIZE, and PLOT directives. Arguments, of directives having multiple arguments, are separated from each other and from the directive with a blank or a comma. Thus the directive

HELP SHOW PAGESIZE PLOT

is equivalent to

HELP,SHOW,PAGESIZE,PLOT

Some directives are special and permit omitted arguments. For example, the directive

FIT SPLINE

asks for a cubic spline fit through the data points. If the directive

FIT

is entered (omitting the fit method argument), the default fit method (Akima's method) is assumed. Remember that this is a special feature allowed by only some of the directives. Note, the HELP facility, Section 6 (Directive Syntax Summary), and Section 7 (Directive Use And Syntax), depict the directives and arguments along with their characteristics.

Abbreviations

Because some of the directives (and their text arguments) are too long to be entered conveniently, all directives and some arguments may be abbreviated. For example, the command

DATA FILE TEST119 FIT SPLINE PLOT

may be abbreviated to:

DAT FIL TEST119 FIT SPL PLO

Most directives may be abbreviated to their first three letters. Exceptions occur where two or more directives have the same first three letters. These directives may be abbreviated to four

letters. In addition, many of the directive arguments may also be abbreviated. For example, "SPLINE" may be abbreviated to "SPL". Notice, however, that in the example above the file name "TEST119" was not abbreviated. The HELP facility will tell the user what abbreviations are allowed for any particular directive.

Chaining Directives

Rather than enter a series of directives one line at a time, it is often quicker and more convenient to enter them all together on a single line. For example, the three directives,

```
FILE TEST119
FIT SPLINE
PLOT
```

each on its own command line, may be combined into the single command line

```
FILE TEST119 FIT SPLINE PLOT
```

with identical results.

There are exceptions. Since some directives allow for a variable number of arguments, the program may not be able to recognize where such a list ends and where the next directive begins. For example, the command

```
HELP PAGESIZE FIT
```

asks for help on the keywords PAGESIZE and FIT, as opposed to asking for help on PAGESIZE only, then processing the FIT directive. To avoid this confusion, some directives require that no other directives follow them on the same command line (HELP is an example). This restriction is represented syntactically in Sections 6 and 7 by an asterisk enclosed in parentheses "(*)". This restriction implies that if the user wishes information on the PAGESIZE directive, and desires to set the default fit method, the user would enter

```
HELP PAGESIZE
FIT
```

as two separate command lines.

Continuations

If a command line is over 80 characters long, it must be continued. In practice, continuations are rarely necessary since nearly all command lines can be broken up into a series of directives, which can be entered one at a time. If a very long command line is necessary, placing a "&" sign as the very last character in the line indicates that the command is incomplete, and that the remainder of the command follows immediately after. For example, the command line

```
DATA FILE TEST119 FORMAT TOAD X 2Y/B Y "CP UPPER" PLOT
```

may also be written as

```
DATA FILE TEST119 FORMAT TOAD &
X 2Y/B Y "CP UPPER" &
PLOT
```

or

```
DATA FILE TEST119 &
FORMAT TOAD X 2Y/B &
Y "CP UPPER" PLOT
```

In this example the continuation characters are unnecessary, as the commands

```
DATA FILE TEST119
FORMAT TOAD X 2Y/B
Y "CP UPPER" PLOT
```

will perform the same function.

The "NO" Prefix

There is a convenient way to "deselect" some directives. For example, the SYMBOL directive in the DATA Subsection, has the following syntax:

```
[NO] SYM[BOL] [ivalue] [EVERY ivalue] [FIRST] [LAST]
```

To deselect a symbol from appearing at the raw data points, the directive is

```
NOSYMBOL
```

Here, the "NO" prefix deselects the symbol option. This feature is generally available for those directives which turn an option ON or OFF. Also, "NO" prefixed directives usually do not require any arguments.

Comments

User comments may be inserted during the construction of the command line or on a separate line. The character indicating comments is the "@" sign. Anything to the right of a "@" is regarded as a comment, and is ignored. If the command line begins with a "@", the entire line is assumed to be a comment and is ignored. Comments are not usually part of an interactive session, unless the user is documenting a session in a log file. Comments, however, are often found in the directive files. Both log and directive files are discussed in Section 5.

Section 4

Single Curve Plots and Basic Operations

Introduction

This section explains how to make a simple single curve plot. The only information required is the raw data file name, the X and Y parameter names, and any raw data subset selection descriptors. The generation of more advanced multiple curve plots is explained in Section 5.

Execution

Because this program can be executed on several computers and operating systems, there is no single way to invoke it. Please refer to Appendix A for specific instructions for your particular installation.

Entering the DATA Subsection

The program begins with a welcoming banner, and then gives a "?", which is the TADPLOT prompt. The user is initially placed in the DATA subsection in order to prepare a set of raw data for plotting. If the user wishes to confirm that the DATA subsection is attached, then enter

```
WHEREAMI
```

and the response will be

```
YOU ARE IN THE DATA SUBSECTION.
```

Raw Data Files

Two different raw data file formats are currently available: SIF and TOAD. SIF, or Standard Interface File, is a common format of raw data collected in NASA Langley's wind tunnels. The SIF format is fully explained in NASA CR-159284. TOAD, or Transferrable Output ASCII Data, is similar to SIF, but is a formatted file, and thus can be easily transmitted between different computer systems. The TOAD format is fully explained in the NASA CR-178361 (TOAD file format description).

Regardless of format, the name of the file containing the raw data must be specified by the FILE directive. For an explanation of the FILE directive while at the terminal, enter

```
HELP FILE
```

and the interactive HELP facility displays a brief description.

To tell the program the name of the disk file containing the raw data the user wishes plotted, enter

```
FILE fname
```

where "fname" is the file name. For example, if the raw data file is called TEST119, the user would enter:

FILE TEST119

To confirm that the file name is stored, enter

SHOW FILE

and the program should display the raw data file name.

Scanning Files

The user may need to know some information about a particular raw data file before the user can extract a data set from it. The SCAN directive performs this function. For example, if the user has a raw data file, and the user wished to know its format and/or a list of the available parameters, enter

SCAN fname

where "fname" is the raw data file name. In response, the program will determine and display the raw data file's format, and then list all of the available parameters. For example, if your raw data file is called "TEST119", the directive

SCAN TEST119

will yield a response similar in nature to the following:

FILE TEST119 IS IN TOAD FORMAT.

FILE TEST119 CONTAINS THE VARIABLES

TEST	CL
RUN	ALPHA
CD	CM

Selecting the X- and Y- Axis Parameters

Once the user knows what data is available from the raw data file, the variable), and a single parameter for the y-axis (the dependent variable). The X and Y directives perform these functions. For example, if the user wishes to use "ALPHA" as the x-axis variable, the

user would enter

X ALPHA

If the user wishes to use "CL" as the y-axis variable, the user would enter

```
Y CL
```

Of course, both directives may be entered on the same command line:

```
X ALPHA Y CL
```

If the user wishes to use "ALPHA" as the x-axis variable, but extract only the values between 0 and 40 degrees, and "CD" as the y-axis variable, the user would enter

```
X ALPHA 0 40 Y CD
```

This introduces the "range" option, a way to limit the set of data extracted from the raw data file. The general syntax of the X and Y directives is:

```
X vname [range]
Y vname [range]
```

where "vname" is a variable name (i.e., "ALPHA"), and the optional range may have any of the following forms:

```
dmin dmax
* dmax
dmin *     * or omitted

value
value / tol [%]
```

where "dmin" and "dmax" are a data minimum and maximum, respectively; "value" is a target value; "tol" is an absolute or relative tolerance; and the percent sign is optional (if it is used, the tolerance is relative; otherwise, the tolerance is absolute).

Those forms using a wildcard ("*") may be interpreted as follows:

```
        * dmax less than or equal to "dmax"

    dmin * greater than or equal to "dmin"

        * or omitted use all available data
```

The target value / tolerance pair acts as an alternate way to specify a range. For example, the range

```
50 / 10
```

asks for an absolute tolerance, and should be read as "50 plus or minus 10". By comparison, the range

50 / 10%

asks for a relative tolerance, and should be read as "50 plus or minus 10% of 50", or "50 plus or minus 5".

Spacing is not significant in a target value / tolerance pair. For example, the range

50 / 10%

may also be entered as

50/10%
50/10 %

The target value / tolerance pair is generally not used for x- and y-axis variables, although it is commonly used for selection variables, discussed next.

Selection Variables

Selection variables offer another way to limit how much and what type of data is extracted from the raw data file. Suppose our previous example file ("TEST119") contains a set of raw data for several values of "RUN". If the user wishes to use "CL" for the x-axis, and "CM" for the y-axis, the user would enter

X CL Y CM

but this would extract many values of "CM" for every value of "CL", as the data across all values of "RUN" would be pooled.
Selection variables can solve this problem. In this case, the parameter "RUN" is independent within the raw data file "TEST119". Thus by using "RUN" as a selection variable, we can extract a single set of data. The SELECT directive performs this function. Its general syntax is similar to X and Y:

SELECT vname [range]

For example, the directive

SELECT RUN 20/.01

sets the selection variable "RUN", and enforces a range of 20 plus or minus the global tolerance of .01 . In other words, only pairs of "CL" and "CM" with a corresponding "RUN" value between 19.99 and 20.01 are to be extracted.

Alternately, since RUN is essentially an integer variable, the directives

```
X CL Y CM SELECT RUN 20
```

could be used to extract the intended raw data set.

Tabulating Raw Data

Before making a plot, the user may wish to "preview" the raw data set with the directive TABULATE. If we again use our example raw data file "TEST119", the directive

```
TABULATE
```

might produce the following table:

```
    X        Y
-------  -------

 -.1      .226
 -.3      .073
 -.5     -.179
 -.7     -.435
 -.9     -.534
```

If there is a problem extracting this data from the raw data file, a brief explanation would be displayed. Note, it is not necessary to TABULATE your data before making a plot, but it is recommended if the data set is relatively small.

Generating a Plot

Once the user has provided a file name (using the directive "FILE"), the X and Y parameter names (using the directives "X" and "Y"), and depending on your data, one or more selection variables (directive "SELECT"), the user may make a plot. The directive to perform this function is

```
PLOT
```

An interactive plot is generated based on the set of TADPLOT default attributes for each contributing subsection. For example, the length of the axes is determined by the XAXIS and YAXIS Subsection default attributes. Section 5, Multiple Curve Plots and Advanced Operations, explains how the user can tailor the plot appearance through the subsection attributes.

Similarly, the SAVE command will write plot information to a file which can then be sent to a postprocessing device for "hard copy" output (see Appendix F).

In general, however, the defaults conform to the LaRC "publication quality standards" as described in the NASA Technical Memorandum 81918 (Revised). The defaults conform to the standard's recommendations and affect the plot in the following manner:

- the size of text and symbols;
- the use of line patterns and symbols;
- the line pattern not being drawn through the symbol;
- the justification of text within the legend;
- the spacing between text and adjacent components (i.e. axes, legend entries, etc.);
- the character font type.

Returning to the Dialog Area

When the program is finished drawing the plot, the terminal will beep, or display a cross hair on the plot. After examining the plot, the user can return back to the interactive dialog area (presumably to enter more directives) by pressing the space bar. Note, on selective graphic devices, a carriage return (CR) will also return the user to the dialog area. The terminal then clears its screen, and the user returns back to the same subsection that the user was in when the user entered the PLOT directive.

Making Additional Plots

Additional PLOT requests produce additional plots. For example, if the user wishes to make three plots:

CL vs. ALPHA
CD vs. ALPHA
CM vs. ALPHA

then the user would set up the DATA subsection to first plot CL vs. ALPHA:

```
DATA
FILE TEST119
X ALPHA Y CL
PLOT
```

Then, after examining the first plot and pressing the space bar, the user would ask for the second plot:

```
Y CD
PLOT
```

and finally, the third plot:

```
Y CM
PLOT
```

Notice that the raw data file name and x-axis parameter name were retained for the second and third plots.

This is only a simple example. The user could have just as easily changed the raw data file name and/or the x-axis parameter name between plots. A more complex example is presented in Section 5.

Stopping the Program

The user may stop the program at any time by entering

STOP or END

This page is intentionally left blank.

Section 5

Multiple Curve Plots and Advanced Operations

Introduction

This section explains how to make multiple curve plots, and how to override "attribute" defaults to tailor the plot appearance. In addition, this section describes advanced operations which will aid the user in manipulating complex plots. This section assumes the reader has read all previous sections, and thus is familiar with the same nomenclature.

Attributes

Each subsection performs specific functions by using its directive syntax. Associated with each directive is a group of characteristics called attributes. Each attribute has a specific type and range. Prior to the change of any attribute in a subsection by a directive, the default value (also called the initial value) is used.

Displaying Attributes

To view these defaults, the user can enter a subsection and display the subsection attributes by using the SHOW directive. For example, to enter and display all the attributes for the XAXIS Subsection, the user could type:

```
XAXIS SHOW
```

The values displayed are the default ones if no other XAXIS directive was previously issued. Otherwise these values represent the current (i.e. modified) values. Similarly, the attributes for other subsections may be displayed (including the EXECUTIVE module's attributes).

Should the user desire to display select attributes, the SHOW command can be invoked with the appropriate arguments. For example, the following command will display the current major and minor tick mark attributes:

```
XAXIS SHOW MINOR MAJOR
```

Changing Attributes

To change attributes from their default values to their current values, the user must again enter the subsection associated with the attributes to be changed. Afterward the directive (and arguments) which control the value(s) of the attribute must be entered. For example, in the DATA Subsection, to change the line pattern and symbol number, the user may enter:

```
DATA LINE 1 SYMBOL 2 EVERY 2 FIRST LAST
```

This directive is to be interpreted as: entering the DATA subsection, setting the line pattern to 1, setting the symbol number to 2, and plotting every other data point (including the first and last). Similarly, the attributes for other subsections may be changed. As previously mentioned, the NO prefix is an alternative method for changing some attributes. This convention allows the user to deselect or disable an attribute.

Subsection Attribute Groups

The attributes are used to describe the specific characteristics of a single entity - either a graphical component like an axis, or a data collection (data set) extracted from a file. Since a single entity is described, it is natural to think of these attributes as a single interrelated set. Thus a collection of attributes are hereafter referred to as attribute groups. For example, the following command creates and opens an attribute group for the TITLE Subsection, and assigns values to attributes in order to generate a specific title.

```
TITLE CREATE T1 X 5 Y 1 TEXT 'F.LC IGURE 1:'
```

The TITLE attribute group T1, is created and consists of the attributes describing the characteristics of a specific title.

Each subsection may have several attribute groups. For example, in order to plot several data sets on a single plot, the user must CREATE separate DATA groups. Since multiple groups are allowed, we must have a means to distinguish between them, and commands/directives which operate on groups as whole entities. Thus each group is identified by a group name, and operations are performed on the group by the use of this name.

In order to understand how groups are named and what operations are permitted with them, it is helpful to discuss the state of the group (group states.)

A group state is an internal condition which describes the status of a group in relationship to the program, and other groups. Groups have two states, they are: either opened or closed, and active or inactive. A group is opened, within a subsection, when its attributes are available to be changed. A group is active when its attributes are to be used during subsequent plot operations. Note, only one group per subsection can be opened at a time (i.e. the user can access and change attributes one group at a time). But, multiple groups, for the DATA subsection,

can be active at a time (i.e. the user can display multiple data curves on a single plot). The use and meaning of these states will be further explained, and will become clear upon examining the sample sessions in Appendix E.

Creating Groups

Initially, the program assigns to a group the name DEFAULT. Thus by entering the XAXIS Subsection, the current 'open' and 'active' group is DEFAULT. However, user defined names can be given to groups by the CREATE command. For example, to create a DATA group with the name DAT1, the following could be used:

CRE DAT1 FILE TEST119 X MACH Y RN NOSYM LINE 1

This command would create a data group DAT1, with an associated file TEST119, and set the attributes for that group accordingly.

Since only one group can be opened at a time per subsection, then creating a group will cause an open group (if any), to be closed. For example, if a DATA group was already open when the above command was issued, that opened DATA group would be closed, and DAT1 would be opened.

Currently, only one plot per frame is available, though each plot could contain more than one curve or data set. Explicitly then only subsections which can have more than one active group are those which provide information for a single plot (i.e. DATA and PHRASE). The XAXIS, YAXIS, LEGEND, and TITLE Subsection can only have one active group.

Using and Altering Groups

While the subsections may vary in the commands/directives that are available, the typical commands which operate on groups include:

CREATE, OPEN, CLOSE, DELETE, and TURN.

With this set of commands, the user can CREATE attribute groups for the subsections, then subsequently DELETE them when no longer needed. During the process of generating plots, the user can reOPEN and CLOSE attribute groups to tailor the plot. The user can TURN ON and OFF groups to exclude the effects of a group upon the plot. For example, multiple DATA groups may have been created, and a plot generated based on these groups. The user may decide to inhibit the plotting of one (or several) groups during subsequent plots.

Often the attributes of groups are very similar. The USE directive allows the user to copy the attributes from one group into another, thus allowing the user to alter the duplicate. For example,

```
DAT CRE DAT1 FILE TEST119 X MACH Y RN SEL CASE 1 NOSYM LIN 1
CRE DAT2 USE DAT1 SEL CASE 2 LIN 2
```

The first CREATE will extract data from file TEST119 using the selection CASE 1, and set various attributes. The second command will CREATE a group DAT2 using DAT1's attributes, except for those changed (including group name).

Text Appearance (Mnemonics and Phrases)

Various subsections allow the user to enter text which will be used as part of the plot (e.g. the text used in the plot title). The Plot Program enables the user to control the text in a variety of ways. First, text may be entered without the use of quotes (single or double). However, multiple blanks will be compressed to a single blank. For example,

TITLE TEXT A BC D

is regarded as:

TITLE TEXT A BC D

If multiple blanks are desired, then quotes will ensure that the enclosed blanks are retained. For example:

TITLE TEXT 'A BC D'

will retain the enclosed blanks.

The user can additionally control text appearance by the use of mnemonics. These mnemonics are described in this document as PHRASE MNEMONICS, however are also applicable to any text to be displayed as part of the plot. The mnemonics enable the user to:

- underline text;
- display the text in upper and lower case;
- change the character font (see Appendix D);
- use superscript and subscript notation.

To use mnemonics, the user must embed them as part of the text. For example, to underline part of a string, the following text may be entered:

TITLE TEXT ABC.UND DEF.EUND GHI

This would set the title to:

ABCDEFGHI

Likewise, the other mnemonics can be embedded and invoked accordingly. However, the changing of fonts, and the use of subscripts and superscripts require further explanation.

By default, "Simplex Block" (font 1 - see Appendix D) is used to display text. The user may change the font by embedding the FONT mnemonic. For example, the text string:

TITLE TEXT F.LC IGURE (A) .UC M.SUB .FONT 9 &.ESUB

would be displayed as:

Figure (a) M_{∞}

This example highlights three important features of text; namely a font change, another use of "&", and a single command line for text. With respect to the change in font, note that the font 9 printed character for "&" is ∞. This also points out that the "&" cannot exclusively be considered for

use as a continuation character since within a text string it is just "&" for most fonts. Therefore, no character is available with which to allow the text to be continued pass the end of the command line. Hence, text strings are limited to a single line. To stay within this constraint, the PHRASE directive can be used and text strings can be composed of multiple PHRASES so long as the combination of characters used to name the PHRASES does not exceed the length of the command line.

Subscripting and superscripting is slightly more complicated due to the inherent generality of this code. A phrase begins writing at the main text level, called the "baseline" level, with a level indicator of 0. If superscripting is requested (.SUP), the text level jumps up one level, the level indicator becomes +1, and any subsequent text is written as a superscript. If the superscript is ended (.ESUP), the text level drops back down one level, the level indicator becomes 0, and any subsequent text is written at the baseline text level.

	level
superscript	1
baseline	0
subscript	-1

The subscripting method is very similar. A phrase begins writing at the main text level, with a level indicator of 0. If subscripting is requested (.SUB), the text level drops down one level, the level indicator becomes -1, and any subsequent text is written as a subscript. If the subscript is ended (.ESUB), the text level jumps back up one level, the level indicator becomes 0, and any subsequent text is written at the baseline text level.

The user may use more than one level of superscripts or subscripts. For example, if subscripting is requested (.SUB), the text level drops down one level, and the level indicator becomes -1. If a second subscripting request is made before the first subscript is ended (i.e., subscripting a subscript), the text level drops down another level, and the level indicator becomes -2.

```
                                  superscript
                    superscript
                                   subscript
baseline
                                  superscript
                    subscript
                                   subscript
```

The user may mix superscripts with subscripts, although caution is advised. For example, the mnemonic sequence:

```
A .SUB B .SUP C .ESUP .ESUB
```

writes "A", then subscripts it with "B", then superscripts this subscript with "C". Similarly, the mnemonic sequence:

A .SUP B .SUB C .ESUB .ESUP

rites "A", then superscripts it with "B", then subscripts this superscript with "C". However, the mnemonic sequence:

A .SUP B .ESUP .SUB C .ESUB

writes "A", then superscripts it with "B", then subscripts it with "C". This last example illustrates that the placement of the "ending" mnemonics (.ESUP, .ESUB) is critical to the finished appearance of the phrase.

In addition to mnemonics, text can also use PHRASES to describe text content. The PHRASE Subsection sets up simple or complex text definitions for use in all subsections generating graphics text. Thus, commonly used text string (with or without embedded mnemonics) can be set up in the PHRASE Subsection. For example, if "Cd" (where the d is to be subscripted) is used in several graphical text strings, then it can be created and displayed in the PHRASE Subsection by:

PHRASE CREATE CSUBD C.SUB .LC D

DISPLAY CSUBD

Once created, phrases can be used in other subsections. For example, assume the phrase CSUBD has already been created, then the title could make use of this phrase as follows:

TITLE TEXT F.LC IGURE (A) .CSUBD VS .UC A.LC LPHA

The title would then be expanded to use CSUBD's text and mnemonics. Note, the control of the mnemonics used in phrases is carried forth, and will affect subsequent text (i.e. in the last example, lower case was still set from the phrase CSUBD to cause "vs" to be displayed in lower case).

The Use of Notes

Users often desire to further characterize details contained within a specific plot. To do this, the text information is written in a nearby region and indicated by a leader (line) to a pertinent location on the plot. Although leaders aren't available, the NOTE Subsection enables the user to position text anywhere on the plot. The text in the NOTE Subsection, as in all subsections, can use mnemonics and phrases.

Directive Files

Often the user may want to perform the same set of commands/directives during different terminal sessions. Instead of retyping these instructions repeatedly, the user can make use of directive files. These files provide the user with the ability to read a set of commands/directives from a file. For example, suppose a frequently used set of PHRASES were required for a series of plots, it may be convenient to establish PHRASES via a predefined (verified) set on a directives file. Then READ these PHRASES during future sessions.

To use a directive file, the user would have to edit (create) a file containing the Plot Program commands/directives. These commands must be entered in the sequence in which the Plot Program would execute them. Thus the commands will be executed as if the user had typed them. For example, to execute a set of commands written on file the SET1, the user could type:

```
READ SET1
```

The Plot Program will read and execute each command from file SET1.

The user must be careful to give appropriate responses to any Plot Program questions, and ensure no errors occur. If not, subsequent commands read from the directive file may be misinterpreted by the Plot Program. Note, the PLOT command will generate output to the current OUTPUT device so that, during an interactive session, a plot will be drawn interactively, then pause for a response. During a batch session, the generated plot is only written to the metafile. Thus in batch mode, no pause occurs, but the program will continue to execute normally.

Log Files

To aid in the development of directive files, as well as subsequent debugging, log files provide the user with a means of retaining a set of commands executed during a session. Log files keep a history of the commands and comments entered, or read from a directive file, during a session. Thus if an error occurs, the error can be documented and easily repeated. Or, if the user wants to create a directive file for future sessions, the user could type:

```
LOG
```

and all subsequent commands the user enters will be recorded on a local file called LOG. The user then could modify this file, and use it during subsequent sessions.

This page intentionally left blank.

Section 6

Directive Syntax Summary

This section strictly shows the syntax summary of the TADPLOT Program directives. The subsections are listed in the following order:

EXECUTIVE, DATA, XAXIS and YAXIS, LEGEND, PHRASE, TITLE, and NOTE.

For each subsection, the directives are listed in alphabetical order. For each directive the following categories are detailed.

The Directive Name

The directive(s) being described is (are) listed and underlined in the leftmost column. Logically similar directives are often mentioned and/or described together.

Syntax

The syntax for the directive is presented, describing both the abbreviations and arguments. The following conventions will be used throughout to aid in representation.

- names in capital letters are used to denote reserved words (words with special meaning to the Plot Program); whereas names in lower case letters are used to denote arguments which are further described or are user supplied;

- brackets [ ] are used to denote optional items. This notation is used both in abbreviating the directives, as well as for indicating optional arguments (see EXECUTIVE for an example).

- ellipses ... denote that additional directives may follow (see EXECUTIVE for an example);

- directives enclosed in parentheses indicate that the directive is not currently implemented (see .BAS in the PHRASE mnemonics Subsection for an example);

- an asterisk in parentheses (*) is used to denote that no additional directive may follow on the same line (see HELP for an example);

- directives with NO preceding them, indicate a method to disable the directive (see AUTO for an example); and

- if an argument is immediately listed under another, then either form of the argument is acceptable (see SHOW for an example).

[redacted] 5-8 [redacted]

EXECUTIVE MODULE

[EXECUTIVE] dir [dir] . . .

dir is an EXECUTIVE directive in one of the following forms:

```
  DAT[A] [dir] [dir] . . .
+ EXE[CUTIVE] [dir] [dir] . . .
  LEG[END] [dir] [dir] . . .
  NOT[E] [dir] [dir] . . .
  PHR[ASE] [dir] [dir] . . .
  TIT[LE] [dir] [dir] . . .
  XAX[IS] [dir] [dir] . . .
  YAX[IS] [dir] [dir] . . .
+ (not as the first directive)
```

When any of the above directives are used, control passes to that subsection.

attribute control
[NO] ECH[O]
[NO] LOG
MAR[KERSIZE] [value]
PAG[ESIZE] [ivalue]
PHE[IGHT] [value]
PWI[DTH] [value]
TOL[ERANCE] [value]

file control
COP[Y] fname
MET[A] fname
PLO[T]
REA[D] fname
SAV[E]

diagnostics
HEL[P] [keyword] [keyword] . . . (*)
SHO[W] [keyword] [keyword] . . . (*)
GRO[UP] [gname] [gname] . . . (*)
LIM[ITS]
PHR[ASE] [pname] [pname] . . . (*)
WHE[REAMI]

termination
STO[P] or END

dir	is a directive appropriate for the controlling subsection.
fname	is a 1-10 character disk file name.
keyword	is an EXECUTIVE directive keyword. If the keyword is a sub section name, control passes to that subsection.
gname	is a group name appropriate for the controlling subsection.
pname	is a phrase name.
value	is an integer or real value.
ivalue	is an integer value.

DATA SUBSECTION

[DATA] dir [dir] . . .

dir is an EXECUTIVE directive or a DATA subsection directive in one of the following forms:

group control

CHE[CK] [gname] [gname] . . . (*)
CLO[SE]
CRE[ATE] gname
DEF[AULT] [gname] [gname] . . . (*)
DEL[ETE] [gname] [gname] . . . (*)
OPE[N] gname
TUR[N] state gname [gname] . . . (*)
USE gname

raw data file control

FIL[E] fname
FOR[MAT] format
SCA[N] [fname] [fname] . . . (*)
X vname [range]
[NO] XMA[X] [value]
[NO] XMI[N] [value]
Y vname [range]
[NO] YMA[X] [value]
[NO] YMI[N] [value]

selection control

[NO] ASE[LECT] vname range [vname range] . . . (*)
[NO] CSE[LECT] vname range [vname range] . . . (*)
[NO] DSE[LECT] vname [vname] . . . (*)
[NO] SEL[ECT] vname range [vname range] . . . (*)

symbol and line control

[NO] KEY [text] (*)
[NO] LIN[E] [ivalue]
[NO] SYM[BOL] [ivalue] [EVERY ivalue] [FIRST] [LAST]

raw data control

[NO] DEH[ANCE] [ivalue]
[NO] ENH[ANCE] [ifactor]
[NO] FIT [ftype]
SIG[MA] [value]
[NO] SOR[T]
TAB[ULATE] [gname] [gname] . . .

[NO] XGA[IN] [value] [order]
[NO] XOF[FSET] [value] [order]
[NO] YGA[IN] [value] [order]
[NO] YOF[FSET] [value] [order]

range is a range specification in one of the following forms:

value value
* value
value *
*
value
value / tol [%]

ftype is a fit type in one of the following forms:

LIN[EAR] | SPL[INE]
TEN[SIONSPLINE] | AKI[MA]

format is an input data file format of the following forms:

SIF | TOAD

order is an order specification in one of the following forms:

BEF[ORE] | AFT[ER]

state is an ON/OFF state in one of the following forms:

ON | OFF

gname is a 1-20 character DATA attribute group name.

fname is a 1-10 character disk file name.

vname is a 1-10 character variable name.

value is an integer or real value.

ivalue is an integer value.

text is a character string of the following forms:

character text | 'text' | "text" | mnemonics | .phrase

XAXIS and YAXIS SUBSECTIONS

[XAXIS] dir [dir] . . .
[YAXIS] dir [dir] . . .

dir is an EXECUTIVE directive, an XAXIS subsection directive, or a YAXIS subsection directive in one of the following forms:

group control

CLO[SE]
CRE[ATE] gname
DEF[AULT] [gname] [gname] . . . (*)
DEL[ETE] gname [gname] . . . (*)
OPE[N] gname
TUR[N] state gname . . . (*)
USE gname

axis format control

[NO] BAS[E] [ifactor]
LEN[GTH] [value]
[NO] LIN[E] [lfactor]
MAX[IMUM] [value]
MIN[IMUM] [value]
POS[ITION] [value]
[NO] SCI[ENTIFIC] [ivalue]

axis tick mark control

tick marks

[NO] FMA[JOR] [ivalue]
[format]
[NO] FMI[NOR] [ivalue]
[format]
LMA[JOR] [value]
LMI[NOR] [value]
MAJ[OR] [ivalue]
[NO] MINO[R] [ivalue]

labels for tick marks

DIR[ECTION] [value]
FON[T] [ivalue]
[NO] GAP [value]
HEI[GHT] [value]
[NO] LDI[GITS] [ivalue]
PRE[CISION] [ivalue]
[NO] RDI[GITS] [ivalue]
WID[TH] [value]

[NO] XOF[FSET] [value]
[NO] YOF[FSET] [value]

axis label control
[NO] LAB[EL] label (*)
LDIR[ECTION] [value]
LFO[NT] [ivalue]
LGA[P] [value]
LHE[IGLHT] [value]
LPR[ECISION] [ivalue]
LWI[DTH] [value]
[NO] LXO[FFSET] [value]
[NO] LYO[FFSET] [value]

additional axis format control
[NO] AUT[O]
[NO] FRA[ME] [ifactor]
[NO] GRI[D]
[NO] ZER[O]

weight — is a line weight in one of the following forms:
LIG[HT] | MED[IUM] | HEA[VY}

format — is a tick mark format in one of the following forms:
IN | OUT | IN/OUT | OUT/IN | BOTH

label — is the axis label in one of the following forms:
character text | 'text' | "text" | mnemonics | .pname

state — is an ON/OFF state in one of the following forms:
ON | OFF

gname — is a 1-20 character XAXIS attribute group name.

text — is a 1-40 character text string.

pname — is a phrase name.

value — is an integer or real value.

lfactor — is one of the following forms:
NONE | LIGHT | MEDIUM | HEAVY

ifactor — is an integer value:
0 - No axis
1 - Line only
2 - Line and tick marks only
3 - Full axis

LEGEND SUBSECTION

[LEGEND] dir [dir] . . .

dir is an EXECUTIVE directive or a LEGEND subsection directive in one of the following forms:

group control

CLO[SE]
CRE[ATE] gname
DEF[AULT] [gname] [gname] . . . (*)
DEL[ETE] gname [gname] . . . (*)
OPE[N] gname
TUR[N] state gname . . . (*)
USE gname

attribute control

[NO] LCO[LUMN] [text] (*)
[NO] LTI[TLE] [text] (*)
[NO] ORD[ER] [gname] [gname] . . . (*)
[NO] FRA[ME]
X [value]
Y [value]

text control

FON[T] [ivalue]
GAP [value]
HEI[GHT] [value]
JUS[TIFY] [jvalue]
[jdescr]
WID[TH] [value]

gname is a 1-20 character LEGEND subsection attribute group name.

jvalue is an integer value denoting text justification.

1 - upper left	6 - center right
2 - upper center	7 - lower left
3 - upper right	8 - lower center
4 - center left	9 - lower right
5 - center center	

jdescr is a justification description. For example:

TOPLEFT | RIGHT-CENTER
CENTER/BOTTOM | MIDDLE_LEFT

PHRASE SUBSECTION

[PHRASE] dir [dir] . . .

dir is an EXECUTIVE directive or a PHRASE subsection directive in one of the following forms:

group control

```
CLO[SE]
CRE[ATE] pname
DEL[ETE] pname [pname] . . . (*)
OPE[N] pname
```

phrase construction control

```
PRE[FIX] item [item] . . . (*)
REP[LACE] item [item] . . . (*)
SUF[FIX] item [item] . . . (*)
USE pname
```

diagnostics

```
DIS[PLAY] [pname]
```

item is a phrase argument in one of the following forms:

mnemonic | string | .pname

mnemonic is a phrase mnemonic in one of the following forms:

```
.DEL dchar
.FONT ivalue
.LC          .ELC
.(NL)
.SUB         .ESUB
.SUP         .ESUP
.UC          .EUC
.UND         .EUND
.(VMA) value
```

string is a text string in one of the following forms:

text |'text' | "text"

pname is a 1-20 character PHRASE name.

ivalue is an integer value.

value is an integer or real value.

dchar is the new delimiter character.

TITLE SUBSECTION

[TITLE] dir [dir] . . .

dir is an EXECUTIVE directive or a TITLE subsection directive in one of the following forms:

group control

```
CLO[SE]
CRE[ATE] gname
DEL[ETE] gname [gname] . . . (*)
OPE[N] gname
TUR[N] state gname . . . (*)
USE gname
```

text control

```
        JUS[TIFY]  [jvalue]
                   [jdescr]
        FON[T] [ivalue]
[NO]    GAP [value]
        HEI[GHT] [value]
[NO]    TEX[T] [text] [text] . . . (*)
        WID[TH] [value]
        X [value]
        Y [value]
```

gname is a 1-20 character TITLE subsection attribute group name.

value is an integer or real value.

jvalue is an integer value denoting text justification.

1 - upper left	6 - center right
2 - upper center	7 - lower left
3 - upper right	8 - lower center
4 - center left	9 - lower right
5 - center center	

jdescr is a justification description. For example:

```
TOPLEFT
RIGHT-CENTER
CENTER/BOTTOM
MIDDLE_LEFT
```

NOTE SUBSECTION

[NOTE] dir [dir] . . .

dir is an EXECUTIVE directive or a NOTE subsection directive in one of the following forms:

group control

```
CLO[SE]
CRE[ATE] gname
DEL[ETE] gname [gname] . . . (*)
OPE[N] gname
TUR[N] state gname . . . (*)
USE gname
```

text control

```
    DIR[ECTION] [value]
    JUS[TIFY]  [jvalue]
               [jdescr]
    FON[T] [ivalue]
[NO] GAP [value]
    HEI[GHT] [value]
[NO] TEX[T] [text] [text] . . . (*)
    WID[TH] [value]
    X [value]
    Y [value]
```

gname is a 1-20 character NOTE subsection attribute group name.

value is an integer or real value.

jvalue is an integer value denoting text justification.

1 - upper left	6 - center right
2 - upper center	7 - lower left
3 - upper right	8 - lower center
4 - center left	9 - lower right
5 - center center	

jdescr is a justification description. For example:

TOPLEFT
RIGHT-CENTER
CENTER/BOTTOM
MIDDLE_LEFT

This page is intentially left blank.

Section 7

Directive Use and Syntax

This section depicts the use and syntax of the TADPLOT Program directives. Specifically, this section explains all subsection syntax for both directives and mnemonics. The subsections are listed in the following order:

EXECUTIVE, DATA, XAXIS and YAXIS, LEGEND, PHRASE, TITLE, and NOTE.

For each subsection, the directives are listed in alphabetical order. For each directive the following categories are detailed.

The Directive Name

The directive(s) being described is (are) listed and underlined in the leftmost column. Logically similar directives are often mentioned and/or described together.

Syntax

The syntax for the directive is presented, describing both the abbreviations and arguments. The following conventions will be used throughout to aid in representation.

- names in capital letters are used to denote reserved words (words with special meaning to the Plot Program);
- names in lower case letters are used to denote arguments which are further described or are user supplied;
- brackets [ ] are used to denote optional items. This notation is used both in abbreviating the directives, as well as for indicating optional arguments (see EXECUTIVE for an example);
- ellipses ... denote that additional directives may follow (see EXECUTIVE for an example);
- directives enclosed in parentheses indicate that the directive is not currently implemented (see .BAS in the PHRASE mnemonics Subsection for an example);
- an asterisk in parentheses (*) is used to denote that no additional directive may follow on the same line (see HELP for an example);
- directives with NO preceding them, indicate a method to disable the directive (see AUTO for an example); and
- if an argument is immediately listed under another, then either form of the argument is acceptable (see SHOW for an example).

Arguments

The arguments are detailed with a brief description, usually including the range, type, and limitations specific to each.

Defaults

The defaults are listed for each directive which assigns values to subsection attributes. Note, however, some directives have multiple arguments, and therefore multiple defaults will be listed.

Notes

Various directives require further explanation, typically describing directive interrelationships, limitations, error messages, and similar related points of interest.

****

* EXECUTIVE Module *

****

COPY

Purpose: To copy the contents of a directive file to the screen. This directive essentially allows the user to view the contents of a directive file while executing TAD-PLOT.

Syntax: COP[Y] fname

fname the name of the disk file containing the input directives to be copied.

DATA

Purpose: To pass control to the DATA subsection.

Syntax: DAT[A] [dir] [dir] . . .

dir optional DATA subsection directive.

ECHO

Purpose: To turn the echo option ON or OFF. ECHO displays the contents of a file from the READ commands and redisplays interactive commands.

Syntax: [NO] ECH[O]

Default: The initial default value for ECHO is OFF.

END

Purpose: To stop the plot program.

Syntax: END

EXECUTIVE

Purpose: To pass control to the EXECUTIVE Module.

Syntax: EXE[CUTIVE] [dir] [dir] . . .

dir optional EXECUTIVE Module directive.

HELP

Purpose: To provide a brief description of a single keyword, a list of keywords, or all keywords.

Syntax: HEL[P] [keyword] [keyword] ... (*)

keyword optional keyword valid for the controlling subsection. If no keywords are given, all keywords are assumed.

No other directives may follow a HELP directive on the same line.

LEGEND

Purpose: To pass control to the LEGEND subsection.

Syntax: LEG[END] [dir] [dir] ...

dir optional LEGEND subsection directive.

LOG

Purpose: To turn the LOG file ON or OFF. The log file records all commands to a file, which can be replayed via the READ command.

Syntax: [NO] LOG

Default: The initial default value for LOG is OFF.

MARKERSIZE

Purpose: To set the marker size.

Syntax: MAR[KERSIZE] [value]

value an integer or real value, greater than zero, but less than one. If omitted, the initial default marker size is restored.

Default: The initial default value for MARKERSIZE is .1

META

Purpose: To set the default metafile name.

Syntax: MET[A] fname

fname file name for the graphics output.

Default: The initial default value for META is metafl.dat.

NOTE

Purpose: To pass control to the NOTE Subsection.

Syntax: NOT[E] [dir] [dir] . . .

dir optional NOTE subsection directive.

PHEIGHT

Purpose: To set the page height.

Syntax: PHE[IGHT] [value]

value an integer or real value, greater than zero. If omitted, the default page height is restored.

Default: The initial default value for PHEIGHT is 11.

PWIDTH

Purpose: To set the page width.

Syntax: PWI[DTH] [value]

value an integer or real value, greater than zero. If omitted, the default page width is restored.

Default: The initial default value for PWIDTH is 11.

PAGESIZE

Purpose: To set the screen page size (the number of lines which fit on the terminal screen).

Syntax: PAG[ESIZE] [ivalue]

ivalue an integer value, greater than zero. If omitted, the initial default page size is restored.

Default: The initial default value for PAGESIZE is 24.

PHRASE

Purpose: To pass control to the PHRASE subsection.

Syntax: PHR[ASE] [dir] [dir] . . .

dir optional PHRASE subsection directive.

PLOT

Purpose: To generate a plot on an interactive plotting device using the current values in the opened subsection groups.

Syntax: PLO[T]

READ

Purpose: To read a set of input directives from a disk file and execute the directives.

Syntax: REA[D] fname

fname the name of the disk file containing the input directives to be read.

SAVE

Purpose: To generate a "metafile" plot frame using the current values in the opened subsection groups. The file name containing the output is determined by the META command in the Executive Subsection. See Appendix F for a description of the use of metafiles.

Syntax: SAV[E]

SHOW

Purpose: To show a variety of session information.

Syntax: SHO[W] [keyword] [keyword] ... (*)
GRO[UP] [gname] [gname] ... (*)
LIM[ITS] ... (*)
PHR[ASE] [pname] [pname] ... (*)

keyword optional keyword valid for the controlling subsection. If omitted, all of the controlling subsection's currently open group's attributes are listed.

gname optional group name within the controlling subsection. If omitted, a summary of all the controlling subsection's groups is given.

pname optional phrase name within the PHRASE subsection. If omitted, a summary of all phrases is given.

No other directives may follow a SHOW directive on the same line.

STOP

Purpose: To stop the TADPLOT program.

Syntax: STO[P]

TITLE

Purpose: To pass control to the TITLE subsection.

Syntax: TIT[LE] [dir] [dir] . . .

dir optional TITLE subsection directive.

TOLERANCE

Purpose: To set the default tolerance, which is used in the range directive (and internally in the AUTO and FIT directives). This value is used to determine if two numbers are to be considered equal (within an acceptable tolerance).

Syntax: TOL[ERANCE] [value]

value an integer or real value, greater than zero. If omitted, the initial default tolerance is restored.

Default: The initial default value for TOLERANCE is .001

WHEREAMI

Purpose: To identify the controlling subsection.

Syntax: WHE[REAMI]

XAXIS

Purpose: To pass control to the XAXIS subsection.

Syntax: XAX[IS] [dir] [dir] . . .

dir optional XAXIS subsection directive.

YAXIS

Purpose: To pass control to the YAXIS subsection.

Syntax: YAX[IS] [dir] [dir] . . .

dir optional YAXIS subsection directive.

****

* DATA Subsection *

****

ASELECT

Purpose: To add new selection variables to the current selection variable list.

Syntax: ASEL[ECT] svname1 range1 [svname2 range2] . . . (*)

svname the new selection variable name(s) to be added to the current list.

range the range(s) associated with the new selection variable(s).

Both types of arguments are required, and must be entered in the order shown above.

Notes: There is a limit on how many selection variables may be used. This limit may be reviewed by entering:

DATA SHOW LIMITS

If a new selection variable is already in the current list, a warning message is written, and the new range specification is ignored.

If a new selection variable cannot be found on the raw data file, a warning message is written, and the new selection variable is ignored. Similarly, if a new selection variable's range specification is improper (i.e., the minimum is greater than the maximum), an error message is written, and the new selection variable is ignored.

To change the selection variable's range use CSELECT, and to delete or create a new list of selection variables use DSELECT and SELECT, respectively.

CHECK

Purpose: To check if DATA Subsection attribute information is complete and proper.

Syntax: CHE[CK] [gname] [gname] . . . (*)

gname the name of an existing DATA subsection attribute group being checked.

If a list of group names are entered, each will be checked in that order. If no group names are entered, the group currently open is checked.

No other directives may follow a CHECK directive on the same line.

CLOSE

Purpose: To close the DATA subsection attribute group currently open.

Syntax: CLO[SE]

Notes: If no group is currently open, a warning message is written.

CREATE

Purpose: To create a new DATA subsection attribute group.

Syntax: CRE[ATE] gname

gname the name of the DATA subsection attribute group being created. Up to 20 characters may be used in the group name. For clarity, we recommend the use of letters, numbers, or a combination of both, and that the name begin with a letter.

This argument is required.

Notes: If the group "gname" already exists, an error message is written, and the original group named "gname" remains unaltered.

If the maximum number of DATA subsection attribute group has already been created, a warning message is written, and the CREATE directive is ignored. To review this maximum enter

DATA SHOW LIMITS

CSELECT

Purpose: To change the range of one or more selection variables in the current list.

Syntax: CSEL[ECT] svname1 range1 [svname2 range2] . . . (*)

svname the name(s) of the selection variable(s) to be changed.

range the new range(s) associated with the selection variable(s).

Both types of arguments are required, and must be entered in the order shown above.

Notes: If a selection variable cannot be found in the current selection variable list, a warning message is written, and the current list remains unaltered.

If a new range specification is improper (i.e., the minimum is greater than the maximum), an error message is written, and the original range specification is retained.

DEFAULT

Purpose: To restore one or more DATA subsection attribute groups' variables to their original default values.

Syntax: DEF[AULT] [gname1] [gname2] . . . (*)

gname the name(s) of the existing DATA subsection attribute group(s) being defaulted.

This argument list is optional. If omitted, the group currently open is defaulted.

Notes: If the group "gname" does not exist, a warning message is written.

If more than one group is named in the argument list, each group named is defaulted.

If the argument list is omitted and no group is currently open, a warning message is written.

No directives may follow a DEFAULT on the same command line except a SHOW.

DEHANCE

Purpose: To set the dehancement factor to be applied to the raw data set. This factor determines which values will be extracted from the raw data set. For example, a factor of 2 will cause the first data value, and every other subsequent data value to be extracted.

Syntax: [NO] DEH[ANCE] [ifactor]

ifactor an integer dehancement factor. If omitted the value 2 is assumed.

Default: The initial default value for DEHANCE is 1.

Notes: Use NODEHANCE to restore the data set back to its content prior to dehancements, and to set DEHANCE to 1.

DELETE

Purpose: To delete one or more existing DATA subsection attribute groups.

Syntax: DEL[ETE] [gname1] [gname2] . . . (*)

gname the name(s) of the DATA subsection attribute group(s) being deleted.

This argument list is optional. If omitted, the group currently open is deleted.

Notes: If more than one group is named in the argument list, each group named is deleted.

If the group "gname" is currently open, it is still deleted, and no group will be open.

If the group "gname" does not exist, a warning message is written.

If the argument list is omitted and no group is currently open, a warning message is written.

It is possible to delete all DATA subsection attribute groups.

No directives may follow a DELETE directive on the same command line except a SHOW.

DSELECT

Purpose: To delete one or more selection variables from the current list.

Syntax: DSEL[ECT] svname1 [svname2] . . . (*)

svname the name(s) of the selection variable(s) to be removed from the current list.

This argument list is required.

No directives may follow a DSELECT on the same command line.

ENHANCE

Purpose: To set the enhancement factor to be applied to the raw data set. This factor, combined with DEHANCE factor, determines which values will be extracted from the raw data set. The ENHANCE factor will be multiplied with the DEHANCE factor. Example: a dehance factor of 4, and an enhance factor of 2, represented every second data point (i.e., 4/2).

Syntax: [NO] ENH[ANCE] [ifactor]

Default: The initial default value for ENHANCE is 1.

Notes: ENHANCE is performed when the command is encountered. Its values are not reflected in the SHOW information directly.

FILE

Purpose: To set the name of the input raw data file.

Syntax: FIL[E] fname

fname the name of the disk file containing the input raw data. This file name must be appropriate for the host computer operating system.

This argument is required.

Default: The initial default value for FILE is none.

Notes: The raw data file named is not actually accessed until the user enters a SCAN, CHECK, TABULATE, or PLOT directive. If there is any problem accessing the file, one of these directives will uncover it.

FIT

Purpose: To set the curve fit method.

Syntax: [NO] FIT [ftype]

ftype a curve fit method. The available methods are:

LIN[EAR]
SPL[INE]
TEN[SIONSPLINE]
AKI[MA]

This argument is optional. If omitted, the FIT is set to AKIMA.

Default: The initial default value for FIT is LINEAR.

Notes: If the sort option is ON (see the SORT directive), the raw data is first sorted, resulting in a monotonic data set, then run through the regular curve fit method. If the sort option is OFF, the raw data is not sorted, and is run through a parametric curve fit method. If raw data has a monotonic x-axis parameter (like time) and the sort option is OFF, the resulting parametric curve fit may not be what the user expects. Some cases may even draw loops. Thus if your data has an ever increasing x-axis parameter, we strongly recommend

to turn on the sort option.

If the spline under tension method is chosen (TENSIONSPLINE), the user may control the tension factor with the SIGMA directive ("sigma" is the Greek letter which commonly represents the weighting factor for a spline under tension analysis).

If more than 100 raw data points are available, a linear curve fit is performed, regardless of the method requested.

The form NOFIT inhibits any curve fitting and removes the current line pattern code.

FORMAT

Purpose: To set the format of the raw data file. This directive only provides information, it does not allow the format of the raw data file to be changed.

Syntax: FOR[MAT] type

type a raw input data file format:

SIF
TOAD

Default: The initial default value for FORMAT is UNKNOWN.

KEY

Purpose: To set the legend data text entry.

Syntax: [NO] KEY [text] (*)

text a text string (with phrase and/or mnemonics). No other directives may follow a KEY directive on the same line.

Default: The initial default value for KEY is '<DEFAULT>'.

LINE

Purpose: To set the pattern of the line connecting the raw data points. See Appendix C for samples of each available line pattern.

Syntax: [NO] LIN[E] [ivalue]

ivalue an integer value expressing the line pattern code. This value must be greater than zero. (see Appendix C)

This argument is optional. If omitted LINE is set to 1 (solid line).

Default: The initial default value for LINE is 2 for the DEFAULT group, and the ordinal value for created groups (i.e. 1 for the first group created, 2 for the second group created, etc.).

Notes: The form NOLINE inhibits drawing a line to connect the raw data points. It is commonly used when plotting experimental data, which is usually displayed as a series of symbols without a line.

OPEN

Purpose: To open an existing DATA subsection attribute group and makes it available for alterations.

Syntax: OPE[N] gname

gname the name of the DATA subsection attribute group being opened.

This argument is required.

Notes: If the group "gname" does not exist, a warning message is written.

If the group "gname" is already open, it remains open.

If a group is already open, and the user asks to open a second group, the first group is closed before the second group is opened.

range

Purpose: The range specification, used with the X, Y, and SELECT DATA Subsection directives, must conform to one of the following forms:

value value a minimum-maximum pair. Only data equal to or between these two values are used.

value* a minimum, with no maximum. Only data greater than or equal to the minimum is used.

* value no minimum, with a maximum. Only data less than or equal to the maximum is used.

* no minimum and no maximum. Use all of the available data.

value / tol [%]

a target/tolerance pair. The range is the target value plus or minus the absolute tolerance. If the optional percent sign is used, a relative tolerance is assumed.

value — a target value, with no tolerance. The range is the target value plus or minus the default absolute tolerance.

If the range specification is omitted, all available data is used.

SCAN

Purpose: To scan one or more input raw data files, determines their format, and lists the available parameters.

Syntax: SCA[N] [fname1] [fname2] . . . (*)

fname — the name(s) of the existing input raw data file(s) being scanned.

This argument list is optional. If omitted, the file specified in the currently open group is scanned.

Notes: If more than one file is named in the argument list, each file named is scanned. If the group "fname" does not exist, a warning message is written.

If the argument list is omitted, SCAN will operate on the current, opened data group's file. If no group is currently open, a warning message is written.

No directives may follow the SCAN command on the same line.

SELECT

Purpose: To create a new list of selection variables.

Syntax: [NO] SEL[ECT] svname1 range1 [svname2 range2] . . . (*)

svname — the selection variable name(s) to control which raw data is to be plotted.

range — the range(s) associated with the selection variable(s). Both types of arguments are required, and must be entered in the order shown above.

Default: The initial default value for SELECT is NONE.

Notes: If a list of selection variables already exists, it is replaced with the new list. If no list exists, a new list is created.

If any selection variable cannot be found on the raw data file, an error message is written, although the selection variable list remains unal-

tered. Similarly, if the range specification for any selection variable is improper (i.e., the minimum greater than the maximum), an error message is written, although the selection variable list remains unaltered.

Even if all selection variables are found on the raw data file, and all of the range specifications are proper, the combination may be so restrictive that none of the raw data qualifies, thus no data is available for plotting. If this is the case, it becomes evident only when the user enters a directive which actually accesses the raw data file (a CHECK, TABULATE, or PLOT directive).

The form NOSELECT deletes the current selection variable list.

No directives may follow the SELECT command on the same line.

SIGMA

Purpose: To set the tension factor for the spline under tension curve fit method (see the directive FIT).

Syntax: SIG[MA] [value]

value — a real value expressing the tension factor. This value must be greater than or equal to zero, but less than or equal to 50.

This argument is optional. If omitted, the initial default tension factor is restored.

Default: The initial default value for SIGMA is 1.

Notes: The tension factor controls how "tight" the spline will be. In general, a small tension factor produces a curve resembling a normal cubic spline, while a large tension factor produces a curve resembling more of a linear fit. On the extremes, a tension factor of zero is a normal cubic spline, and a tension factor of infinity is a linear fit. Experimentation may be required to produce the desired effect.

SORT

Purpose: To turn the sort option on or off. The sort option enables the sorting of the data elements.

Syntax: [NO] SOR[T]

Default: The initial default value for SORT is OFF.

Notes: If the sort option is ON (see the SORT directive), the raw data is first sorted,

resulting in a monotonic data set, then run through the regular curve fit method. If the sort option is OFF, the raw data is not sorted, and is run through a parametric curve fit method. If the raw data has a monotonic x-axis parameter (like time) and the sort option is OFF, the resulting parametric curve fit may not be what the user expects. Some cases may even draw loops. Thus if your data has an ever increasing x-axis parameter, we strongly recommend to turn on the sort option.

SYMBOL

Purpose: To set the symbol used to mark the raw data points. Optionally, the user can adjust a "skip" factor, so that a symbol is drawn at every "Nth" data point. See Appendix B for samples of each available symbol.

Syntax: [NO] SYM[BOL] [ivalue1] [EVERY ivalue2] [FIRST] [LAST]

ivalue1 an integer value expressing the symbol code. This value must be greater than zero. (see Appendix B)

This argument is optional. If omitted, SYMBOL is set to 2. By default, the first group created uses symbol code 1, the second group uses symbol code 2, the third uses symbol code 3, and so on.

ivalue2 an integer value expressing the "skip" factor. For example, the subdirective

EVERY 2

draws a symbol at every other raw data point. The subdirective

EVERY 5

draws a symbol at every fifth raw data point.

This argument is optional. If omitted, the initial default skip factor (1) is restored, and a symbol is drawn at every raw data point.

Default: The initial default value for SYMBOL is 2 for the DEFAULT group, and the ordinal value for created groups (i.e. 1 for the first group created, 2 for the second group created, etc.). The default value for EVERY is 1. The default value for FIRST is ON. The default value for LAST is ON.

Notes: If the skip factor is greater than the number of raw data points, the resulting plot may or may not display any symbols. Relatively large skip factors may

not display a symbol at the first or last raw data points.
The subdirective FIRST forces a symbol to be drawn at the first raw data point, regardless of the skip factor. Similarly, the subdirective LAST forces a symbol to be drawn at the last raw data point, regardless of the skip factor.

The subdirectives EVERY, FIRST, and LAST may be entered in any order.

The form NOSYMBOL inhibits drawing symbols at any of the raw data points. This representation is often used when plotting theoretical data, which is usually displayed as a line without any symbols.

TABULATE

Purpose: To tabulate one or more groups' raw data.

Syntax: TAB[ULATE] [gname] [gname] ...

gname the name of an existing DATA subsection attribute group to be tabulated.

If no DATA subsection attribute group names are used, the group currently open is tabulated.

TURN

Purpose: To turn one or more DATA subsection attribute groups on or off.

Syntax: TUR[N] state gname [gname] ... (*)

state an ON/OFF state:

ON
OFF

gname the name of an existing DATA subsection attribute group being turned ON or OFF.

DATA subsection attribute groups which are ON are called "active", and will be plotted when a PLOT directive is entered. Attribute groups which are OFF are called "inactive", and will not be plotted.

No other directives may follow a TURN directive on the same line.

USE

Purpose: To copy the information contained in an existing data subsection attribute group and writes it over the currently open group.

Syntax: USE gname

gname the name of an existing DATA subsection attribute group containing the desired information set.

The USE directive may only be used when a DATA subsection attribute group is currently open.

X

Purpose: To set the name and range of the x-axis parameter.

Syntax: X vname [range]

vname the name of the parameter containing the x-axis data. If it contains embedded blanks, it must be enclosed within quotes.

This argument is required.

range the range of values allowed for this data set. See the notes for range in the DATA Subsection for more information.

This argument is optional. If omitted, all of the available data is used.

Default: The initial default value for X is NONE.

Notes: If the parameter named is unavailable from the input raw data file, an error message is written (see CHECK).

If the parameter is available but the range specification disqualifies all of the available raw data, a warning message is written (see CHECK).

For an easy way to alter the range without having to re-enter the parameter name, see the directives XMIN and XMAX.

XGAIN

Purpose: To set the x-axis variable's gain and order. This directive multiplies a specified value to the data set.

Syntax: [NO] XGA[IN] [value] [order]

value an integer or real value. If omitted, the initial default gain is restored.

order an optional before/after order to be used:

BEF[ORE]
AFT[ER]

The keyword BEFORE forces the gain to be applied before the offset. The keyword AFTER forces the gain to be applied after the offset. If neither keyword is used, AFTER is assumed. If both the offset and the gain directives use or default to the keyword AFTER, the most recently entered operation is performed last. If both directives use the keyword BEFORE, the most recently entered operation is performed first.

Default: The initial default value for XGAIN is 1 (applied after XOFFSET).

Notes: The offset and gain factors allow the user to alter the raw data set being used without actually changing the raw data file.

The result of gains and offsets are reflected by the TABULATE directive.

The form NOGAIN will set XGAIN to 1 (applied after XOFFSET).

XMAX

Purpose: To set the maximum value of the x-axis parameter.

Syntax: [NO] XMA[X] [value]

value a maximum threshold value for the incoming raw data. Only raw data which is equal to or less than this maximum value is retained.

This argument is optional. If omitted, the initial default maximum is restored.

Default: The initial default value for XMAX is NONE.

Notes: If the new maximum value is equal to or less than the current minimum value, a warning message is written. The user must correct this situation before the user may tabulate or plot the data.

If the new maximum disqualifies all of the available raw data,

a warning message is written (see CHECK).

The form NOXMAX removes the maximum-value restriction from the incoming raw data.

XMIN

Purpose: To set the minimum value of the x-axis parameter.

Syntax: [NO] XMI[N] [value]

value a minimum threshold value for the incoming raw data. Only raw data which is equal to or greater than this minimum value is retained.

This argument is optional. If omitted, the initial default minimum is restored.

Default: The initial default value for XMIN is NONE.

Notes: If the new minimum value is equal to or greater than the current maximum value, a warning message is written. The user must correct this situation before the user may tabulate or plot the data. If the new minimum disqualifies all of the available raw data, a warning message is written (see CHECK).

The form NOXMIN removes the minimum-value restriction from the incoming raw data.

XOFFSET

Purpose: To set the x-axis variable's offset and order. This directive adds (subtracts) a specified value to the data set.

Syntax: [NO] XOF[FSET] [value] [order]

value an integer or real offset. For example, an offset of 19 would add 19 to all incoming raw x-axis data.

This argument is optional. If omitted, the initial default offset (0.) is assumed.

order a keyword indicating if the offset is to be applied before or after the gain. The possible keywords are:

BEF[ORE]
AFT[ER]

This argument is optional. If omitted, the offset will be applied after the gain.

Default: The initial default value for XOFFSET is 0 (applied before XGAIN).

Notes: The offset and gain factors allow the user to alter the raw data set being used without actually changing the raw data file.

The result of gains and offsets are reflected by the TABULATE directive.

The form NOXOFFSET will set XOFFSET to 0.

Y

Purpose: To set the name and range of the y-axis parameter.

Syntax: Y vname [range]

vname the y-axis variable name. If it contains embedded blanks, it must be enclosed within quotes.

range the range of values allowed for this data set. See the notes for range in the DATA Subsection for more information.

Default: The initial default value for Y is NONE.

YGAIN

Purpose: To set the y-axis variable's gain and order. This directive multiples a specified value to the data set.

Syntax: [NO] YGA[IN] [value] [order]

value an integer or real value. If omitted, the initial default gain is restored.

order an optional before/after order to be used:

BEF[ORE]
AFT[ER]

The keyword BEFORE forces the gain to be applied before the offset. The keyword AFTER forces the gain to be applied after the offset. If neither keyword is used, AFTER is assumed. If both the offset and the gain directives use or default to the keyword AFTER, the most recently entered operation is performed last. If both directives use the keyword BEFORE, the most recently entered operation is performed first.

Default: The initial default value for YGAIN is 1 (applied after YOFFSET).

Notes: The offset and gain factors allow the user to alter the raw data set being used

without actually changing the raw data file.

The result of gains and offsets are reflected by the TABULATE directive.

The form NOYGAIN will set YGAIN to 1.

YMAX

Purpose: To set the y-axis variable's maximum value.

Syntax: [NO] YMA[X] [value]

value the y-axis variable's new maximum value.

If the value is omitted, the maximum is restored to its initial default value. NOYMAX removes any existing maximum.

Default: The initial default value for YMAX is NONE.

YMIN

Purpose: To set the y-axis variable's minimum value.
Syntax: [NO] YMI[N] [value]

value the y-axis variable's new minimum value.

If the value is omitted, the minimum is restored to its initial default value. NOYMIN removes any existing minimum.

Default: The initial default value for YMIN is NONE.

YOFFSET

Purpose: To set the y-axis variable's offset and order. This directive adds (subtracts) a specified value to the data set.

Syntax: [NO] YOF[FSET] [value] [order]

value an integer or real offset. If omitted, the initial default offset is restored.

order an optional before/after order to be used:

BEF[ORE]
AFT[ER]

The keyword BEFORE forces the offset to be applied before the gain. The keyword AFTER forces the offset to be applied after the gain. If neither keyword is used, AFTER is assumed. If both the offset and the gain directives

use or default to the keyword AFTER, the most recently entered operation is performed last. If both directives use the keyword BEFORE, the most recently entered operation is performed first.

Default: The initial default value for YOFFSET is 0 (applied before YGAIN).

Notes: The offset and gain factors allow the user to alter the raw data set being used without actually changing the raw data file.

The result of gains and offsets are reflected by the TABULATE directive.

The form NOYOFFSET will set YOFFSET to 0.

****

* XAXIS and YAXIS Subsections *

****

AUTO

Purpose: To turn the automatic axis scaling option ON or OFF. When this option is turned ON, the current axis group's tick mark labels will be adjusted with maximum and minimum values which will both encompass the data set and provide suitable numeric tick mark labels. Thus, the attributes, MINIMUM, MAXIMUM, MAJOR, MINOR, LDIGITS, and RDIGITS, will be adjusted for the current open axis group.

Syntax: [NO] AUT[O]

Default: The initial default value for AUTO is ON.

BASE

Purpose: To set the axis characteristics of the base axis.

Syntax: [NO] BAS[E] [ifactor]

ifactor 0 - no axis
1 - line only
2 - line and tick marks
3 - full axis

If omitted the initial default value is restored.

Default: The initial default value for BASE is 3 - full axis.

CLOSE

Purpose: To close the open XAXIS or YAXIS subsection attribute group.

Syntax: CLO[SE]

CREATE

Purpose: To create and name a new XAXIS or YAXIS subsection attribute group.

Syntax: CRE[ATE] gname

gname the name of the XAXIS or YAXIS subsection attribute group being created.

DEFAULT

Purpose: To reset one or more groups' attributes to their original default values.

Syntax: DEF[AULT] [gname] [gname] ... (*)

gname the name of an existing XAXIS or YAXIS subsection attribute group to be defaulted.

If no XAXIS or YAXIS subsection attribute group names are used, the group currently open is defaulted.

No other directives may follow a DEFAULT directive on the same line.

DELETE

Purpose: To delete one or more XAXIS or YAXIS subsection attribute groups from the current session and from the permanent database.

Syntax: DEL[ETE] gname [gname] ... (*)

gname the name of the XAXIS or YAXIS subsection attribute group to be deleted.

All groups may be deleted by using the group name "all".

No other directives may follow a DELETE directive on the same line.

DIRECTION

Purpose: To set the direction (or path) along which the numerical values used in the tick mark labels. For example, a direction of 45-degrees, will cause the numerical tick mark labels to be drawn on a 45-degree angle.

Syntax: DIR[ECTION] [value]

value an integer or real angle (in degrees), from the positive x-axis, in a counter-clockwise direction. If omitted, the initial default direction angle is restored.

Default: The initial default value for DIRECTION is 0.

FMAJOR

Purpose: To set the format of the major tick marks.

Syntax: [NO] FMA[JOR] [ivalue]
[format]

ivalue a format indicator:

0 - no tick mark
1 - inside the axis
2 - outside the axis
3 - inside and outside the axis

format a format indicator:

IN - inside the axis
OUT - outside the axis
IN/OUT - inside and outside the axis
OUT/IN - inside and outside the axis
BOTH - inside and outside the axis

For either syntax, if the argument is omitted, the initial default format is restored. The form NOFMAJOR will set FMAJOR to 0.

Default: The initial default value for FMAJOR is 1.

FMINOR

Purpose: To set the format of the minor tick marks.

Syntax: [NO] FMI[NOR] [ivalue]
[format]

ivalue a format indicator:

0 - no tick mark
1 - inside the axis
2 - outside the axis
3 - inside and outside the axis

format a format indicator:

IN - inside the axis
OUT - outside the axis
IN/OUT - inside and outside the axis
OUT/IN - inside and outside the axis
BOTH - inside and outside the axis

For either syntax, if the argument is omitted, the initial default format is restored. The form NOFMINOR will set FMINOR to 0.

Default: The initial default value for FMINOR is 1.

FONT

Purpose: To set the character font for the numeric labels.

Syntax: FON[T] [ivalue]

ivalue an integer font code (site dependent - see Appendix D). If omitted, the initial default font code is restored.

Default: The initial default value for FONT is 1.

FRAME

Purpose: To set the axis characteristics of the "framing" axis.

Syntax: [NO] FRA[ME] [ifactor]

ifactor 0 - no axis
1 - line only
2 - line and tick marks
3 - full axis

If omitted the initial default value is restored.

Default: The initial default value for FRAME is 1 - line only.

GAP

Purpose: To set the inter-character gap for the numerical values used in the axis tick mark labels. The character gap is specified as a fraction of the average character width. The gap of 0.0 implies that adjacent character boxes are the abutting. A gap of 1.0 implies that a gap, equivalent to the size of the average character box, is to be inserted between adjacent characters. A gap of -0.5 implies that character boxes are to partially overlap.

Syntax: [NO] GAP [value]

value an integer or real gap ratio:

<0 - compressed/overlapped
0 - normal
>0 - expanded

If omitted, the initial default gap is restored.

Default: The initial default value for GAP is 0.

GRID

Purpose: To set the grid option.

Syntax: [NO] GRI[D]

The grid option draws solid lines from each major tick mark to the opposite side of the plot. If both the XAXIS and the YAXIS subsection grid options are used, the plot is cross-hatched.

Default: The initial default value for GRID is OFF.

Notes: If the legend is drawn in the data area, then the grid lines will refrain from entering the legend.

HEIGHT

Purpose: To set the height for the numerical values used in the axis tick mark labels.

Syntax: HEI[GHT] [value]

value an integer or real height. If omitted, the initial default height is restored.

Default: The initial default value for HEIGHT is .15

LABEL

Purpose: To set the axis label.

Syntax: [NO] LAB[EL] text (*)

text the axis label. If it contains embedded blanks, it must be enclosed within single (') or double (") quotes.

Default: The initial default value for LABEL is the group name.

Notes: The form NOLABEL will set the label to the null string (i.e. no label will appear).

No directives may follow the command LABEL on the same line.

LDIGIT

Purpose: To set the number of digits to the left of the decimal place in the numeric labels.

Syntax: [NO] LDI[GITS] [ivalue]

ivalue the integer number of digits. If omitted, the initial default number

of digits is restored.

Default: The initial default value for LDIGITS is 4.

Notes: The form NOLDIGITS will set LDIGITS to 0.

LDIRECTION

Purpose: To set the direction (or path) along which the axis label is to be written.

Syntax: LDIR[ECTION] [value]

value an integer or real angle (in degrees), from the positive x-axis, in a counter-clockwise direction. If omitted, the initial default direction angle is restored.

Default: The initial default value for LDIRECTION is 0.

LENGTH

Purpose: To set the axis length.

Syntax: LEN[GTH] [value]

value an integer or real length. If omitted, the initial default axis length is restored.

Default: The initial default value for LENGTH is 6.

LFONT

Purpose: To set the character font for the axis label.

Syntax: LFO[NT] [ivalue]

ivalue an integer font code (site dependent - see the user's guide, Appendix D). If omitted, the initial default font code is restored.

Default: The initial default value for LFONT is 1.

LGAP

Purpose: To set the inter-character gap for the axis label. The character gap is specified as a fraction of the average character width. The gap of 0.0 implies that adjacent character boxes are to be abutting. A gap of 1.0 implies that a gap, equivalent to the size of the average character box, is to be inserted between adjacent characters. A gap of -0.5 implies that character boxes are to partially overlap.

Syntax: [NO] LGA[P] [value]

value an integer or real gap ratio:

<0 - compressed/overlapped
0 - normal
>0 - expanded

If omitted, the initial default gap is restored.

Default: The initial default value for LGAP is 0.

LHEIGHT

Purpose: To set the height for the axis label.

Syntax: LHE[IGHT] [value]

value an integer or real height. If omitted, the initial default height is initial default height is restored.

Default: The initial default value for LHEIGHT is .15

LINE

Purpose: To set the line intensity of the axis lines.

Syntax: [NO] LIN[E] [lfactor]

lfactor a line intensity factor:

NON[E]
LIG[HT]
MED[IUM]
HEA[VY]

If omitted, the initial default value is restored.

Default: The initial default value for LINE is MEDIUM.

Notes: This option is only supported on devices which support differing line intensities.

LMAJOR

Purpose: To set the length of the major tick marks.

Syntax: LMA[JOR] [value]

value an integer or real length. If omitted, the initial default length is restored.

Default: The initial default value for LMAJOR is .15

LMINOR

Purpose: To set the length of the minor tick marks.

Syntax: LMI[NOR] [value]

value an integer or real length. If omitted, the initial default length is restored.

Default: The initial default value for LMINOR is .075

LPRECISION

Purpose: To set the precision for the axis label.

Syntax: LPR[ECISION] [ivalue]

ivalue an integer precision code (site dependent - see the user's guide, Appendix D). If omitted, the initial default precision level is restored.

Default: The initial default value for LPRECISION is 4.

LWIDTH

Purpose: To set the width of the axis label.

Syntax: LWI[DTH] [value]

value an integer or real width. If omitted, the initial default width is restored.

Default: The initial default value for LWIDTH is .15

LXOFFSET

Purpose: To set the x-axis offset for the axis label. The label is initially centered along the axis between the extreme values. This directive allows the user to shift the label based on this initial starting location. The form NOLXOFFSET will set XOFFSET to 0.

Syntax: [NO] LXO[FFSET] [value]

value an integer or real distance along the XAXIS which the axis label is to be offset. A negative value moves the label to the left. A positive value moves it to the right. If omitted, the initial default offset is restored.

Default: The initial default value for LXOFFSET is 0.

LYOFFSET

Purpose: To set the y-axis offset for the axis label. The label is initially centered along the axis between the extreme values. This directive allows the user to shift the label based on this initial starting location. The form NOYOFFSET will set YOFFSET to 0.

Syntax: [NO] LYO[FFSET] [value]

value an integer or real distance along the y-axis which the axis label is to be offset. A negative value moves the label down. A positive value moves it up. If omitted, the initial default offset is restored.

Default: The initial default value for LYOFFSET is -.15

MAJOR

Purpose: To set the number of major tick mark intervals.

Syntax: MAJ[OR] [ivalue]

ivalue the integer number of major intervals. If omitted, the initial default number of major tick intervals is restored.

Default: The initial default value for MAJOR is 5.

MAXIMUM

Purpose: To set the axis maximum.

Syntax: MAX[IMUM] [value]

value an integer or real maximum value. If omitted, the initial default maximum is restored.

Default: The initial default value for MAXIMUM is 100.

Notes: If the MAXIMUM equals the MINIMUM, AUTO is set to ON (which will reset both MINIMUM and MAXIMUM upon plotting). If the MAXIMUM is less than the MINIMUM, a decreasing axis will be plotted.

MINIMUM

Purpose: To set the axis minimum.

Syntax: MIN[IMUM] [value]

value an integer or real minimum value. If omitted, the initial default minimum is restored.

Default: The initial default value for MINIMUM is 0.

Notes: If the MINIMUM equals the MAXIMUM, AUTO is set to ON (which will reset both MINIMUM and MAXIMUM upon plotting). If the MINIMUM is greater than the MAXIMUM, a decreasing axis will be plotted.

MINOR

Purpose: To set the number of minor tick mark intervals within reach major tick mark interval.

Syntax: [NO] MINO[R] [ivalue]

ivalue the integer number of minor intervals within each major interval. If omitted, the initial default number of minor tick marks is restored.

Default: The initial default value for MINOR is 5.

Notes: The form NOMINOR will set MINOR to 1.

OPEN

Purpose: To open an existing XAXIS or YAXIS subsection attribute group, allowing it to be altered.

Syntax: OPE[N] gname

gname the name of an existing XAXIS or YAXIS subsection attribute group.

only one XAXIS or YAXIS subsection attribute group may be open at any given time. If a group is already open and a second group is opened, the first group will be closed.

POSITION

Purpose: To set the axis's position on the page.

Syntax: POS[ITION] [value]

value a page or world coordinate used to determine the position of the axis. The position values of both axes determine the X and Y location of the lower-left corner of the axes intersection. Although this directive can be used in an interactive mode, its

primary usage is in the postprocessing mode with the metafile. See Appendix F for a description of the use of metafiles.

Default: The initial default value for POSITION is 2.

PRECISION

Purpose: To set the precision for the numeric labels.

Syntax: PRE[CISION] [ivalue]

ivalue an integer precision code (site dependent see the user's guide, Appendix D). If omitted, the initial default precision level is restored.

Default: The initial default value for PRECISION is 4.

RDIGITS

Purpose: To set the number of digits to the right of the decimal place in the numeric labels.

Syntax: [NO] RDI[GITS] ivalue

ivalue the integer number of digits. If omitted, the initial default number of digits is restored.

Default: The initial default value for RDIGITS is 1.

Notes: The form NORDIGITS will set RDIGITS to 0.

SCIENTIFIC

Purpose: To set the scientific notation for the numeric axis labels.

Syntax: [NO] SCI[ENTIFIC] [ivalue]

ivalue the integer number representing the power of 10 to be displayed on the axis. NOSCI, or an ivalue of zero, will suppress the power of 10 on the axis.

If omitted the initial default value is restored.

Default: The initial default value for SCI is NO.

Notes: The value of SCI will shift the decimal point appropriately.

TURN

Purpose: To turn a XAXIS or YAXIS subsection attribute group on or off.

Syntax: TUR[N] state gname (*)

state an ON/OFF state:
ON
OFF

gname the name of an existing XAXIS or YAXIS subsection attribute group being turned on or off.

An XAXIS or YAXIS subsection attribute group which is ON is called "active", and will be used to format all subsequent plots made with the PLOT directive. Only one subsection attribute group may be active at any time. Once a group is turned on, all others are automatically turned off.

No other directives may follow a TURN directive on the same line.

USE

Purpose: To copy the information contained in an existing XAXIS or YAXIS subsection attribute group to help create a new group.

Syntax: USE gname

gname the name of an existing axis subsection attribute group containing the desired information set.

WIDTH

Purpose: To set the width of the numerical values used in the axis tick mark labels.

Syntax: WID[TH] [value]

value an integer or real width. If omitted, the initial default width is restored.

Default: The initial default value for WIDTH is .15

XOFFSET

Purpose: To set the x-axis offset for the numeric labels.

Syntax: [NO] XOF[FSET] [value]

value an integer or real distance along the x-axis which each numeric label is to be offset. A negative value moves the la-

bels to the left. A positive value moves them to the right. If omitted, the initial default offset is restored.

Default: The initial default value for XOFFSET is 0.

Notes: The form NOXOFFSET will set XOFFSET to 0.

YOFFSET

Purpose: To set the y-axis offset for the numeric labels.

Syntax: [NO] YOF[FSET] [value]

value an integer or real distance along the y-axis which each numeric label is to be offset. A negative value moves the labels down. A positive value moves them up. If omitted, the initial default offset is restored.

Default: The initial default value for YOFFSET is -.15

Notes: The form NOYOFFSET will set YOFFSET to 0.

ZERO

Purpose: To set the zero-line option.

Syntax: [NO] ZER[O]

The zero-line option draws a solid line perpendicular to the zero position on the axis. For example, the zero-line option on the horizontal axis draws a vertical line from the horizontal axis. If the grid option is used, the zero-line is emphasized.

Default: The initial default value for ZERO is OFF.

****

* LEGEND Subsection *

****

CLOSE

Purpose: To close the open LEGEND subsection attribute group.

Syntax: CLO[SE]

CREATE

Purpose: To create and name a new LEGEND subsection attribute group.

Syntax: CRE[ATE] gname

gname the name of the LEGEND subsection attribute group being created.

DEFAULT

Purpose: To reset one or more groups' attributes to their original default values.

Syntax: DEF[AULT] [gname] [gname] ... (*)

gname the name of an existing LEGEND subsection attribute group to be defaulted.

If no LEGEND subsection attribute group names are used, the group currently open is defaulted.

No other directives may follow a DEFAULT directive on the same line.

DELETE

Purpose: To delete one or more LEGEND subsection attribute groups from the current session and from the permanent database.

Syntax: DEL[ETE] gname [gname] ... (*)

gname the name of the LEGEND subsection attribute group to be deleted.

No other directives may follow a DELETE directive on the same line.

FONT

Purpose: To set the legend character font.

Syntax: FON[T] [ivalue]

ivalue an integer font code (site dependent - see Appendix D). If omitted, the initial default font code is restored.

Default: The initial default value for FONT is 1.

FRAME

Purpose: To set the legend frame option ON or OFF. This option will draw a box around the legend contents.

Syntax: [NO] FRA[ME]

Default: The initial default value for FRAME is ON.

GAP

Purpose: To set the inter-character gap for the LEGEND characters.

Syntax: [NO] GAP [value]
value an integer or real gap ratio:

<0 - compressed/overlapped
0 - normal
>0 - expanded

Default: The initial default value for GAP is 0. If omitted, the initial default gap is restored.

HEIGHT

Purpose: To set the legend character height.

Syntax: HEI[GHT] [value]

value an integer or real height. If omitted, the initial default height is restored.

Default: The initial default value for HEIGHT is .15

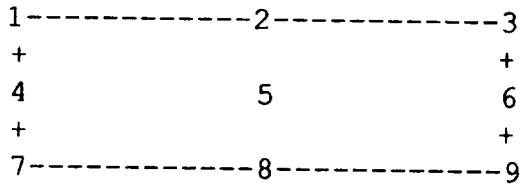
JUSTIFY

Purpose: To set the legend's justification. The placement of the legend, of a particular plot, is based on the justification (supplied by JUSTIFY) about the legend position (specified by X and Y). Thus a justification of 3 (upper right) will position the upper right corner of the legend at the position denoted by the legend X and Y.

Syntax: JUS[TIFY] [ivalue]
[text]

ivalue an integer justification indicator:

1 - upper left	6 - center right
2 - upper center	7 - lower left
3 - upper right	8 - lower center
4 - center left	9 - lower right
5 - center center	



text a justification description using one or more of the keywords.

TOP	UPPER
MIDDLE	CENTER
BOTTOM	LOWER

for example:
TOPCENTER
RIGHT/LOWER
CENTER-LEFT

Default: The initial default value for JUSTIFY is 3.

LCOLUMN

Purpose: To set the legend column header (i.e. the title over the legend column).

Syntax: [NO] LCO[LUMN] [text] (*)

text a character string which may include text, mnemonics, and phrases.

Default: The initial default value for LCOLUMN is '<LEGENDCOLHEADER>'.

Notes: The form NOLCOLUMN will suppress the legend column header (and space within the legend box), and set LCOLUMN to the null string.

LTITLE

Purpose: To set the legend header (i.e. the title centered over the entire legend area).

Syntax: [NO] LTI[TLE] [text] (*)

text a character string which may include text, mnemonics, and phrases.

Default: The initial default value for LTITLE is '<LEGENDTITLE>'.

Notes: The form NOTITLE will suppress the legend title (and space within the legend box), and set LTITLE to the null string.

OPEN

Purpose: To open an existing LEGEND subsection attribute group, allowing it to be altered.

Syntax: OPE[N] gname

gname the name of an existing LEGEND subsection attribute group.

Only one legend subsection attribute group may be open at any given time. If a group is already open and a second group is opened, The first group will be closed.

ORDER

Purpose: To set the "explicit" ordering of the data sets in the legend.

Syntax: [NO] ORD[ER] [gname] [gname] . . . (*)

gname The name of data groups to be ordered.
If omitted the initial default value is restored.

Default: The initial default value for ORDER is none, that is the order of data groups is "implicit" (i.e., the order of the data groups as shown is SHOW of the DATA subsection).

Notes: Only active and checked data groups with a key entry are displayed in the legend.

PRECISION

Purpose: To set the precision for legend characters.

Syntax: PRE[CISION] [ivalue]

ivalue an integer precision code (site dependent see the us-

er's guide, Appendix D). If omitted, the initial default precision level is restored.

Default: The initial default value for PRECISION is 4.

TURN

Purpose: To turn a LEGEND subsection attribute group ON or OFF.

Syntax: TUR[N] state gname [gname] ... (*)

state an ON/OFF state:

ON
OFF

gname the name of an existing LEGEND subsection attribute group being turned on or off.

A LEGEND Subsection attribute group which is ON is called "active", and will be used to format all subsequent plots made with the PLOT directive. Only one LEGEND subsection attribute group may be active at any time. Once a group is turned on, all others are automatically turned off.

No other directives may follow a TURN directive on the same line.

USE

Purpose: To use the information set contained in an existing legend subsection attribute group to help create a new group.

Syntax: USE gname

gname the name of an existing LEGEND subsection attribute group containing the desired information set.

WIDTH

Purpose: To set the legend character width.

Syntax: WID[TH] [value]

value an integer or real width. If omitted, the initial default width is restored.

Default: The initial default value for WIDTH is .12

X

Purpose: To set the legend's location in the x-direction of the page.

Syntax: X [value]

value a x-direction location. This value used with the LEGEND Subsection command JUSTIFY positions the legend in the x-direction.

Default: The initial default value for X is 8.

Y

Purpose: To set the legend's location in the y-direction of the page.

Syntax: Y [value]

value a y-direction location. This value used with the LEGEND Subsection command JUSTIFY positions the legend in the y-direction.

Default: The initial default value for Y is 8.

****

* PHRASE Subsection - Directives *

****

CLOSE

Purpose: To close the open phrase.

Syntax: CLO[SE]

CREATE

Purpose: To create and name a new phrase in the PHRASE subsection.

Syntax: CRE[ATE] pname

pname the name of the phrase being created.

DELETE

Purpose: To delete one or more PHRASE subsection attribute groups from the current session and from the permanent database.

Syntax: DEL[ETE] pname [pname] . . . (*)

pname the name of the PHRASE subsection attribute group to be deleted.

No other Directives may follow a DELETE directive on the same line.

DISPLAY

Purpose: To display the graphic representation a phrase.

Syntax: DIS[PLAY] [pname]

pname the name of the phrase being displayed.

This argument list is optional. If omitted, the phrase currently open is displayed.

Notes: If the phrase "pname" does not exist, a warning message is written.

If the argument list is omitted and no phrase is currently open, a warning message is written.

No directives may follow a DISPLAY directive on the same command line.

OPEN

Purpose: To open an existing phrase, allowing it to be altered.

Syntax: OPE[N] pname

pname the name of an existing phrase.

PREFIX

Purpose: To insert a block of text and/or mnemonics at the beginning of a phrase.

Syntax: PRE[FIX] item [item] . . . (*)

item a text string or a mnemonic. At least one item must be used.

Notes: A phrase must be open before the PREFIX directive is used. If not, a warning message is written.

If the open phrase is already filled to capacity, or if the combined length of the original phrase and the new text exceeds the phrase's capacity, a warning message is written. The user is then asked if new text is to be inserted. If the user chooses to insert the new text, some of the trailing text of the original phrase is lost. If not, the original phrase remains unaltered.

No other directive may follow a PREFIX directive on the same command line.

To determine the phrase capacity, the user may interactively type SHOW LIMITS from within the PHRASE Subsection.

REPLACE

Purpose: To replace the phrase's current contents with the given text and/or mnemonics.

Syntax: REP[LACE] item [item] . . . (*)

item is a text string or a mnemonic. At least one 1 item must be used.

Notes: A phrase must be open before a REPLACE directive is used. If not, a warning message is written.

If the length of the new text exceeds the phrase's capacity, a warning message is written. The user is then asked if the phrase is be be replaced. If the user chooses to replace the phrase, some of the new text's trailing characters are lost. If not, the original phrase remains unaltered.

No other directives may follow a REPLACE directive on the same command line.

To determine the phrase capacity, the user may interactively type SHOW LIMITS from within the PHRASE Subsection.

SUFFIX

Purpose: To insert a block of text and/or mnemonics at the end of a phrase.

Syntax: SUF[FIX] item [item] . . . (*)

item is a text string or a mnemonic. At least one item must be used.

Notes: A phrase must be open before an SUFFIX directive is used. If not, a warning message is written.

If the open phrase is already filled to capacity, or if the combined length of the original phrase and the new text exceeds the phrase's capacity, a warning message is written. The user is then asked if new text is to be added. If the user chooses to add the new text, some of the new text's trailing characters are lost. If not, the original phrase remains unaltered.

No other directives may follow a SUFFIX directive on the same command line.

To determine the phrase capacity, the user may interactively type SHOW LIMITS from within the PHRASE Subsection.

USE

Purpose: To copy information from one phrase into the open phrase.

Syntax: USE pname

pname the name of an existing phrase containing the information to be used.

This argument is required.

Notes: If a phrase is not currently open, a warning message is written.

If the phrase "pname" does not exist, a warning message is written, and the currently open phrase remains unaltered.

If the phrase "pname" is the phrase currently open, the directive has no effect.

This directive can only be used in conjunction with the PHRASE directives CREATE and REPLACE.

****

* PHRASE Subsection - MNEMONICS *

****

.(BAS)

Purpose: To reset the text level back to the original baseline level.

Syntax: .BAS

Notes: For more information about subscripting and superscripting, see "Text Appearance (Mnemonics and Phrases)" in Section 4.

This mnemonic performs the following three functions:

- moves the vertical position back to its original default baseline location (Note: the current horizontal position is not changed).

- restores the original default character height, width, and gap factor (Note: the current font index is not changed).

- resets the superscript and subscript indicators back to 0 (Note: the current scale and translation factors are not changed).

.ELC

Purpose: To turn off the lowercase indicator.

Syntax: .ELC

Notes: The .UC mnemonic performs the same function.

.ESUB

Purpose: To end the current subscript level, and jump back up to the next higher text level.

Syntax: .ESUB

Notes: For more information about subscripting and superscripting, see "Text Appearance (Mnemonics and Phrases)" in Section 4.

If this mnemonic is used while at the baseline level, it is ignored. If .ESUB is used when no subscripting is present in the current text, then it has no effect.

.ESUP

Purpose: To end the current superscript level, and drops back down to the next lower text level.

Syntax: .ESUP

Notes: For more information about subscripting and superscripting, see "Text Appearance (Mnemonics and Phrases)" in Section 4.

If this mnemonic is used while at the baseline level, it is ignored. If .ESUP is used when no superscripting is present in the current text, then it has no effect.

.EUC

Purpose: To turn off the uppercase indicator.

Syntax: .EUC

Notes: The .LC mnemonic performs the same function.

.EUND

Purpose: To turn off the underline indicator.

Syntax: .EUND

.FONT

Purpose: To set the font for all subsequent text.

Syntax: .FONT [index]

index an integer font index.

This argument is optional. If omitted, the initial default font is restored.

Notes: For a description of the fonts, please refer to the Appendix D.

.LC

Purpose: To turn on the lowercase indicator, so that all subsequent text is written in lowercase letters.

Syntax: .LC

Notes: Use .ELC or .UC to turn off the lowercase indicator.

.(NL)

Purpose: To begin a new line of text.

Syntax: .NL

Notes: The .NL mnemonic resets the vertical and horizontal positions such that any subsequent text is written on a new line. The vertical position is returned to the baseline level (see .BAS), then moved down the current character height plus the product of this height and the current vertical margin (see .VMA). The horizontal position is reset to its original location (i.e. the new line is left justified).

.SUB

Purpose: To begin writing any subsequent text at the next lower subscript level.

Syntax: .SUB

Notes: For more information about subscripting and superscripting, see "Text Appearance (Mnemonics and Phrases)" in Section 4.

Use .ESUB to return to the next higher text level.

.SUP

Purpose: To begin writing any subsequent text at the next higher superscript level.

Syntax: .SUP

Notes: For more information about subscripting and superscripting, see "Text Appearance (Mnemonics and Phrases)" in Section 4.

Use .ESUP to return to the next higher text level.

.UC

Purpose: To turn on the uppercase indicator, so that all subsequent text is written in uppercase letters.

Syntax: .UC

Notes: Use .EUC or .LC to turn off the uppercase indicator.

.UND

Purpose: To turn on the underline indicator, so that all subsequent text is underlined.

Syntax: .UND

Notes: Use .EUND to turn off the underline indicator.

.(VMA)

Purpose: To set the vertical margin factor between text lines.

Syntax: (VMA) [value]

value an integer or real value expressing the vertical margin factor. This value must be greater than or equal to zero.

This argument is optional. If omitted, the initial default vertical margin factor is restored.

Notes: The vertical margin factor is expressed as a ratio of the vertical space between two stacked text lines divided by the character height of the upper text line. For example, a vertical margin factor of 0.5 produces a vertical margin of half the upper text line's character height, which is single line spacing. A factor of 1.5 produces a vertical margin of one and a half the upper text line's character height, which is double line spacing.

The vertical margin factor only applies after a .NL mnemonic is used.

****

* TITLE Subsection *

****

CLOSE

Purpose: To close the open TITLE subsection attribute group.

Syntax: CLO[SE]

CREATE

Purpose: To create and name a new TITLE subsection attribute group.

Syntax: CRE[ATE] gname

gname the name of the TITLE subsection attribute group being created.

DELETE

Purpose: To delete one or more TITLE subsection attribute groups from the current session and from the permanent database.

Syntax: DEL[ETE] gname [gname] . . . (*)

gname the name of the TITLE subsection attribute group to be deleted.

No other directives may follow a DELETE directive on the same line.

FONT

Purpose: To set the character font for the TITLE.

Syntax: FON[T] [ivalue]

ivalue an integer font code (site dependent - see Appendix D). If omitted, the initial default font code is restored.

Default: The initial default value for FONT is 1.

GAP

Purpose: To set the inter-character gap for the TITLE.

Syntax: [NO] GAP [value]

value an integer or real gap ratio:

<0 - compressed/overlapped

0 - normal
>0 - expanded

Default: The initial default value for GAP is 0. If omitted, the initial default gap is restored.

HEIGHT

Purpose: To set the height for the TITLE.

Syntax: HEI[GHT] [value]

value an integer or real height. If omitted, the initial default height is restored.

Default: The initial default value for HEIGHT is .2

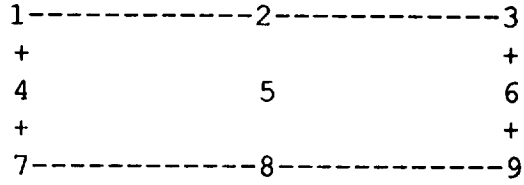
JUSTIFY

Purpose: To set the title's justification. The placement of the title of a particular plot is based on the justification (supplied by JUSTIFY) about the title position (specified by X and Y). Thus a justification of 3 (upper right) will position the upper right corner of the legend at the position denoted by the title X and Y.

Syntax: JUS[TIFY] [ivalue]
[text]

ivalue an integer justification indicator:

1 - upper left	6 - center right
2 - upper center	7 - lower left
3 - upper right	8 - lower center
4 - center left	9 - lower right
5 - center center	



text a justification description using one or more of the keywords:

TOP	UPPER
MIDDLE	CENTER
BOTTOM	LOWER

for example:

TOPCENTER
RIGHT/LOWER
CENTER-LEFT

Default: The initial default value for JUSTIFY is 2.

OPEN

Purpose: To open an existing TITLE subsection attribute group, allowing it to be altered.

Syntax: OPE[N] gname

gname the name of an existing TITLE subsection attribute group.

Only one TITLE subsection attribute group may be open at any given time. If a group is already open and a second group is opened, The first group will be closed.

TEXT

Purpose: To set the TITLE text.

Syntax: [NO] TEX[T] [text] [text] ... (*)

text the title text. If it contains embedded blanks, it must be enclosed within single (') or double (") quotes.

Default: The initial default value for TEXT is '<TITLE>'.

No directives may follow a TEXT directive on the same command line.

TURN

Purpose: To turn a TITLE subsection attribute group ON or OFF.

Syntax: TUR[N] state gname ... (*)

state an ON/OFF state:

ON
OFF

gname the name of an existing TITLE subsection attribute group being turned on or off.

A TITLE Subsection attribute group which is ON is called

"active", and will be used to format all subsequent plots made with the PLOT directive. Only one TITLE subsection attribute group may be active at any time. Once a group is turned on, all others are automatically turned off.

No other Directives may follow a TURN directive on the same line.

WIDTH

Purpose: To set the title character width.

Syntax: WID[TH] [value]

value an integer or real width. If omitted, the initial default width is restored.

Default: The initial default value for WIDTH is .2

USE

Purpose: To use the information set contained in an existing TITLE subsection attribute group to help create a new group.

Syntax: USE gname

gname the name of an existing TITLE subsection attribute group containing the desired information set.

X

Purpose: To set the TITLE's location in the x-direction of the page.

Syntax: X [value]

value a x-direction location. This value used with the TITLE Subsection command JUSTIFY positions the title in the x-direction.

Default: The initial default value for X is 5.5

Y

Purpose: To set the TITLE's location in the y-direction of the page.

Syntax: Y [value]

value a y-direction location. This value used with the TITLE Subsection command JUSTIFY positions the title in the y-direction.

Default: The initial default value for Y is 9.

****

* NOTE Subsection *

****

CLOSE

Purpose: To close the open NOTE subsection attribute group.

Syntax: CLO[SE]

CREATE

Purpose: To create and name a new NOTE subsection attribute group.

Syntax: CRE[ATE] gname

gname the name of the NOTE subsection attribute group being created.

DELETE

Purpose: To delete one or more NOTE subsection attribute groups from the current session and from the permanent database.

Syntax: DEL[ETE] gname [gname] . . . (*)

gname the name of the NOTE subsection attribute group to be deleted.

No other directives may follow a DELETE directive on the same line.

FONT

Purpose: To set the character font for the NOTE.

Syntax: FON[T] [ivalue]

ivalue an integer font code (site dependent - see Appendix D). If omitted, the initial default font code is restored.

Default: The initial default value for FONT is 1.

GAP

Purpose: To set the inter-character gap for the NOTE.

Syntax: [NO] GAP [value]

value an integer or real gap ratio:

<0 - compressed/overlapped

0 - normal
>0 - expanded

Default: The initial default value for GAP is 0. If omitted, the initial default gap is restored.

HEIGHT

Purpose: To set the height for the NOTE.

Syntax: HEI[GHT] [value]

value an integer or real height. If omitted, the initial default height is restored.

Default: The initial default value for HEIGHT is .2

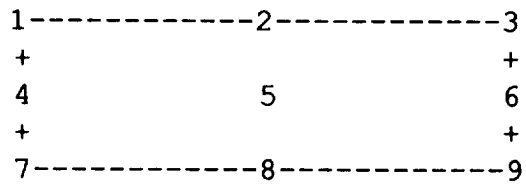
JUSTIFY

Purpose: To set the note's justification. The placement of the note of a particular plot is based on the justification (supplied by JUSTIFY) about the note position (specified by X and Y). Thus a justification of 3 (upper right) will position the upper right corner of the note at the position denoted by the NOTE X and Y.

Syntax: JUS[TIFY] [ivalue]
[text]

ivalue an integer justification indicator:

1 - upper left	6 - center right
2 - upper center	7 - lower left
3 - upper right	8 - lower center
4 - center left	9 - lower right
5 - center center	



text a justification description using one or more of the keywords:

TOP	UPPER
MIDDLE	CENTER
BOTTOM	LOWER

for example:

TOPCENTER
RIGHT/LOWER
CENTER-LEFT

Default: The initial default value for JUSTIFY is 2.

OPEN

Purpose: To open an existing NOTE subsection attribute group, allowing it to be altered.

Syntax: OPE[N] gname

gname the name of an existing NOTE subsection attribute group.

Only one NOTE subsection attribute group may be open at any given time. If a group is already open and a second group is opened, The first group will be closed.

TEXT

Purpose: To set the NOTE text.

Syntax: [NO] TEX[T] [text] [text] . . . (*)

text the NOTE text. If it contains embedded blanks, it must be enclosed within single (') or double (") quotes.

Default: The initial default value for TEXT is '<NOTE>'.

No directives may follow a TEXT directive on the same command line.

TURN

Purpose: To turn a NOTE subsection attribute group ON or OFF.

Syntax: TUR[N] state gname . . . (*)

state an ON/OFF state:

ON
OFF

gname the name of an existing NOTE subsection attribute group being turned on or off.

A NOTE Subsection attribute group which is ON is called

"active", and will be used to format all subsequent plots made with the PLOT directive. Only one NOTE subsection attribute group may be active at any time. Once a group is turned on, all others are automatically turned off.

No other Directives may follow a TURN directive on the same line.

WIDTH

Purpose: To set the NOTE character width.

Syntax: WID[TH] [value]

value an integer or real width. If omitted, the initial default width is restored.

Default: The initial default value for WIDTH is .2

USE

Purpose: To use the information set contained in an existing NOTE subsection attribute group to help create a new group.

Syntax: USE gname

gname the name of an existing NOTE subsection attribute group containing the desired information set.

X

Purpose: To set the NOTE's location in the x-direction of the page.

Syntax: X [value]

value a x-direction location. This value used with the NOTE Subsection command JUSIFY positions the note in the x-direction.

Default: The initial default value for X is 5.5

Y

Purpose: To set the NOTE's location in the y-direction of the page.

Syntax: Y [value]

value a y-direction location. This value used with the NOTE Subsection command JUSIFY positions the note in the y-direction.

Default: The initial default value for Y is 9.

Appendix A

Program Execution

The TADPLOT Program is written in FORTRAN 77 and can be used on any platform that supports DI-3000 (its underlying graphics package). The program can be used in an interactive or batch capacity. The program requires a minimum field length of approximately 250K (usually more). Appendix A describes how to run TADPLOT, using the Convex, a UNIX operating system, as the example architecture.

Interactive Usage

To use the TADPLOT Program interactively, an interactive plotting device must be used. Since the TADPLOT Program uses DI-3000 as its underlying graphics package, only those graphics devices which have a locally supported DI-3000 device driver can be used. For a list of locally supported device drivers, call the Plot Program Administrator (or for Langley-NASA personnel, contact the Computer Complex, Central Scientific: User Consultation Office).

Select the appropriate device driver for your terminal according to how DI-3000 is implemented on your machine. On the Convex, the device driver is selected by setting the environment variable PVI_DEV_1 to the appropriate device:

```
setenv PVI_DEV_1 407
```

The statement selects the Tektronix 4107/4109 device driver.

To invoke the Plot Program interactively, determine where the TADPLOT absolute file resides on your system, then invoke the program:

```
~ntflib/tadplot
```

where ~ntflib is the directory containing the tadplot executable.

Batch Usage

To use the TADPLOT Program in a batch mode, an interactive device is not necessary. Thus the user can omit selecting the device driver during invocation.

The plot program commands are read from INPUT (as during the interactive use), and thus the commands must be entered appropriately in the batch job stream.

The following example is one method to invoke the plot code in the batch environment on Convex. For additional information pertaining to the batch environment, see the appropriate documentation for your machine.

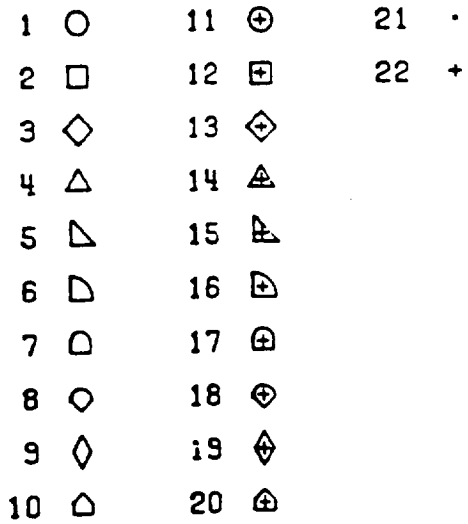
```
#
# Example of executing TADPLOT in the batch mode on Convex
#-----------------------------------------------------------------
# # This section sets Convex control options.
#
# @$-lt "1:00"
# # Set the maximum per-process CPU time limit to 1 minute.
#
# @$-mb -me
# # Send mail at beginning and end of the request execution.
#
# @$-l
# # Ensure the batch mode is LOGIN.
#
# @$-y
# # Append accounting information at the end of output.
#
# @$ # No more embedded flags.
#-----------------------------------------------------------------
cd /scr/temparea
# Change directory to a temporary area
cp ~ntflib/TAU4A /scr/temparea
# Copy the data file into the temporary area
~ntflib/tadplot < ~ntflib/SESIN4
# Invoke the tadplot program and read the commands from SESIN4
mfcal11 -i ~ntflib/cont -h11.0 -w11.0 metafl.dat | lpr -Pcal11 -C "delivery_information"
# Send metafile to plotting device (i.e., CalComp 11" plotter)
rm TAU4A metafl.dat
# Remove file TAU4A and metafl.dat. Leave LOG for verification.
```

Appendix B

Symbol Codes

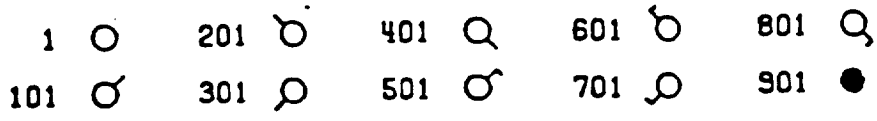
The symbol codes available are implemented by the Plot program and thus are not limited to the underlying graphics package's symbols. The symbols included are those recommended in the publication quality standards as presented at NASA Langley Research Center.

The symbols include:



Note:

Alternative symbols (see below) may be obtained by adding the appropriate multiple of a hundred. Symbols greater than 900, will be solid filled with the color oppose the device background (i.e. "device normal").



C-2

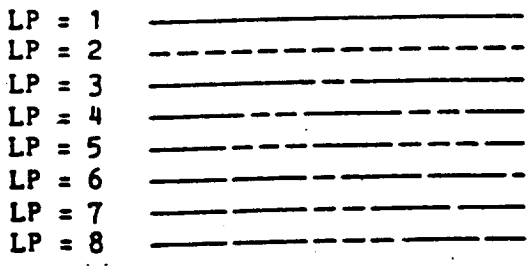
This page is intentionally left blank.

Appendix C

Line Patterns

The line patterns available are implemented by the Plot program and thus are not limited to the underlying graphics package's line patterns. The patterns included are those recommended in the publication quality standards as presented at NASA Langley Research Center.

The patterns include:



Note:

The line pattern is uniquely identified by one unit of length (i.e. one inch). Thus to fully distinguish lines, based on the line pattern alone, one whole unit is necessary (and is the default).

This page is intentionally left blank.

Appendix D

Character Fonts

This appendix describes the fonts available within the TADPLOT Program, and the effect "precision" has on the graphical character output. The number and types of fonts available is wholly dependent on the underlying graphics package being used.

Currently, the TADPLOT Program is using DI-3000 as its underlying graphics package. The following excerpts are taken from the DI-3000 (version 4) User's Guide.

The following table describes the relationship between text precision and text attributes. Note, precision 4 (Graphics Art) supports all text attributes, and is therefore the default throughout the Plot Program.

	(1) String (Low Quality)	(2) Character (Medium Quality)	(3) Stroke (High Quality)	(4) Graphic Arts (Highest (Quality)
Drawing Speed	Fast and Efficient	Less efficient than string	Less efficient than character	As slow as 'stroke' or slower, depending on font
Implementation	Hardware	Hardware	Software	Software
Path	Ignored	Exact adherence	Exact adherence	Exact adherence
Font	Device dependent	Device dependent	Plain, simple Uppercase only No special symbols	Varied[a]
Justification	Adheres closely	Adheres closely	Exact adherence	Exact adherence
Character Size	Approximate[b] adherence	Approximate[b] adherence	Exact adherence	Exact adherence
Gap	Device dependent	Approximate[b] adherence	Exact adherence	Exact adherence
Proportional Spacing	No[c]	No[c]	No	Yes
Character Base	Ignored	Exact Adherence	Exact Adherence	Exact Adherence
Character Plane	Ignored	Ignored	Exact Adherence	Exact Adherence

[a]See Figure 3.5.6 for Fonts.
[b]Depends on hardware character sizes available in the device, the closest hardware size equal to or less than the specified character size will be used
[c]Some high-quality output devices provide proportionately spaced hardware fonts

Currently, DI-3000 supports twenty-four graphics arts precision fonts, of which twenty-two are text, and two are symbol.

Font Index	Graphic Arts Font	Font Index	Graphic Arts Font
1	Simplex block	13	Complex Cyrillic block
2	Simplex italics	14	Complex Cyrillic italics
3	Duplex block	15	Gothic English block
4	Duplex italics	16	Gothic English italics
5	Complex block	17	Gothic German block
6	Complex italics	18	Gothic German italics
7	Triplex block	19	Gothic Italian block
8	Triplex italics	20	Gothic Italian italics
9	Greek block	21	Swedish block
10	Greek italics	22	Swedish italics
11	Script block	23	Symbol Font I
12	Script italics	24	Symbol Font II

The remainder of Appendix D provides a complete list of the contents of these fonts.

Decimal Character Number	Font 1 — Simplex	Font 3 — Duplex	Font 5 — Complex	Font 7 — Triplex	Font 9 — Greek	Font 11 — Script	Font 13 — Complex Cyrillic	Font 15 — Gothic English	Font 17 — Gothic German	Font 19 — Gothic Italian	Font 21 — Swedish	Font 23 — Symbol I	Font 24 — Symbol II
32													
33	!	!	!	!	∫	!	!	!	!	!	!	○	
34	"	"	"	"	∮	"	"	"	"	"	"	⊕	
35	#	#	#	#	#	#	#	#	#	#	#	◇	
36	$	$	$	$	✓	$	Ы	$	$	$	$	◇	
37	%	%	%	%	%	%	Ц	%	%	%	%	¤	
38	&	&	&	&	∞	&	ы	&	&	&	&	∞	
39	'	'	'	'	'	'	'	'	'	'	'	○	
40	(	(	(	(	(	(	(	(	(	(	(	∞	
41	)	)	)	)	)	)	)	)	)	)	)	□	
42	*	*	*	*	*	*	*	*	*	*	*	⌒	
43	+	+	+	+	+	+	+	+	+	+	+	⌣	
44	,	,	,	,	,	,	,	,	,	,	,	⌒	
45	—	—	—	—	—	—	—	—	—	—	—	⌒	
46	.	.	.	.	.	.	.	.	.	.	.	∽	
47	/	/	/	/	/	/	/	/	/	/	/	∩	
48	0	0	0	0	0	0	0	0	0	0	0	∪	≺
49	1	1	1	1	1	1	1	1	1	1	1	⊂	≺
50	2	2	2	2	2	2	2	2	2	2	2	⊃	
51	3	3	3	3	3	3	3	3	3	3	3	⊔	
52	4	4	4	4	4	4	4	4	4	4	4	⌓	◇
53	5	5	5	5	5	5	5	5	5	5	5	⊠	◇
54	6	6	6	6	6	6	6	6	6	6	6	♣	┼
55	7	7	7	7	7	7	7	7	7	7	7	⌂	┼

Decimal Character Number	Font 1 — Simplex	Font 3 — Duplex	Font 5 — Complex	Font 7 — Triplex	Font 9 — Greek	Font 11 — Script	Font 13 — Complex Cyrillic	Font 15 — Gothic English	Font 17 — Gothic German	Font 19 — Gothic Italian	Font 21 — Swedish	Font 23 — Symbol I	Font 24 — Symbol II
56	8	8	8	8	8	8	8	8	8	8	8	♀	
57	9	9	9	9	9	9	9	9	9	9	9	♂	
58	:	:	:	:	≠	:	:	:	:	:	:	♁	
59	;	;	;	;	≡	;	;	;	;	;	;	♀	
60	<	<	<	<	<	§	<	<	<	<	<	♀	
61	=	=	=	=	=	=	=	=	=	=	=	♄	
62	>	>	>	>	>	†	>	>	>	>	>	○	
63	?	?	?	?	≦	?	?	?	?	?	?	△	?
64	@	@	@	@	≧	@	@	●	●	●	●	▽	
65	A	A	A	A	A	𝒜	A	𝔄	𝔄	𝔄	A	◁	—
66	B	B	B	B	B	ℬ	Б	𝔅	𝔅	𝔅	B	▷	|
67	C	C	C	C	Г	𝒞	Э	ℭ	ℭ	ℭ	C	△△	┼
68	D	D	D	D	Δ	𝒟	Д	𝔇	𝔇	𝔇	D	○	└
69	E	E	E	E	E	ℰ	Й	𝔈	𝔈	𝔈	E	▱	┘
70	F	F	F	F	Z	ℱ	Ф	𝔉	𝔉	𝔉	F	▭	┐
71	G	G	G	G	H	𝒢	Г	𝔊	𝔊	𝔊	G	▯	┌
72	H	H	H	H	Θ	ℋ	Ж	𝔎	𝔎	𝔎	H	∪	┼
73	I	I	I	I	I	ℐ	И	𝔍	𝔍	𝔍	I	∧	┼
74	J	J	J	J		𝒥	Ч	𝔍	𝔍	𝔍	J	∨	┴
75	K	K	K	K	K	𝒦	К	𝔎	𝔎	𝔎	K	<	┤
76	L	L	L	L	Λ	ℒ	Л	𝔏	Ω	𝔏	L	>	┬
77	M	M	M	M	M	ℳ	M	𝔐	𝔐	𝔐	M	◇	├
78	N	N	N	N	N	𝒩	H	𝔑	𝔑	Ω	N	Y	╪
79	O	O	O	O	Ξ	𝒪	O	𝔒	𝔒	O	o	∟	┈

Decimal Character Number	Font 1 — Simplex	Font 3 — Duplex	Font 5 — Complex	Font 7 — Triplex	Font 9 — Greek	Font 11 — Script	Font 13 — Complex Cyrillic	Font 15 — Gothic English	Font 17 — Gothic German	Font 19 — Gothic Italian	Font 21 — Swedish	Font 23 — Symbol I	Font 24 — Symbol II
80	P	P	P	P	O	$\mathcal{P}$	П	𝔓	𝔓	𝔓	P	⊤	⋕
81	Q	Q	Q	Q	Π	$\mathcal{Q}$	Ш	𝔔	𝔔	𝔔	Q	⊥	⋔
82	R	R	R	R	P	$\mathcal{R}$	Р	𝔑	𝔑	𝔑	R	⊢	⋕
83	S	S	S	S	Σ	$\mathcal{S}$	С	𝔖	𝔖	𝔖	S	⫟	⊢
84	T	T	T	T	T	$\mathcal{T}$	Т	𝔗	𝔗	𝔗	T	⌶	↓
85	U	U	U	U	ϒ	$\mathcal{U}$	Ю	𝔘	𝔘	𝔘	U	∣	⊸
86	V	V	V	V	Φ	$\mathcal{V}$	В	𝔙	𝔙	𝔙	V	—	↑
87	W	W	W	W		$\mathcal{W}$	Ш	𝔚	𝔚	𝔚	W	⧵	⇃
88	X	X	X	X	X	$\mathcal{X}$	Х	𝔛	𝔛	𝔛	X	≀	⊬
89	Y	Y	Y	Y	Ψ	$\mathcal{Y}$	У	𝔜	𝔜	𝔜	Y	∼	
90	Z	Z	Z	Z	Ω	$\mathcal{Z}$	З	ℨ	𝔜	𝔜	Z	⑥	
91	[	[	[	[	[	[	Е	[	[	[	Ä	↯	
92	\	\	\	\	⊂	\	\	\	\	\	Ö	⧘	
93	]	]	]	]	]	]	Ъ	]	]	]	Å	∥	
94	⌢	⌢	⌢	⌢	∪	⌢	Я	⌢	⌢	⌢	^	⩵	
95					⊃							⧵⧵	
96	‘	‘	‘	‘	∩	‡	‘	‘	‘	‘	⧸	#	∘
97	a	a	a	a	α	α	а	𝔞	𝔞	𝔞	a	⫽	♦
98	b	b	b	b	β	$\mathcal{b}$	б	𝔟	𝔟	𝔟	b	⫷	⋄
99	c	c	c	c	γ	$\mathcal{c}$	Э	𝔠	𝔠	𝔠	c	↑	*
100	d	d	d	d	δ	$\mathcal{d}$	д	𝔡	𝔡	𝔡	d	↓	✦
101	e	e	e	e	ε	$\mathcal{e}$	й	𝔢	𝔢	𝔢	e	←	⚠
102	f	f	f	f	ζ	$\mathcal{f}$	ф	𝔣	𝔣	𝔣	f	→	➧
103	g	g	g	g	η	$\mathcal{g}$	г	𝔤	𝔤	𝔤	g	↖	⚠

Decimal Character Number	Font 1 — Simplex	Font 3 — Duplex	Font 5 — Complex	Font 7 — Triplex	Font 9 — Greek	Font 11 — Script	Font 13 — Complex Cyrillic	Font 15 — Gothic English	Font 17 — Gothic German	Font 19 — Gothic Italian	Font 21 — Swedish	Font 23 — Symbol I	Font 24 — Symbol II
104	h	h	h	h	ϑ	*h*	ж	ℌ	𝔥	*h*	h	⌐	♦
105	i	i	i	i	ι	*i*	и	i	i	*i*	ı	⇈	♦
106	j	j	j	j		*j*	ч	j	i	*j*	ȷ	↕	♦
107	k	k	k	k	κ	*k*	к	k	𝔨	*k*	k	⇉	♣
108	l	l	l	l	λ	*l*	л	l	I	*l*	ı	↔	♦
109	m	m	m	m	μ	*m*	м	m	m	*m*	m	→	♦
110	n	n	n	n	ν	*n*	н	n	n	*n*	n	⇳	♠
111	o	o	o	o	ξ	*o*	о	o	o	*o*	o	+	♠
112	p	p	p	p	o	*p*	п	p	𝔭	*p*	p	++	•
113	q	q	q	q	π	*q*	ш	q	q	*q*	q	✕	
114	r	r	r	r	ρ	*r*	р	r	r	*r*	r	✳	
115	s	s	s	s	σ	*s*	с	s	ſ	*s*	s	✩	
116	t	t	t	t	τ	*t*	т	t	t	*t*	t		
117	u	u	u	u	υ	*u*	ю	u	u	*u*	u		
118	v	v	v	v	φ	*v*	в	v	v	*v*	v		
119	w	w	w	w		*w*	щ	w	w	*w*	w		
120	x	x	x	x	χ	*x*	х	x	x	*x*	x		
121	y	y	y	y	ψ	*y*	у	y	y	*y*	y		
122	z	z	z	z	ω	*z*	з	z	z	*z*	z		
123	{	{	{	{	∂	{	е	{	{	{	ä		
124	|	|	|	|	∇	|	ъ	|	|	|	ö		
125	}	}	}	}	∈	}	я	}	}	}	å		
126	~	~	~	~	~	~	ь	~	~	~	~		
127													

Appendix E

Sample Sessions

The following sample sessions are to be regarded as distinct and independent from one another. Thus, the results from one session is not referenced by subsequent sessions. Each session assumes that the user has gotten the data files necessary, and invoked TADPLOT.

```
cd /scr/myarea (change directory to temporary area)
cp ~/ntflib/TAU4A /scr/myarea      (the sessions' data files)
cp ~/ntflib/J45CDM8 /scr/myarea    (into the temporary area)
```

On the Convex, the device driver is selected by setting the environment variable PVI_DEV_1 to the appropriate device:

```
setenv PVI_DEV_1 407
```

The statement selects the Tektronix 4107/4109 device driver.

To invoke the Plot Program interactively, determine where the TADPLOT absolute file resides on your system, then invoke the program:

```
~ntflib/tadplot
```

where ~ntflib is the directory containing the tadplot executable.

TADPLOT

PRODUCTION VERSION 2.00

OCTOBER 1991

This is the Plot Program introduction. It informs the user of the current version number, and the last revision date.

The program prompts the user with a "?", then waits for a command.

Sample session 1 illustrates how to generate a plot easily (i.e. only using default attributes).

TADPLOT

PRODUCTION VERSION 2.00

OCTOBER 1991

? CREATE C1 FILE TAU4A X T Y PDEG SELECT CASE 2
? CRE C2 FIL TAU4A X T Y PDEG SEL CASE 1
? PLOT

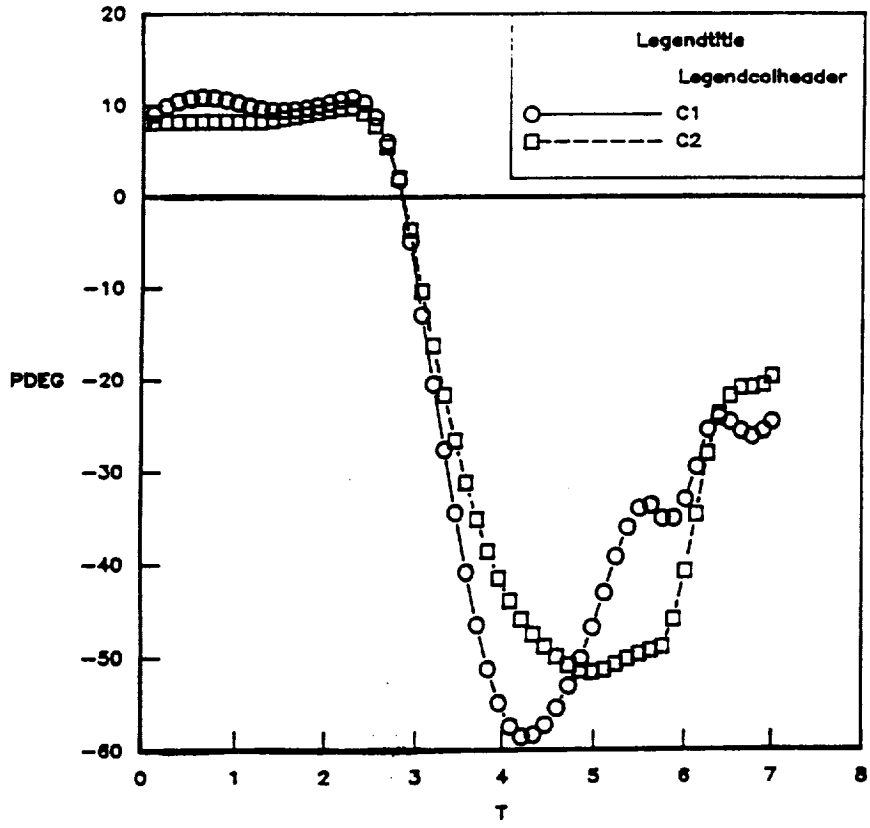


Figure:

Sample session 2 acquaints the user with TADPLOT in terms of structure and execute flow.

TADPLOT

PRODUCTION VERSION 2.00

OCTOBER 1991

? @ COMMENTS CAN BE ENTERED WITH COMMANDS/DIRECTIVES.

? @ WHERE AM I IN THE PROGRAM?

? WHEREAMI

YOU ARE IN THE DATA SUBSECTION.

? @ WHAT COMMANDS/DIRECTIVES ARE CURRENTLY AVAILABLE?

? HELP

EXECUTIVE DIRECTIVES:

COPY	LOG	PLOT	TITLE
DATA	MARKERSIZE	PWIDTH	TOLERANCE
ECHO	META	READ	WHEREAMI
END	NOTE	REMARK	XAXIS
EXECUTIVE	PAGESIZE	SAVE	YAXIS
HELP	PHEIGHT	SHOW	
LEGEND	PHRASE	STOP	

DATA SUBSECTION DIRECTIVES:

CHECK	FIT	SIGMA	XMAX
CLOSE	FORMAT	SORT	XMIN
CREATE	KEY	SYMBOL	XOFFSET
DEFAULT	LINE	TABULATE	Y
ENHANCE	SCAN	X	YMIN
FILE	SELECT	XGAIN	YOFFSET

? @ WHAT ARE THE CURRENT ATTRIBUTES OF THIS SUBSECTION?

? SHOW

DATA GROUP: DEFAULT OPEN ACTIVE
KEY : <DEFAULT>

FILE: FORMAT: UNKNOWN

X: MIN: NONE MAX: NONE
Y: MIN: NONE MAX: NONE

SELECTION VARIABLES: NONE

SYMBOL CODE: 2 EVERY POINT FIRST AND LAST POINTS
LINE CODE: 2 CURVE FIT: LINEAR

X-AXIS: OFFSET: 0. GAIN: 1. OFFSET GAIN
Y-AXIS: OFFSET: 0. GAIN: 1. OFFSET GAIN

DEHANCEMENT FACTOR: 1 UNSORTED NOT CHECKED

? @ CREATE 2 DATA SETS AND PLOT THEM.

? CRE C1 FILE TAU4A X T Y PDEG SEL CASE 2 NOSYM

? CRE C2 USE C1 SEL CASE 1

? PLOT

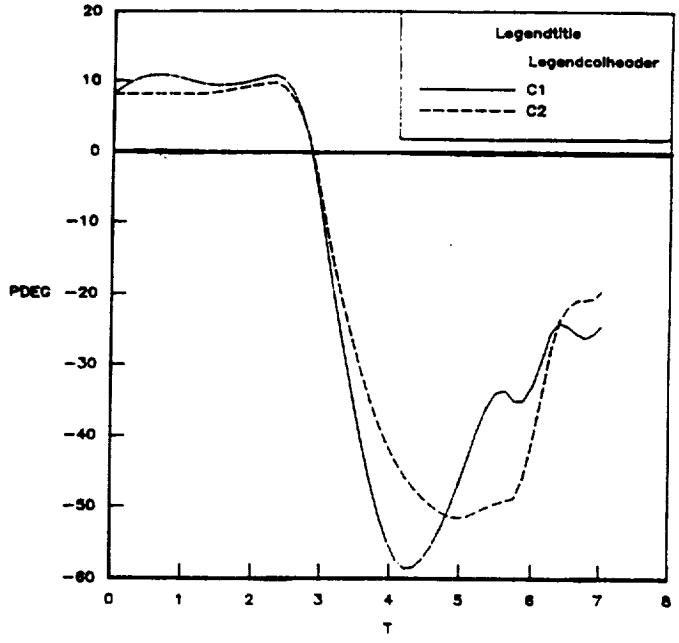


Figure:

? @ SUPPOSE WE WANT TO CHANGE THE TITLE. FIRST ENTER THE TITLE SUBSECTION.

? TITLE

? @ WHAT ARE THE CURRENT ATTRIBUTES?

? SHO

TITLE GROUP: DEFAULT OPEN ACTIVE

TEXT : <T.LC ITLE.UC >

FONT: 1
HEIGHT: .2 WIDTH: .16 GAP: 0.

X: 5.5 Y: .5 JUSTIFICATION: UPPER-CENTER

? @ WHAT ARE THE AVAILABLE COMMANDS/DIRECTIVES?

? HELP

EXECUTIVE DIRECTIVES:

COPY	LOG	PLOT	TITLE
DATA	MARKERSIZE	PWIDTH	TOLERANCE
ECHO	META	READ	WHEREAMI
END	NOTE	REMARK	XAXIS
EXECUTIVE	PAGESIZE	SAVE	YAXIS
HELP	PHEIGHT	SHOW	
LEGEND	PHRASE	STOP	

TITLE SUBSECTION DIRECTIVES:

CLOSE	GAP	TEXT	X
CREATE	HEIGHT	TURN	Y
DELETE	JUSTIFY	USE	
FONT	OPEN	WIDTH	

? @ MORE DETAILED INFORMATION ABOUT TITLE JUSTIFICATION.

? HELP JUST

JUSTIFY SETS THE TITLE'S JUSTIFICATION.

SYNTAX: JUS[TIFY] [IVALUE]
[TEXT]

IVALUE - INTEGER JUSTIFICATION INDICATOR:

1 - UPPER LEFT	6 - CENTER RIGHT
2 - UPPER CENTER	7 - LOWER LEFT
3 - UPPER RIGHT	8 - LOWER CENTER
4 - CENTER LEFT	9 - LOWER RIGHT
5 - CENTER CENTER	

TEXT - A JUSTIFICATION DESCRIPTION USING ONE OR MORE OF THE KEYWORDS:

TOP	UPPER
MIDDLE	CENTER
BOTTOM	LOWER

FOR EXAMPLE: TOPCENTER
RIGHT/LOWER
CENTER-LEFT

? @ CHANGE THE TITLE BOX JUSTIFICATION AND POSITION.

? JUST 8 X 5 Y 1

? @ CHANGE THE TEXT

? TEXT F.LC IGURE: .UC PDEG .LC VS .ELC T

? @ CHECK ATTRIBUTES.

? SHO

TITLE GROUP: DEFAULT OPEN ACTIVE

TEXT : <F.LC IGURE: .UC T .LC VS .ELC PDEG>

FONT: 1
HEIGHT: .2 WIDTH: .2 GAP: 0.

X: 5. Y: 1. JUSTIFICATION: LOWER-CENTER

? @ CHANGE SYMBOL NUMBER AND LINE PATTERN FOR DATA SET 'C1'

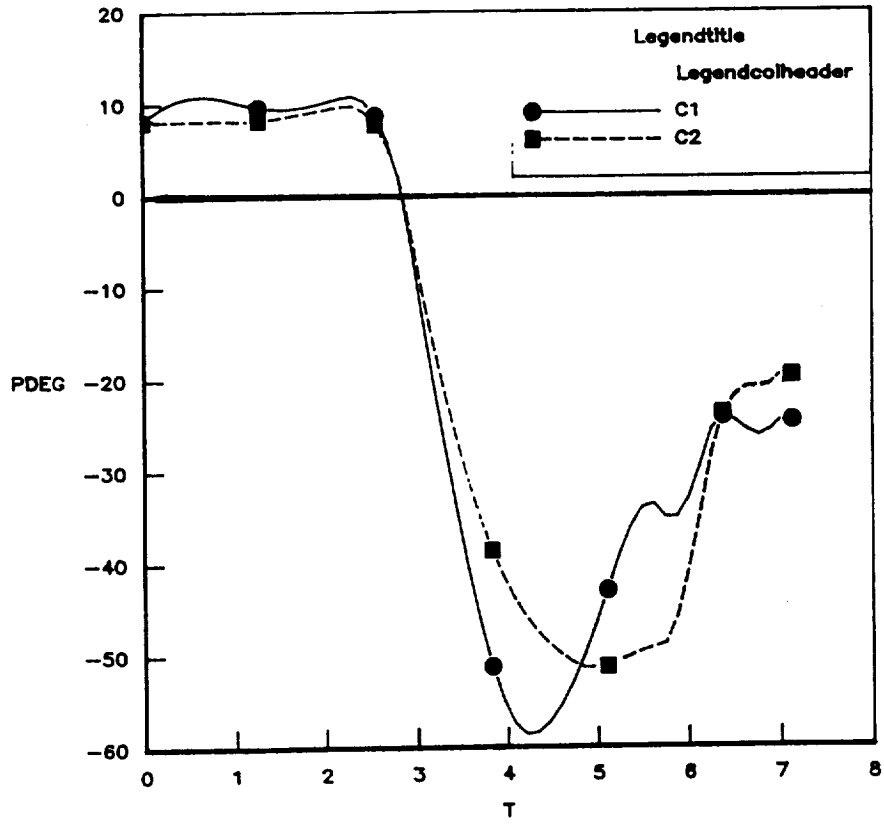
? DATA OPEN C1 SYM 901 EVERY 10 FIRST LAST

? @ CHANGE SYMBOL NUMBER AND LINE PATTERN FOR DATA SET 'C2'

? DATA OPEN C2 SYM 902 EVERY 10 FIRST LAST

? @ REPLOT CURVES WITH NEW ATTRIBUTES.

? PLOT



? STOP

Figure: PDEG vs T

Sample session 3 illustrates a complex plot using PHRASES and multiple data sets.

TADPLOT

PRODUCTION VERSION 2.00

OCTOBER 1991

```
? @-----------------------------------------------------------------
? @ ENTER THE PHRASE SUBSECTION, AND CREATE SEVERAL PHRASES
?      CRE CSUBD .LC C.SUB D
?      CRE CSUBN .LC C.SUB N.ESUB =0.60
?      CRE PWR X10.SUP 6.ESUP
?      CRE TFXED .LC .FONT 1 FIXED
? @ -------------------------------------------------------------------
? @ ENTER THE TITLE SUBSECTION, AND SET X, Y, HEIGHT, AND TEXT
? TITLE  JUST 2 X 5. Y 1 HEI .25 TEXT F.LC IGURE (A) .CSUBN
? @ -------------------------------------------------------------------
? @ ENTER THE LEGEND SUBSECTION, AND SUPPRESS THE LEGEND TITLE
 AND FRAME
? LEGEND NOLTITLE NOFRAME X 8 Y 9 JUST 3
? @ SET THE LEGEND COLUMN HEADER USING MNEMONICS
? LCOL .UC .FONT 9 JJ.FONT 1 R.LC E ' ' T.LC RANSITION
? @ -------------------------------------------------------------------
? @ ENTER THE DATA SUBSECTION, CREATE A DATA GROUP, AND SET
    ATTRIBUTES
? DAT CRE D1 FILE J45CDM8 X MINF Y FCD FIT LINEAR SEL INDEX1 1
?    SYM 1 LINE 1 FORM TOAD
?    XMAX .8 XMIN .5 YMAX .020 YMIN .006
?    KEY ' 4.4'.PWR ' ' .TFREE
? DAT CRE D2 USE D1 SYM 2 LINE 1 SEL INDEX1 2
?    KEY ' 7.7'.PWR ' ' .TFREE
? DAT CRE D3 USE D1 SYM 3 LINE 1 SEL INDEX1 3
?    KEY 14.0.PWR ' ' .TFREE
? DAT CRE D4 USE D1 SYM 4 LINE 1 SEL INDEX1 4
?    KEY 22.0.PWR ' ' .TFREE
? DAT CRE D5 USE D1 SYM 5 LINE 1 SEL INDEX1 5
?    KEY 30.0.PWR ' ' .TFREE
? DAT CRE D6 USE D1 SYM 6 LINE 1 SEL INDEX1 6
?    KEY 42.0.PWR ' ' .TFREE
? DAT CRE D7 USE D1 SYM 901 LINE 1 SEL INDEX1 7
```

```
?    KEY ' '4.4.PWR ' ' .TFXED
? DAT CRE D8 USE D1 SYM 902 LINE 1 SEL INDEX1 8
?    KEY ' '7.7.PWR ' ' .TFXED
? DAT CRE D9 USE D1 SYM 903 LINE 1 SEL INDEX1 9
?    KEY 14.0.PWR ' ' .TFXED
?@ ------------------------------------------------------------------
?@ ENTER THE XAXIS SUBSECTION, AND SET XAXIS ATTRIBUTES
? XAX MAX .8 MIN .5 LEN 6. RDI 1 GRID LAB .MINF
?    MAJOR 3 MINOR 5 HEIGHT .2 WIDTH .2 LHEIGHT .2 LWIDTH .2
?    PRE 4 LPRE 4 ZERO
?@ ------------------------------------------------------------------
?@ ENTER THE YAXIS SUBSECTION, AND SET ATTRIBUTES
? YAX MAX .020 MIN .006 LEN 3.5 RDI 3 GRID LAB .CSUBD
?    MAJOR 7 MINOR 2 HEIGHT .2 WIDTH .2 LHEIGHT .2 LWIDTH .2
?    PRE 4 LPRE 4 ZERO
?@ ------------------------------------------------------------------
?@ NOW GENERATE A PLOT WITH THE CURRENT DATA SET, AND ACTIVE
   SUBSECTIONS
? PLOT
```

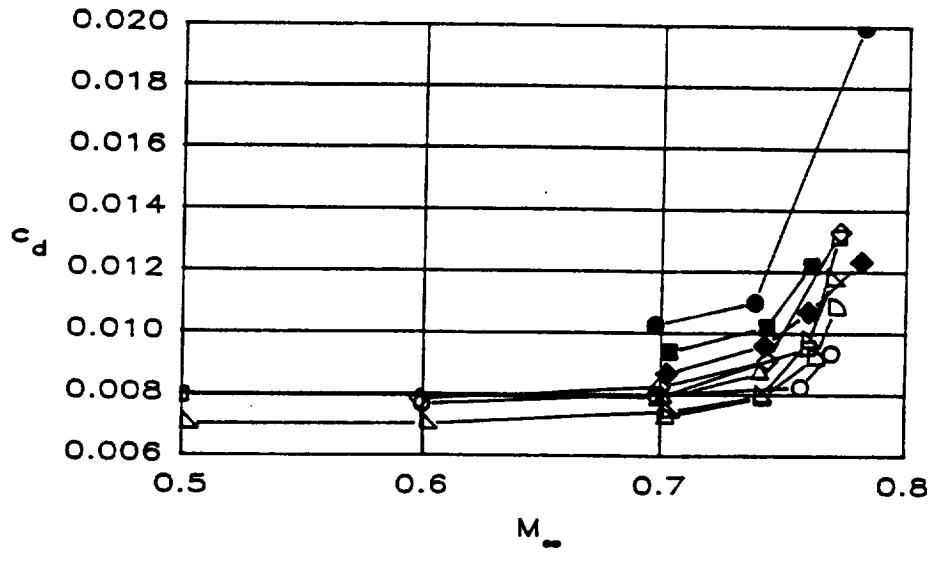


Figure (a) c_n=0.60

```
? STOP
```

This page intentionally left blank.

Appendix F

Postprocessing (Plotting Devices)

This section describes how to postprocess plots generated by the TADPLOT Program, what graphical capabilities are available, and what information is necessary in order to interface with the postprocessors.

Most large-scale, general-purpose graphics packages provide the ability to generate and manipulate graphical information in an external file. The user can generate interactive plots, and save the equivalent graphical representation on a file for later use. However, the ability to postprocess depends on a translator to convert the external file information into appropriate instructions to drive the physical device. The translator can take the same external file and plot (or display) it on several different display devices.

TADPLOLT Program uses DI-3000 as its underlying graphics package, and generates external files (called metafiles) which can be postprocessed using the Metafile Translator. The same DI-3000 metafile can be displayed on several different devices. The availability of the devices is installation dependent, while the device limitations (i.e. color, selective erase, etc.) are device dependent. Information on the availability of device drivers (graphics filters) is available through on-line help (i.e., man mftran) and through manuals describing DI-3000 on your system. Information specific to a particular device driver is only available in the device driver guides provided to your facility with the driver's installation.

The TADPLOT Program

Using the SAVE command within the TADPLOT Program will generate a metafile and write the graphical external file to the default file called "metafl.dat". In order to postprocess this metafile, the user must know the frame dimensions. This information is needed to ensure compatible aspect ratios between the world coordinate system and the devices coordinate system (i.e., a plot generated in an 11-by-11 inch coordinate system should be plotted in an area of equal aspect ratio to avoid distortion).

Currently, the Plot Program establishes a default world coordinate system of 11-by-11. This implies the aspect ratio is 1. The world coordinate boundaries, and therefore the aspect ratio, can be changed with the TADPLOT EXECUTIVE commands PHEIGHT and PWIDTH.

DI-3000 Environment Variables

The Metafile Translator is controlled by several variables set in the operating system (these are called "environment variables"). The variable PVI_DEV_number associates a specific graphical device driver with graphical output generated by TADPLOT. For example, the following statement would associate interactive graphics output with a Tektronix 4109:

```
PVI_DEV_1        409
```

The variable PVI_OUT_number associates the file name with metafile output. For example, the following statement would save the metafile in file "posts".

```
PVI_OUT_1          posts
```

The variable PVI_CFG_number associates a "configuration file" with the graphical output. A configuration file controls aspects of the device driver when plotting graphics output. The following properties are examples of device driver control: page orientation, page margins, color mapping, output resolution, number of copies, etc. The configuration files are of a specific format, and each device driver has a default file associated with it. Consult the DI-3000 installation notes for more information on configuration files.

Interactive Postprocessing

The Metafile Translator can be invoked interactively to display and manipulate the metafile. The user is referred to the Metafile System User's Guide for a detailed description on using the Metafile Translator. This document is written by Precision Visuals Incorporated (PVI), and is available from OCO as Document G-6.

Note, since the Metafile Translator loads the device drivers, only those graphics devices which have a locally supported device driver can be used. For a list of available device drivers locally supported, call the Plot Program Administrator (or for Langley-NASA personnel, contact "User Support").

In the following example the Metafile Translator is used to convert already generated DI-3000 metafiles to PostScript files with the use of DI-3000 related "environment variables".

```
#!
# Set the DI-3000 environment variables.
setenv PVI_DEV_1 pst              # Set the interactive output device to PostScript
setenv PVI_OUT_1 posts            # Set the interactive output file name to "posts"
setenv PVI_CFG_1 drvpst.cfg       # Read the configuration file to "drvpst.cfg"

foreach f ($*)                    # For each metafile entered
 cp $f\_meta DIMETA               # Copy the metafile to "DIMETA"
 mftran < mf_cmds >& mftranout_$f # Read metafile commands from the file "mf_cmds"
                                  # and output verbal information to "mftranout_$f"
 mv posts $f\_ps                  # Move PostScript output to a file name with _ps
                                  # appended
 rm DIMETA                        # Remove the intermediate file "DIMETA"
end                               # end of loop for each file
```

Plot Postprocessing

To postprocess the metafile using the ACD Production Graphics Output Devices, the user must generate a metafile, and then execute the Metafile Translator and selected device.

The metafile translator and device driver is loaded and executed with a "graphics filter" control statement. Since each driver provides a variety of capabilities controlled through command options, the user must provide the appropriate options although defaults are available.

Since the availability of device drivers is installation dependent, and each device driver has it own inherent limitations, the user is referred to the ACD Production Plotters : Device Driver Guide Document. Some typical control statements include:

mfcal11	CALCOMP 11" drum plotter
mfcal34	CALCOMP 34" drum plotter
mfvera	VERSATEC thermal plotter (A size drawing)
mfver39	VERSATEC 39" color electrostatic plotter

The following example is one method to submit a metafile generated by the Plot Program on the CalComp 11" plotter:

```
#
# Example of executing TADPLOT in the batch mode on Convex
#-----------------------------------------------------------------
# # This section sets Convex control options.
#
# @$-lt "1:00"
# # Set the maximum per-process CPU time limit to 1 minute.
#
# @$-mb -me
# # Send mail at beginning and end of the request execution.
#
# @$-l
# # Ensure the batch mode is LOGIN.
#
# @$-y
# # Append accounting information at the end of output.
#
# @$ # No more embedded flags.
#-----------------------------------------------------------------
cd /scr/temparea
# Change directory to a temporary area
cp ~ntflib/TAU4A /scr/temparea
# Copy the data file into the temporary area
~ntflib/tadplot < ~ntflib/SESIN4
# Invoke the tadplot program and read the commands from SESIN4
mfcal11 -i ~ntflib/cont -h11.0 -w11.0 metafl.dat | lpr -Pcal11 -C "delivery_information"
# Send metafile to plotting device (i.e., CalComp 11" plotter)
rm TAU4A metafl.dat
# Remove file TAU4A and metafl.dat. Leave LOG for verification.
```

Creating Composite Plots

The metafiles generated through TADPLOT consist of one or more frames each containing a single plot. If multiple plots are to be combined onto a single frame (composite plot) for display purposes, then the Metafile Translator can be used. The user is referred to the Metafile Translator System User's Guide for a complete description of how to manipulate metafiles.

The following session shows an example of how to combine the use of the TADPLOT POSITION command, and the abilities of the Metafile Translator. Each plot is positioned such that they can be combined on a single display area without overlapping. In this case, the positions and lengths of the axes are constructed to fill a 9 inch page, stacking vertically the three plots (see the axis directives POSITION and LENGTH).

```
cp ~ntflib/SESIN5 .          (* copy SESIN5 into the current directory *)
cp ~ntflib/SES5INP .         (* copy SESINP5 into the current directory *)
~ntflib/tadplot              (* Invoke TADPLOT *)
```

TADPLOT

PRODUCTION VERSION 2.00
OCTOBER 1991

```
@
@ SET TITLE
@
TITLE CRE T1
TEXT F.LC IGURE: .UC E.LC FFECT OF .UC LEVF
@
@ SET LEGEND
@
LEGEND CREATE L1 X 5.5 Y 8.5 NOLTI
LCO .FONT 9 .LC D .SUB .UC .FONT 1 TEF, .ESUB .LC .FONT 1 DEG
@
@ SET UP AN X-AXIS FOR ALL CHARTS
@
XAXIS MIN -.2 MAX 1.2 MAJ 7 NOFRAME NOAUTO LAB C.SUB L
@
@ SET UP FIRST DATA SET AND YAXIS
@
DATA CRE C1 FILE SES5INP X CL Y ADEG LINE 1 SYM 2 KEY -30
YAXIS POS 2 LEN 1.5 MIN -10 MAX 30 MAJ 4 NOMINOR NOFRAME
LAB .FONT 9 .LC A, .FONT 1 DEG
```

```
SAVE
@
@ REMOVE SUBSEQUENT XAXIS LABEL
@
XAXIS NOLAB
@
@ TURN OFF THE TITLE FOR SUBSEQUENT PLOTS
@
TITLE TURN OFF T1
CLOSE
@
@ TURN OFF THE LEGEND FOR SUBSEQUENT PLOTS
@
LEGEND TURN OFF L1
CLOSE
@
@ SET UP 2ND DATA SET AND YAXIS
@
DATA Y CM
YAX POS 4. NOMINOR AUTO LAB C.SUB .LC M
SAVE
@
@ SET UP 3RD DATA SET AND YAXIS
@
DATA Y CD
YAX POS 6 LAB C.SUB D
SAVE
@
@ STOP
@
STOP
```

Next, the Metafile Translator is invoked as before, and the three frames plotted onto a single display device.

```
mftran
S MF 1 metafl.dat
D P 1 P 2 P 3     (* DRAW PICTURES 1, 2, AND 3 *)
QUIT
```

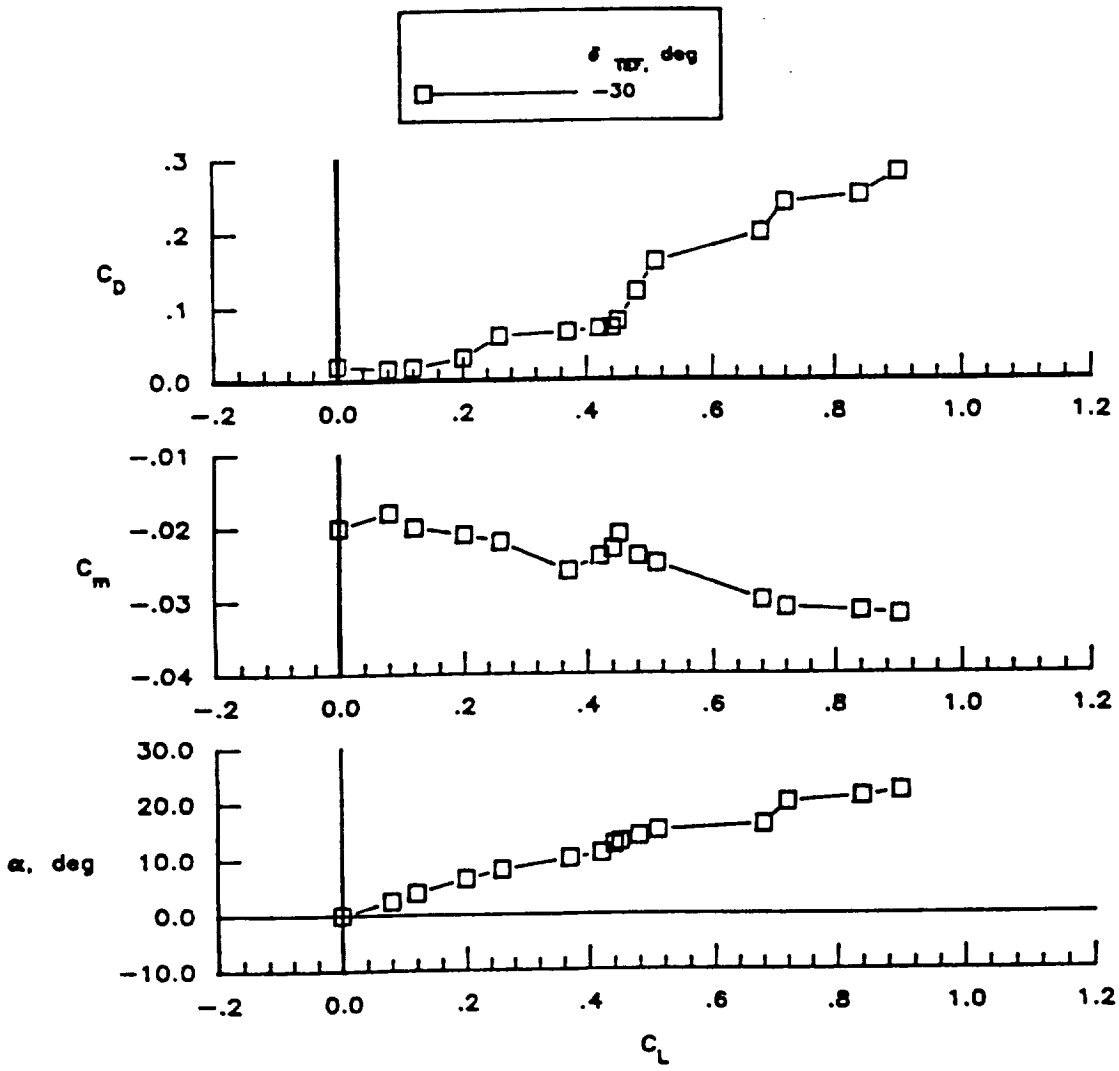


Figure: Effect of LEVF

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY(Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 1991	Contractor Report

4. TITLE AND SUBTITLE
TADPLOT Program - Version 2.0
User's Guide

5. FUNDING NUMBERS
C NAS1-19038
WU 505-59-53-05

6. AUTHOR(S)
Dana P. Hammond

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computer Sciences Corporation
3217 N. Armistead Avenue
Hampton, VA 23666-1379

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Adminsitration
Langley Research Center
Hampton, VA 23665-5225

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-189553

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Dr. James R. Schiess (COTR)
Final Report Dr. Neal T. Frink

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited

Subject Category 61

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This document describes the TADPLOT Program, Version 2.0. The TADPLOT program is a software package coordinated by a single, easy-to-use interface, enabling the researcher to access several standard file formats, selectively collect specific subsets of data, and create full-featured publication and viewgraph quality plots.

The user-interface was designed to be independent from any file format, yet provide capabilities to accommodate highly specialized data queries. Integrated with an applications software network, data can be accessed, collected, and viewed quickly and easily.

Since the commands are data independent, subsequent modifications to the file format will be transparent, while additional file formats can be integrated with minimal impact on the user-interface. The graphical capabilities are independent from the method of data collection, thus the data specification and subsequent plotting can be modified and upgraded as separate functional components. The graphics kernel selected adheres to the full functional specifications of the CORE standard. Both interface and postprocessing capabilities are fully integrated into TADPLOT.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Computer Graphics, Computer Programs	130
	16. PRICE CODE A07

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		