UNLIMITED DISTRIBUTION

(1)

**National Defence** Research and Development Branch

**Défense nationale** Bureau de recherche et développement

TECHNICAL COMMUNICATION 89/309

September 1989

AD-A232 524

# SAPLOT
# SCIENTIFIC GRAPHIC SOFTWARE HISTORY AND OVERVIEW

Philip R. Staal



**Defence Research Establishment Atlantic**

DEFENSIO PER SCIENTIAM

**Centre de Recherches pour la Défense Atlantique**

Canada



91 3 04 095

**DEFENCE RESEARCH ESTABLISHMENT ATLANTIC**

9 GROVE STREET

P.O. BOX 1012
DARTMOUTH, N.S.
B2Y 3Z7

TELEPHONE
(902) 426-3100

**CENTRE DE RECHERCHES POUR LA DÉFENSE ATLANTIQUE**

9 GROVE STREET

C.P. 1012
DARTMOUTH, N.É.
B2Y 3Z7

UNLIMITED DISTRIBUTION

National Defence
Research and Development Branch

Défense nationale
Bureau de recherche et développement

# SAPLOT
## SCIENTIFIC GRAPHIC SOFTWARE
## HISTORY AND OVERVIEW

Philip R. Staal

September 1989

Approved by H.M. Merklinger
H/Surveillance Acoustics Section

Distribution Approved by

A/D/UAD

TECHNICAL COMMUNICATION 89/309

Defence Research Establishment Atlantic

Centre de Recherches pour la Défense Atlantique

Canada

## ABSTRACT

The SAPLOT software system provides a simple means by which scientists or engineers can produce graphical output. SAPLOT is designed to meet the standards of the American Institute of Physics, and to produce graphic output suitable for publications, slides, or overhead transparencies. Careful attention is given to producing high quality axis markings. The SAPLOT software system has defined a standard input data file. The SAPLOT software reads data from the input file, and produces graphical output representing those input data. Finally, SAPLOT is still evolving. It is becoming more portable as new versions are written for different machines, and as SAPLOT to PostScript file conversion software becomes available. It is likely that SAPLOT will continue to fill a need for scientific graphics for years to come. This document provides a brief history of SAPLOT and a brief description of its features. Reference is made to other documents where more detail may be found.



## RÉSUMÉ

Le logiciel SAPLOT constitue un moyen simple pour les scientifiques et les ingénieurs de produire des graphiques. SAPLOT, qui satisfait aux normes de American Institute of Physics, permet de produire des graphiques adaptés à la publication ainsi qu'à la présentation de diapositives ou de transparents pour rétroprojection. Il permet entre autres de produire des repères axiaux de grande qualité. Le système logiciel SAPLOT a défínu un fichier de données d'entrée standard. Le logiciel SAPLOT lit les données d'entrée de ce fichier, et produit un graphique qui représente ces données. Enfin, SAPLOT est un logiciel évolutif. Sa transférabilité s'améliore constamment au fur et à mesure que de nouvelles versions sont rédigées pour différentes machines; en outre, il existe maintenant un logiciel de conversion de fichiers SAPLOT/PostScript. Selon toute probabilité, SAPLOT va demeurer la norme en matière de logicial graphique pour les scientifiques pour encore de nombreuses années. Ce document présente un bref historique de SAPLOT ainsi qu'un résumé de ses caractéristiques. On y mentionne d'autres documents qu'on peut consulter pour de plus amples détails.

# TABLE OF CONTENTS

# TABLE OF CONTENTS (cont'd)

# LIST OF FIGURES

**Figure** **Page**

# 1. INTRODUCTION

While conducting research, scientists and engineers often produce results best presented in graphic form. Much of this graphical information is produced by computers. In the past, standard plotting packages have been used, but these seldom produce results in a form acceptable for publication. Thus, re-plotting, often done manually, was needed. Computer software capable of giving the graphical output in publication form would be very useful. SAPLOT (Surveillance Acoustics PLOTting) is such a software package.

The SAPLOT scientific graphic software, which was developed in the Surveillance Acoustics Section at Defence Research Establishment Atlantic (DREA), has had extensive use but has not been described in a report until now. For those who use and maintain SAPLOT, this document describes the SAPLOT software and philosophy. This document is not intended to be a detailed user manual *per se*, but rather an overview of the general function of the software; hence, not all commands or options of SAPLOT are described. The history of SAPLOT and the reasons for its development are described first. Then, the input data file format is described, followed finally by a description of the graphic output. For more detailed information, see the "User's Manual" **[Young, 1989].**

# 2. HISTORY OF SAPLOT

## 2.1. PROBLEMS TO SOLVE

### 2.1.1. Easy Data Plotting

One reason for the development of the SAPLOT plotting software was the need to plot small amounts of data from hastily written programs or from hand-prepared data records. Most plotting software available at the time SAPLOT was developed took more effort to use than simply plotting the data by hand on graph paper. Often, simple programs were written and linked to a graphics library for each data set to be plotted. Preparing all these programs was a time consuming duplication of effort.

The solution that enabled easy data plotting was a standard ASCII (American Standard Code for Information Interchange) input data file format and a program (SAPLOT) that could interpret this format and produce plots suited to the input data. This solution involved no programming by the user, only the writing of standard ASCII files from a program and/or with a text editor.

Software similar in some ways to the above solution was available at the time SAPLOT was developed, but was too inflexible, had poor output quality, and/or required too many commands for a simple plot. Significant effort was put into SAPLOT in order to make it do the "right" thing by *default*, instead of requiring a multitude of commands. Only if the users wanted to do something unusual with their plots would complex commands be necessary.

### 2.1.2. Document Preparation

Another need that led to the development of the SAPLOT plotting software, was the need to produce high quality figures for publications and presentations. At the time SAPLOT was developed, software could not be found that would automatically produce figures adhering to the required standards. Even at the time of this writing, there are few software packages that produce graphic output that meets acceptable graphic standards (e.g. [**Tufte, 1983**]), let alone the standards required by journals.

SAPLOT was designed from conception to meet the standards of the American Institute of Physics [**AIP Style Manual, 1978**], and to produce graphic output suitable for publications, slides, or overhead transparencies, without the need for any manual retouching.

## 2.2. LINE-DRAWING AXIS LABELLING

Easily understood axis markings are not hard to produce, but many graphics packages fail to give clear markings. Often, the axis markings appear to have been chosen for the ease of the computer programmer, rather than to satisfy presentation standards. Because of the importance that we attach to clear markings, axis marking algorithms were the first part of SAPLOT to be designed. The rest of this subsection describes the methods used by SAPLOT to produce axis markings, and subsection 4.3 gives further details.

### 2.2.1. Axis Ticks

Ticks and sub-ticks denoting numerical values along axes are chosen with a reasonable density (enough markings to make the numerical scale clear but not so many markings that they obscure the data or become unreadable), with sizes that are as resolvable as the other graphical elements (symbols, letters, etc.), and with a scale that is naturally easy to interpret. (This is the antithesis of an "easy programming" technique such as setting the axis limits then dividing by ten to define eleven numbered ticks; see subsection 4.3 for the simple scales used in SAPLOT).

A choice of axis marking density is provided with SAPLOT. Dense markings are generally used for graphs from which individual measurements may be extracted. Sparse markings are often used for papers or presentations where general trends in the data are being illustrated, and a multitude of axis markings would only obscure the trend.

### 2.2.2. Axis Numbers

Easily interpreted numbers are essential for labelling axis ticks, so careful attention is given to numeric formatting in SAPLOT. The simplest-possible number formats are used. Decimal points and plus signs are omitted where possible. Exponential notation is only used as a last resort. The always confusing "unit ($*10^x$)" type of axis label is *never* used.

## 2.3. USER CALLABLE SUBROUTINES

The initial part of the SAPLOT software that was written was a set of FORTRAN subroutines that could be easily linked into a user's program. These subroutines followed the philosophy that default parameters would normally be sufficient, so that the users ordinarily only needed to make two subroutine calls in order to plot their data. Further subroutine calls were only necessary for changing the default parameters.

## 2.4. ASCII-FILE READING PROGRAM

It was very quickly decided that a stand-alone program, that could read an ASCII file containing the data, was desirable. This program simply added a file reading and interpreting framework on top of the user callable subroutines that had been developed previously. This first version of SAPLOT was written in 1983 for a PDP 11/34 computer, at sea, in the fog, on the Grand Banks of Newfoundland.

A stand-alone plotting program is usually preferable to a subroutine library for several reasons. The time to compile, link, and debug a plotting program is eliminated. Also, users often have data that are either input by hand into an ASCII file, or are already in an ASCII file format suitable for input to a stand-alone plotting program. It has generally been found at DREA that a clear separation between the data file generation procedure and the well-established SAPLOT plotting software has eliminated a great deal of the effort required in adding graphic capabilities to software.

## 2.5. VERSIONS FOR A VARIETY OF COMPUTERS

As interest in using the SAPLOT software grew, and as computers began to be linked in networks, versions of SAPLOT for other computers were developed. This improved the accessibility to quality graphic plotting, eliminated some bottlenecks, but continually requires vigilance in keeping a standard set of commands in all implementations of SAPLOT. In order to help check new versions of SAPLOT, there is a set of standard test input files that is normally kept with the on-line documentation for SAPLOT.

### 2.5.1. PDP-11/34 With RT-11

The first version of SAPLOT was developed in 1983 for the Digital Equipment Corporation PDP-11/34 computer running the RT-11 operating system, and was written in a FORTRAN-66 superset. Two variants of this version of SAPLOT were produced. One variant used a VERSATEC VERSAPLOT electrostatic plotter, and the other variant produced TEKTRONIX 4010 graphic output. The on-line documentation for this version is in a file called "PLTFIL.DOC".

This first version of SAPLOT (VERSATEC variant) solved some of the problems described in Section 2.1, but did not provide high-resolution plotting (VERSATEC hardcopy was only 200 dots/inch) and was available on only a few single-user computers.

### 2.5.2. DECSYSTEM-20 With TOPS-20

The second version of SAPLOT was produced in 1983 for DREA's mainframe computer, a DECSYSTEM-20 running the TOPS-20 operating system, and was written in a FORTRAN-77 superset. This version solved all the problems described in Section 2.1. The high quality graphic output was obtained from pen plotters and laser printers driven by the TEKTRONIX Interactive Graphics Language (IGL). The on-line documentation for this version is in a file called "SAPLTF.DOC". The final modifications to this currently used version were completed in early 1985.

### 2.5.3. VAX With VMS

The third version of SAPLOT was produced in 1984 for the VAX computer running the VMS operating system, and was written in a FORTRAN-77 superset (slightly different from the DECSYSTEM-20 version). The on-line documentation for this version is in a file called "SAPLOT.DOC". Some features were added to this version, and it became the most frequently used version in about 1986.

### 2.5.4. Macintosh

The fourth version of SAPLOT was produced for the Macintosh computer in 1985, and was written in a FORTRAN-77 superset (slightly different from the DECSYSTEM-20 and VAX-VMS versions). This version of SAPLOT was then supplanted in 1988 by a complete re-write of the SAPLOT code in the "C" language. The on-line documentation for this version is in a file called "MacSaplot doc".

Some special features have been added to the Macintosh version of SAPLOT, because of the Macintosh computer's inherent graphics capabilities. Originally, there was little impetus to add bar graphs to SAPLOT, since there are many "business graphics" packages that will produce bar graphs, pie charts, etc. There are also pitfalls to be avoided in producing such "business graphics" (see **[Tufte, 1983]**, particularly the chapters on "graphical integrity" and "chartjunk"). However, for completeness, bar graph options were added to the Macintosh version of SAPLOT.

## 2.6. ADDITION OF PATTERN-FILL AND GRAY SCALE

In most cases, two dimensional graphic plots were sufficient. However, there were a growing number of cases where a third dimension was desired. Certain types of data were amenable to a "waterfall" type of plot (e.g. multiple spectral plots), for which the standard two-dimensional SAPLOT would suffice. On the other hand, there were cases that required a better three dimensional capability. Two types of three dimensional graphics were developed and are discussed in the following two sections.

### 2.6.1. Pattern Fill

For cases in which the data density is fairly low (e.g. up to ≈ 40 x 40 cells in the plot), a pattern-fill capability was added to the DECSYSTEM-20 version of SAPLOT. This capability has been used for representing data such as that shown in geographic maps, in which a particular area can be identified by its pattern (or colour). Some caution had to be taken in choosing patterns, since poorly designed patterns can produce moiré patterns that can be quite disconcerting (see **[Tufte, 1983]**, particularly the chapter on "chartjunk").

### 2.6.2. Gray Scale

For cases in which the data density is fairly high (e.g. higher than ≈ 40 x 40 cells in the plot), a gray-scale capability was added to SAPLOT. This capability has been used for representing data such as that shown in sonograms (plots of spectrum level vs time and frequency). Typically only about eight levels of gray can be identified and related to a legend, whereas many more patterns (or colours) can be uniquely identified. Also, patterns will likely be more easily photocopied than gray levels. However, the order of the gray intensity scale is intuitively obvious, unlike the order of patterns or some colour scales.

## 2.7. DATA MANIPULATION

The SAPLOT software has been designed with the primary aim of producing graphic output from data, not manipulating or processing that data. However, experience using SAPLOT has shown that incorporating a limited set of data manipulation commands into SAPLOT is beneficial. The two most used data manipulation commands are described in the following two paragraphs.

The first command, "ADJUST", has the effect of scaling and/or moving a curve. Both the $x$ and $y$ values of all data points in a curve are multiplied by a constant and then added to a constant before the curve is plotted.

The second command is "LOG10". The base-ten logarithm is taken of the $x$ and/or $y$ values of all data points in a curve before the curve is plotted.

## 2.8. POSTSCRIPT OUTPUT

The latest development of SAPLOT, has been its conversion to a nearly computer-independent program. This program (called SAPLOT_PS) is simply a file-conversion utility. It changes a SAPLOT ASCII text input file to a PostScript output file. SAPLOT_PS is written in a standard form of the "C" language, and has been tested successfully on Apple Macintosh, ATARI ST, IBM PC, DEC VAX, Silicon Graphics, MASSCOMP, and other computers.

# 3. SAPLOT INPUT DATA FILE FORMAT

The SAPLOT input data file is an ASCII text file that appears (while using an editor for the computer system in use) as shown in the "Input File" box in the examples that follow. This type of file is easy to inspect, edit, and understand. Disadvantages of an ASCII input file include a doubling (or so) of the storage space required, and an increased computer access time to read the data. The advantages have been found to highly outweigh the disadvantages.

The format described in this section has become a *de facto* standard for the storage of small sets of processed data at DREA, because of the heavy use of SAPLOT. Much larger quantities of digital data are normally stored in another DREA standard format **[Caldwell, 1987]**.

In the figures that follow, the output of SAPLOT is shown under the "Graphic Output" heading. Most of these output examples were produced with the program MacSAPLOT on a Macintosh computer, and transferred to Microsoft Word which was used to produce this report. The plotting resolution in this document suffers slightly because of limitations due to the process used to insert graphics into Microsoft Word documents.

## 3.1. DATA STORAGE FORMAT IN THE SAPLOT INPUT FILE

Two-dimensional data to be plotted are stored in the SAPLOT ASCII text input file following a file line containing the character string "CURVE". The data are in two columns, with $x$ values in the left column and $y$ values in the right column. The values are interpreted as "free format" real numbers. The separator between the values can be any one of: one or more spaces or tabs; a comma, with or without surrounding spaces or tabs. Any characters to the right of the second column are ignored, and may be used for comments. In Fig. 1, one curve is drawn, with data points (0,0) and (1,1).

Multiple curves of data may be stored in the SAPLOT ASCII text input file, by placing them one after another. The number of data points in each curve is arbitrary, but there may be some limits due to memory size etc. which can be implementation dependent. A curve is considered to have ended if a file line is encountered whose contents cannot be interpreted as a data point (e.g. a file line containing "CURVE", some other command, or a blank line). An example showing two data curves is presented in Fig. 2.

Multiple plots may be combined in one SAPLOT ASCII text input file, by separating two normal single plots by a file line containing the command "PLOT". In conjunction with plot size and position commands, the "PLOT" command allows the production of multiple plots on one page. For example, the graphic outputs shown in Figs. 1 and 2 could have been produced by a single input file consisting of the contents of the two input files shown, separated by a file line containing "PLOT", along with two positioning commands (see Appendix A for command descriptions).

Input File

```
CURVE
0      0
1      1
```

Graphic Output

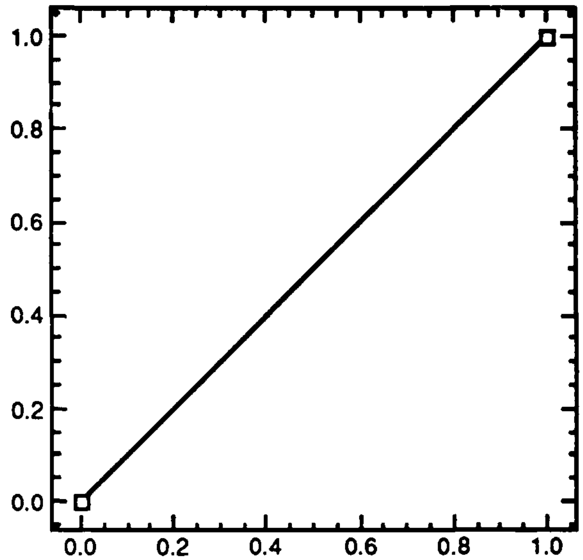



*Fig. 1. SAPLOT ASCII text input file and graphic output for a curve with two points.*

Input File

```
CURVE
0      0
1      1
CURVE
1.5    1.5
2.0    0
2.5    1.5
1.5    1.5
```

Graphic Output

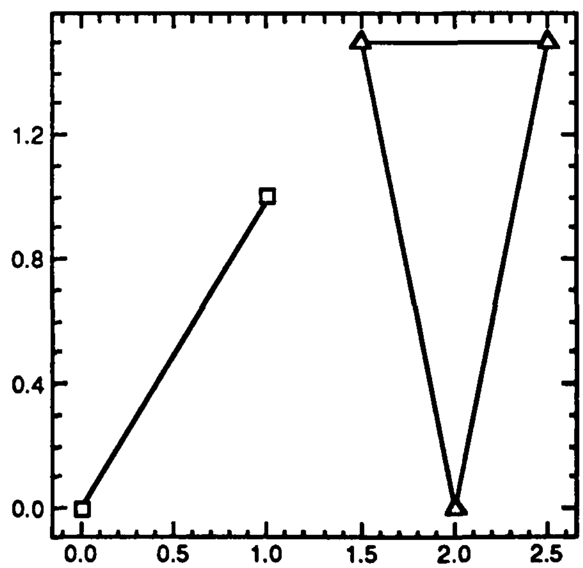



*Fig. 2. SAPLOT ASCII text input file and graphic output showing two curves.*

### 3.2. SAPLOT COMMANDS

If the SAPLOT ASCII text input file contains only the data points and "CURVE" commands (e.g. Fig.1), then the graphic output is produced using a set of default parameters for such things as tick size, axis range and line type. In order to change from this default format, additional commands must be given in the SAPLOT ASCII text input file. All of these commands start in the first column with alphabetic characters, and are placed on lines by themselves. These command lines may be placed anywhere in the ASCII file for the current generation of SAPLOT programs, but are usually placed before the curve data. The commands may be UPPER-CASE or lower-case for current SAPLOT programs, but are normally UPPER-CASE. Any characters to the right of a complete command are ignored, and may be used for comments. The default parameters for the commands are described with the individual SAPLOT commands in Section 4.

The number of different commands has been kept to a minimum by avoiding unnecessary or redundant commands. The resulting simplicity of SAPLOT has made it more attractive for the average user. The commands have been developed over a period of more than six years. In order to maintain compatibility, new commands have been added, but old commands have not been changed. Consequently, some commands are slightly anachronistic.

## 4. SAPLOT GRAPHICAL OUTPUT

The end product of the SAPLOT software is graphical output. Examples of this output can be found in **[Staal, 1983]**, **[Walker and Ashley, 1984]**, **[Cotaras, Fraser, and Merklinger, 1988]**, **[Staal and Hughes, 1988]**, **[Chapman, 1989]**, **[Swingler and Walker, 1989]**.

### 4.1. STANDARDS USED FOR DEFAULT PLOTS

As mentioned in Section 2.1.2., SAPLOT was designed from conception to meet the standards of the American Institute of Physics **[AIP Style Manual, 1978]**. Local DREA standards for papers and overhead transparencies were also consulted in creating the SAPLOT output format.

In the most recent versions of SAPLOT, a choice of pre-defined formats has been made available. These include "viewgraph", "single-column", "double-column", and "thesis" formats. Other formats are being added as the need arises.

## 4.2. DATA PRESENTATION

### 4.2.1. Data-point Symbols and Lines

By default, if there are *less than 41 points* in all curves of a plot, then the points are represented by symbols joined by straight continuous lines. The symbols are the same for all points in a curve, and the symbol type is changed for each new curve. Such a curve is shown in the middle of Fig. 3.

By default, if there are *more than 40 points* in any curve of a plot, then the points are joined by straight lines. The line type is changed for each new curve. Such a curve is shown as the bottom curve of Fig. 3.

The defaults for the curve types are changed by the command "CSET", as shown in Fig. 3. The number following "CSET" is the curve number (where the first curve in the file is curve number 1, etc.), and the number following that defines the curve type. Curve type "0" consists of straight lines, and type "1" consists of symbols joined by straight continuous lines, as described above for the default cases. A third curve type ("2") consists of symbols marking the data points with no lines joining these points. Such a "curve" is shown as the top curve of Fig. 3. As shown in Fig. 4, it is also possible to

Input File

```
CSET 1 0
CSET 2 1
CSET 3 2
CURVE
0    0
1    1
3    0
CURVE
0    1
1    2
3    1
CURVE
0    2
1    3
3    2
```

Graphic Output

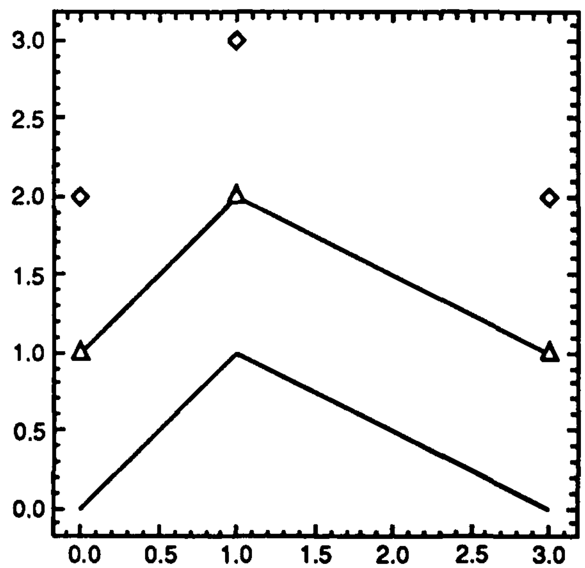



*Fig. 3. SAPLOT ASCII text input file and graphic output showing three curve types. The bottom curve consists of lines joining the data points. The middle curve consists of lines joining symbo!s at the data points. The top curve consists of symbols at the data points.*

produce error bars along with the data points. These bars are produced by using curve type "3", which consists of lines drawn between successive pairs of points in the curve (line from point 1 to point 2, line from point 3 to point 4, etc.), and special symbol number "256" (top and bottom horizontal lines of the error bars. Horizontal error bars are also possible, using special symbol number "257".

Unless "RANGE" commands are inserted in the input file, the axis ranges are chosen by SAPLOT so that the data-point symbols and lines are separated from the axis ticks by a small margin. By default, the data are thus expanded to nearly fill the plot frame.

### 4.2.2. Line Types

As mentioned in the previous section, the line type may be changed for each curve. The first 17 line types are normally as shown in Fig. 5. These line types have been chosen to be easily distinguishable and resolvable, while presenting as continuous a trace as possible. It is also possible to change the line width and/or colour using the commands "WSET" and "BSET" (see Appendix A for details).

To keep the example input file of Fig. 5 acceptably short, curves of only two data points have been used. These two-point curves have been represented by line types by using the "CSET" command. The special curve number of "-1 " is used to represent all curve numbers *in* a plot. The curves are placed in order from the bottom of Fig. 5 to the top, since by default the *y* axis increases upwards.

Input File

```
CSET 2 3
MSET 2 256
CURVE
1     1
2     2
3     3
CURVE
1     1.2
1     0.8
2     2.5
2     1.5
3     3.7
3     2.3
```

Graphic Output

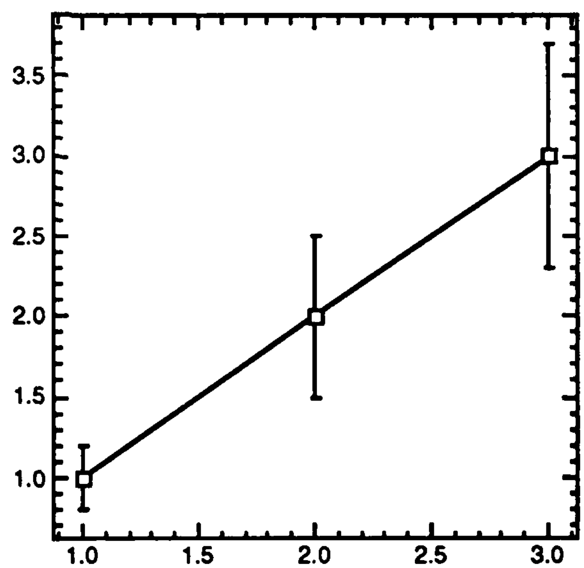



*Fig. 4. SAPLOT ASCII text input file and graphic output showing the error bar curve and symbol type.*

Input File

```
CSET -1 0
CURVE
0       0
1       0
CURVE
0       1
1       1
CURVE
0       2
1       2
CURVE
0       3
1       3
CURVE
0       4
1       4
CURVE
0       5
1       5
CURVE
0       6
1       6
CURVE
0       7
1       7
CURVE
0       8
1       8
CURVE
0       9
1       9
CURVE
0       10
1       10
CURVE
0       11
1       11
CURVE
0       12
1       12
CURVE
0       13
1       13
CURVE
0       14
1       14
CURVE
0       15
1       15
CURVE
0       16
1       16
```

Graphic Output

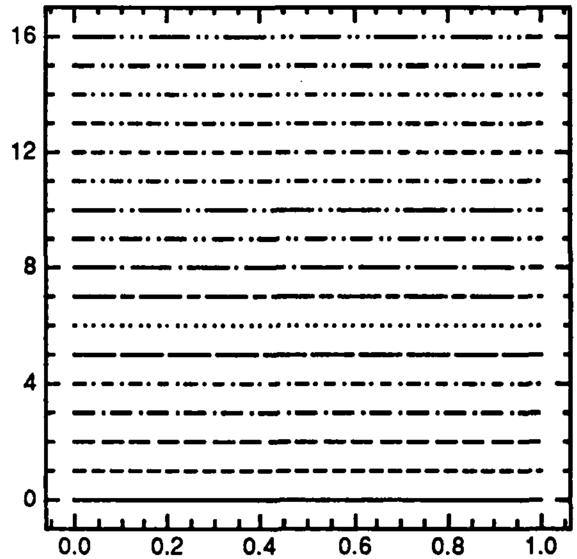



*Fig. 5. SAPLOT ASCII text input file and graphic output showing the default line types.*

### 4.2.3. Symbol Types

As mentioned in Section 4.2.1, the symbol type may be changed for each new curve. The first 16 symbol types are normally as shown in Fig. 6. These symbol types have been chosen to be easily distinguishable and resolvable.

<u>Input File</u>

```
CURVE
1     1
CURVE
2     2
CURVE
3     3
CURVE
4     4
CURVE
5     5
CURVE
6     6
CURVE
7     7
CURVE
8     8
CURVE
9     9
CURVE
10    10
CURVE
11    11
CURVE
12    12
CURVE
13    13
CURVE
14    14
CURVE
15    15
CURVE
16    16
```

<u>Graphic Output</u>

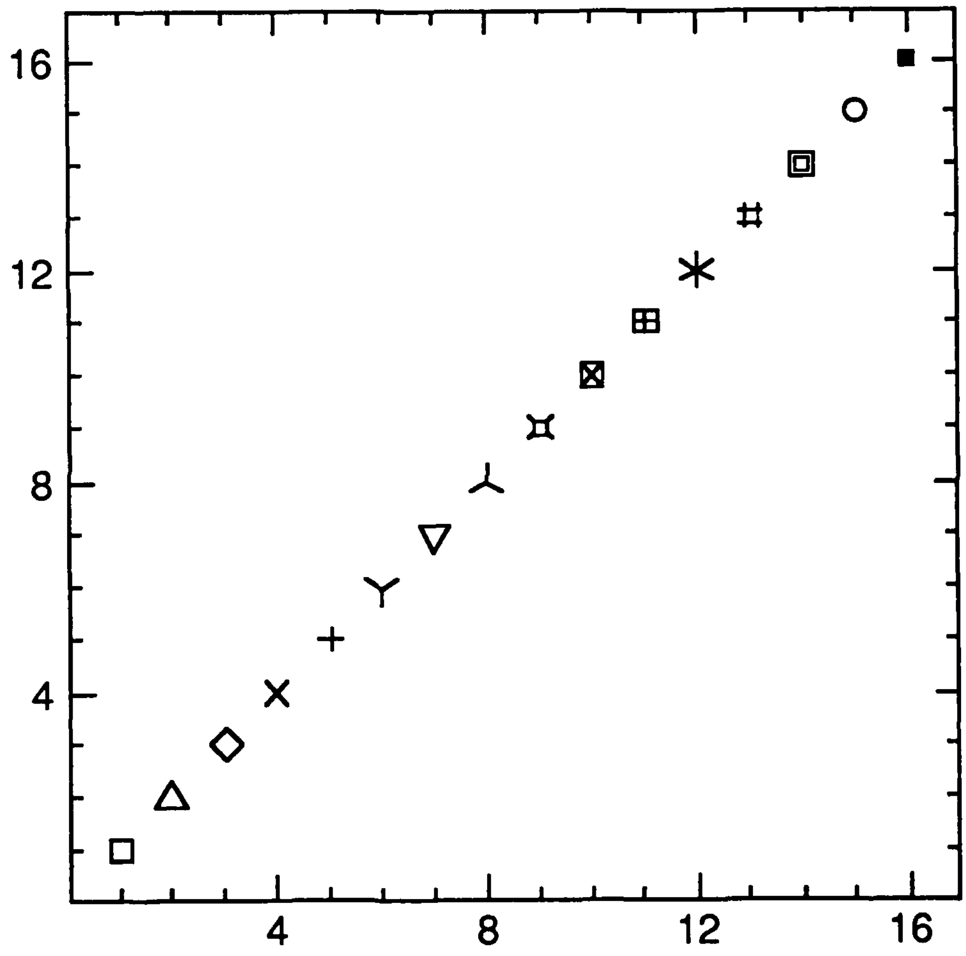



*Fig. 6. SAPLOT ASCII text input file and graphic output showing the default symbol types.*

### 4.2.4. Pattern Fill

As mentioned in section 2.6.1., fill patterns can be applied to a plot as shown in Fig. 7. The data representing the areas to be filled are placed after the command "PATPLT" which is on a file-line by itself. The command "PATPLT" can have following parameters (described in Appendix A) that may define the location and size of the pattern plot.

The area fill data are (in their simplest form) a direct map of the pattern type to a rectangular grid of ASCII characters. The normal set of ASCII characters used is: "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", ":", ";", "<", "=", ">", "?", and "@". These 17 characters have ASCII decimal equivalents that start at 48 and go monotonically to 64 (in the order shown). Such a form for the area fill data may be converted to a pattern plot on a Macintosh computer by changing the font type to a special pattern font ("Pattern8", available on "SERVER 1" at DREA, or from the author).

The area fill data characters may be in a compressed form. A "REPEAT" ... "END" construct may be used to repeat file lines. Also, any character in a line may be repeated up to 999 times by following the character to be repeated by the special character "*" then a three digit number. For example, "2*0103*010" is the same as writing "22222222223333333333". Long data segments may be continued on the next line by ending the line with the continuation character "-". Further details may be found in the SAPLOT user's manual.

### 4.2.5. Gray Scale

As mentioned in section 2.6.2., gray scale fill can be applied to a plot. An example is shown in Fig. 8. The data representing the gray levels are placed after the command "PTPLT" which is on a file-line by itself. The command "PTPLT" may have following parameters (described in Appendix A) that may define the location and size of the pattern plot.

The gray level data are (in their simplest form) a direct map of the gray level to a rectangular grid of ASCII characters. The normal set of ASCII characters used is: "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", ":", ";", "<", "=", ">", "?", and "@". These 17 characters have ASCII decimal equivalents that start at 48 and go monotonically to 64 (in the order shown). The gray levels change monotonically from white ("0") to black ("@"). Such a form for the gray level data may be converted to a gray scale plot on a Macintosh computer by changing the font type to a special gray scale font ("Gray4", available on "SERVER 1" at DREA, or from the author).

The gray level data characters may be in the same compressed form as described in the previous section for fill patterns.

Input File

```
CURVE
0 0
1 1
PATPLT 0 0
0000000000
0000000000
1111111111
1111111111
2222222222
2222222222
3333333333
3333333333
4444444444
4444444444
5555555555
5555555555
6666666666
6666666666
7777777777
7777777777
8888888888
8888888888
9999999999
9999999999
::::::::::
::::::::::
;;;;;;;;;;
;;;;;;;;;;
<<<<<<<<<<
<<<<<<<<<<
==========
==========
>>>>>>>>>>
>>>>>>>>>>
??????????
??????????
@@@@@@@@@@
@@@@@@@@@@
```

Graphic Output

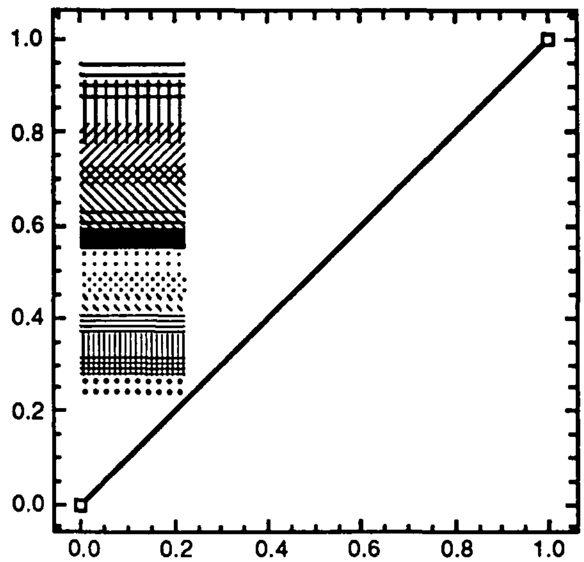



*Fig. 7. SAPLOT ASCII text input file and graphic output showing pattern fill data.*

Input File

```
CURVE
0 0
1 1
PTPLT 0 0
0000000000
0000000000
1111111111
1111111111
2222222222
2222222222
3333333333
3333333333
4444444444
4444444444
5555555555
5555555555
6666666666
6666666666
7777777777
7777777777
8888888888
8888888888
9999999999
9999999999
::::::::::
::::::::::
;;;;;;;;;;
;;;;;;;;;;
<<<<<<<<<<
<<<<<<<<<<
==========
==========
>>>>>>>>>>
>>>>>>>>>>
??????????
??????????
@@@@@@@@@@
@@@@@@@@@@
```

Graphic Output

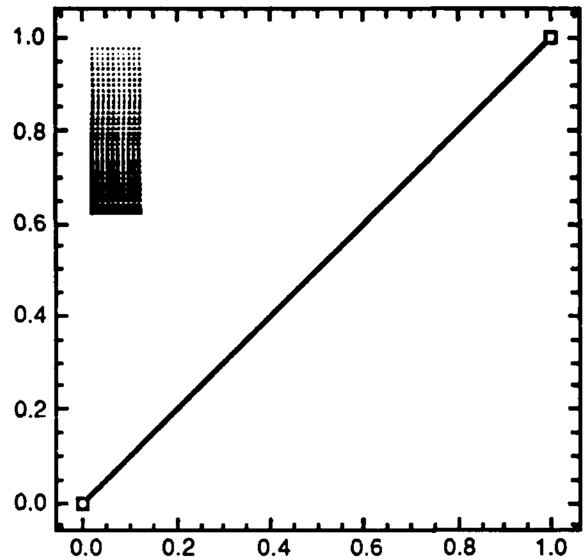



*Fig. 8. SAPLOT ASCII text input file and graphic output showing gray-scale data.*

## 4.2.6. Bar Graphs

As mentioned in section 2.5.3., a bar graph capability was added to SAPLOT. This capability involved a new curve type ("4") that produces bars, and a new axis type ("5") that provides for "categories" along an axis.

Input File

```
FORMAT 3
LABEL 1
Year
LABEL 2
Provincial Grant ($)
LABEL 4
Hours
RANGE 2 0 50000
RANGE 4 0 50000
LFSET 2 1
PSET 50 25 1 0
LEGEND 3
 Grant (actual $)
 Grant (constant $)
 Hours
ASET 1 5 Category axis
CSET 1 4 Column graph
CSET 2 4
CSET 3 1 Symbols and lines
MSET 3 0
CURVE
1984 25000
1985 26500
1986 30000
1987 30000
1988 30000
CURVE
1984 25000
1985 25175
1986 27075
1987 25721
1988 24435
CURVE
1984 30627
1985 40667
1986 32401
1987 43205
1988 48000
```

Graphic Output

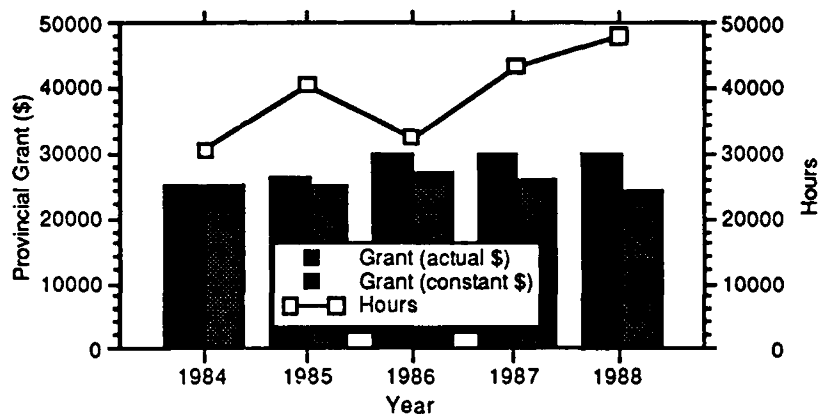
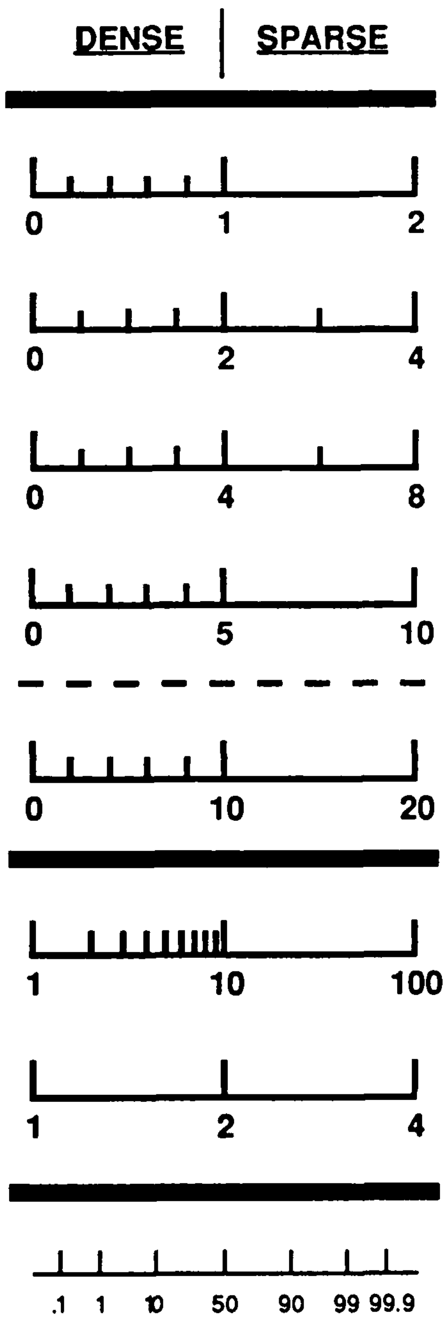



*Fig. 9. SAPLOT ASCII text input file and graphic output showing a bar graph.*

## 4.3. AXES

### 4.3.1. Ticks and Numbers

As mentioned in Section 2.2., SAPLOT produces easily understood axis markings. Dense markings are demonstrated on the left half of these examples, and sparse markings are on the right.




LINEAR AXES

Four axis models are used in order to span all possible linear numbered-tick intervals. These intervals are always chosen to be multiples (or sub-multiples) of 1, 2, 4, or 5. If the number of labelled ticks along an axis is allowed to vary by up to a factor of two, then these four axis models are sufficient to cover all possible linear numbered-tick intervals.

For these examples, the next higher numbered-tick interval is 10 (shown below the dashed line), which is easily seen to be derived from the same model as the interval 1. Similarly, the next lower numbered-tick interval from 1 is 0.5, which is easily seen to be derived from the same model as the interval 5.

LOGARITHMIC AXES

One axis model for base 10 is usually sufficient. However, if the axis spans one or less decade marks, then more than just the decade ticks should be labelled.

One axis model for base 2 is also usually sufficient.

PROBABILITY AXES

A fixed axis (as shown to the left) accommodates most applications.

### 4.3.2. Linear Axes

Four linear axes are shown in Fig. 10. The axis limits have been set by the "RANGE" commands. The curve data are always plotted according to the bottom and left axes, although it is planned to add the capability to plot data according to any two selected axes. This capability would make it easier to plot data of different types on the same plot; however, it is possible to do this now by use of the "ADJUST" command or by overlaying two plots with the use of the "PLOT" command. The same four axes shown in Fig. 10 are shown in Fig. 11 with sparse ticks (described in the previous section). The plot frame may be removed, as shown in Fig. 12.

Three types of axis marking are shown in Fig. 13. These axis markings are changed from their default values by the "NSET" command, whose final arguments are in parentheses in the following sentence. The bottom axis has no markings (-1), the left and right axes have ticks only (0), and the top axis has ticks and numbers (1). The right axis (axis "4") is not changed from its default markings. Axis markings are sometimes changed to accommodate immediately adjacent plots, for which a common scale may be used.

Input File

```
RANGE 1 -1   2
RANGE 2  3  -2
RANGE 3  8   9
RANGE 4 1.5 2.5
LABEL 1
Range 1
LABEL 2
Range 2
LABEL 3
Range 3
LABEL 4
Range 4
CURVE
0  0
1  1
```

Graphic Output

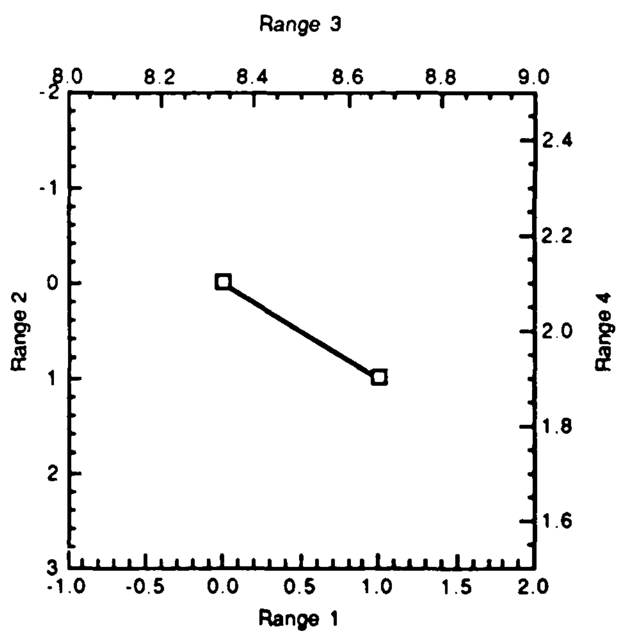



*Fig. 10. SAPLOT ASCII text input file and graphic output showing linear axes.*

Input File

```
DENSIT 0
RANGE 1 -1    2
RANGE 2 -2    3
RANGE 3  8    9
RANGE 4  1.5 2.5
CURVE
0    0
1    1
```

Graphic Output

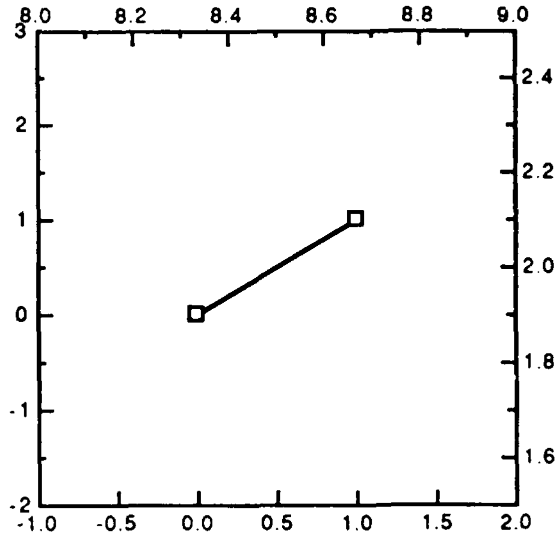



*Fig. 11. SAPLOT ASCII text input file and graphic output showing axes with sparse tick marks.*

Input File

```
FSET 0
CURVE
0      0
1      1
```

Graphic Output

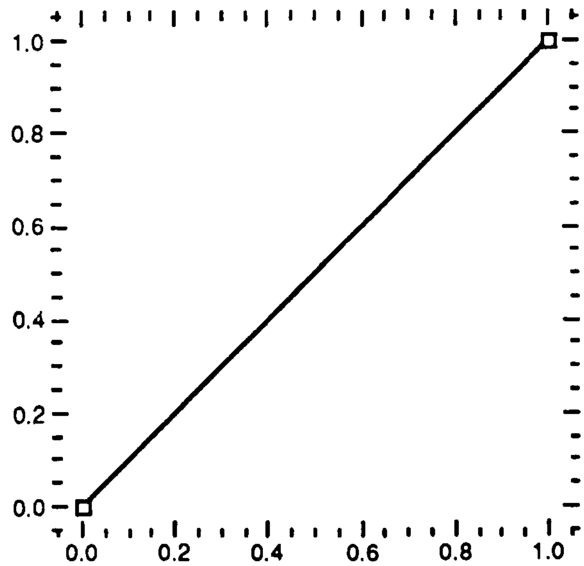



*Fig. 12. SAPLOT ASCII text input file and graphic output showing no frame.*

Input File

```
NSET 1 -1
NSET 2 0
NSET 3 1
CURVE
0      0
1      1
```

Graphic Output

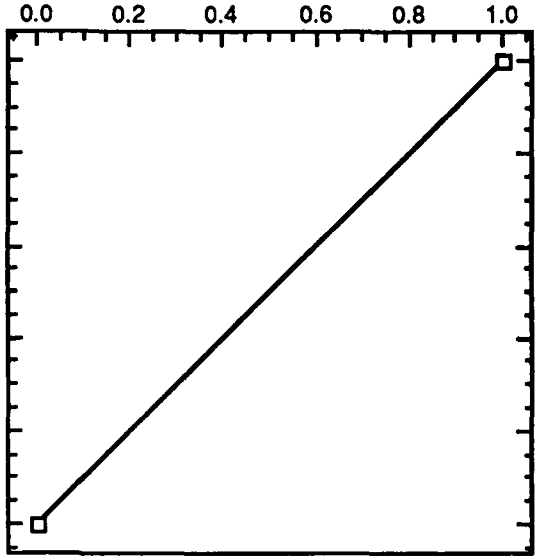



*Fig. 13. SAPLOT ASCII text input file and graphic output showing three types of axis marking. The bottom axis has no markings, the left and right axes have ticks only, and the top axis has ticks and numbers.*

### 4.3.3. Logarithmic Axes

Two types of logarithmic axes are shown in Fig. 14. These axis markings are changed from their default values by the "ASET" command, whose final arguments are in parentheses in the following sentence. The bottom and top axes are log base 10 (2), and the left and right axes are log base 2 (3).

Input File

```
ASET 1 2
ASET 2 3
CURVE
1     1
2     2
4     4
8     8
16    16
32    32
```

Graphic Output

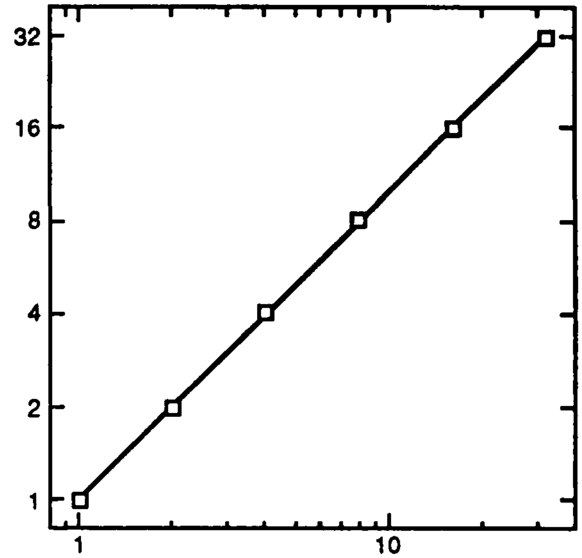



*Fig. 14. SAPLOT ASCII text input file and graphic output showing logarithmic axes. The bottom and top axes are log base 10, and the left and right axes are log base 2.*

### 4.3.4. Probability Axes

The bottom and top axes of Fig. 15 are probability axes. This type of axis marking is produced by the "ASET" command, with the final argument "4".

Input File

```
ASET 1 4
CURVE
.001    1
.01     2
.1      3
.5      4
.9      5
.99     6
.999    7
```

Graphic Output

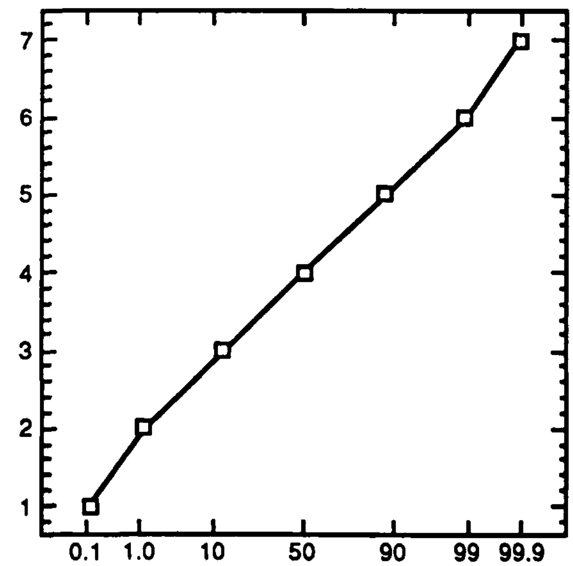



*Fig. 15. SAPLOT ASCII text input file and graphic output showing probability axes.*

### 4.3.5. Hours, Minutes, Seconds Axes

Although some of DREA's data are in the form of hours, minutes and seconds, an appropriate axis type has not yet been developed for SAPLOT. In most cases, a simple decimal axis (e.g. decimal hours or decimal degrees) has been sufficient. It is planned to add an hours, minutes, and seconds axis type in the future.

## 4.4. LABELS

A plotting package would be incomplete without a method of labelling. The labelling commands in SAPLOT can be divided into three types that are described in the following sections: axis labels, legends, and positioned labels.

### 4.4.1. Axis Labels

The most important labels for a line drawing are the axis labels. The very simple way that SAPLOT uses to specify these axis labels is shown in Fig. 16. The axis label is the contents of the file line (with leading and trailing tabs and/or spaces removed) following a "LABEL" command line. The axis label is centred on the axis specified by a numerical argument following "LABEL". Arguments greater than "4" spiral out beyond the first four axis labels.

Input File

```
LABEL 1
Here's Label #1
LABEL 2
Here's Label #2
LABEL 3
Here's Label #3
LABEL 4
Here's Label #4
LABEL 5
Here's Label #5
LABEL 6
Here's Label #6
LABEL 7
Here's Label #7
LABEL 8
Here's Label #8
CURVE
0     0
1     1
```

Graphic Output

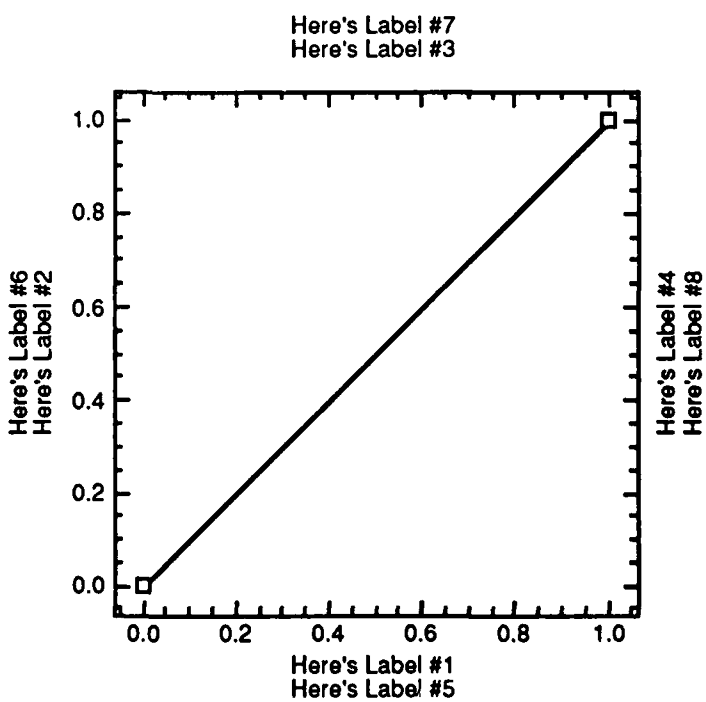



*Fig. 16. SAPLOT ASCII text input file and graphic output showing axis labels.*

### 4.4.2. Data Symbol and Line Legends

Although it is clearer and therefore preferable to label curves directly in many cases, it is possible to use a legend to identify curves by their line types or data point symbols. The legend labels follow a "LEGEND" command line, with normally one label corresponding to each curve in the plot. In Fig. 17, two curves are shown with their corresponding legend. The "LEGEND" command interpreter in SAPLOT tries to find a blank area of the plot for the legend, with the inside corners given highest priority. It is also possible to force the legend position with a "PSET" command.

Input File

```
RANGE 2  -.2  2
CSET 2 0
LEGEND
First Curve
Second Curve
CURVE
0    0
1    0
CURVE
0    1
1    1
```

Graphic Output

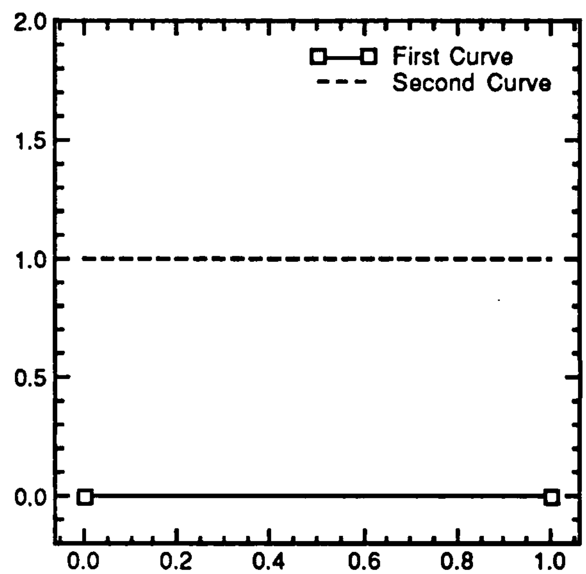



*Fig. 17. SAPLOT ASCII text input file and graphic output showing a curve legend.*

### 4.4.3. Randomly Positioned Labels

With the advent of "what you see is what you get" (WYSIWYG) computer systems such as the Apple Macintosh, final touchups of computer generated plots (such as adding randomly positioned labels) are easily done with the computer. Nonetheless, there are still cases where it is useful to be able to place a label at an arbitrary location on a plot using SAPLOT. Figure 18 shows the use of the "PLABEL" command, which places the label that is on the following file line at a location specified by three arguments following "PLABEL". The first and second arguments specify the percent of the way from the left to the right axis, and the percent of the way from the bottom to the top axis respectively (the values of the arguments can lie outside the range of 0 to 100). The third argument specifies the angle of the line of text in degrees relative to the horizontal.

Input File

```
PLABEL 50 50 45
Label Text
CURVE
0    0
1    1
```

Graphic Output

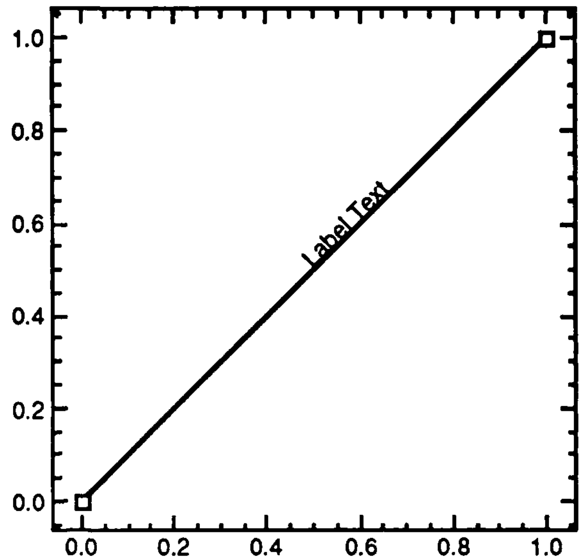



*Fig. 18. SAPLOT ASCII text input file and graphic output showing a positioned label.*

## 5. CONCLUSIONS

The SAPLOT software system described in this Technical Communication provides a simple means by which scientists or engineers can produce graphical output. Significant effort was put into SAPLOT in order to make it do the "right" thing by *default*, instead of requiring a multitude of commands. Only if the users want to do something unusual to their plots are commands necessary. The resulting simplicity of SAPLOT has made it attractive for the average user.

SAPLOT is designed to meet the standards of the American Institute of Physics, and to produce graphic output suitable for publications, slides, or overhead transparencies, without the need for any manual retouching. Careful attention is given to producing high quality axis markings.

The SAPLOT software system has become, in part, a file standard. The SAPLOT software takes a standard ASCII input data file, and produces graphical output representing that input data. The SAPLOT ASCII input file format has become a *de facto* standard for the storage of small sets of data at DREA, because of the extensive use of SAPLOT.

The extensive use of SAPLOT at DREA is reducing the overload on our professional illustrators. This speeds up the production of reports, journal articles, etc., and puts illustration tools directly in the hands of scientists and engineers. There is a danger in this, since scientists and engineers do not necessarily understand the art of illustration. Fortunately, SAPLOT tries to automatically produce graphical output to accepted standards by default.

Finally, SAPLOT is still evolving. New capabilities such as hours, minutes and seconds axes are being added. It is becoming more portable as new versions are written for different machines, and as the SAPLOT to PostScript file conversion software becomes available. It is likely that SAPLOT will continue to fill a need for scientific graphics for years to come.

## ACKNOWLEDGEMENTS

Richard Hughes produced the axis marking algorithms, Ed Chaulk produced the VAX/VMS version of SAPLOT, and Steven Hughes created the Macintosh and PostScript versions of SAPLOT. Their help has been invaluable.

# Appendix A

## SAPLOT Commands

The following is an alphabetically ordered list of the SAPLOT commands available at the time that this document was written.

**ADJUST** `<curve number> <*x> <*y> <+x> <+y>`

This command is used to scale and/or move curves on the plot. The values which will be plotted are determined by:

```
<x plot value> = <x value> * <*x> + <+x>
<y plot value> = <y value> * <*y> + <+y>
```

The whole range of curves may be adjusted by the same <+x> and <+y> values by specifying a curve number of 0. A waterfall plot may be set up with the <+x> and <+y> values by specifying a negative curve number. The curves to be waterfallen go from the absolute value of the curve number to the highest curve number. The first curve in the waterfall will have no offset added.

**ASET** `<axis number> <axis type>`

This command specifies the axis type for the given axis.

The `<axis type>` parameter may be one of the following:

| | |
|---|---|
| 1 | linear axis (DEFAULT) |
| 2 | log base 10 |
| 3 | log base 2 |
| 4 | probability |
| 5 | category |

**AXSET** `<x-axis length> <y-axis length>`

This command specifies lengths, in inches, for the x-axis and the y-axis. The default axes lengths are determined by the plot format.

## BSET `<curve number> <pen number>`

This command forces the curve to be drawn with the specified pen. The action of this command is dependent on the plotting device.

## CHSET `<% x-width> <% y-width>`

This command sets the x and y scale factors for text drawing.

## CSET `<curve number> <curve type>`

This command specifies the curve type for the given curve. A range of curves may be set to the same type by using a negative curve number. If negative, the range of curves affected will be from the absolute value of the curve number to the highest numbered curve (e.g. if `<curve number> = -1` then all curves are affected).

The `<curve type>` parameter may be one of the following:

| | |
|---|---|
| 0 | draw lines only |
| 1 | draw symbols joined by lines |
| 2 | draw symbols only |
| 3 | draw error bar lines, with symbols |
| 4 | draw column-type bar graph (vertical bars) |
| 5 | draw row-type bar graph (horizontal bars) |

The default curve type depends on the number of points in the curve. Curves with less than 41 points default to curve type 1. Curves with 41 or more points default to curve type 0.

Curve type 3 is used for drawing error bars. It causes a line to be drawn between successive pairs of points in the curve. Symbol number 256 has been designed for use as the top/bottom symbols of vertical error bars. Symbol number 257 has been designed for use as the left/right symbols of horizontal error bars.

## CURVE

```
<x value-1>    <y value-1>
<x value-2>    <y value-2>
.    .
.    .
.    .
<x value-N>    <y value-N>
```

This command precedes the numeric data that specify the curve to be plotted. The *x* and *y* values may be integral or floating point, and may be separated by space, tab, or comma (",") field delimiters. The number of points allowed in a curve is limited only by the amount of memory allocated to the application.

## DENSIT `<density>`

This command specifies the axis tick density.

The `<density>` may be one of the following:

| | |
|---|---|
| 0 | do not draw minor tick marks |
| 1 | do draw minor tick marks (DEFAULT) |

By default, minor tick marks will be drawn.

## END

This command is used to delimit the range of grey scale lines to be repeated in conjunction with the `REPEAT` command in the context of a `PATPLT` or `PTPLT` command. See `PTPLT` and `REPEAT` for more details.

## EXCH `<curve>`

This command causes the *x* and *y* data values to be interchanged for the specified curve. The data for a range of curves may be interchanged by using a negative curve number. If negative, the range of curves affected will be from the absolute value of the curve number to the highest numbered curve.

## FORMAT `<style number>`

This command specifies the style, or format, of the plot.

The `<style number>` may be one of the following:

1 VAX SAPLOT Figure
2 Viewgraph
3 Single Column Figure
4 Double Column Figure
5 Thesis Figure

VAX SAPLOT Figure:

This format is the same as the old SAPLOT program on the DREA VAX systems. This is the default format. Data are plotted in a 5.5 by 5.5 inch frame with 18-point text in portrait mode. The text size will be scaled smaller if the 'AXSET' command is used to decrease the axes sizes.

Viewgraph Figure:

This format creates a plot for use as a viewgraph, following the DREA drafting office guidelines. Data are plotted in a 7.0 by 5.5 frame with 18-point text in landscape mode. The default line width is 2 points.

Single Column Figure:

This format creates a plot suitable for use in a DREA Report or Technical Memorandum. Data are plotted in a 5.0 by 3.0 inch frame with 12-point text in portrait mode.

Double Column Figure:

This format creates a plot suitable for use in a JASA Report. The figure is intended for 50% reduction before publication. This format is very similar to the VAX SAPLOT format. Data are plotted in a 5.5 by 5.5 inch frame with 14-point text in portrait mode.

Thesis Format:

This format creates a plot suitable for use in a postgraduate thesis. It has been set up to follow the guidelines of the Dean of Graduate Studies at Dalhousie University. Data are plotted in a 12.0 by 8.5 cm frame with 12-point text in portrait mode.

See Appendix B for a list of the layouts used in each of these plot formats.

## FSET `<frame type>`

This command is used to determine the frame type. The `<frame type>` parameter may be one of the following. (By default, a frame will be drawn for every plot.)

| | |
|---|---|
| 0 | do not draw the frame |
| 1 | draw the frame (DEFAULT) |

## GFILL `<curve> <greyLevel>`

This command specifies the grey fill level for the given bar curve. A range of curves may be set to the same level by using a negative curve number. If negative, the range of curves affected will be from the absolute value of the curve number to the highest numbered curve.

The <greyLevel> parameter ranges from 0 to 100:

| | |
|---|---|
| 0 | white |
| 100 | black |

Note that most laser printers have a resolution of only 8-10 grey levels. By default, the bars will be filled in uniform increments of grey from black to white.

## LABEL `<label number>`

`<label text>`

This command specifies the text to be drawn as a label. Labels are numbered according to the axis, as below:

```
    7
    3
6 2[ ]4 8
    1
    5
```

Labels 3, 7, etc. are considered as titles, and may be drawn in a larger font, depending on the plot format. Leading spaces are stripped from the label text string. Labels may be positioned randomly (see also PLABEL) by specifying a negative label number of the form `-aaaxxyy`, where `aaa` is the string rotation angle in degrees (may be omitted if zero), `xx` is the percentage across the frame, `yy` is the percentage up the frame. For example, `<label number>` = `-5050` will cause the string to be plotted horizontally, with the lower left hand corner of the text in the centre of the frame.

**LEGEND** `[<legend count>]`

```
    <legend_string-1>
    <legend_string-2>
    .
    .
    .
    <legend_string-N>
```

This command specifies specifies the text to be drawn as the plot legend. The `<legend count>` parameter is optional in SAPLOT. It specifies the number of legend strings following the `LEGEND` command. If the `<legend count>` parameter is not specified, SAPLOT assumes that there is one legend string for each curve in the plot. Use a legend string of "`!NOL;`" to ignore the string for any given curve.

SAPLOT attempts to find an empty region of the plot in which to place the legend. Use the `PSET` command to override the default positioning. Use the `LFSET` command to draw a box around the legend.

**LFSET** `<frame> <fill>`

This command specifies whether or not the legend area will have a frame drawn around it, and whether or not the area will be blank filled before drawing the legend (this hides any part of the plot behind the legend).

The `<frame>` parameter may be one of the following:

| | |
|---|---|
| 0 | don't draw the legend box frame (DEFAULT) |
| 1 | draw a plain box around the legend area |
| 2 | draw a shadow box around the legend area |

The `<fill>` parameter may be one of the following:

| | |
|---|---|
| 0 | don't blank fill the legend area (DEFAULT) |
| 1 | blank fill the legend area |

NOTE: The legend box will always be blank filled when `<frame>` = 2.

**LINCLR** `<pen number>`

This command selects the pen for drawing on a Tektronix 4662 pen plotter.

## **LOG10** `<curve number> <xLog> <yLog>`

This command specifies whether or not the base 10 logarithm of the *x* and/or *y* data values should be calculated for plotting the given curve. If `<xLog>` or `<yLog>` are non-zero, then the base 10 logarithm of the respective data will be plotted for that curve.

## **LSET** `<curve> <line type>`

This command specifies the line type for the given curve.

The `<line type>` parameter may be one of the following:

0 => solid line
1 => short-dash
2 => medium-dash
3 => medium-dash dot
4 => short-dash dot
5 => long-dash
6 => dot
7 => long-dash short-dash
8 => long-dash dot
9 => medium-dash dot dot
10 => long-dash dot dot
11 => short-dash dot dot
12 => short-dash short-dash dot
13 => short-dash short-dash dot dot
14 => short-dash dot dot dot
15 => medium-dash dot dot dot
16 => long-dash dot dot dot

Values of `<line type>` less than 0 or greater than 16 are interpreted as a 2 to 9 digit sequence to define a pattern,described by a DRAW-MOVE sequence, in units of tenths of inches. For example, the integer 41 dictates a DRAW of 0.4 inches followed by a MOVE of 0.1 inches. Zero means a dot. A negative sign can be used to force a leading zero. The maximum number of digits is nine.

If `<curve>` is less than zero, all curves from abs(`<curve>`) are affected by this command.

## MSET `<curve number> <symbol type>`

This command specifies the number of the symbol to be draw for the given curve. The symbols available are shown in Fig. 6.

If `<curve>` is less than zero, all curves from abs(`<curve>`) are affected by this command.

## NCLIP

This command is used to turn off the default clipping of the curve data to be within the frame border.

## NEWPAGE

This command causes the plots following it to be drawn on a new page. The effect is equivalent to starting a completely new plot job on the next page.

## NSET `<axis number> <label type>`

This command specifies the type of marking to be applied to the given axis. Axes are numbered 1 through 4 as below:

```
  3
2[ ]4
  1
```

The `<label type>` parameter may be one of the following:

-1 => no marking
0 => ticks only
1 => ticks and numbers

By default, ticks and numbers are drawn for axes 1 and 2, while ticks only are drawn for axes 3 and 4. A range of axes may be affected by using a negative axis number. If negative, the axes affected will range from the absolute value of the axis number to axis number 4.

## ORSET `<x origin> <y origin>`

This command sets the plot origin, in inches, with respect to the lower left hand corner of the page. The plot origin is the location of the lower left hand corner of the plot frame. The default orientation of the page is portrait unless the `THSET` command is used, in which case landscape becomes the default.

## PATPLT `<x offset> <y offset> [<x replication> <y replication>]`

This command initiates pattern plotting for devices that have this capability. The default pixel size is 2/75 inches (8 dots on a 300 dot-per-inch (dpi) laser printer). There are 17 patterns, specified by the ASCII characters "0" through "@". Each character denotes one pattern pixel. The offset parameters specify the offset (in inches) from the upper left corner of the plot frame.

The lines of pattern data must be continuous and delimited by a new line. Any number of lines may follow the `PATPLT` command, and each line may be up to 800 characters long. The "*" repeat operator may be used to repeat any pixel up to 999 times. Long data segments may be continued on the next line by ending the line with the continuation character "-".

For example:

```
PATPLT 0 0
222222333333
```

is the same as

```
PATPLT 0 0
2*0063*006
```

The optional `<x replication>` and `<y replication>` parameters may be used to repeat the pattern pixels in the *x* and *y* directions, respectively.

For example:

```
PATPLT 0 0 3 2
0123
```

is the same as

```
PATPLT 0 0
000111222333
000111222333
```

## PLABEL `<x position> <y position> <angle> [<x justification> <y justification>]`

`<label text>`

This command specifies the text to be drawn as a label. The label text is positioned `<x position>` percent across the frame, `<y position>` percent up the frame, and is plotted at the angle specified by `<angle>`. See also the `LABEL` command. The parameters `<x justification>` and `<y justification>` are optional. If given, they specify the justification point for positioning the label.

The `<x justification>` parameter may be one of the following:

| | |
|---|---|
| 0 | left justify (DEFAULT) |
| 1 | centre justify |
| 2 | right justify |

The `<y justification>` parameter may be one of the following:

| | |
|---|---|
| 0 | top justify |
| 1 | centre justify |
| 2 | bottom justify (DEFAULT) |

## PLOT

This command separates logically distinct plots contained in the same text file. It may also be used to separate groups of curves for the same plot. (This was used on older versions on `SAPLOT` with limits on the number of curves per plot.) `ORSET`, `AXSET`, `THSET` and `PVSET` parameters are retained across plots, and so do not need to be re-entered if unchanged.

## PSET `<x position> <y position> [<x justification> <y justification>]`

This command specifies the positioning of the legend. The parameters `<x position>` and `<y position>` are mandatory. They specify the position of the legend justification point (default is upper left-hand corner) in a percentage of the plot width (for `<x position>`) and the plot height (for `<y position>`). Values of `<x position>` and `<y position>` which cause the legend to be placed outside the plot frame are permitted. The parameters `<x justification>` and `<y justification>` are optional. If given, they specify the justification point for positioning the legend.

The `<x justification>` parameter may be one of the following:

| | |
|---|---|
| 0 | left justify (DEFAULT) |
| 1 | centre justify |
| 2 | right justify |

The `<y justification>` parameter may be one of the following:

| | |
|---|---|
| 0 | top justify (DEFAULT) |
| 1 | centre justify |
| 2 | bottom justify |

## PTPLT `<x offset> <y offset> [<x replication> <y replication>]`

This command initiates grey scale plotting for devices that have this capability. The default pixel size is 1/75 inches (4 dots on a 300 dpi laser printer). The grey scale is 17 shades from white to black, specified by the ASCII characters "0" through "@". Each character denotes one grey scale pixel. The offset parameters specify the offset (in inches) from the upper left corner of the plot frame.

The lines of grey scale data must be continuous and delimited by a new line. Any number of lines may follow the `PTPLT` command, and each line may be up to 800 characters long. The "*" repeat operator may be used to repeat any pixel up to 999 times. Long data segments may be continued on the next line by ending the line with the continuation character "-".

For example:

```
PTPLT 0 0
222222333333
```

is the same as

```
PTPLT 0 0
2*0063*006
```

The optional `<x replication>` and `<y replication>` parameters may be used to repeat the gray scale pixels in the *x* and *y* directions, respectively.

For example:

```
PTPLT 0 0 3 2
0123
```

is the same as

```
PTPLT 0 0
000111222333
000111222333
```

## PVSET `<x pivot> <y pivot>`

This command sets the rotation pivot point, in inches, with respect to the lower left hand corner of the page. If the plot is rotated using the THSET command, it will rotate about the specified pivot point. The default pivot point is the centre of the first plot frame. The default orientation of the page is portrait unless the THSET command is used, in which case landscape becomes the default.

## PWSET `<pen width>`

This command specifies the pen width to be used for the plot. The `<pen width>` parameter is in units of 1/300 inches. The default pen width is dependent on the format.

## RANGE `<axis> <lower / left value> <upper / right value>`

This command specifies the lower and upper, or left and right, values for the specified axis. Axes are numbered 1 through 4 as below:

```
  3
2[ ]4
  1
```

The lower and left values may be numerically less or more than the upper and right values, respectively.

## REPEAT `<repeat count>`

This command is used within the grey scale data to repeat some number of lines the specified number of times. The grey scale data to be repeated are delimited by the END command.

For example:

```
PTPLT 0 0
REPEAT 30
22344556678899000@@@@@::::;;;>>>????
77744803><<<<@@@@@>>>>><<<<<::::;;;;
END
```

In the example, the grey scale lines between REPEAT and END will be plotted 30 times. The REPEAT - END commands will be recognized only within the context of a PATPLT or PTPLT command.

## **SETSPD** `<unit type> <speed>`

This command sets the plotting speed of a Tektronix 4662 pen plotter.

```
<unit type> = 1  is mm
            = 2  is inches
```

`<speed>` is units per second (4662 range is 10 to 570 mm/s).

## **SSET** `<axis number> <start> <spacing> <subdivisions>`

This command specifies the start value, tick spacing and number of subdivisions for the given axis. Axes are numbered 1 through 4 as below:

```
  3
2[ ]4
  1
```

By default, these parameters are determined automatically for a given axis type, based on the range of the data.

## **THSET** `<angle>`

This command specifies the angle of rotation (in degrees) of the plot about the pivot point (see also `PVSET`). The default orientation of the page is landscape when the `THSET` command is used.

## **TSET** `<tick in> <tick out> <sub-tick in> <sub-tick out>`

This command specifies the length in tenths of inches for the major and minor ticks. The default size is format dependent.

## **TXTCLR** `<pen number>`

This command sets the pen for drawing text with a Tektronix 4662 pen plotter.

## WSET `<curve> <pen width>`

This command specifies the pen width to be used for plotting the curve. The `<pen width>` parameter is in units of 1/300 inches. The default pen width is dependent on the format. The pen width for a range of curves may be set by using a negative curve number. If negative, the range of curves affected will be from the absolute value of the curve number to the highest numbered curve.

## XLABEL `<x position> <y position> <angle> [<x justification> <y justification>]`

`<label text>`

This command specifies the text to be drawn as a label. The label text is positioned at the point (`<x position>`,`<y position>`) with respect to data values along axes 1 ($x$) and 2 ($y$), and is plotted at the angle specified by `<angle>`. See also the `LABEL` command. The parameters `<x justification>` and `<y justification>` are optional. If given, they specify the justification point for positioning the label.

The `<x justification>` parameter may be one of the following:

| | |
|---|---|
| 0 | left justify (DEFAULT) |
| 1 | centre justify |
| 2 | right justify |

The `<y justification>` parameter may be one of the following:

| | |
|---|---|
| 0 | top justify |
| 1 | centre justify |
| 2 | bottom justify (DEFAULT) |

# Appendix B

## Format Defaults

Each of the plot formats recognized by SAPLOT specifies a set of defaults which define the size of the plot frame, the location of the frame on the page, text size, etc. The defaults for each format are given below (all dimensions are in inches unless otherwise noted):

| | Format | | | | | |
|---|---|---|---|---|---|---|
| | VAX | Viewgraph | Double | Single | Thesis | VAX |
| format # | 1 | 2 | 3 | 4 | 5 | 6 |
| x-axis length | 5.5" | 7.0" | 5.5" | 5.0" | 12.0 (cm) | 5.5" |
| y-axis length | 5.5" | 5.5" | 5.5" | 3.0" | 8.5 (cm) | 5.5" |
| x-origin | 2.0" | 2.5" | 2.0" | 2.0" | 6.0 (cm) | 2.0" |
| y-origin | 2.5" | 1.5" | 2.5" | 2.5" | 15.0 (cm) | 2.5" |
| label size | 18 | 18 | 14 | 12 | 12 | 18 (points) |
| title size | 18 | 24 | 14 | 14 | 14 | 18 (points) |
| font scaling | yes | no | no | no | no | yes |
| major tick | 0.14 | 0.17 | 0.15 | 0.08 | 0.08 | 0.14 |
| minor tick | 0.07 | 0.08 | 0.07 | 0.04 | 0.04 | 0.07 |
| line width | 1 | 2 | 1 | 1 | 1 | 1(points) |
| orientation | portrait | landscape | portrait | portrait | portrait | landscape |

## REFERENCES

American Institute of Physics *Style Manual* (1978). Third edition, A.I.P., New York, N.Y.

Caldwell, D. A. (1987). "A standard for the DREA data descriptor block," DREA Technical Communication 87/302.

Chapman, D. M. F. (1989). "A simple estimate of propagation loss fluctuations due to modal interference," J. Acoust. Soc. Am., **85**, 1097-1106.

Cotaras, F. D., Fraser, I. A., and Merklinger, H. M. (1988). "Near-surface ocean ambient noise measurements at very low frequencies," J. Acoust. Soc. Am. **83**, 1345-1359.

Staal, P. R. (1983). "Acoustic propagation measurements with a bottom mounted array," in: "Acoustics and the Sea-Bed," N.G. Pace, ed., Bath University Press, Bath, UK, pp. 289-296.

Staal, P. R., and Hughes, S. J. (1988). "Shallow-water Geo-acoustics," article in CRAD 87 - Research and Development Branch Review 1987, p. 7.

Swingler, D. N., and Walker, R. S. (1989). "Line-array beamforming using linear prediction for aperture interpolation and extrapolation," IEEE Trans. Acoust. Speech and Sig. Proc. **37**, pp. 16-30.

Tufte, E. R. (1983). *The visual display of quantitative information,* (Graphics Press, Cheshire, Connecticut,1983).

Walker, R. S., and Ashley, A. T. (1984). "Modal decomposition for detection of plane waves with a line array receiver," proceedings of ICASSP 84 - IEEE International Conference on Acoustics, Speech, and Signal Processing, March 19-21, 1984, pp. 40.10.1 to 40.10.4.

Young, C. A. (1989). "SAPLOT Scientific Graphic Software - User's Manual," DREA Technical Communication (in preparation).

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM
(highest classification of Title, Abstract, Keywords)

## DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

| | |
|---|---|
| 1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)<br>Defence Research Establishment Atlantic | 2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)<br>UNCLASSIFIED |

3 TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title.)

SAPLOT Scientific Graphic Software - History and Overview

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Staal, Philip R.

| | | |
|---|---|---|
| 5. DATE OF PUBLICATION (month and year of publication of document)<br>September 1989 | 6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)<br>48 | 6b. NO. OF REFS (total cited in document)<br>10 |

6. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

DREA Technical Communication

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

| | |
|---|---|
| 9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)<br>Project 1AD | 9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written) |
| 10a. ORIGINATOR 'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be uniqué to this document.)<br>DREA Technical Communication 89/309 | 10b. OTHER DOCUMENT NOS. (any other numbers which may be assigned this document either by the originator or by the sponsor) |

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

( X ) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12.. DOCUMENT ANNOUNCEMENT (any limitations to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where futher distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

UNCLASSIFIED

SECURIITY CLASSIFICATION OF FORM

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

The SAPLOT software system provides a simple means by which scientists or engineers can produce graphical output. SAPLOT is designed to meet the standards of the American Institute of Physics, and to produce graphic output suitable for publications, slides, or overhead transparencies. Careful attention is given to producing high quality axis markings. The SAPLOT software system has become, in part, a file standard. The SAPLOT software takes a standard ASCII input data file, and produces graphical output representing those input data. Finally, SAPLOT is still evolving. It is becoming more portable as new versions are written for different machines, and as SAPLOT to PostScript file conversion software becomes available. It is likely that SAPLOT will continue to fill a need for scientific graphics for years to come.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is *required*. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

computer graphics
computer programs
PDP computers
VAX computers
Macintosh computers
PostScript
graphic arts
charts
diagrams
axes (reference lines)
data storage
SAPLOT
SAPLOT_PS

(from T.E.S.T.)