Forestry Canada Forêts Canada

PLT: An interactive graphics package for VAX/VMS

Jacques Régnière

Information Report LAU-X-88E
Quebec Region

CENTRE DE FORESTERIE DES LAURENTIDES

Le Centre de foresterie des Laurentides (CFL) est un des six établissements régionaux de Forêts Canada (ForCan). Le Centre poursuit des travaux de recherche et de développement dans le but de promouvoir un aménagement et une utilisation plus rationnels des ressources forestières du Québec. En collaboration avec divers groupes et organismes québécois, les chercheurs du CFL visent à acquérir une meilleure compréhension du fonctionnement des écosystèmes forestiers en vue de contribuer à solutionner les problèmes majeurs en foresterie au Québec et à développer des méthodes acceptables pour l'amélioration et la sauvegarde de l'environnement forestier.

Au Québec, les activités de ForCan portent sur trois éléments majeurs: la recherche dans le domaine des ressources forestières, la recherche dans le domaine de la protection et enfin, le développement forestier. La plupart des travaux sont entrepris pour répondre aux besoins des organismes intéressés à l'aménagement forestier, surtout dans le but d'améliorer la protection, la croissance et la qualité de la ressource forestière de la région. Les résultats de ces travaux sont diffusés sous forme de rapports techniques et scientifiques ou autres publications, permettant de rejoindre toutes les catégories d'utilisateurs.

LAURENTIAN FORESTRY CENTRE

The Laurentian Forestry Centre (LFC) is one of six regional establishments of Forestry Canada (ForCan). The Centre's objective is to promote through research and development the most efficient and rational management and use of Quebec's forest. In cooperation with several Quebec agencies, scientists at LFC seek to acquire a better understanding of how the forest ecosystem works with a view to solving major forestry problems and developing methods to improve and to protect the forest environment.

In the province of Quebec, ForCan's program consists of three major elements: forest resources research, forest protection research, and forest development. Most of the research is undertaken in response to the needs of forest management agencies, with the aim of improving the protection, growth, and quality of the region's forest resource. The results of this research are distributed to potential users through scientific and technical reports and other publications.

PLT: AN INTERACTIVE GRAPHICS PACKAGE FOR VAX/VMS

Jacques Régnière

Information Report LAU-X-88E
1989

Forestry Canada
Quebec Region

Catalog No. Fo46-18/88E
ISSN 0835-1570
ISBN 0-662-17281-7
Printed in Canada

Limited additional copies of this publication are available at no charge from:

Forestry Canada, Quebec Region
Laurentian Forestry Centre
1055 du P.E.P.S.
Sainte-Foy, Quebec
G1V 4C7

Copies or microfiches of this publication may be purchased from:
Micromedia Inc.
Place du Portage
165, Hôtel-de-Ville
Hull, Quebec
J8X 3X2

Aussi disponible en français sous le titre «PLT: un logiciel graphique interactif pour VAX/VMS».

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

LIST OF TABLES AND FIGURES

Page

ABSTRACT

PLT is a computer graphics program aimed at easing the task of preparing two-dimensional line graphs, bar charts and other diagrams for scientific publications and presentations. PLT offers several advantages over large statistical packages with incorporated graphics capabilities: it is fully interactive, very fast and quite simple to use. It provides great flexibility in the development of publication-quality illustrations with mixed line and bar graphs, multiple figures, regression analysis, and more. PLT allows complete control over the graph's shape, axes, ticmarks, line and symbol types, bar shadings and font styles. PLT possesses many of the attributes which make microcomputer graphics popular: extensive on-screen drawing and lettering capabilities, interaction through the keyboard, mouse or digitizing tablet, progressive development of illustrations, on-line help, etc. PLT is a FORTRAN-77 program. At this point, it is designed to run on VAX computers under the VMS operating system with graphic terminals and printers compatible with standard TEKTRONIX-4014 technology (such as VT/240-compatible devices). PLT can also be used on microcomputers connected to a VAX through VT/240-emulator software. PLT can also be used to complete illustrations prepared with TEKTRONIX-4014 protocol by other software such as SAS or MINITAB.

*****

*****

To obtain a copy of the PLT software for VAX/VMS, communicate directly with the author of this document.

RÉSUMÉ

PLT est un logiciel graphique visant à faciliter la préparation de graphiques en deux dimensions, histogrammes et autres illustrations destinées à la publication ou à la présentation de données scientifiques. PLT offre de nombreux avantages par rapport aux logiciels statistiques incorporant des capacités graphiques: il est totalement interactif, ultra rapide et très simple à utiliser. Il offre une grande flexibilité dans le développement d'illustrations de qualité supérieure, avec la combinaison de lignes et histogrammes, figures multiples, analyse de régression et plus. PLT permet le contrôle total de la forme du graphique, des axes, des graduations, des types de symboles ou traits, du remplissage des bandes d'histogrammes et du style de lettrage. PLT possède plusieurs des attributs qui ont rendu populaires les logiciels graphiques sur micro-ordinateurs: capacité étendue de dessin et lettrage à l'écran, interaction au clavier, à la souris ou à la table de numérisation, développement progressif des illustrations, aide interactive, etc. PLT est écrit en FORTRAN-77. Pour le moment, il a été mis au point pour usage sur un ordinateur VAX sous le système d'exploitation VMS, muni de terminaux et imprimantes compatibles avec le protocole standard TEKTRONIX-4014 (par exemple, les appareils compatibles avec le VT/240). PLT peut également être utilisé sur un micro-ordinateur branché au VAX par le truchement d'un logiciel émulateur du VT/240. PLT peut aussi être utilisé pour compléter des illustrations préparées selon le protocole TEKTRONIX-4014 par d'autres logiciels tels que SAS ou MINITAB.

*****

L'exclusion de certains produits manufacturés ne signifie pas nécessairement que Forêts Canada les désapprouve et le fait que d'autres produits soient mentionnés ne signifie pas nécessairement qu'il les approuve.

*****

Pour obtenir une copie du logiciel PLT pour VAX/VMS, communiquer avec l'auteur de ce document.

INTRODUCTION

PLT is a bilingual interactive graphic program which was designed to prepare easily and rapidly most commonly-encountered types of 2-dimensional line graphs and histograms. PLT can be used to prepare maps, diagrams, charts, tables and other visuals, with its DRAw facility. It can make use of a mouse or digitizing tablet. Hard copies of graphs can be obtained on laser printers and digital plotters as well as on printers connected to the terminal. A series of standard maps and other drawings are provided with PLT. Consult your system manager to know which of these are on your system and how to use them (see **SYSTEM DRAWING FILES**).

PLT, a FORTRAN-77 program, was developed on VAX/VMS. To run it on most systems, the PLT command is issued in response to the VMS $ prompt. PLT is based on the standard TEKTRONIX 4014 graphics protocole. For details on how to use PLT and compatible equipment, see the **USE OF PLT ON VAX/VMS** section.

Any type of experimental data consisting of a series of measurements repeated several times in a sequence of observations is especially suitable for graphing with PLT. However, with a bit of ingenuity there is usually a way to prepare any kind of data for 2-dimensional graphing by PLT.

PLT responds to a series of commands which it reads from the terminal keyboard or from command files stored on disk (see the **COMMAND FILES** section). Graphs can be built gradually and can be displayed at any time during a session. Once a graph has been prepared to the user's satisfaction, it is possible to save on disk the parameters needed to redo or further develop a graph. A major advantage of PLT is that data files can be substituted, with the FILe command, without any other alteration to do a series of identical graphs of several data sets. PLT has extensive on-line documentation in the form of an interactive HELp facility which explains syntax and use of the commands. It is therefore not necessary to have this manual in hand to use the program.

PLT can draw up to 8 distinct curves and/or histograms per graph on 1 or 2 Y-axes (left-hand and right-hand side of the graph). Axes can be positioned anywhere on the graph and the graph can be placed anywhere on the screen. By using the SAVe and RETrieve commands (see descriptions elsewhere in this text), PLT can draw up to 6 graphs on the same screen or page.

About the notation used in this manual to describe command syntax: Only the first 3 letters of command verbs are necessary (capitals or lower case). Other words within command lines can be abbreviated to their first letter. Brackets (round, square, etc.) indicate optional parts of commands. In examples, user entries are printed in bold characters.

DATA PREPARATION AND ENTRY

The Structure Of PLT Data Sets

PLT handles data which can be represented in a matrix or table, where columns are different measurements (hereafter **variables**) and lines are repetitions (hereafter **observations**). An example of a data set for PLT is given in Table 1.

Table 1. Hypothetical data set suitable for PLT

Julian Date	Proportion of maximum Shoot		Relative concentration			
	Weight	Length	Nitrogen	Phosphor	Fats	Sugars
106.	0.000	0.000	1.000	0.800	0.600	0.200
121.	0.031	0.015	0.995	0.790	0.577	0.650
136.	0.162	0.070	0.972	0.780	0.555	0.860
151.	0.376	0.166	0.918	0.750	0.533	0.945
166.	0.606	0.298	0.832	0.694	0.511	0.948
181.	0.792	0.456	0.719	-1	0.488	0.914
196.	0.913	0.626	0.589	0.486	0.466	0.865
211.	0.975	0.789	0.450	0.335	0.444	0.810
226.	0.997	0.924	0.308	0.240	0.422	0.755
241.	1.000	1.000	0.250	0.200	0.400	0.700

Usually, PLT reads data sets from sequential, ASCII files stored on disk. It can also read data from the terminal screen. The data files must reflect the line and column structure illustrated in Table 1 and can be created with a variety of tools (editor, programs, etc.).

File Format

PLT reads data files in **list-directed** format. This means that no specific format is required except that all values must be separated by at least one space or a comma. Observations can be split into as many lines as necessary or useful provided that each new observation starts on a new line. The data files can also contain table titles, column headings, or footnotes at the top and bottom of the file. PLT cannot handle alphanumeric data. Therefore, data lines must be kept free of alphanumerics unless these are at the end of observations, beyond the last numerical variable.

Size Of Data Sets

PLT processes files which contain between 2 and 100 measurements per observation (thus, up to 100 variables in a file). However, PLT cannot store more than 25 variables in memory. There is also a limit of 10 000 data items per file and a limit of between 2 and 5 000 to the number of observations (and thus points per curve). Whenever the product of the number of variables by the number of observations is greater than 10 000, the input file cannot be processed all at once by PLT.

The SELect command described in this manual, is intended to get around these size limitations, allowing only some of the columns of a file to be read at a time.

Missing Values

PLT handles missing data. Missing values must be entered in the data file by giving them a value below the minimum valid datum (e.g. missing values at or below -1, as in Table 1). Once this missing value code has been given to PLT with the MISsing command, any value at or below this specified code is considered missing. Missing data are not used in drawing, nor in automatic evaluation of axis ranges or in other operations. Missing values can be used to cause interruptions in the drawing of a curve (series of X-Y coordinates) and thus allow a "pen-up" operation. This can be used to draw complex objects on the screen.

THE PLT COMMANDS

The following sections describe the various PLT commands, their syntax and usage. PLT commands are divided into 5 groups: (1) Data definition and manipulation; (2) Graph structure; (3) Lettering; (4) DRAw mode; and (5) Program execution.

Data Definition And Manipulation Commands

These 14 commands are used to tell PLT where to find the data (FILe command), which subset of the data to use for large data sets (SELect command), or to enter data via the terminal (ENTer command). Missing values are identified by the MISsing command. Graph variables are defined by the VARiable, PREcision and CLAsses commands. Data can be sorted with the SORt command, and transformed to logs or antilogs with the LOG and ANTilog commands. They can be transformed to percentages (PERcent command) or smoothed (SMOoth command). Regression lines can be obtained with the REGress and POLynomial commands.

The FILe Command - Syntax: FIL [FILENAME N (FORMAT)]

The FILe command is used to define the name of the disk file (**FILENAME**) where the data are stored and the number of columns or variables (N) per observation in the file (between 2 and 100). The FILENAME parameter may be any valid file specification including other than default disk drive, directory or version number. The default file type is .DAT.

When the input file has been successfully found and opened, PLT reads it to the end or until a maximum of 10 000 values is reached, whichever comes first. Upon successful completion of a FILe command, PLT prints the number of observations read on the terminal screen. For more details on PLT input file structure, see the **DATA PREPARATION AND ENTRY** section. Because

PLT can only store 25 variables, it is necessary to use the SELect command to process files containing more than 25 variables.

Each variable read in by PLT is automatically identified by a number between 1 and N (the second parameter of the FILe command). Whenever a PLT command requires the I.D. number of a variable, it is the POSITION of this variable in the input file that is required. The only exceptions are when the SELect, PERcent or SMOoth commands have been issued.

The FILe command is often the first command issued in a PLT session. However, it can be issued at any time. For example, several similar graphs with different data sets may be prepared by building up the graph on the first data set and simply issuing a new FILe command to change data sets.

The optional **FORMAT** parameter is used to describe the format of the data lines when list-directed input cannot apply. Only X and F field descriptors can be used. For example, the two digits in 123 A 765.321 could be read by PLT using the format (f3.0,3x,f7.3). Consult a FORTRAN manual for more details on format specification.

When the FILENAME and N parameters are omitted, PLT prints the current data definition: file name, number of variables, number of columns in the file, number of columns selected (see SELect command), the I.D. numbers of variables for PLT and the corresponding columns in the file (identical when the SELect, PERcent or SMOoth commands are not used).

```
Example: PLT > FIL JUNK.DAT 10
         NUMBER OF OBSERVATIONS ENTERED: 5
         PLT > FIL
           FILE: JUNK.DAT
  NB OF COLUMNS: 10      OBSERVATIONS:  5
    NB SELECTED: 10
     NO FOR PLT:  1  2  3  4  5  6  7  8  9 10
     NO IN FILE:  1  2  3  4  5  6  7  8  9 10
```

The SELect Command - SYNTAX: SELect NN

The SELect command is used whenever a file contains more than 10 000 values or more than 25 variables. If PLT cannot read all the observations while executing a FILe command (because the product of the number of observations and variables exceeds 10 000), the SELect command can be used to read only part of the file. After the SELect command is issued, PLT prompts the user for NN columns to be selected in the input file (NN between 2 and 25). PLT then reads the file again, skipping columns which were not specified in the selected list. The net effect of this is similar to having a new input file containing only the columns specified. Variable I.D. numbers are changed to reflect the new data structure. The SELect command can be used repeatedly during a PLT session

to change the selected columns. Use the FILe command with no parameters for a list of columns currently selected.

```
Example: PLT > SEL 4
         ENTER ID OF COLUMNS TO BE SELECTED: 1 4 7 9
         NUMBER OF OBSERVATIONS ENTERED: 5

         PLT> FIL
           FILE: JUNK.DAT
  NB OF COLUMNS: 10
    NB SELECTED:  4
     NO FOR PLT:  1  2  3  4
     NO IN FILE:  1  4  7  9
```

The ENTer Command - Syntax: ENT FILENAME NVARS

In terms of syntax and usage, this command resembles the FILe command. It is issued to enter data on the terminal, via PLT, and to store them in an ASCII file named **FILENAME** (PLT uses .DAT as the default when a file type is not specified). The **NVARS** parameter must be between 2 and 10. Only one line per observation is allowed. As data are entered, PLT checks for typographical errors. When an error is found, PLT asks the user to re-enter the whole line.

Once data entry is complete, type CTRL Z, or the END keyword. PLT treats the newly-created file FILENAME as if a FILe command had been issued.

```
Example: PLT> ENT JUNK.DAT 3
         ENTER DATA. CTRL Z OR E: END / ENTRER DONNEES...
         ENT> 1 2 3
         ENT> 3 4 5
         ENT> END
         NUMBER OF OBSERVATIONS ENTERED: 2
         PLT> FIL
           FILE: JUNK.DAT
  NB OF COLUMNS: 3
    NB SELECTED: 3
     NO FOR PLT:  1  2  3
     NO IN FILE:  1  2  3
```

The MISsing Command - Syntax: MIS [V (SKIP)]

Missing values can be present in any column of an input file (e.g., Table 1) as long as they are identified by the same value (or any value lower than V) within the file.

By default, PLT draws lines connecting non-missing values (interpolation) when a line type other than 0 is specified. The **SKIP** instruction (any character) is used to cause breaks in lines when missing values occur. This feature is most useful in drawing broken line segments from a single set of X-Y coordinates. It permits drawing of complex objects by causing pen-up moves.

When all parameters are omitted, PLT prints the current missing value indicator which is set by default at -999999.

```
Example: PLT> MIS -1
         PLT> MIS
         CURRENT MISSING VALUE INDICATOR: -1.0000
```

The VARiable Command - Syntax: VAR { AXIS [ID (NOMNMX)] }

The VARiable command defines the variables forming the curves or histograms of the graph and thus is one of the most important PLT commands (Figures 1 and 2). Up to 8 X and Y variables can be defined by repeating the VARiable command. The **AXIS** parameter is either X (abscissa), Y or Y1 (left-hand ordinate), or Y2 (right-hand ordinate). The **ID** parameter refers to the I.D. number of a variable (see FILe command).

PLT evaluates the corresponding axis minimum/maximum automatically with each valid use of the VARiable command unless a character (the **NOMNMX** instruction) is typed after the ID parameter.

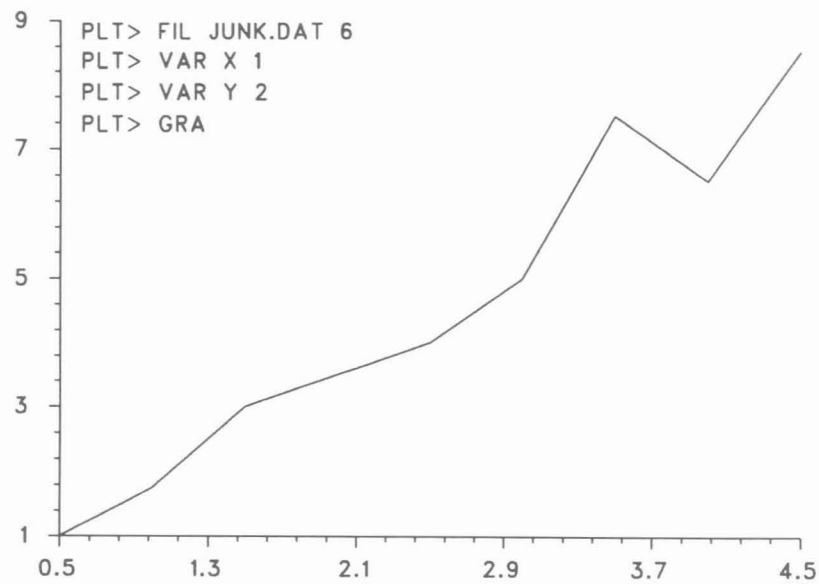


Figure 1. Use of FILe and VARiable commands.

An X variable must always be defined BEFORE the corresponding Y variable (Figure 2). If a VAR X command is not issued prior to a VAR Y command, the previously defined X is assumed (or 1, by default). This allows for a single use of the VAR X command when the graph includes only one abscissa.

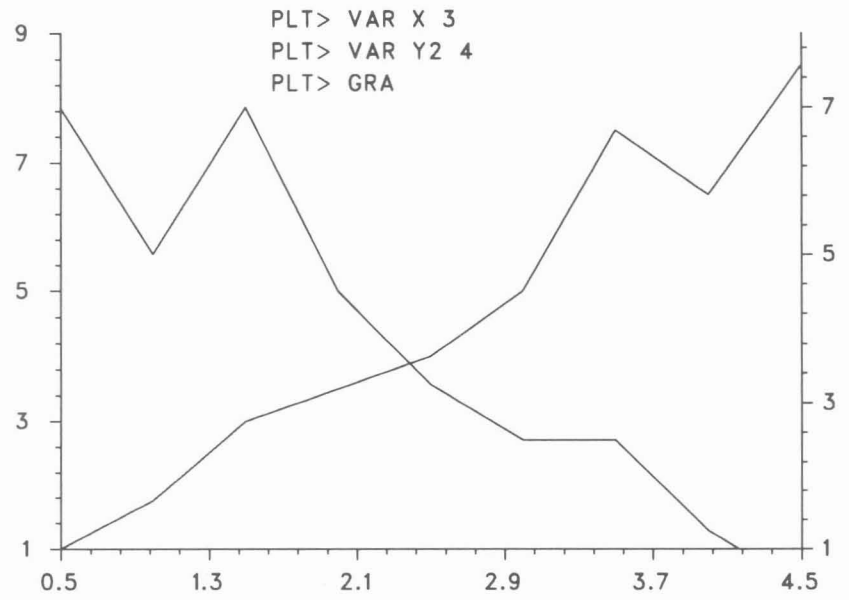


Figure 2. Use of the VARiable command.

Each Y-variable definition is accumulated in a separate plot-queue for each Y-axis. When the ID and NOMNMX parameters are omitted, the appropriate plot-queue is erased (Figure 3) and new Y-variables for that axis can be defined (starting a new plot queue). Individual Y-variables may be removed from either plot queue (Y1 or Y2) by issuing a VAR command with the ID number of the variable to be removed with a negative sign.

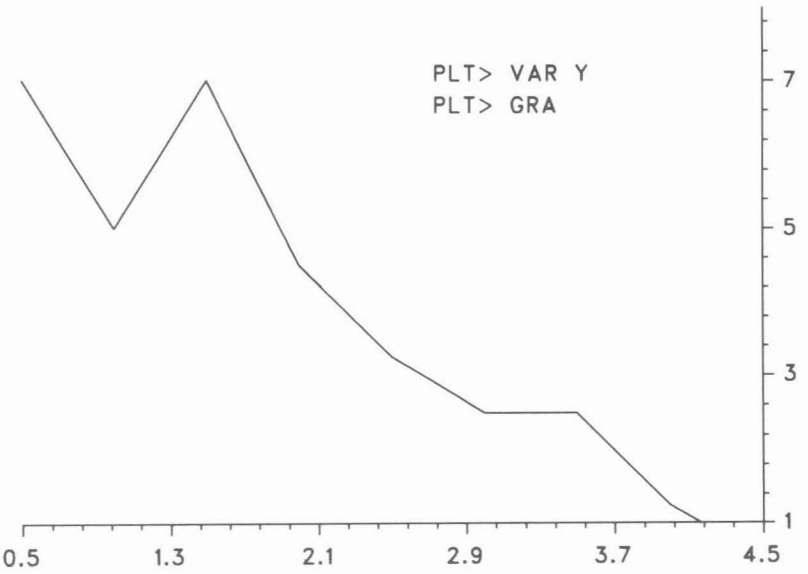


Figure 3. Use of the VARiable command.

When all parameters are omitted, PLT prints a complete description of the curves currently defined: X and Y variable I.D.s, the appropriate Y-axis, precision intervals defined, line types and symbol types (see the PREcision, LINe and SYMbol commands later).

PLT refers to the curves in the plot queues by the order number of their definition, taking plot-queue erasals into consideration. This number is used in assigning line and symbol types, etc.

Examples: (see also Figures 1-3)

```
PLT> VAR X 1
PLT> VAR Y 2
PLT> VAR Y2 3
PLT> VAR X 4
PLT> VAR Y 5
PLT> VAR
      CURVE:   1   2   3
  VARIABLE X:   1   1   4
  VARIABLE Y:   2   3   5
AXIS(1 OR 2):   1   2   1
   PRECISION:   0   0   0
        LINE:   5   5   5
      SYMBOL:
     SHADING:

PLT> VAR Y -2
PLT> VAR
      CURVE:   1   2
  VARIABLE X:   1   4
  VARIABLE Y:   3   5
AXIS(1 OR 2):   2   1
   PRECISION:   0   0
        LINE:   5   5
      SYMBOL:
     SHADING:

PLT> VAR Y
PLT> VAR
      CURVE:   1
  VARIABLE X:   1
  VARIABLE Y:   3
AXIS(1 OR 2):   2
   PRECISION:   0
        LINE:   5
      SYMBOL:
     SHADING:
```

The PREcision Command - Syntax: PRE CURV [ID]

The PREcision command instructs PLT to use the values in variable **ID** as precision intervals (e.g., standard error, 95% confidence interval) to be applied to the values of Y in curve **CURV**. For

example, ID could contain the standard error of an other variable. A vertical line of twice the value of variable ID is drawn, centered around the ordinate of points in CURV (Figure 4).

NOTE: Axis minimum/maximum are not adjusted for these intervals.

When parameter ID is omitted, PLT forgets whatever precision-interval variable was defined for curve CURV.

Example: (see also Figure 4)

```
PLT> PRE 1 3
PLT> VAR
       CURVE:    1
  VARIABLE X:    1
  VARIABLE Y:    3
AXIS(1 OR 2):    2
   PRECISION:    3
        LINE:    5
      SYMBOL:
    SHADING:
```

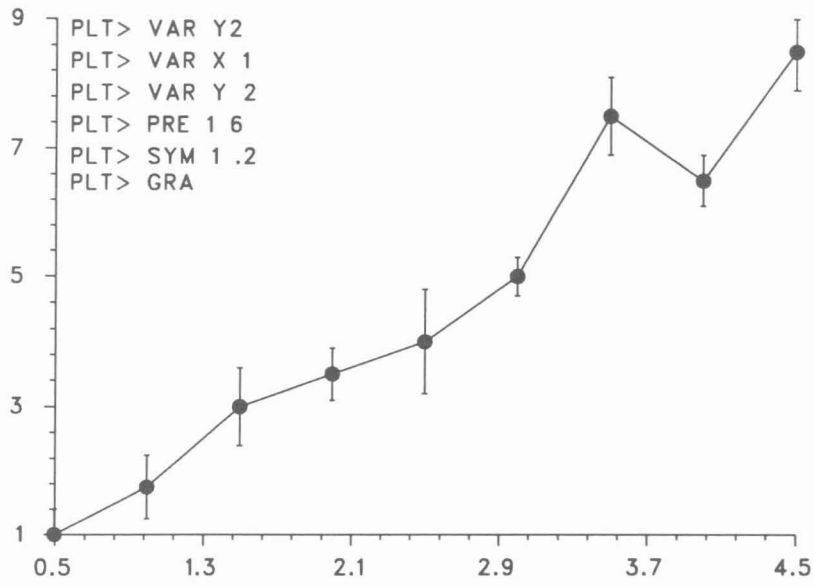


Figure 4. Use of the PREcision command.

The CLAsses Command - Syntax: CLA CURVE [ID (N)]

The CLAsses command associates different symbols to ranges of values of variable **ID** when representing the points of curve **CURVE**. The **N** parameter (between 2 and 20) is optional. It indicates that the user wants PLT to define automatically N even-sized classes within the range of values of variable ID. Missing values are not considered. When this parameter is omitted, PLT requests a number of classes from the user and requires that the upper limit of each class be provided (this is useful when uneven classes are desired). In any case, the symbol to be associated with each

class is defined by the user. Any keyboard character, as well as PLT's 10 special symbols (see the SYMbol command), can be used. In particular, a space is used to skip representation of a particular class.

```
Example: PLT> CLA 3 2
Number of classes (2 to 20; CTRL Z to abort): 4

                        CLASS(E)      LIMIT(E)     SYMBOL(E)
                           1             5             .
                           2            15             +
                           3            21             o
                           4            30            .2
```

Class limits here are user-defined because optional parameter N was omitted from the CLAsses command. Points of curve 3 would bear different symbols depending on values of variable 2.

Omission of the ID and N parameters erases any previously defined classes associated to the values of CURVE.

The SORt Command - Syntax: SOR ID

The SORt command is used to sort the data in PLT in increasing order according to the values of variable **ID**. This command is extremely useful when points are to be connected by a line and the X-coordinates of the curve are not ordered. Missing data within variable ID are not sorted.

NOTE: **PLT does not sort the input file.** Therefore, the SORt command has to be issued every time the data are read from the file (FILe command).

```
Example:  PLT> SOR 1
```

The LOGarithm Command - Syntax: LOG ID

The LOG command is used to take the base-10 logarithm of variable **ID** in PLT memory (Figure 5). **Values in the original data file are not modified.** Missing, zero or negative values are not transformed. The latter two are reset to missing values.

A LOG command cannot be issued twice for the same variable unless an ANTilog command was used to restore values to normal scale, first. Thus the LOG of LOG of ID is not possible, but the LOG of ANTilog of LOG of ID is fine. **Axes are NOT modified by this command.** Use the AXIs command or redefine the graph's curves with the VARiable command, after the transformation. To display a logarithmic scale, use the TICmark command.

```
Example:  PLT> LOG 1 (see Figure 5)
```

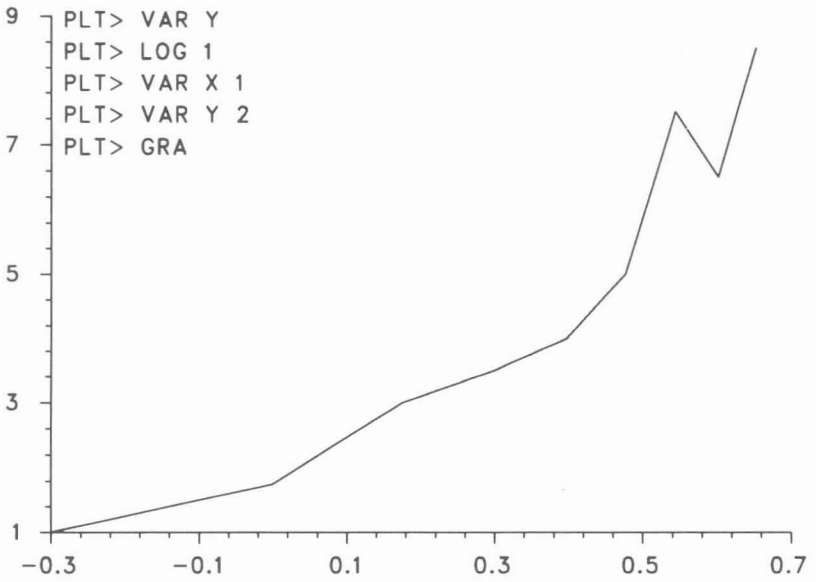


Figure 5. Use of the LOG command.

The ANTilogarithm Command - Syntax: ANT ID

The ANTilog command is used to take the exponent base-10 (10 exp X) of variable **ID** in PLT memory. **Values in the original data file are not modified.** Missing values are not transformed. An ANTilog command cannot be issued twice for the same variable unless a LOG command was used to restore values to normal scale, first. Thus the ANTilog of the ANTilog of ID is not possible, but the ANTilog of LOG of ANTilog of ID is fine.

Axes are NOT modified by this command. Use the AXIs command or redefine the graph's curves with the VARiable command, after the transformation.

Example: PLT> **ANT 1**

The PERcent Command - Syntax: PER ID [DENOM] [SE]

The PERcent command creates a new variable (unless 25 variables already exist) beyond the last variable in PLT memory containing percentages calculated by dividing the values of ID either by the sum of its non-missing values (default) or by the values of denominator variable DENOM (optional). Missing values (or zeros) in DENOM cause missing values in the resulting percent variable.

The SE keyword (any non-numeric character as the last word of the command line) is used to request that standard errors of the percentages be computed and stored in an additional variable.

This form of the command is only valid for counts (frequency variables) because the calculation used is: SE = $100(pq/n)^{1/2}$ where n is either the sum of ID or a value of DENOM.

The I.D. number of new variables created by the PERcent command are reported on the sreen. To delete variables created by the PERcent command, the FILe command must be re-issued. When that is done, however, it is important to remove from the plot queues any reference to the variables erased.

```
Example: PLT> FIL JUNK.DAT 3
         PLT> PER 2 3 SE
         % IN VARIABLE 4 CONTIENT LES %
         STANDARD ERRORS IN VARIABLE 5 CONTIENT LES ECARTS TYPES
```

The REGress command - Syntax: REG [CURVE (LINTYP) (CONFID) (ZERO)]

The REGress command is used to perform regression analysis on the non-missing data of curve **CURVE**. Regression parameters are printed immediately and the regression is displayed on the graph upon execution.

The **LINTYP** parameter is optional. By default, the regression line and associated confidence interval are displayed with a solid line. Any line type described under the LINe command can be specified. When it is specified this parameter (a value between 1 and 5) must always follow immediately the CURVE parameter.

The **CONFID** keyword (first letter is sufficient) is used to request that a 95% confidence sheath be drawn around the regression line. This confidence interval can be applied to predicted mean values (CONFID = C) or to predicted Y values (CONFID = CY). This keyword can be specified anywhere after the CURVE parameter or the LINTYP parameter when the latter is specified.

The **ZERO** keyword (abbreviate to Z if desired) is also optional. It is used to request a zero intercept for the regression line. PLT assumes that variance is independent of the mean, and estimates the line accordingly. Thus, this option should be used with caution. This parameter can be specified anytime after CURVE or LINTYP.

When all parameters are omitted, PLT erases any previously-defined regression lines including those produced by the POLynomial command.

```
Example: (see also Figure 6)
         PLT> REG 1 C
```

```
Y = 1.567 + 2.436 * X   R2 = 0.961  t = 6.751 (df=6)
```

would produce a solid regression line for curve 1, with 95% confidence interval on mean predicted values.

PLT > **REG 2 1 Z**
Y = 0.000 + -0.452 * X R2 = 0.255 t = 1.751 (df=10)

would produce a dotted (style 1) regression line with zero intercept and no confidence sheath for curve 2.

PLT > **REG 2 1 Z CY**
Y = 0.000 + -0.452 * X R2 = 0.255 t = 1.751 (df=10)

would produce the same, but with a 95% confidence sheath on predicted Y values.

PLT > **REG**

would cancel any previously requested regression lines.

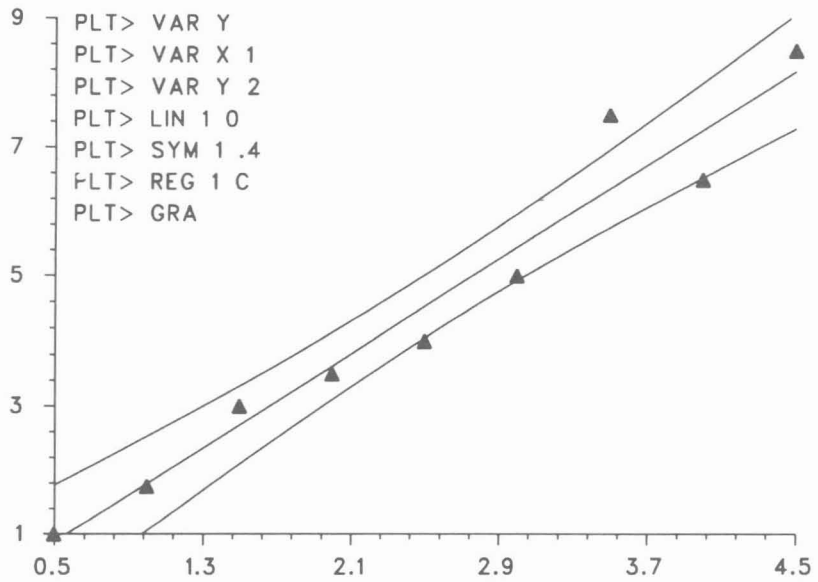


Figure 6. Use of the REGress command.

The POLynomial Command - Syntax: POL CURVE [ORDER (LINE) (ORIGIN)]

The POLynomial command fits a polynomial of **ORDER** 2 to 5 to the non-missing observations of **CURVE** (between 1 and the number of curves in the plot-queues). The **LINE** keyword is optional and is used to specify a line style (see LINe command). By default, a regular solid line is drawn. A bold line is obtained by adding 5 to the desired line style (e.g., 10 is a bold solid line). The **ORIGIN** keyword (any non-numeric character as the last word of the command line) is also optional. It

instructs PLT to fit the polynomial using the transform X' = X - Xmin as independent variable. Parameter values and fit statistics are printed on the screen.

When only the CURVE parameter is given in the command, any previously defined regression for that curve number is deleted including those defined by the REGress command. This command requires that enough PLT memory be free (i.e., 10 000 - observations(variables + ORDER + 1) > 0).

```
Example: PLT> POL 1 3 1 ORIGIN  (See Figure 7)
         Y  = b0 + b1X + b2X² + b3X³
         b0 = 12.34
         b1 =  1.35 ± 0.32
         b2 = -0.63 ± 0.12
         b3 =  0.03 ± 0.01
         R = 0.983   F = 12.5   df = ( 3,  15)
         ORIGIN/ORIGINE X = 150
```

The SMOoth Command - SYNTAX: SMO ID [N]

The SMOoth command applies a running-average filter to the non-missing values of variable **ID** and stores the smoothed values in a variable beyond the last variable in PLT memory (if fewer than 25 variables already exist). Use of this command requires that enough PLT memory be available (i.e., 10 000 - observations(variables+2) > 0).

Normally, smoothing is applied to a sorted time series. The running average is Y'(t)=.25 Y(t-1) + .5 Y(t) + .25 Y(t+1) except at either end where Y'(1)=.75 Y(1) + .25 Y(2) and Y'(n)=.25 Y(n-1) + .75 Y(n). By default, the process is repeated twice. The optional **N** parameter is used to specify any number of iterations between 1 and 38.

The I.D. number of new variables created by the SMOoth command are reported on the screen. To delete variables created by the SMOoth command, the FILe command must be re-issued. When that is done, however, it is important to remove from the plot queues any reference to the variables erased.

```
Example: PLT> FIL JUNK.DAT 2 (See Figure 7)
         PLT> SMO 2 4
       SMOOTHED VALUES IN VARIABLE 3 CONTIENT LES DONNEES FILTREES
```

Graph Structure Commands

Most of the aspects of a graph, including shape and structure, as well as data presentation, can be controlled by a PLT graph-structure command.

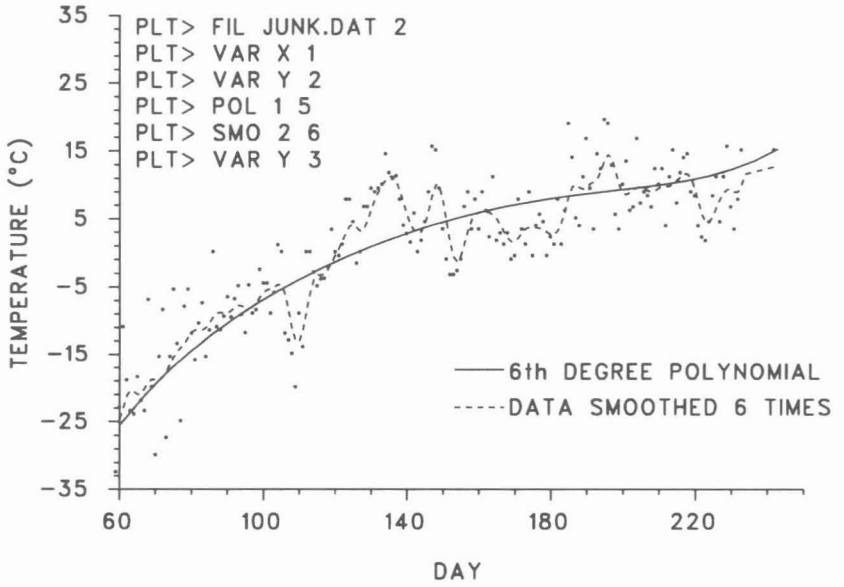


Figure 7. Use of the POLynomial and SMOoth commands.

The SCReen Command - Syntax: SCR [ASPECT MIN MAX] or SCR INVERT [MULTIPLY]

First syntax. The SCReen command controls the SIZE and SHAPE of the graph on the terminal screen. The command is useful in changing the aspect ratio of axes, or in allowing more than one graph to fit on the same page (screen). The **ASPECT** keyword is either H for horizontal or V for vertical. **MIN** and **MAX** are the address limits within which to draw the graph. They are defined in the following paragraphs.

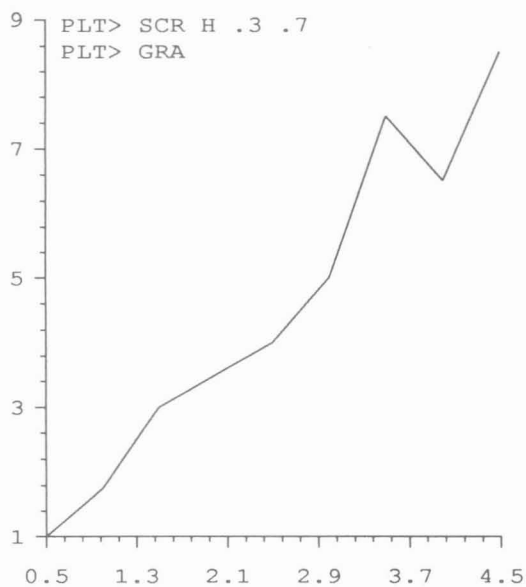
Positions on the screen arbitrarily range from 0 to 1 along both aspects. Points lying outside this range cannot be displayed.

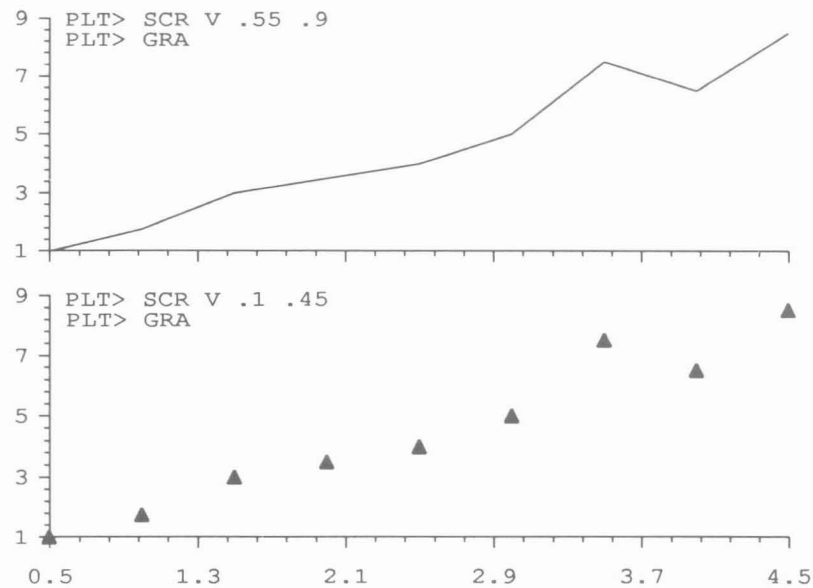
By default, the screen limits of the graph, or the extent of the axes, are [0.2, 0.85] in both directions. These values leave sufficient space on all sides of the graph for ticmark labels and axis titles. When modifying these limits, consider that axis labeling requires approximately 0.1 screen units.

It should be noted that modifications of graph coordinates through the SCReen command cause an equivalent change in the coordinates of features already defined in DRAw Mode (see the DRAw command).

Example: PLT> **SCR H .3 .7** (see Figure 8)

Drawing several graphs per page with PLT requires some practise. For example, a figure with two graphs could use vertical limits [.55, .85] for the upper graph and [.1, .45] for the lower. The result would be two thin graphs on top of each other (Figure 9). It should be noted that each graph must be SAVed separately and that both must be RETreived simultaneously to be drawn on the same screen.





Figures 8, 9. Use of the SCReen command.

When the parameters of the SCReen command are omitted, PLT prints the current screen limits on the terminal:

```
Example: PLT> SCR H .3 .7
         PLT> SCR V .175 .475
         PLT> SCR
         PRESENT LIMITS          X              Y

                      MIN      0.300          0.175
                      MAX      0.700          0.475
```

Second syntax

The SCReen **Invert** command is used to invert the graph on the screen. This produces a printout (via the LASer or DISk command) which is centered and upright on the output page printed landscape. This is a toggle command.

```
Example:  PLT> SCR I
          Inverted image inversée
          PLT> SCR I
          Regular image normale
```

For a graph to occupy as much as possible of the space available on an 8½ x 11 page printed lengthwise, it is first important that the graph be completely drawn within the horizontal range [0,.58]. Then, the SCR I M command is issued to invert and multiply it so that its vertical aspect spans the length of the printed output. Any portion of the graph exceeding .58 of the screen horizontally is clipped.

```
Example:  PLT> SCR I M
          Inverted image inversée
          Any element outside of horizontal [0,.58] will be clipped
```

The AXIs Command - Syntax: AXI [ID MIN MAX]

This command allows the user to choose the minimum (**MIN**) and maximum (**MAX**) values to be displayed on the graph along axis **ID** (either X, Y or Y1, or Y2). When the parameters of the AXIs command are omitted, PLT prints the current axis conditions: minimum, maximum, position, ticmarks, starting value of ticmarks, print format of the ticmark values (see the POSition and TICmark commands later). Data values outside ranges specified by the AXIs command is not displayed and curves are clipped at the axis limits.

```
Example: PLT> AXI X 50 100
         PLT> AXI Y 0 10
         PLT> AXI
              PARAMETER       X          Y          Y2
                   MIN:   50.000      0.000      0.000
                   MAX:  100.000     10.000      1.000
              POSITION:    0.000     50.000     50.000
            TIC SPACING:   10.000      2.000      0.200
            TIC   START:   50.000      0.000      0.000
            TIC  FORMAT:    4.0        3.0        3.1
          AXIS OMISSION:    -1         -1         -1
```

The JULian command - Syntax: JUL [LANGUAGE (LEAPYEAR) (CASE)]

The JULian command produces the printing of monthly ranges underneath Julian dates on the X-axis. Month names are abbreviated to 1 or 3 letters or spelled out completely depending on the range of dates and available screen space. **The LANGUAGE** keyword is either E (English) or F (French). The **LEAPYEAR** keyword (L is enough) specifies a leap year. The **CASE** keyword (C is enough) requests lower-case letters (except the first).

When all parameters are omitted, printing of monthly ranges is cancelled.

```
Example: PLT> JUL E L C
```

The POSition Command - Syntax: POS AXIS LOC

The POSition command is used to change the default position of **AXIS** (X, Y or Y1 left, or Y2 right) on the graph. Normally (by default) the X-axis is positioned at the lower end of the left-hand Y axis (Y1); the Y (or Y1, left-hand ordinate) is placed at the lower end of the X-axis; and the Y2 (right-hand ordinate), when present, is placed at the upper end of the X-axis.

The **LOC** parameter is given in values of the relevant axis and can have any value even if it is outside of the range covered by the perpendicular axis. Vertical **histogram bars** are always drawn in relation to the position of the X-axis, wherever it is. Similarly, horizontal bars follow the Y axis. **Ticmark values** are always printed outside of the range of axes. When a Y axis does not exist (and only a right-hand Y2 axis does), PLT assumes a range of 0 to 1 for the purpose of positioning the X axis. Any time that a VARiable, AXIs, or MISsing command is issued, the default positions of axes are reinstated.

```
Example:  PLT> POS X 5 (see also Figure 10)
```

This would position the X axis right in the middle of the left Y axis.

```
PLT> AXI
     PARAMETER        X          Y          Y2
          MIN:   50.000      0.000      0.000
          MAX:  100.000     10.000      1.000
     POSITION:    5.000     50.000     50.000
  TIC SPACING:   10.000      2.000      0.200
  TIC   START:   50.000      0.000      0.000
  TIC  FORMAT:    4.0        3.0        3.1
AXIS OMISSION:    -1         -1         -1
```

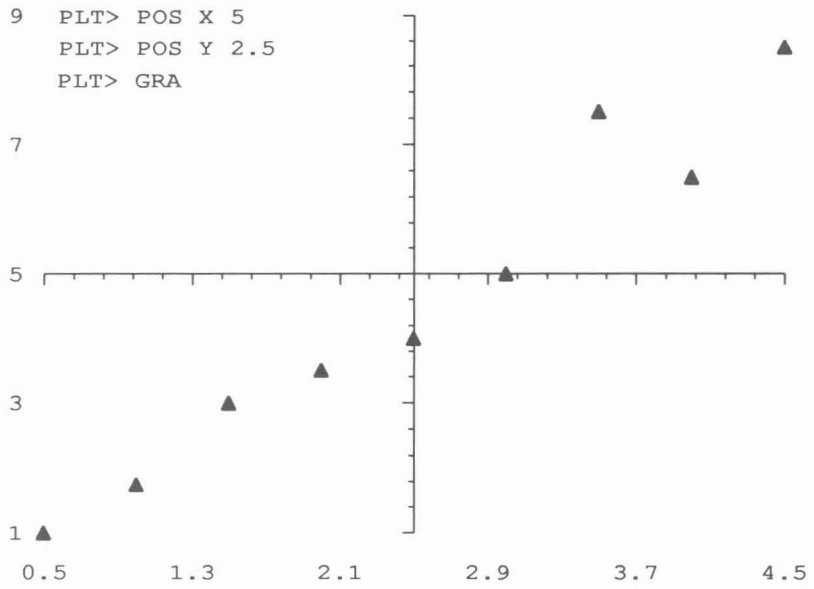


Figure 10. Use of the POSition command.

The OMIt Command - Syntax: OMI AXIS

The OMIt command makes PLT skip the drawing of the specified **AXIS** (X, Y or Y1 left, or Y2 right), except for an axis title if one was defined (see TITle command). This affects only the display of the axis. To resume drawing of the axis, re-issue the OMIt command. PLT reports the omission status of the axes with the AXIs command.

Example: PLT> **OMIT X** (see also Figure 11)

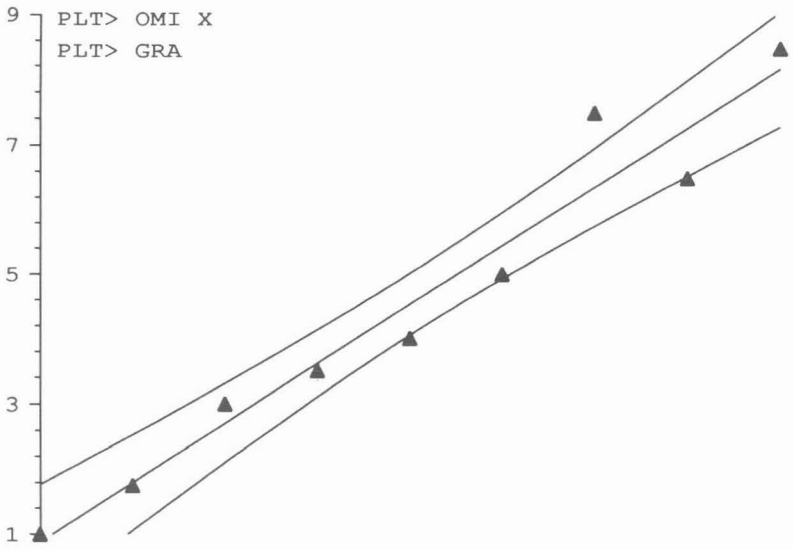


Figure 11. Use of the OMit command.

The TICmark Command - Syntax: TIC AXIS Minor N
or TIC X LOG
or TIC Position
or TIC AXIS START DIST [FORMAT]

The TICmark command provides flexible control of the ticmarks. It has 4 different syntaxes.

The first syntax is used to change the number of minor ticmarks on **AXIS** (X, Y or Y1, or Y2). The Minor keyword indicates that **N** is a number of minor ticmarks (between 0 and 9). **Note that n minor marks divide major ticmarks in n+1 segments.** The default is 4 minor marks. There is a lower limit of 10 pixels between minor ticmarks which PLT does not break unless LOG scale is requested.

```
Example:  PLT> TIC X M 0 (see also Figure 12)
```

The second syntax allows for the representation of **logarithmic** axes with non-linear minor ticmarks and tic values printed as powers of 10 (.1, 1, 10, 100...). Axis limits must lie in the range [-7, 8] (0.0000001 to 10000000). To revert to a standard scale, use another syntax of the TICmarks command, or the AXIs command. Note that only the drawing of the axis is altered with this command. Data are not actually plotted on log scale. To do this, you must use the LOGarithm command. Log-scale axes always come with 9 minor ticmarks.

```
Example:  PLT> TIC X L (see also Figure 12)
```

By default, ticmarks are drawn outside of the axes. The third syntax is used to change the position of the ticmarks relative to the axis: the Position keyword can be one of the following (abbreviated to first letter): Inside, Outside, or Centered.

```
Example: (see also Figure 12)
         PLT> TIC I
         TIC MARKS INSIDE AXES
         PLT> TIC O
         TIC MARKS OUTSIDE AXES
         PLT> TIC C
         TIC MARKS CENTERED ON AXES
```

The fourth syntax is used to define the position of the first ticmark (START), the distance between major ticmarks (**DIST**) (both in units of the corresponding axis) and the printing of tic

values (FORMAT parameter). There is a basic restriction that START+DIST < Maximum of AXIS. By default, PLT places 5 major ticmarks on the axis. When **DIST=0, no ticmarks are drawn** and no tic values are displayed along the axis. Printing formats for the major ticmark values are also decided according to "neatness" rules. The **FORMAT** parameter is useful in changing the automatic printing format for tic values. It is specified in a manner similar to FORTRAN formats for REAL numbers: N.M where N is the number of characters required to print the LARGEST ABSOLUTE value (including space for a negative sign if needed, maximum N=9) and M is the number of decimal places (maximum M=8). When **FORMAT=0** is specified, ticmarks appear in accordance with the START and DIST parameters, but **numerical values are not printed.** This is especially useful when several graphs with identical axes are to be placed on the same page.

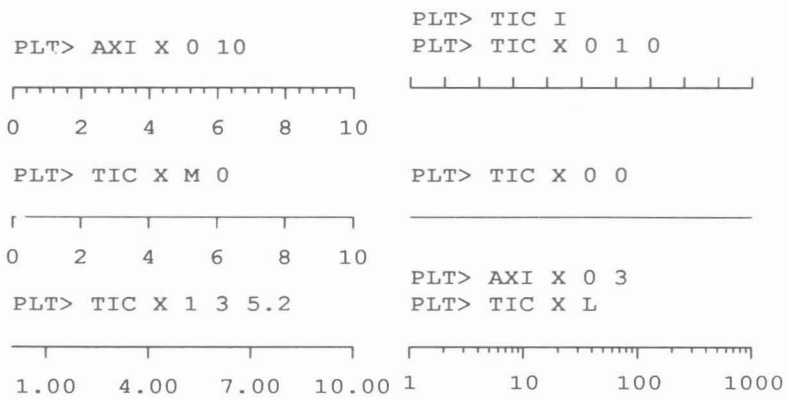


Figure 12. Uses of the TICmarks command.

```
Example: PLT> TIC Y 0 1 4.1  (see also Figure 12)
         PLT> AXI
              PARAMETER       X          Y          Y2
                   MIN:   50.000      0.000      0.000
                   MAX:  100.000     10.000      1.000
              POSITION:    5.000     50.000     50.000
            TIC SPACING:   10.000      1.000      0.200
            TIC   START:   50.000      0.000      0.000
            TIC  FORMAT:    4.0        4.1        3.1
           AXIS OMISSION:   -1         -1         -1
```

would produce ticmarks from zero, in steps of 1, with three significant digits, one period, and one decimal place.

The FRAme Command - Syntax: FRA [Ticmarks]

The FRAme command is used to initiate or suppress (toggle) the drawing of a frame around a graph (Figure 13). After issuing the FRAME command, PLT reports whether a frame or no frame

is in effect. The **Ticmarks** keyword (actually, any character), produces ticmarks on the frame.

```
Example: PLT> FRA    (see also Figure 13)
         FRAME ON
         PLT> FRA
         FRAME OFF
         PLT> FRA T
         FRAME WITH TICMARKS
```

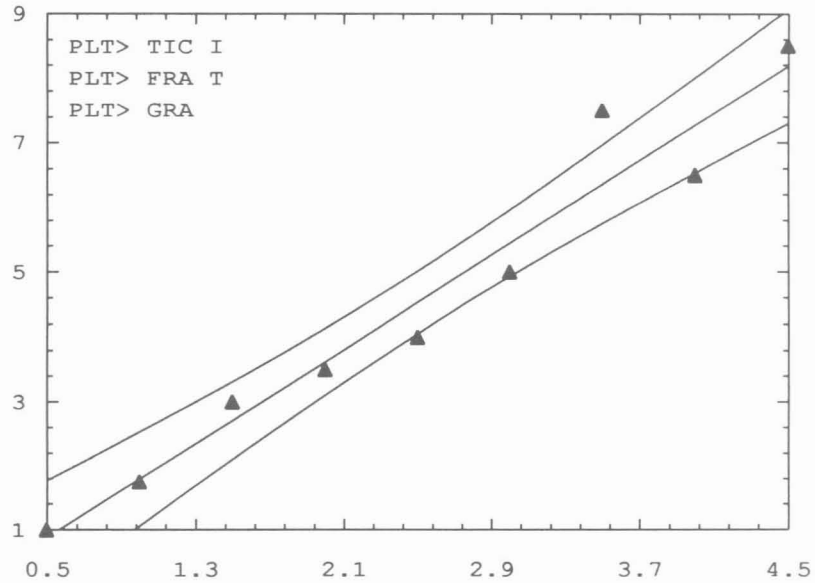


Figure 13. Use of the FRAme command.

The SYMbol Command - Syntax: SYM ID [KEY]

The SYMbol command is used to change the symbol representing data points for curve **ID** in the plot queue. By default, no symbol is displayed. Any printable keyboard character can be specified. Upper case and lower case characters are distinct. In addition, there are 10 SPECIAL characters that produce DRAWN symbols rather than printed characters. The special symbols can be specified by typing a number between **0** and **9** preceded by a decimal point:

Symbol	Empty	Full
Circle	.1	.2
Triangle	.3	.4
Square	.5	.6
Diamond	.7	.8
Star	.9	.0

To change the SIZE of symbols, use the SIZe command.

If the **KEY** parameter is a space or is omitted, any symbol associated to values of curve ID is dropped.

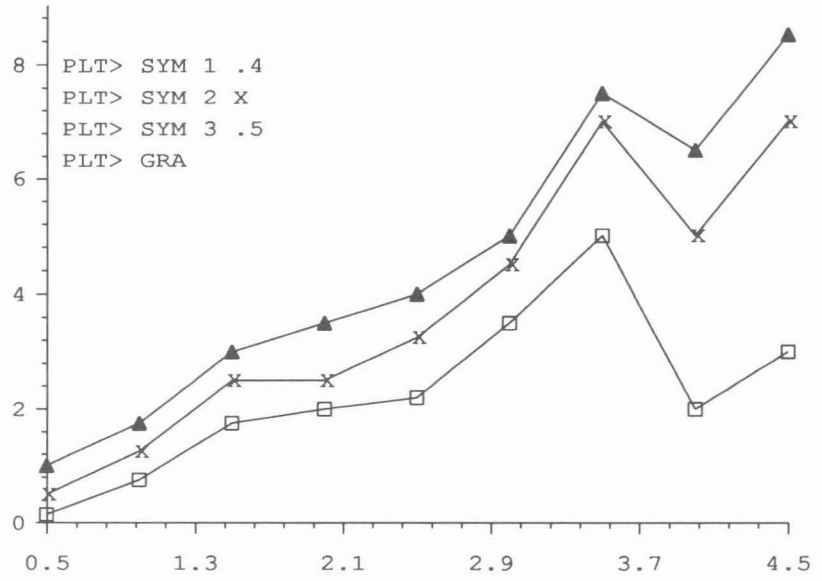


Figure 14. Use of the SYMbol command.

```
Example: PLT> SYM 1 x  (see also Figure 14)
         PLT> SYM 2 .4
         PLT> VAR
                CURVE:   1  2
           VARIABLE X:   1  1
           VARIABLE Y:   2  3
         AXIS(1 OR 2):   1  1
            PRECISION:   0  0
                 LINE:   5  5
               SYMBOL:   x  4
```

The LINe Command - Syntax: LIN ID TYPE

The LINE command is used to change the type of line connecting the points of curve **ID**. By default, PLT draws solid lines for all curves. Select any of the following **types**:

```
 0: no line
 1: dotted line              .............      6: same as 1, bold
 2: dot-dash line            .__.__.__.__       9: same as 2, bold
 3: short dash line          _ _ _ _ _ _        8: same as 3, bold
 4: long dash line           __ __ __ __        9: same as 4, bold
 5: solid line (default)     ___________       10: same as 5, bold
```

```
Example: PLT> LIN 1 1          (see also Figure 15)
         PLT> LIN 2 0
         PLT> VAR
                CURVE:   1  2
           VARIABLE X:   1  1
           VARIABLE Y:   2  3
         AXIS(1 OR 2):   1  1
            PRECISION:   0  0
                 LINE:   1  0
               SYMBOL:   x  4
```

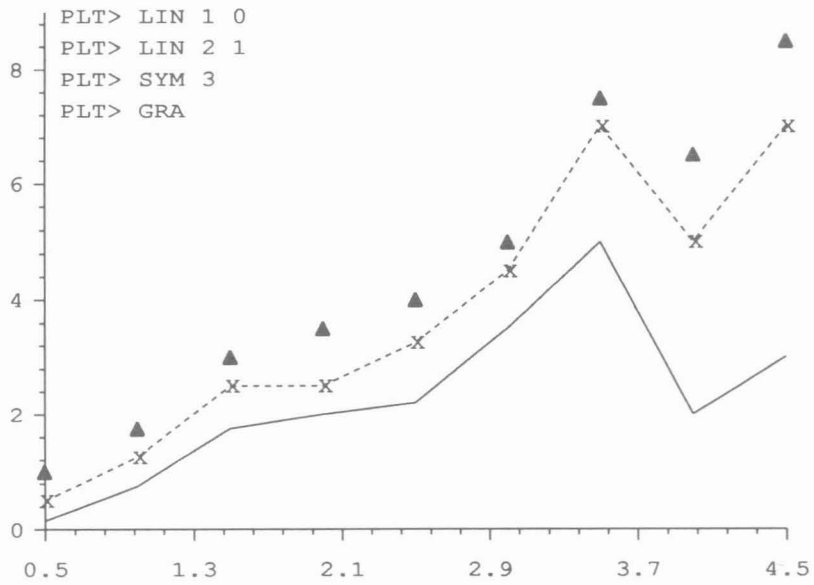


Figure 15. Use of the LINe command.

The SHAde command - Syntax: SHA CURVE [DENSITY (DOTS)]

The SHAde command is used to shade the area under the specified **CURVE** with vertical lines ranging from the curve to the minimum of the Y-axis or to the next-lowest, preceding curve. Shading only occurs for the **EXPOSED** segments of a curve. It is assumed that a curve being shaded is behind all curves drawn before it, but in front of those drawn after it. Right-hand Y-axis curves are drawn last and, thus, are always behind those of the left-hand Y-axis.

The **DENSITY** of shading lines can range from 1 (total shading) to 9 pixels apart. The lines can be replaced by rows of dots with the DOTS keyword (any character).

When all but the CURVE parameter are omitted, any shading defined for the curve is cancelled.

```
Example: PLT> SHA 1 5 D
```

The HIStogram Command - Syntax: HIS CURV [SHAD (WIDTH) (SUPERIMPOSE)]

The HIStogram command is used to define curve **CURV** in the plot-queues as a histogram (Figure 17). By default, the histogram bars are left empty (**SHAD=0**). To change this default, you must specify one of the following shading patterns:

0: empty
1: horizontal lines
2: completely shaded ("black")
3: vertical lines
4: upright grid pattern (Types 1 + 3)
5: oblique lines (right-handed)
6: oblique lines (left-handed)
7: oblique grid pattern (Types 5 + 6)

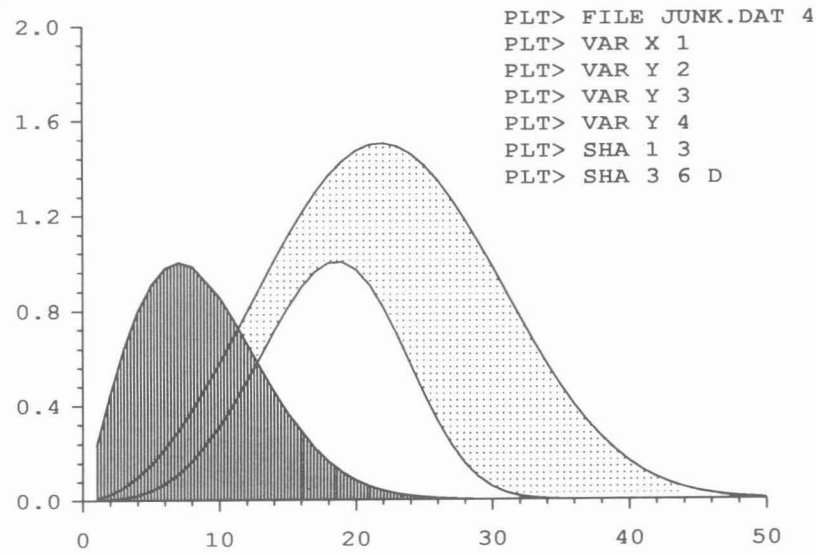


Figure 16. Use of the SHAde command.

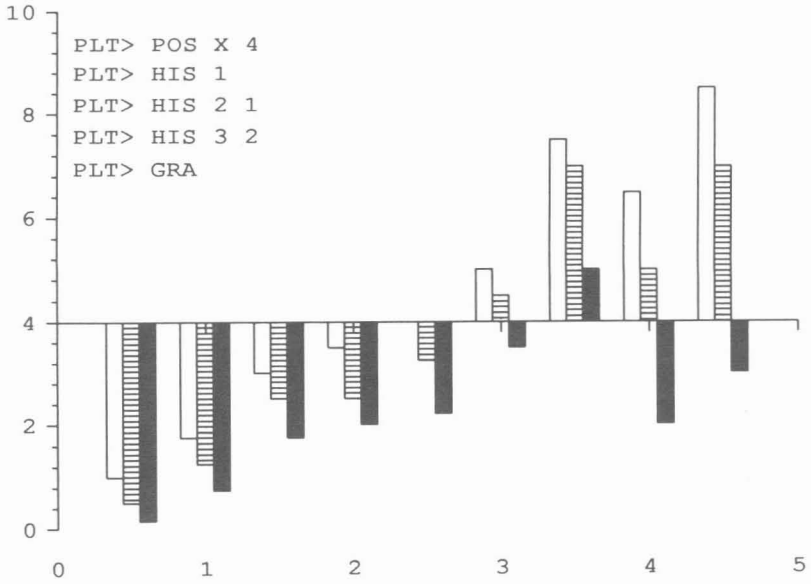


Figure 17. Use of the HIStogram command.

To specify additional parameters, SHAD must be specified. PLT computes an optimum width for histogram bars based on the range of X values, the number of points per histogram, and the

number of histograms (Figure 17). The **WIDTH** parameter (expressed in units of the X-axis) has 2 uses: (1) to alter this default width (the last HIStogram command determines the width of all bars; any width may be specified), and (2) to make PLT draw the histogram bars **HORIZONTALLY by specifying WIDTH<0** in units of the Y1 (left-hand) axis (Figure 18).

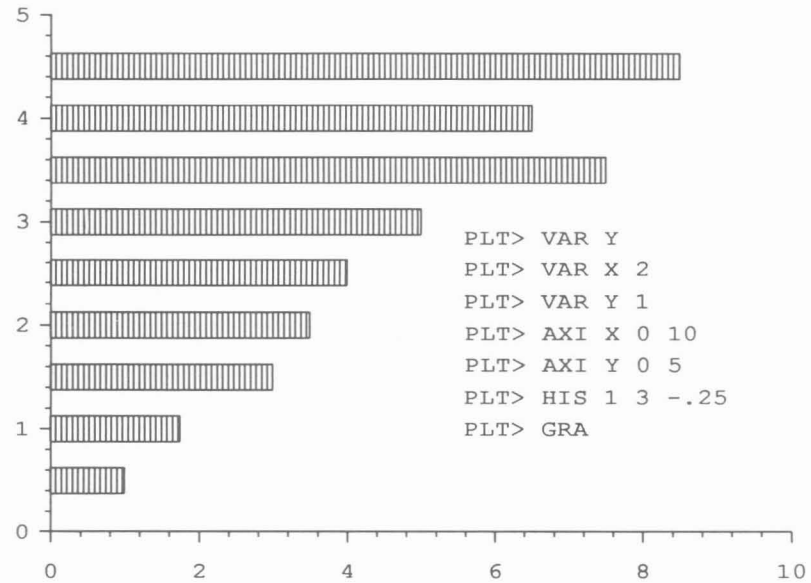


Figure 18. Histogram with horizontal bars.

PLT automatically centers and shifts histogram bars to accommodate as many bars per X value as you have defined histograms in the graph. To make PLT **SUPERIMPOSE** the bars rather than place them side by side, enter any character after the SHAD parameter or the WIDTH parameter if it is specified (Figure 19). Any further use of the HIStogram command without a SUPERIMPOSE keyword returns the graph to bars placed side by side.

The base of vertical bars is drawn on the X axis or at the minimum of the Y1 axis, whichever is larger. Similarly, the base of horizontal bars is drawn on the Y1 axis or at the minimum of the X axis. Thus, bars can point on either side of a centrally-located axis (POSition command. Figure 19).

To go from a histogram back to a line graph, use the LINe command.

Example: `PLT> HIS 1 (see also Figures 17-19)`

would display curve 1 as empty histogram bars.

```
PLT> HIS 2 1 .25 S
```

would display curve 2 as histogram bars filled with horizontal lines of width 0.25 (in X-axis units) for all histograms defined and superimposed bars if more than one curve were defined as histograms.

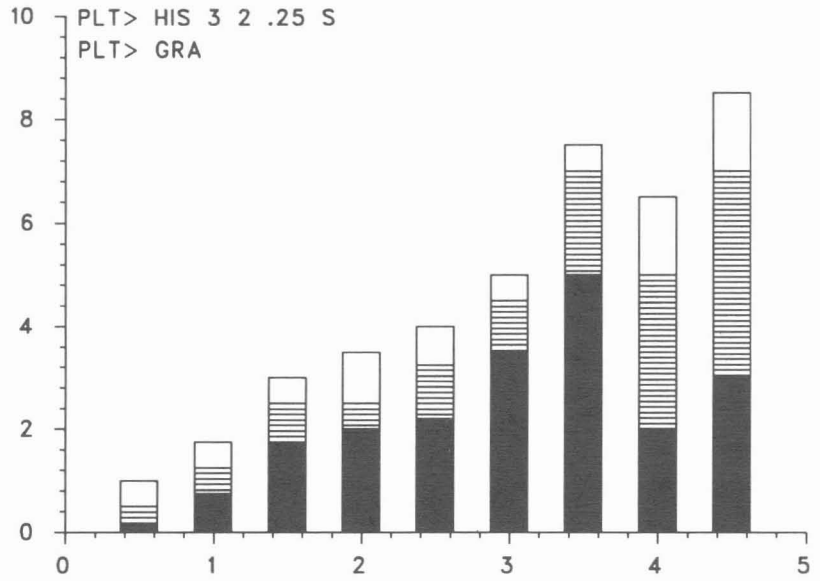


Figure 19. Histogram with superimposed bars.

Lettering Commands

Two types of lettering are offered by PLT: axis titles which are positioned automatically (TITle command) and free text placed with the graphic cursor in DRAw mode or with the PUT command.

The FONt Command - Syntax: FON [ID]

The FONt command changes the character set used by PLT for all printing (ticmark values, axis titles and DRAw-mode text). By default, PLT uses the hardware character set of the device being used (**ID=0**). In addition, there are a number of software fonts (**ID=1 to 9**) that are drawn rather than typed on the output screen or page (Figure 18). Only one font can be used in a given PLT graph. However, up to 6 graphs with independent fonts may be drawn on the same surface (see the SAVe and RETrieve commands). Because of the inherent limitations of the TEKTRONIX 4014 protocol, composed characters are not printable when the hardware font is used.

Each software font contains most keyboard characters plus the common French accents. **FONT 1** also has all composed characters in the DEC Multinational Character Set (Table 2). **FONTS 2 to 9** contain the standard ASCII character set (except \, |, ‘, ^, and ~), plus all French accents and most of the Greek alphabet (Table 3).

Accents (all fonts) and composed characters (font 1) can be obtained in two ways: in axis titles (TITle command), the **<Comp.Char>** key can be used. In draw mode, however, the <Comp.Char>

key does not work and is replaced by the character \ which PLT interprets as a <Comp.Char> unless the next two characters do not constitute a valid composed character. For simplicity, the Greek alphabet in fonts 2-9 is accessed by typing the introducer **\g** followed by the desired letter (Table 3). For example, character μ is obtained by typing **\gm**.

When **ID** is omitted, PLT reports the current font definition and lists the fonts available on the system.

The main advantages of using software fonts include: device-independence in character size and style, italics and angled printing. The major disadvantage, apart from poor reproduction on low-resolution equipment, is slower performance (which can be considerable when large amounts of text are involved).

A number of the available fonts are illustrated in Figure 20.

Figure 20. Examples of software fonts.

```
Example: PLT> FON 1
         PLT-1 CHARACTER SET LOADED / JEU DE CARACTERES CHARGE
```

The TITle Command - Syntax: TIT AXIS [WORDS]

The TITle command is used to place axis titles (**WORDS**, up to 50 characters) centered automatically about the specified **AXIS** (X, Y or Y1 left, or Y2 right) (Figure 21). Titles of ordinate (Y) axes are printed vertically. When a software font is in use, characters are truly rotated on ordinate axes. Otherwise, letters appear one below the other. Centering is done automatically, regardless of the graph's position on the screen. The SIZe command controls the size of letters.

Table 2. Composed characters available with software Font 1 obtained by typing either <Comp Char> (TITle command) or the character \ (DRAw Mode) followed by the two characters indicated

Ä	A"	ª	a_	ï	i"	ó	o'	§	OS	°	^0
ä	a"	Ç	C,	Í	I'	Ô	O^	ß	SS	¹	^1
Á	A'	ç	c,	í	i'	ô	o^	Ü	U"	²	^2
á	a'	¢	c|	Î	I^	Ò	O`	ü	u"	³	^3
Â	A^	Ë	E"	î	i^	ò	o`	Ú	U'	½	12
À	A`	ë	e"	Ì	I`	Œ	OE	ú	u'	¼	14
à	a`	É	E'	ì	i`	Õ	O~	Û	U^	±	+-
Æ	AE	é	e'	£	L-	õ	o~	û	u^	µ	/u
æ	ae	Ê	E^	Ñ	N~	Ø	O/	Ù	U`	«	<<
Ã	A~	ê	e^	ñ	n~	ø	o/	ù	u`	»	>>
ã	a~	È	E`	Ö	O"	¤	OX	Ÿ	Y"	¿	??
Å	A*	è	e`	ö	o"	º	o_	ÿ	y"	¡	!!
å	a*	Ï	I"	Ó	O'	¶	P!	¥	Y-	·	^.

Table 3. Special characters available with Fonts 2 to 9. Type \ followed by one of the combinations below

Special characters & accents

0+	♀	U"	Ü
0>	♂	a`	à
+-	±	a'	á
^0	°	a^	â
^1	¹	a*	å
^2	²	c,	ç
^3	³	e`	è
A`	À	e'	é
A^	Â	e^	ê
A*	Å	e"	ë
C,	Ç	i^	î
E`	È	i"	ï
E'	É	o^	ô
E^	Ê	u`	ù
E"	Ë	u^	û
I^	Î	u"	ü
I"	Ï		
O^	Ô		
U`	Ù		
U^	Û		

Greek alphabet

ga	α	gv	∂
gb	β	gw	ω
gc	ξ	gx	χ
gd	δ	gy	ψ
ge	ε	gz	ζ
gf	φ	GC	Ξ
gh	η	GD	Δ
gi	ι	GF	Φ
gj	ϵ	GG	Γ
gk	κ	GL	Λ
gl	λ	GP	Π
gm	μ	GQ	Θ
gn	ν	GS	Σ
go	o	GU	Υ
gp	π	GV	∇
gq	θ	GW	Ω
gr	ρ	GY	Ψ
gs	σ		
gt	τ		
gu	υ		

When no **WORDS** (or only blanks) are specified, the current axis title is forgotten.

Example: PLT> **TIT X Surface Area of Leaflet (cm²)**

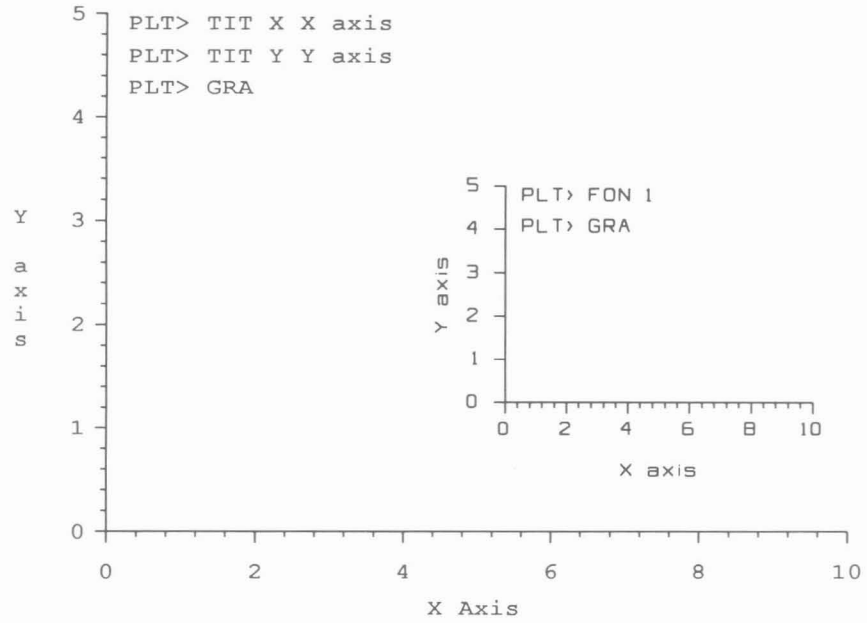


Figure 21. Use of the TITle command.

The SIZe Command - Syntax: SIZe V [ITEM]

The SIZE command is used to change the size of curve symbols (**ITEM = S or omitted**) (see SYMbol command) or lettering (**ITEM = L**) (axis titles and ticmark labels) on terminals which have this capability. Size **V=1** is the smallest; **V=4** is the largest (default) when using hardware characters, but V can go up to 8 when using a software font. This command does not affect text written in DRAw mode.

Example: PLT> **SIZ 2**

would produce smaller than default curve symbols, but would not affect lettering.

PLT> **SIZ 3 L**

would modify lettering without altering curve symbols.

PLT> **SIZ 4 S**

would restore only symbols to default (4) size.

The DRAw Command - Syntax: DRA

PLT's DRAw mode is a visual development tool which can be used on its own or in combination with standard PLT graphs (or other software packages). It is used to position text,

vectors (line drawings), arrows, circles or arcs of circles, special symbols, histogram samples, and rectangles (boxes) anywhere on the screen and to move them about.

The DRAw facility is accessed by issuing the DRAw command before a GRAph or VIEw command. It is cancelled by issuing the command a second time (toggle) without doing the graph. The DRAw command must be issued everytime the facility is to be accessed after a graph is drawn (by the GRAph or VIEw commands).

Once in DRAw mode, the graphic cursor appears on the screen. The cursor can be moved with a mouse, a digitizing tablet, the thumbwheels on TEKTRONIX terminals, or the arrow keys. The travel speed of the latter is accelerated by pressing the **SHIFT** and arrow keys simultaneously.

The terminal bell rings whenever you are attempting an illegal function, option, or action (e.g., maximum number of text strings has been reached, no feature close enough to be erased).

Once in DRAw mode, commands are transmitted to PLT by single key-strokes which are usually **not followed by a carriage return** unless otherwise indicated in this manual. If the graphic cursor (a cross-hair on most terminals) does not reappear after a key-stroke, the terminal is probably improperly set-up (the terminal should transmit a <CR> automatically as graphic input terminator). **Avoid typing ahead of the graphic cursor because that causes transmission errors.**

There are 13 keys which select as many primary DRAw mode functions. Once one has been typed, an array of secondary keys take on a specific meaning (sub-functions). Consult Table 4 for a summary of DRAw mode functions and their sub-functions. The following is a detailed description of each function.

M: Menus (function reminders) can be printed at the top of the screen during a DRAw session by typing **M** in response to the graphic cursor. The menus are helpful for the beginning or occasional user. However, they slow down the response of PLT and can soon become irritating. Type M again to disable this function.

T: Defines and positions up to 40 text strings of up to 50 characters each (Figure 22). PLT rings the bell when the limit number of strings is reached. After a **T** is punched, the graphic cursor disappears and PLT waits at the lower left of the screen for you to type the string and press return. Then the graphic cursor returns to its initial position and you can type any of the following, in any position, on the screen:

1 to 8: change the character size on devices supporting this feature or with sofware fonts on any device (default 4).

A: Printing angle (software characters, fonts 1 to 9). A prompt for the angle (±360°) appears at the lower left-hand corner of the screen. Characters may seem somewhat distorted on low-resolution screens, but results are excellent on devices like laser printers.

H or V: Set printing direction: Vertical or Horizontal (default) for hardware characters (font 0).

I: Italics printing (software characters, fonts 1 to 9). This is a toggle sub-function.

D: Draw string in current position (cursor is at lower left of first letter).

C: Cancel entry and return to DRAW mode.

V: To draw vectors starting at the cursor when **V** is punched and ending anywhere on the screen (2000 vectors maximum). Vectors can be used to draw **OBJECTS** (Figure 22). Move the cursor to the desired position of the vector's other end and type any of the following:

0 to 5: Set line style, as in LINe command. Line style **0** appears dotted in DRAW mode, but is not drawn at all (dark vectors) when DRAw mode is not invoked. Thus, it is possible to define objects with invisible contour lines (Figure 23).

B: Bold line style (applies to all line styles). This is a toggle sub-function.

A: Draws an arrow pointed towards the vector's second end point. This is a toggle sub-function.

D: Draws the vector to current cursor position and exit the **V** function to return to DRAw mode.

V: An uninterrupted series of **V**s draws an **OBJECT**. In PLT, an object is a collection of contiguous vectors whose extremities match exactly. Thus the end point of vector i equals the origin of vector i+1 and so forth. The definition of an object is terminated (cut) when any character other than **V** or one of its sub-functions is typed. Objects which are more or less closed (origin of vector 1 near end point of vector n) can be filled (see **P**, later). To pursue the drawing of an object after an interruption (before the definition of any new vector not part of the object) use the sequence **V** then **I** (see below).

I: Inserting a vector. This function has two uses. First, when drawing of an object (series of V's) has been interrupted, it can be restarted by getting the cursor near the end point of the last vector in the object and typing **V** then **I**. This defines the origin of the next vector as the end point of the last which is the criterion PLT uses to distinguish object limits. The drawing can then be pursued by typing a sequence of Vs. This only works when the last vector defined belongs to the object whose drawing is to be pursued. If this is not the case, any vector defined after the end of the object must be deleted first. The **I** function also allows for the insertion of a segment within an object. This is the case, for example, when one has been deleted by error and de-erasal cannot be reinstituted. Get the cursor near the end of one of the vectors adjacent to the missing segment in the object, then type **V** and **I**. By extension, the I sub-function can be used to insert a vector between two existing vectors. This is invisible until the **R** function (relocation) is used. With repeated insertions and relocations objects can be completed or modified.

C: Cancels the vector and returns to DRAW mode.

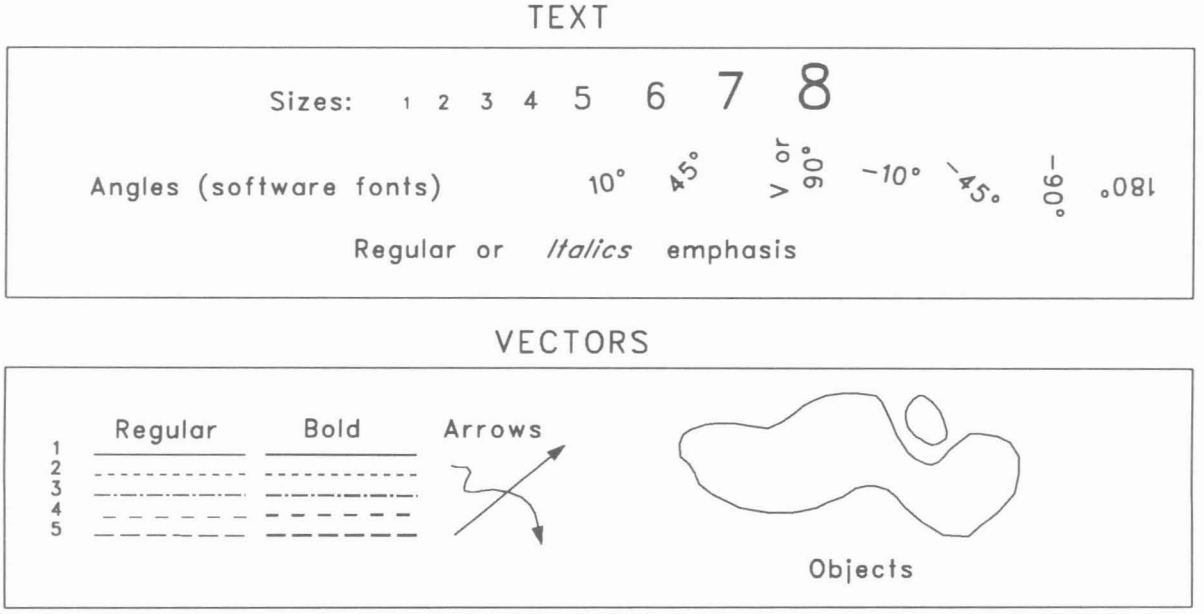


Figure 22. Examples of text and vectors.

C: Draws complete or incomplete circles (arcs). The center of the circle is indicated by a temporary dot where the C was typed. Move to the outside edge and type **D** to get immediate drawing of a 360° circle or type **A** to define the beginning point of an arc. Move the cursor counterclockwise to the approximate position of the arc's end-point and type **D** (the position of the **D** defines the circumscribed angle, rather than the actual end point of the arc). Remember that arcs are always drawn counterclockwise.

Circles and arcs are collections of uninterrupted vectors and as such can form objects which can be filled with the **P** function (Figure 23). Arcs can be completed by vectors to form closed objects by using **V** and **I** near the arc's counterclockwise end-point, as discussed under the **V** function.

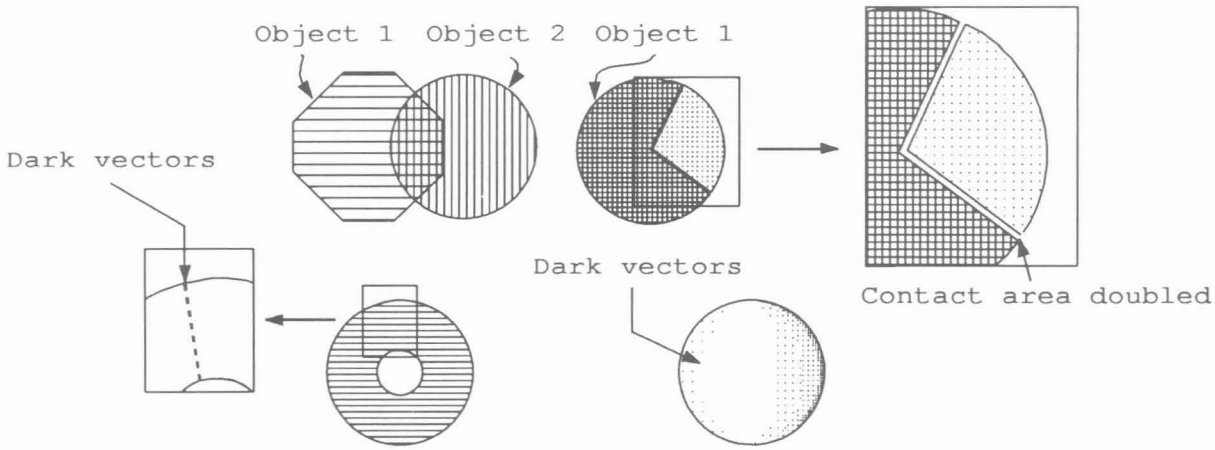


Figure 23. Examples of painted objects.

P: Paint or Measure functions.

Paint - Up to 60 OBJECTS (collections of contiguous vectors) can be painted (filled) with vertical (**V**, default), horizontal (**H**), cross-hatched (**C**) lines, or dotted patterns (**D**) (Figure 23). Filling density (**0** to **9**) is defined last. For terminals with this capability, replace-mode filling allows emptying the inside of an object before painting (which is nice for drawing imbedded objects). Similarly, an object can simply be emptied (density = 0). Painting of objects with dotted patterns can be performed with densities 2-9 only.

To use the **P** function properly, it is useful to understand how PLT fills objects on the screen. Remember that PLT objects are not related to geometric forms that appear on the screen. PLT has no way of knowing what is on the screen at any given time. Rather, PLT defines objects as **series of uninterrupted vectors whose extremities match EXACTLY.** The only way to produce such objects is through a continuous series of **V** instructions (see **V** above). First, PLT finds the nearest vector end-point to the position where **P** is typed. It then searches for the limits of the exactly-matched series which this vector belongs to. THAT becomes the object to be filled. Finally, it computes the periphery of that object in order to fill it. PLT does not consider vectors which are not strictly part of the object, no matter what they look like on the screen. Therefore, to fill two objects which share an edge, BOTH OBJECTS MUST BE COMPLETE, which implies doubling vectors along their common edge (Figure 23).

NOTE: in DRAw mode, an * appears at the paint-definition point to assist in erasing filled areas if the need should arise. The "*" does not appear when the DRAw mode is not invoked.

H, V, C or D: Horizontal (default), Vertical, Cross or Dotted filling

R: Replace (empty before filling). **This feature unfortunately does not work on laser printers.**

0 to 9: Spacing of lines or dots (pixels). **1** is completely full and is not allowed with dotted patterns because it would take too long. Use horizontal or vertical filling for this. **0** erases the inside of the object (so far, this doesn't work on laser printers).

Measure. PLT measures tree area and perimeter of an object (see definition below) when an **M** typed after the **P** function is invoked. The object is filled temporarily and PLT requests a scale vector (entered by typing **V** at each end of the scale). Finally, PLT prompts for the length of the scale (arbitrary units, e.g. km or cm).

S: Symbol definition. Up to 40 symbols can be defined (Figure 22) beyond which point the terminal rings to indicate that the limit has been reached. You can specify the size of the symbol to be drawn (size is always specified before the symbol type) or the symbol itself.

The size is specified by typing a key between ! (shift 1, smallest) and * (shift 8, largest). Size is 4 by default and need not be entered.

The symbol is defined last, (see SYMbol command): **0** to **9** produce a circle, a triangle, a square, a diamond or a star that is empty (odd numbers) or full (even numbers). The symbol is drawn immediately after the type has been specified. **C** cancels and returns to DRAW mode.

H: Histogram sample definition (to draw a legend, for example). Up to 40 samples can be defined (Figure 24). The size of the sample (always square) is specified by entering a key between ! (shift 1, smallest) and (shift 9, largest). The default size is $ (shift 4) and need not be specified. The shading type is defined last by typing a character between 0 and 7 corresponding to histogram shading types (see HIStogram command). The cursor indicates the lower left-hand corner of the sample. The sample is drawn immediately after the shading type is specified (Figure 24). C cancels and returns to DRAW mode.

B: Box definition. Up to 40 rectangles are defined by specifying the position of two opposite corners. The first corner is placed at the point where **B** is typed (a point appears temporarily to remind you of where this corner was located). Move the cursor to the desired position of the opposite corner and press any of the following:

0 to 7: Filling type, as for **H**istogram samples.

D: Draws the box.

C: Cancels if the box is not to be defined (return to DRAw mode) (Figure 24).

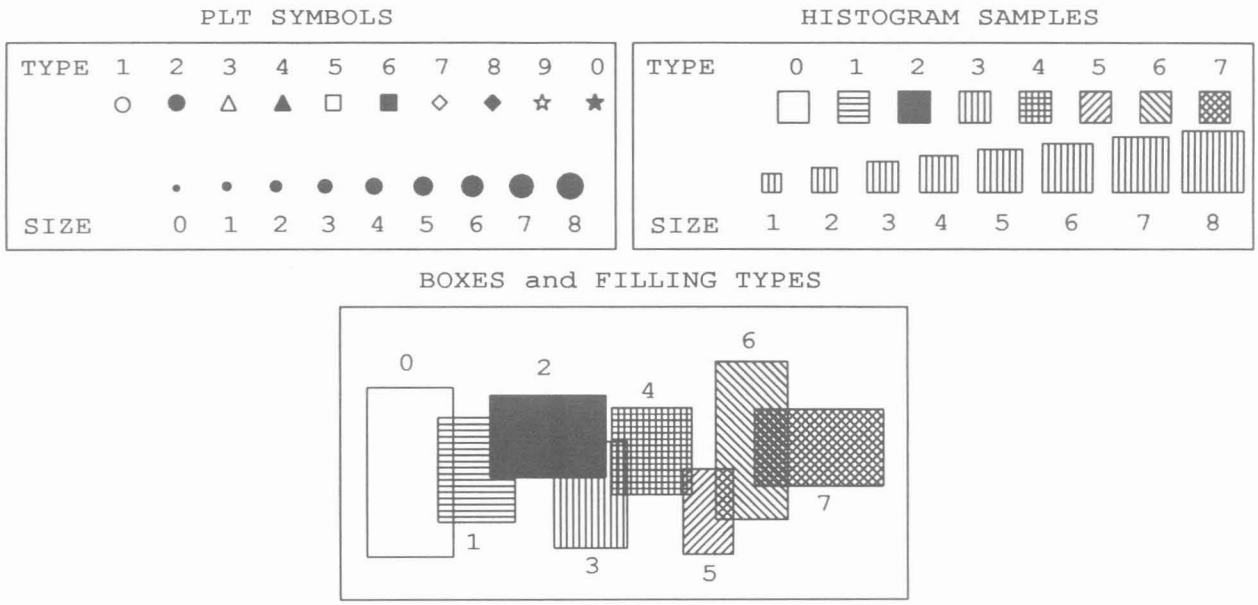


Figure 24. Symbols, histogram samples and boxes.

B and M: If an **M** is typed after a **B**, PLT enters **Block Mode**. A rectangle appears, as defined by points **B** and **M**, and a dot is placed at the center of it. Block mode allows functions to be performed on all DRAw features whose definition points lie inside the rectangle (e.g., the first letter of strings, one or the other end of a vector). Block mode functions are:

E: Erasal, pure and simple. There is no way to reverse the action of this sub-function.

M: Multiply. This allows changes in the proportions and size of the defined area, and the relative positions of everything in it. This is useful for distorting, reducing, magnifying or inverting objects. PLT prompts for two multiplication factors (for X and Y coordinates). For example, a 50% reduction is performed by specifying .5 .5 as multipliers. A mirror image is created with -1 1 or 1 -1. Permutations are infinite.

Multiplication may be preceded or followed by turning (**T**). Execution awaits relocation (**R**) or duplication (**D**) instructions. Note that histogram samples, symbols and text do not change size. Only their definition points are altered.

T: Turn. This allows rotation. A prompt for the angle (±360°) appears in the lower left corner of the screen. Text, boxes, histogram samples and symbols do not rotate, but their definition points do. Vectors, and therefore objets, can be truly rotated. Turning may be preceded or followed by multiplication (**M**). Execution awaits relocation (**R**) or duplication (**D**) instructions.

R: Relocate all elements inside the rectangle. This can follow multiplication and turning, and execution is immediate. The **center of the rectangle** moves to the spot where **R** is typed and all elements follow suit.

D: Duplicate. This makes a copy of all features within the rectangle. The **center of the rectangle** moves to the spot where **R** is typed, and all elements follow suit. Note: limits as to the number of each type of DRAw feature are enforced, so do not be surprised if copying stops at some point.

C: Cancel Block Mode and return to standard Draw mode.

L: Locating points on the screen (digitizing). A request for a file name is made at the lower left corner of the screen. If no output file is to be created for storage of coordinates, press RETURN and proceed with locating. Every time **L** is typed, the coordinates of the cursor are typed on the screen and logged into a previously defined input file. This file is closed when any command other than **L is issued**.

Z: Zoom function. A dot appears on the screen at the position of the cursor. Move the cursor the the desired position of the opposite corner of the region to be zoomed in on then type **Z** for immediate drawing of the region. To return to standard scale, press **Z** again.

E: Erase the DRAw feature closest to the cursor position. Erasing only works in the vicinity of DRAw features. The terminal bell rings if no feature is found close enough to the cursor. On a TEKTRONIX 4010/4014 terminal, this effect is immediate, but will only be apparent when the graph is redrawn.

Get the graphic cursor as close as possible to one extremity of vectors, to the first letter of a string, to the center of symbols, to the lower left corner of histogram samples, to one of the

defined corners of boxes or to the position of the Paint instructions (*s) to be erased so that PLT does not find something closer to erase by error. PLT remains in DRAw mode.

When an error is made, it can be corrected by typing D to reinstitute the last-erased feature. This must be done **immediately** after the incorrect erasal for guaranteed results.

R: Relocate a DRAw feature. As in the **E** keyword above, relocating only works in the immediate vicinity of DRAw features. The bell rings if no feature is found close enough to the cursor. Get as close as possible to one extremity of the vectors, to the first letter of a string, to the center of symbols, to the lower left of histogram samples or to one of the defined corners of boxes to be relocated so that PLT does not find something closer to relocate by error. Paint instructions cannot be relocated (erase and redefine them, instead). Vectors are relocated one end at a time if defined singly or for those at either end of an object. Vectors within objects are moved one VERTEX at a time. Other DRAw features move all at once.

Once you have punched in **R**, move the cursor to the desired position of the other end of the vector and type **R** to draw or **C** to cancel and return to DRAw mode.

F: (Finish) exits DRAw mode and returns to normal PLT execution. The screen is cleared and the PLT> prompt reappears. All DRAw features are kept in memory unless they are specifically erased with the DRAw mode **E** function or the CLEar command.

```
Example: PLT> DRA
         WILL ENTER DRAW MODE AT END OF GRAPHING.
         PLT> DRA
         WILL NOT ENTER DRAW MODE
```

The TERminal Command - Syntax: TER [TYPE]

The TERminal command is used to set the terminal type, so that the system, particularly the DRAw facility, can take advantage of terminal-specific characteristics such as selective erasing or high resolution. The terminal types available may vary from site to site and time to time. However, PLT lists the current setting as well as the list of different terminal types selectable on the current version of PLT when the TERminal command is issued without a TYPE parameter. The **TYPE** parameter, if given, cannot be abbreviated.

If strange characters appear on the screen while using the Erase or Relocate subcommands of DRAw, return to PLT> and issue the TERminal command.

```
Example: PLT> TER
         CHOOSE AMONG / CHOISIR PARMIS
         TEK4010 TEK4014 TEK4015 HDS
         VT100   VT240   PT100G  CONCEPT
         TERMINAL TYPE SET TO: PT100G
         PLT> TER VT240
```

Table 4. Summary of **DRAw** mode functions. Note that **C** cancels most functions unless otherwise indicated.

Functions	Sub-functions
M: Menus	M: Discontinue (not C)
T: Text	1-8: Character size
	H,V: Hardware string orientation)
	A: Software string orientation
	I: Italics (software fonts)
	D: Draw string
V: Vectors	0-5: Line type (**B**: Bold)
	B: Bold lines
	A: Arrow
	D: Draw vector and exit function
	V: Draw vector and start new one
C: Circles	D: Draw (360°)
	A: Arc (**D** defines angle)
P: Paint	V: Vertical lines (default)
	H: Horizontal lines
	C: Crossed lines (no cancel)
	D: Dotted pattern
	R: Replace (erase then fill)
	0-9: Line or dot spacing (pixels)
P then M: Measure	V: Scale vector
S: Symbols	!-*: Size
	0-9: Type
H: Histograms	!-*: Size
	1-7: Type
B: Boxes	0-7: Filling type
	D: Second corner
B then M: Block Mode	E: Erase
	M: Multiply
	T: Turn
	R: Relocate
	D: Duplicate
L: Locating	Name: file spec
Z: Zoom	Z: Second corner of area or exit
E: Erase	D: De-erase (not **C**)
R: Relocate	R or D: Execute
F: Finished	

The PUT Command - Syntax: PUT ITEM X Y [parameters]

The PUT command is used to position DRAw-mode features (text, vectors, symbols, histogram samples, boxes, circles and paint-definition points) on a graph without having to access DRAw-mode. This is especially useful when PLT is used in conjunction with command files. The syntax varies with the **ITEM** being positioned.

X and Y coordinates are expressed in units of the X and left-hand Y axes of the graph (by default, from 0 to 1).

-PUT TEXT(i) X Y SIZE ANGLE STRING

Positions a **STRING** of up to 50 characters starting at X Y. The **SIZE** parameter is the same as described in the SIZe command (1 is smallest, 8 is largest, 4 is default). The **ANGLE** parameter (±360°) must always be specified but is represented only when a software font is in use (see FONt command). The i (TEXTI or TI instead of TEXT or T) option produces italics (only for software fonts).

```
Example: PLT> PUT T .5 .65 2 0  Small letters, printed horizontally
         PLT> PUT TI .5 .5  6 90 Large italics, printed vertically
```

-PUT VECTOR X1 Y1 X2 Y2 [LINE] [ARROW]

Positions a vector starting at **X1 Y1** and ending at **X2 Y2**. By default, a solid line is represented. The **LINE** parameter is used to request another line style (see LINe command). A bold line is obtained by adding 5 to the desired line style (e.g., 10 is a bold solid line). The **ARROW** keyword (any non-numeric character appearing as the last word of the command) produces an arrow pointing to X2, Y2.

```
Example: PLT> PUT V .2 .2  .7 .6  10 A     Arrow, bold solid line
```

-PUT SYMBOL X Y TYPE [SIZE]

Places a symbol at **X Y**. The **TYPE** must correspond to one of those described in the SYMbol command (from 0 to 9). The **SIZE** parameter is optional: 4 by default, it can be between 1 (smallest) and 8 (largest).

```
Example: PLT> PUT S .5 .5 2 1      Small, full circle
```

-PUT HISTOGRAM X Y TYPE [SIZE]

Places a histogram sample at **X Y**. The **TYPE** must be one of those described in the HIStogram command. **SIZE** is an optional parameter: 4 by default, it can be between 1 (smallest) and 8 (largest).

```
Example: PLT> PUT H .3 .3 5 6     large histogram with oblique filling
```

-PUT BOX X1 Y1 X2 Y2 [FILLING]

Places a box with its lower left-hand corner at **X1 Y1** and the opposite corner at **X2 Y2**. With the optional **FILLING** parameter (a value from 0 to 7), the box can be filled with one of the patterns described in the HIStogram command.

```
Example: PLT> PUT B .2 .2  .7 .7 7   Oblique-grid rectangle
```

-PUT CIRCLE X1 Y1 X2 Y2 [LINE (ANGLE)]

Draws a circle with its center at **X1 Y1**, its radius extending to **X2 Y2**. By default, a solid-line, 360° circle is drawn. The circle drawn forms an object as defined in the DRAw command. To obtain another line style, the optional **LINE** parameter is used (see LINe command). A bold line is obtained by adding 5 to the desired line style (e.g., 10 is a bold solid line). The **ANGLE** parameter (±360°) is optional and is specified after the LINE parameter to request an arc of a circle, starting at X2 Y2 and going counterclockwise for ANGLE degrees.

```
Example: PLT> PUT C .5 .5  .75 .5  10 180   Bold, solid line half circle
```

-PUT PAINT X Y DENSITY [STYLE]

Positions a paint-definition point (see DRAw command) at X Y. This causes the closest (if any) collection of vectors forming an object (in the sense of PLT) to be filled with a pattern. The optional **DENSITY** parameter specifies the distance (in pixels) between filling lines or dots. The **STYLE** parameter is used to request a filling pattern other than the default vertical lines. It can be H (horizontal lines), C (crossed lines) or D (dots).

```
Example: PLT> PUT P .5 .5 3 D     dots, 3 pixels apart
```

-PUT NOTHING X Y

Erases the DRAw-mode feature with its definition point nearest to **X Y**, if one is near enough.

```
Example: PLT> PUT N .5 .5
```

Program Execution Commands

The following commands do not act on the graph but give instructions to PLT as to what to do.

The GRAph Command - Syntax: GRAph

The GRAph command instructs PLT to produce the graph on the terminal screen. It can be issued repeatedly at any stage to display the graph as it is developed. Once the graph has been drawn, the user must press return to continue. The screen is cleared following a carriage return.

```
Example:  PLT> GRA
```

The PENplot Command - Syntax: PENplot

This command is used to produce copies of the graph on the TEKTRONIX 4662 penplotter or compatible devices. When the plotter is not available (e.g., in use by someone else), PLT issues a warning and does not execute the command.

Execution of the PENplot command is immediate and cannot be interrupted.

```
Example:  PLT> PEN
```

The DISk command - Syntax: DISk FILENAME [LASER]

This command causes PLT output to be stored on disk in sequential file **FILENAME** in the user's directory. The output is in the form of ASCII control or escape sequences. This file can then be output to any suitable device, including the terminal screen (e.g., using the DCL TYPE or PRINT commands). A file type of .TEK is used by default.

The optional **LASER** keyword (any character) indicates that the file is destined to be printed on a VT240-compatible high-resolution laser printer. This form of the DISk command, or better still, the LASer command, should be used to format output for such devices.

```
Example:  PLT> DIS JUNK.TEK L
```

The LASer Command - Syntax: LAS FILENAME [APPEND]

This command produces output suitable for high-resolution, VT240-compatible devices, such as LN03 laser printers. The output goes in the sequential file **FILENAME** which can then be printed on the device (DCL PRINT command). A file type of .TEK is used by default. This command is in all ways equivalent to the DISk command with L keyword. The **APPEND** keyword (any character)

is used to request that the graphics code generated be appended to the end of file FILENAME if it exists. If it does not, a new file is created. This allows what is produced in DRAw mode to be appended to an image produced by another program such as SAS (see the VIEw command).

Example: PLT> **LAS JUNK.TEK**

The SAVe Command - Syntax: SAVe [FILENAME]

This command writes into file **FILENAME** all the parameters needed to reproduce the graph in its present form at some later time (during the same or another PLT session). A file type of .PLT is used by default.

NOTE: The data for the graph are NOT saved by a PLT SAVe command. Rather, the original data file for the graph should be kept unaltered. The SAVe command is not equivalent to the DISk or LASer commands. SAVing means preserve the parameters which PLT uses to compose a graph and return to the state it was in at the time of the SAVe command. PLT cannot modify files produced by the DISk or LASer commands. The SAVe command, in conjunction with the RETrieve command (below) are used to produce several graphs on a single screen (or page).

If the FILENAME parameter is omitted, PLT saves the data under the last filename given in the most recent RETrieve command.

Example: PLT> **SAV JUNK.PLT**

The RETrieve Command - Syntax: RET FILENAME or RET N

This command is used to retrieve parameters necessary to redo graphs previously saved by the PLT SAVe command in file **FILENAME**. The default file type is .PLT. One file or N files can be retrieved.

If several graphs are to be retreived at once (**N=2** to **6**), use the second syntax. In response, PLT will prompt you for N file names. PLT will retrieve and draw (DRAw, PENplot, DISk or LASer commands) all N graphs, one after the other, without clearing the screen or stopping.

When the last graph is finished, only the parameters of the LAST graph drawn remain in memory. If changes are made to the last graph in a multiple figure (through DRAw or otherwise), they **must be SAVed before the next GRAph, DISk, LASer or PENplot command**. Otherwise they will not be retained. It is not necessary to issue a new RETrieve command until a new series of graphs is desired.

The DRAw mode can be accessed once a group of graphs is prepared by issuing the DRAw command prior to the GRAPH command. If added features are part of the last graph of the series, one should be preserved by the SAVe command before a GRAph, DISk, LASer or PENplotter command is issued again.

```
Example: PLT> RET 3
         NAME OF PARAMETER FILE 1: JUNK.PL1
         NAME OF PARAMETER FILE 2: JUNK.PL2
         NAME OF PARAMETER FILE 3: JUNK.PL3
```

The VIEw Command - Syntax: VIEw FILENAME [TEK]

This command displays file **FILENAME** on the terminal screen, for convenient viewing without having to leave PLT (see also the SYStem command).

The optional **TEK** keyword (any character) indicates that FILENAME containts TEKTRONIX protocole graphic instructions produced by PLT or other software such as a SAS (.GSF files). This allows display of a graph from disk. It is also possible to add elements on top of such an image by issuing a DRAw command before viewing a TEK or GSF file. Then, to incorporate added features into the initial image on disk, a LASER command can be issued to store new graphic instructions at the end of the viewed file.

```
Example: PLT> VIE JUNK.DAT
         File opened: JUNK.DAT
         1 2 3
         3 4 5
         7 8 9
```

The CLEar Command - Syntax: CLEar

This command is equivalent to starting PLT anew. Initial values are restored, and the plot queues, line and symbol types are obliterated.

```
Example:  PLT> CLE
```

The HELp Command - Syntax: HELp [COMmand]

This facility provides most of the information in this manual, command-by-command, on the screen. If **COMmand** is not a valid PLT command, the HELp command will not be executed. If the COMmand parameter is omitted, the HELp facility will list the PLT commands available. The HELp facility is a subprogram of PLT and the prompt HELP> will appear at the end of a HELp printout, awaiting another COMmand parameter. To exit the HELp facility, simply press return.

```
Example:  PLT> HEL FIL
```

The SYStem command - Syntax: SYS [any DCL command string]

The SYStem command gives access, from PLT, to the VAX/VMS operating system. When a **DCL command string** is given with the SYStem command VMS executes that command and then returns control to PLT.

```
Example:  PLT> SYS EDIT JUNK.DAT
```

allows the users to edit file JUNK.DAT without having to leave PLT.

When the SYStem command is issued alone, the VMS $ prompt appears and any number of DCL commands can be executed. To return to PLT, the user must issue a DCL LOGOUT command.

```
Example:  PLT> SYS
          LOGOUT command returns to PLT / La commande LOGOUT revient à PLT
          $ ... any number of DCL commands ...
          $ LOGOUT
          PLT>
```

The BYE Command - Syntax: BYE

This command terminates PLT and returns control to the operating system of your computer. Synonyms are STOp, END, QUIt, OFF or FINished.

```
Example:  PLT> BYE
```

COMMAND FILES

PLT can read commands from files stored on disk rather than from the terminal's screen. This is very useful when PLT is to be used batch as part of DCL procedures or to perform highly automated or repetitive series of commands. At the onset, it should be clear that command files cannot be nested (one command file cannot invoke another command file).

All PLT commands can be contained in command files, one command per line. However, use of the GRAph command requires interaction with the user, because a carriage return must be sent from the terminal for processing to continue. The DRAw command is screen-oriented and its use in a command file is rather limited. Answers to interactive command prompts (SELect, CLAsses, RETrieve and ENTer) must be entered in the command file, on separate lines, just as would be done if the command were issued from the terminal. For example:

```
ENT JUNK.DAT 3
1 2 3
2 5 6
3 8 7
4 1 0
5 2 6
END
VAR X 1
VAR Y 2
CLA 1 3 3
.2
.4
.6
LAS JUNK.TEK
BYE
```

Command files can be used in two different ways. First, during an interactive PLT session, a command file is executed by typing the file name preceded by "@" (the default type is .COM):

PLT> **@FILE.COM**

Second, a command file can be executed without having to issue an @ instruction by associating the name of the command file (e.g., FILE.COM) with the logical symbol PLT_BATCH, prior to running PLT. When PLT starts its session, it automatically attempts to open and execute the file pointed to by logical symbol PLT_BATCH. Thus:

```
$DEFINE PLT_BATCH File.com
$PLT
```

By extension of this feature, PLT can be used in batch mode, using the SUBMIT command under DCL. In this case, the DCL command file submitted should contain at least the following lines:

```
$SET DEF [appropriate directory]:
$DEFINE PLT_BATCH File.com
$PLT
```

In a batch (SUBMIT) context, the GRAph and DRAw commands should not be issued in a PLT command file.

PLT ON VAX/VMS

Running PLT On VAX/VMS

PLT-compatible equipment and its normal configuration are illustrated in Figure 25. Compatible devices must emulate the TEKTRONIX 4010/4014 terminal (or the VT240, which allows selective erasing among other things). Thus, a PC-type microcomputer running the SMART-TERM/240 software constitutes a PLT-compatible terminal.

A terminal running PLT can use a mouse or graphic tablet to manage the graphic cursor in graphic input (GIN) mode, as long as the terminal itself does the handling of these devices. PLT does not distinguish between mouse, tablet or arrow keys. The terminal sould be set up so as to transmit a <CR> automatically as graphic input terminator.

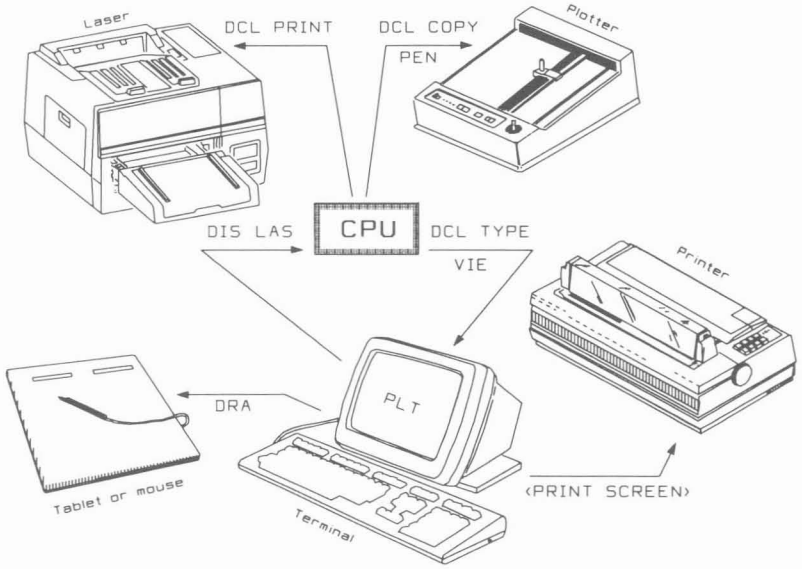


Figure 25. Ideal hardware configuration for PLT use.

A printer connected to the terminal is the simplest way to get a hard copy of a graph, with the terminal's <PRINT SCREEN> key. Graphic instructions (TEKTRONIX 4014 protocole) can also be routed to the disk into a sequential file (DISk and LASer commands). Such a file can be printed on suitable printers (e.g. LN03) via a DCL PRINT command. It can also be displayed on a screen with the DCL TYPE command, provided that the line does not wrap automatically ($SET TERM/NOWRAP) nor block formfeeds ($SET TERM/FORM).

Through the PENplot command, PLT uses a TEKTRONIX 4662 plotter hooked up to a separate line which must allow two-way communication between the task and the plotter.

The user can access several types of disk files. Command files, with a suggested **.COM** file extension (PLT @ command), can contain PLT commands and associated response to prompts.

Parameter-files written by the PLT SAVe command and read by the RETrieve command should have the **.PLT** file type to facilitate their identification. Editing of these files is not recommended as any error in format will render them useless to PLT. That is why their structure is not documented.

Files containing TEKTRONIX-protocole graphic instructions, suggested type **.TEK**, are the only ones which will produce a graph outside of PLT. They are created by the PLT DISk or LASer commands, and can be used by the PLT VIEw command, or the DCL TYPE and PRINT commands. To use the PLT VIEw or DCL TYPE commands to display these files, the system must not generate <CR><LF>'s ($SET TERM/NOWRAP/FORM).

Data files, often with a **.DAT** extension, can be read by the PLT FILe or SELect commands, or written by the PLT ENTer command. Relationships between the user, PLT, and data on disk are illustrated in Figure 26.

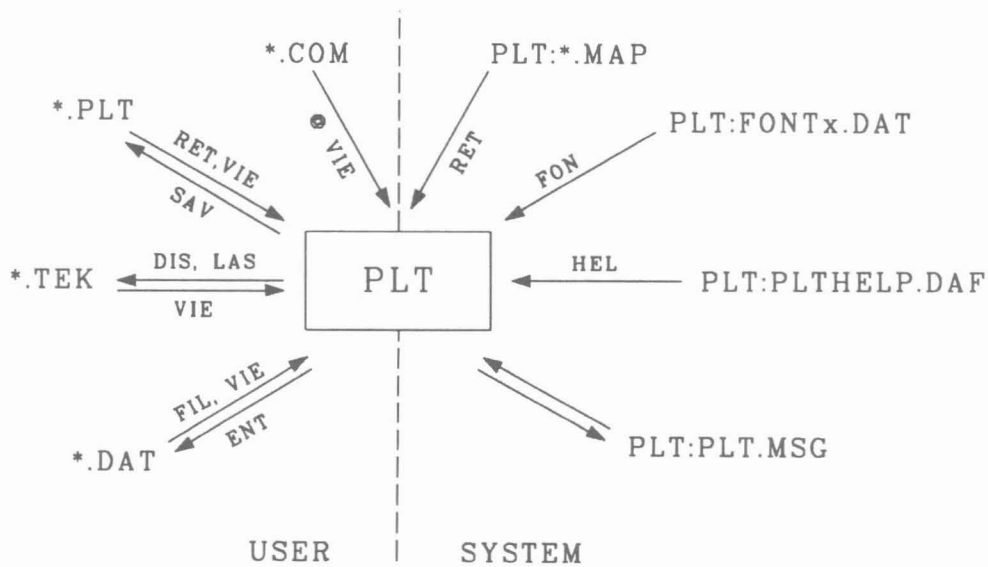


Figure 26. PLT and information on disk.

PLT uses several system files which must be accessible to all users. The file PLT.MSG contains system messages as well as an access-counter (protection WO:RW). The direct-access file PLTHELP.DAF contains PLT's HELp documentation (protection WO:R). PLT's character fonts are contained in sequential files named FONT1.DAT, FONT2.DAT, ..., FONT9.DAT (protection: WO:R). Moreover, PLT drawing files (maps, etc.), all with **.PLT** extensions, should also be known and accessible to users (protection: WO:R).

Installing PLT On VAX/VMS

It is necessary that the person responsible for PLT's implementation designate a directory containing PLT and its files on the system. This directory should be pointed to by logical symbol PLT in the system's logical table. Thus, users can access PLT through the DCL command $RUN PLT:PLT.

Also, it may be convenient to define a system's PLT command (global symbol equivalent to $RUN PLT:PLT) or, better still, leading to the execution of a PLT.COM command file which would in turn execute a $RUN PLT:PLT command after setting-up the terminal ($SET TERM/NOWRAP/FORM).

An implementation suggestion is given below:

1. In the system boot-up sequence, define the PLT logical as the directory containing PLT.EXE and its various utility files.

   $DEFINE/SYSTEM PLT disk:[directory]

2. In the SYSLOGIN.COM file or its equivalent pointed to by SYS$LOGIN define symbol PLT:

   $PLT == @PLT:PLT.COM

3. In the PLT directory, define command file PLT.COM as follows:

```
$! PLT.COM Implementation of PLT
$WRAP=F$GETDVI("SYS$OUTPUT","TT_WRAP")
$FORM=F$GETDVI("SYS$OUTPUT","TT_MECHFORM")
$SET TERM/FORM/NOWRAP
$DEFINE/USER SYS$INPUT SYS$COMMAND
$RUN PLT:PLT
$IF WRAP THEN SET TERM/WRAP
$IF .NOT.FORM THEN SET TERM/NOFORM
$EXIT
```

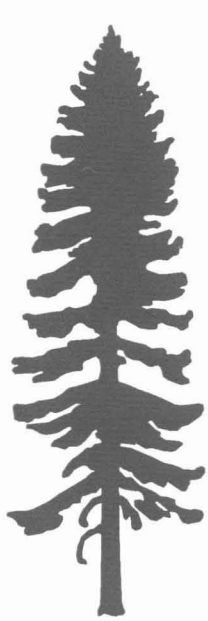
4. In the PLT directory, establish the following file protections:

```
$SET PROT=W:RE PLT.COM,PLT.EXE
$SET PROT=W:RW PLT.MSG
$SET PROT=W:R PLTHELP.DAF
$SET PROT=W:R FONT*.DAT
$SET PROT=W:R *.PLT
```

SYSTEM DRAWING FILES

The following figures are currently available with PLT. They can be modified and SAVed in the user's directory. They can be accessed by issuing the command:

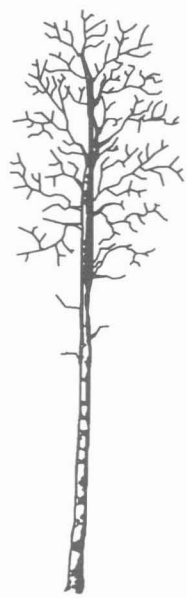
PLT> **RET PLT:name.PLT**



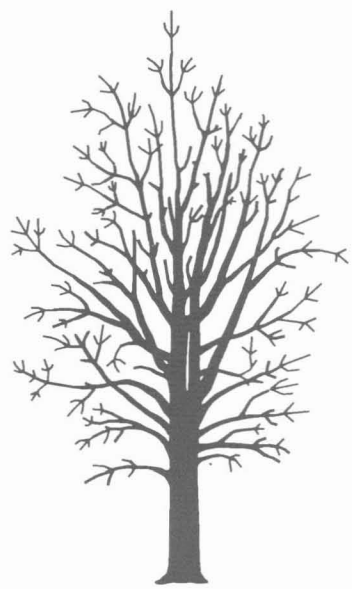
PLT:Whitspruc.plt

PLT:Balsamfir.plt

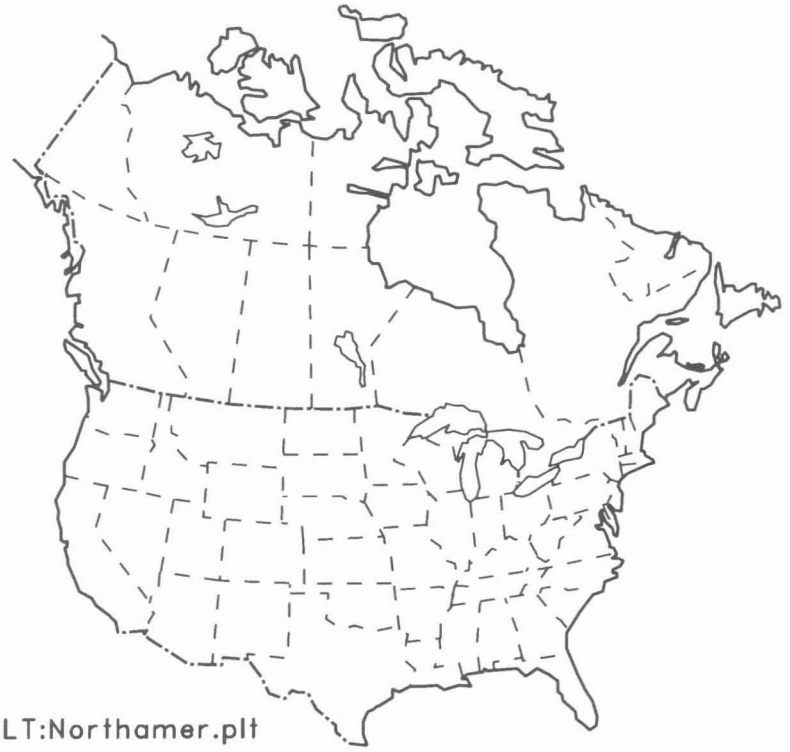
PLT:Blackspruc.plt



PLT:Poplar.plt

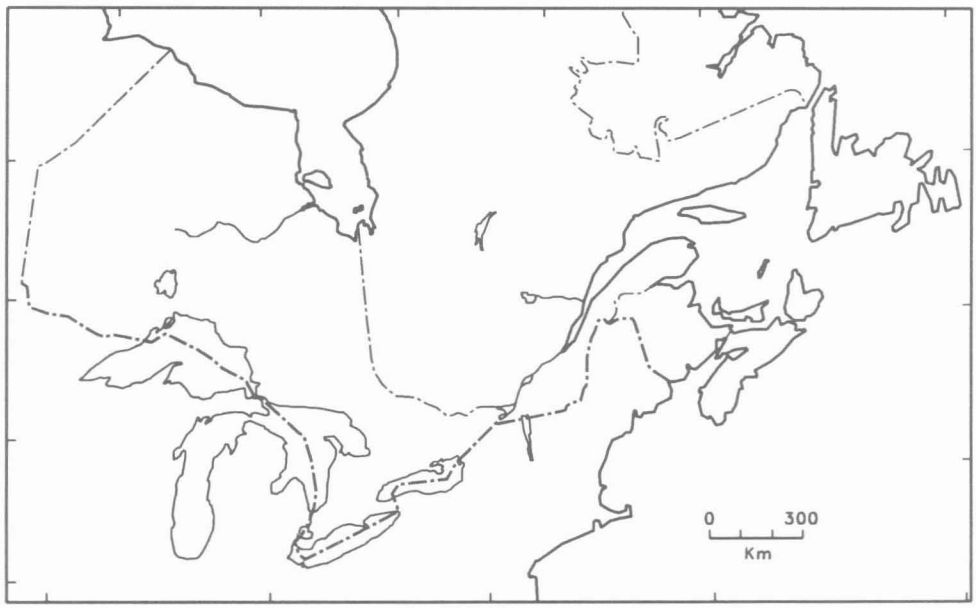


PLT:Sugarmapl.plt



PLT:Northamer.plt

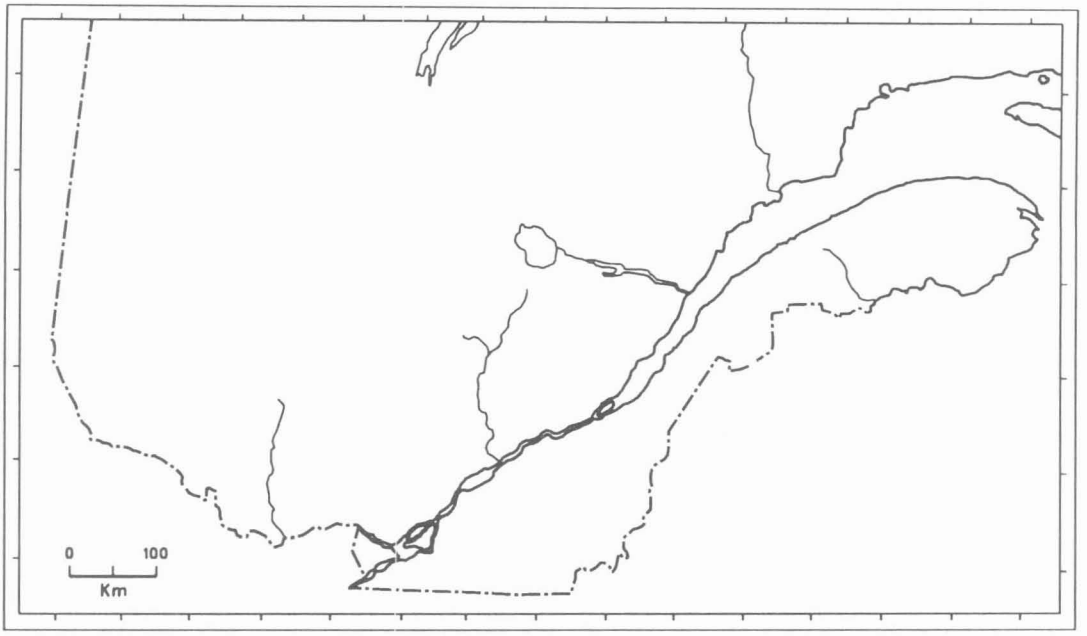
PLT:Canada.plt



PLT:Northeast.plt

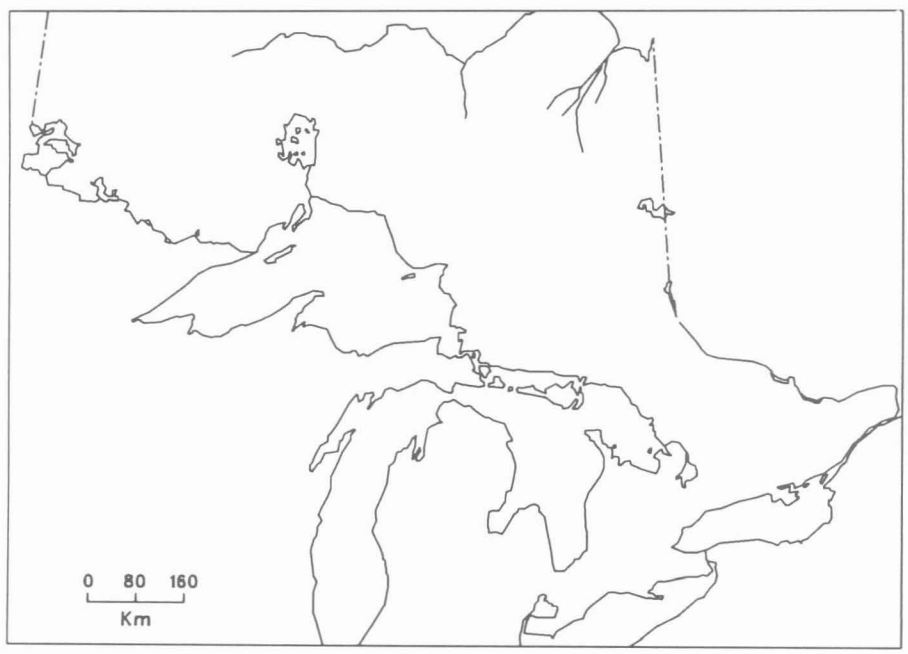
PLT:Quebec.plt

PLT:Quebec2.plt



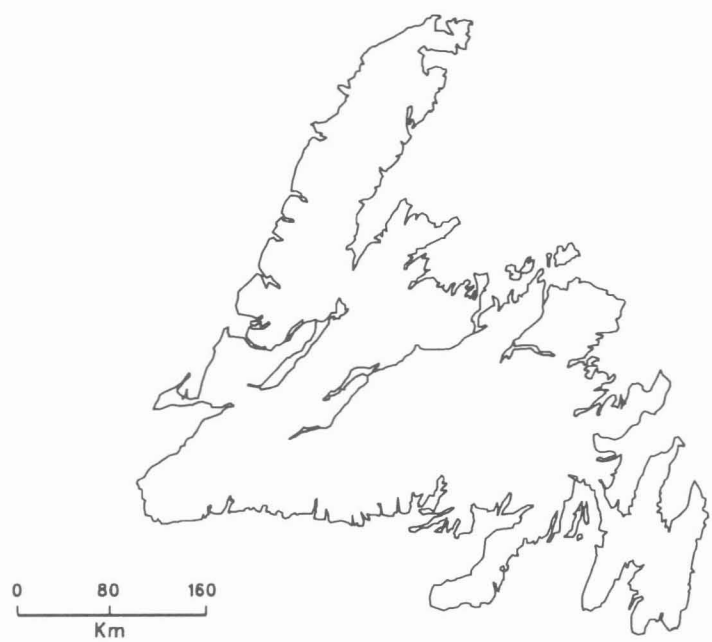
PLT:Quebec3.plt

PLT:Ontario1.plt



PLT:Ontario2.plt

PLT:Maritim.plt



PLT:Newfoundl.plt

Canada