TEKTRONIX®

PLOT 10

4010A03

INTERACTIVE G. APHING PACKAGE

USER MANUAL

TEKTRONIX®

PLOT 10

4010A03

INTERACTIVE GRAPHING PACKAGE

USER MANUAL

Tektronix, Inc.
P.O. Box 500
Beaverton, Oregon 97077

070-2247-00

First Printing Feb 1977

CONTENTS

CONTENTS

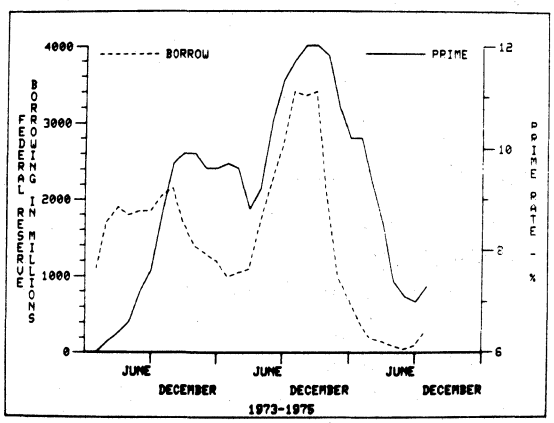
Throughout history man's social and technological progress has been largely the result of his ability to understand and represent the nature of problems. A simple problem consists of a small number of variables. How do these variables interact? A complex problem has so large a number of variables that it is difficult to see which are significant. How do we know? Graphing is a basic tool for problem solving, because it helps us to answer these questions.

A graph is a pictorial representation or map of abstract relationships. Because a graph is a picture rather than a verbal description, its meaning is immediately clear to anyone familiar with a few, simple conventions. The information in a graph can be used by anyone concerned with a problem, regardless of the depth of his or her familiarity. The graph is, therefore, not only a way of representing information but also of reducing its complexity. It allows the problem solver to move from a situation where the outlines of the problem are immediately apparent to one where he can begin to perceive actual solutions.

In the figure below showing federal reserve borrowing vs. prime interest rate, we are able to see the inherent power of a graph to communicate information as compared with columns of data. The graph in the figure shows clearly the trends which are not obvious in the associated data table.

FIGURE 1-1

COLUMN -- PRDATA		COLUMN -- BORODATA	
6.00	10.50	1100.00	2200.00
6.20	11.30	1700.00	2700.00
6.40	11.70	1900.00	3400.00
6.60	12.00	1800.00	3350.00
7.20	12.00	1850.00	3400.00
7.60	11.80	1850.00	2000.00
8.70	10.80	2050.00	1000.00
9.70	10.20	2150.00	700.00
9.90	10.20	1700.00	400.00
9.90	9.40	1400.00	200.00
9.60	8.50	1300.00	150.00
9.60	7.40	1200.00	100.00
9.70	7.10	1000.00	50.00
9.60	7.00	1050.00	100.00
8.80	7.30	1100.00	300.00
9.20		1700.00	



INTRODUCTION

Traditionally, one of the problems with drawing a graph like the one in figure 1-1 has been the amount of time involved. Today we can use the tremendous processing capability of a computer to reduce that time to insignificance. In addition, a powerful software package like Interactive Graphing enables a person with little or no computer or graphing experience to use, after a minimum of work, all the help that graphing can lend to problem solving.

When you create a graph using a computer, you expect the computer to do most of the work. Ideally, you would enter some data (for example, sales records from July through December of a given year), type DO IT on your terminal keyboard, and see a labled graph displayed on the screen. Interactive Graphing has been designed with this perhaps unattainable goal in mind.

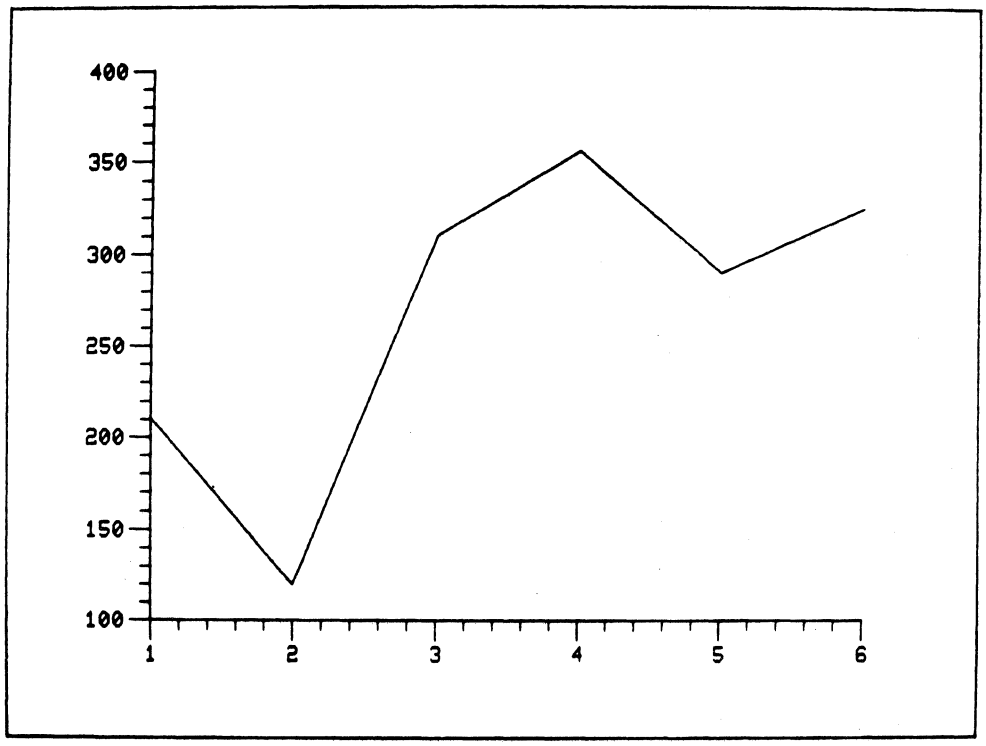
The package extends the capabilities of the fundamental Tektronix graphing software for 4000-Series graphics terminals to non-programmers as well as to programmers. By means of easy to learn, English-language structured commands, you will be able to construct, edit, save, and recall graphs in a conversational relationship with the computer. Interactive Graphing guides you through the creation of graphs by means of messages which indicate errors or inform you about required input. The package also provides a library of commands which you can access at any time either in part or *in toto* by typing HELP.

One of the traditional problems with computer languages is that the syntax (word order) of the commands you type in tends to be inflexible. Interactive Graphing allows you to type in commands in any of several ways. As a general rule, as long as you begin a command with a verb, as you would do in English, the subsequent word order is your decision.

A SAMPLE GRAPH

Assuming that Interactive Graphing has already been implemented on your computer system, let us create the graph in figure 1-2. To create this graph, we will use the simplest possible Interactive Graphing commands and allow the package to do most of the work. We will not worry about defining terms at this point.

FIGURE 1-2



To begin a session with Interactive Graphing, use the RUN command for your particular system, followed by IGRAPH and a carriage return (∂).

Example:

RUN IGRAPH (∂)

When Interactive Graphing is ready to go, a mark (>) will appear on your terminal screen. Type the sequence of commands as shown below, and the same graph shown in figure 1-3 will be displayed on your screen.

ENTER DATAX:1, 2, 3, 4, 5, 6∂
ENTER DATAY:211, 114, 306, 354, 291, 325 ∂

You have just entered the X and Y values which will be used in the curve of your graph.

DEFINE A CURVE CALLED C AS DATAX, DATAY ∂

The curve in your graph is named C; its X (horizontal) values come from the data in DataX; its Y (vertical) values come from the data in DataY.

DEFINE A GRAPH CALLED TEST ∂

You have defined the existence of a graph, which you have named Test. The graph, Test, has an X axis and a Y axis.

INCLUDE C IN TEST ∂

You have included your curve, C, in the graph, Test. You may now display the finished graph. The ERASE command below clears the screen first.

ERASE; DISPLAY TEST ∂

Your screen should have the sample graph displayed on it exactly as shown in figure 1-2.

THE MANUAL

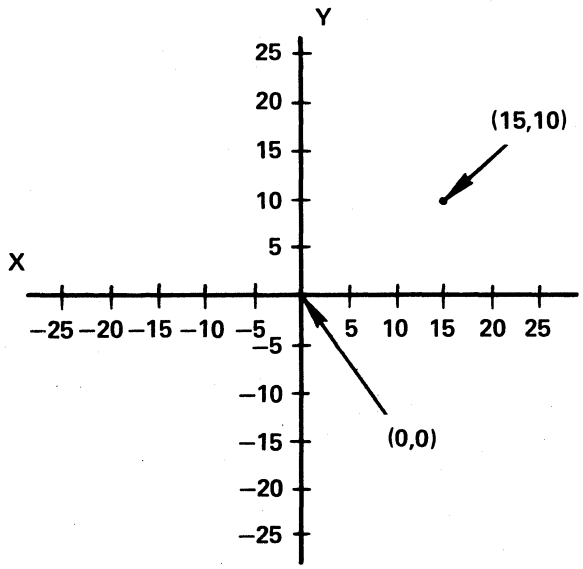
The Interactive Graphing User's Manual is intended both as a tutorial in the use of the software and as a reference guide for experienced users. The format of the manual includes:

Section 2.	General operating information and the discussion of basic concepts of the software.
Section 3.	A section designed to guide you through the interactive commands, beginning with simple graphing.
Section 4.	An advanced section which demonstrates some of the more sophisticated capabilities of the software.
Section 5.	A reference section which includes definitions and tables helpful to a user already familiar with the software.

WHAT IS A GRAPH

Like a map a graph is, for our purposes, a two dimensional surface with a horizontal direction and a vertical direction. Any point on this surface can be uniquely specified by measuring its distance along a horizontal and along a vertical axis. The distance along these axes is measured from the point where the axes intersect. This point is the origin. The figure below shows a typical set of axes. The horizontal axis is conventionally called the X axis; the vertical axis is called the Y axis. When the position of a point is specified, the distance in the X (horizontal) direction is specified first, followed by the distance in the Y (vertical) direction (that is, the X and Y coordinates of the point). The origin of the axis below is at (0,0); the arrow shows a point whose coordinates are (15, 10), or 15 units to the right of the origin in the X direction and 10 units up from the origin in the Y direction.

FIGURE 2-1

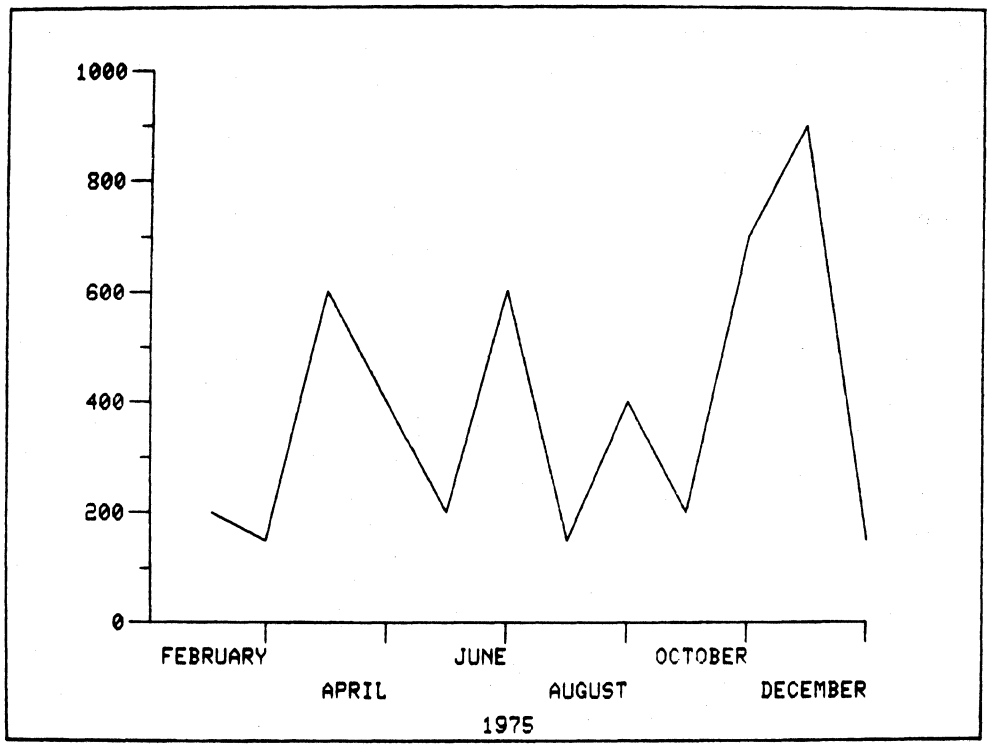


Most graphs with practical applications do not look exactly like the one in figure 2-1. The kind of graph you will for the most part create with Interactive Graphing looks like the one in figure 2-2.

GENERAL INFORMATION

WHAT IS A GRAPH

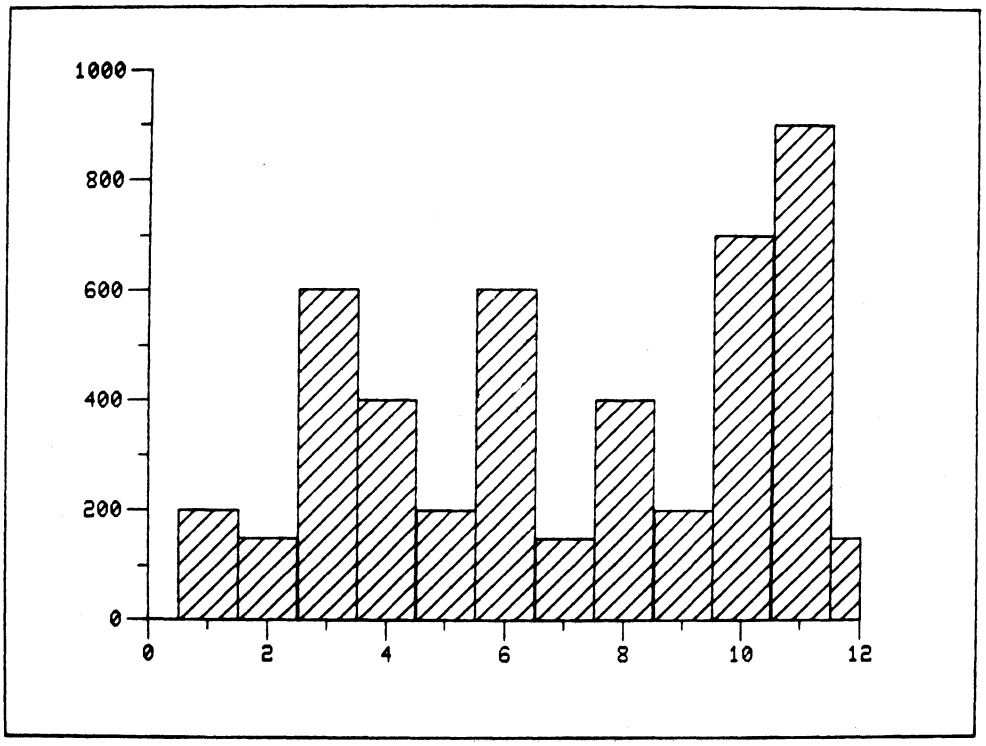
FIGURE 2-2



Notice that the origin of this graph (the point where the X-axis and the Y-axis intersect) is at the lower left corner. The numbers and months on the graph are tic mark *labels*. The tic marks the labels refer to are the *major tic* marks, and the non-labeled tics are the *minor tic* marks. The *curve* included in this graph is represented by a solid line. You might also wish to include a curve with a dashed line. For this reason Interactive Graphing gives you the option of many dashed *line styles* for curve representation.

Another frequently encountered kind of graph is the bar graph, or bar chart, which uses either vertical or horizontal bars. Figure 2-3, a bar chart with vertical bars is illustrated.

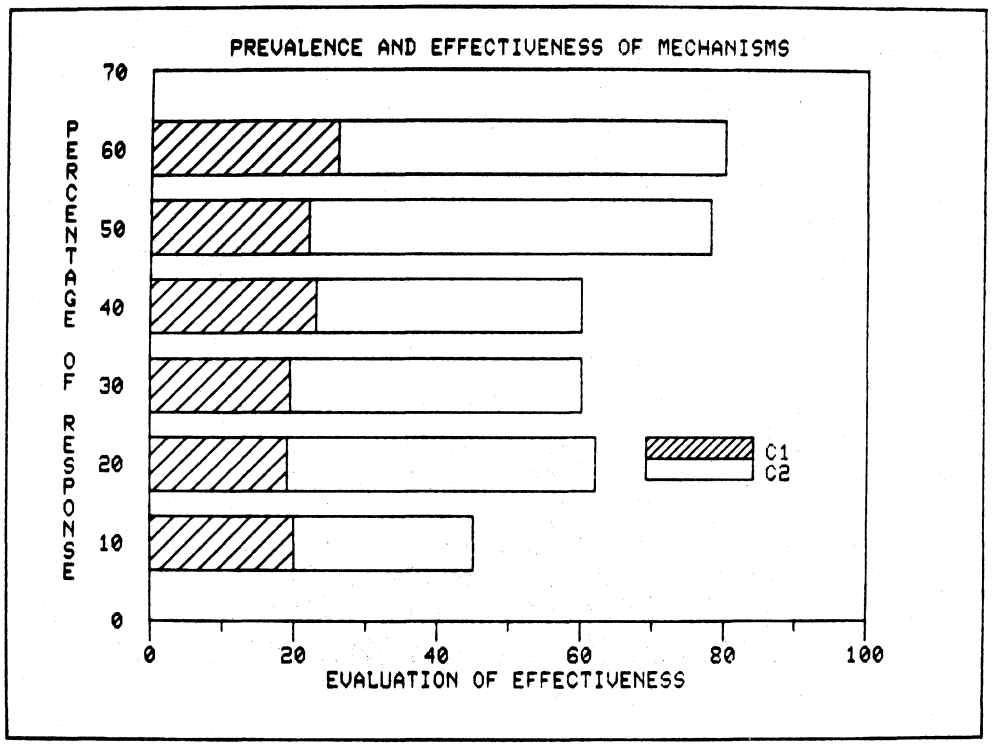
FIGURE 2-3



A bar chart also represents a curve, even though the bars are not continuous. For a bar chart you may imagine that the curve is drawn through the top of each bar at its center point. One of the advantages of a bar chart is that two or more curves with different shadings can be superimposed on each other. This effect is shown in the horizontal bar chart, figure 2-4, which illustrates two curves, C1 and C2.

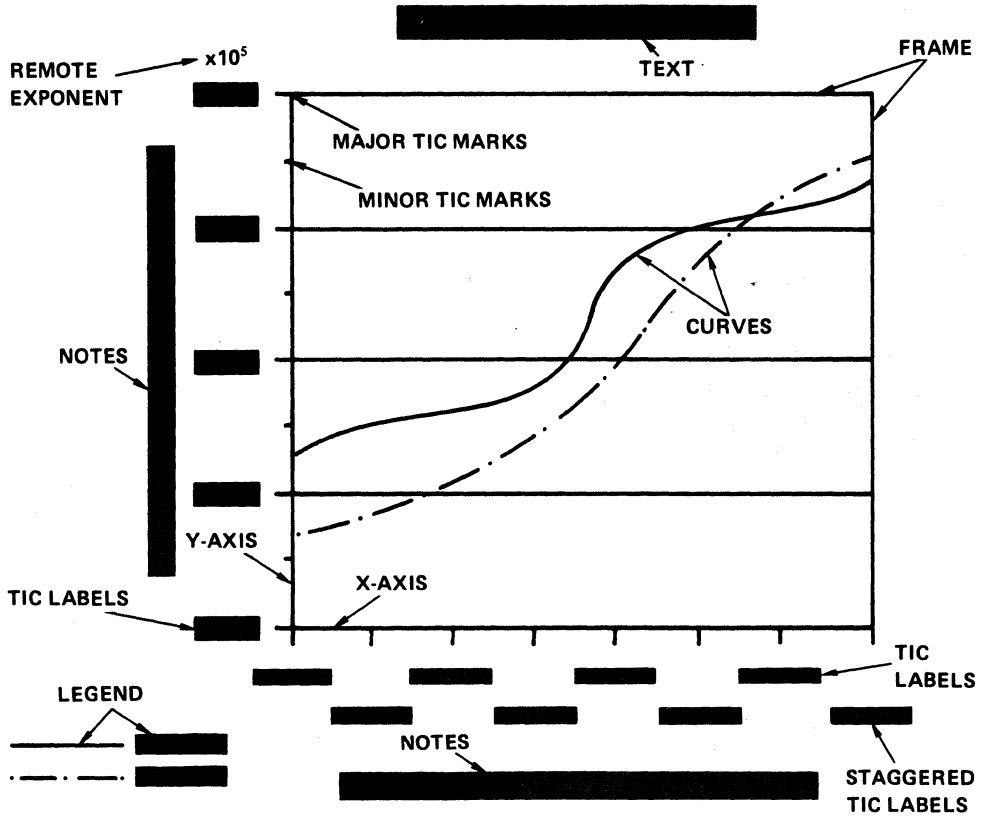
WHAT IS A GRAPH

FIGURE 2-4



Other additions to this graph are the use of axis labels (called *notes* by Interactive Graphing) and a *legend* which tells which shading belongs with which curve. Figure 2-5 shows the names used by Interactive Graphing for the parts of a graph.

FIGURE 2-5



The variety and complexity of graphing terminology may seem confusing at this point. However, the next section, entitled Using Interactive Graphing, is organized so that terminology is introduced as gradually as possible. Figure 2-5 will serve as a useful reference throughout that section.

Once you have become familiar with some simple concepts and how they work, Interactive Graphing is easy to learn.

GENERAL INFORMATION

PARTS AND PARAMETERS

The Interactive Graphing commands are sentences consisting of command verbs and the graphing parts and parameters on which they act. A part is a class or element of a graphics display. The most inclusive (global) part is the page. If you wish to save and restore a display, you must do this by including the elements of the display in a named page. Henceforth, all of the sub-parts of the page are associated with it until you specify otherwise.

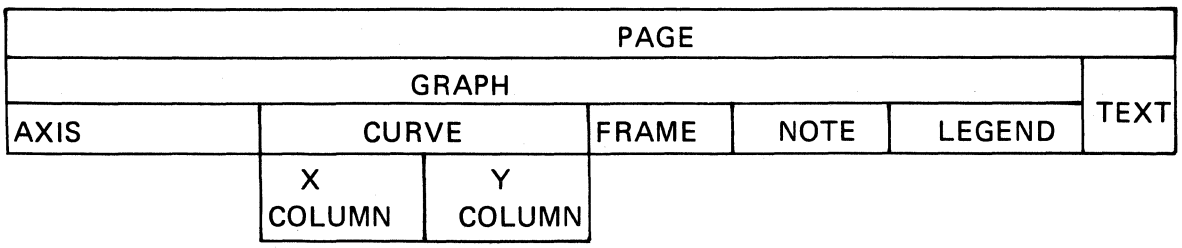
Within the page a hierarchy of parts prevails. For example, a graph is included within a page, a curve within a graph, and an X column of data and a Y column of data within a curve. Every part must be given a unique name to distinguish it from other, similar parts. Partnames may use up to eight alphanumeric characters.

Those characteristics of a part which can change (set) are called parameters. For example, if a dashed line style is necessary for a curve, you can set the line style to the desired numerical value. Line style is one of the parameters of the curve.

Parts include:

PAGE
GRAPH
TEXT
AXIS
CURVE
X COLUMN
Y COLUMN
FRAME
NOTE
LEGEND

Note that every part (except for a page) can be a sub-part of another part, as shown below:



The above parts and their definitions are discussed in Section 3 of this manual.

CONTEXT

One of the major conveniences offered by Interactive Graphing is its English language style of communication with a user. The basis of this communication is a set of command verbs with which you establish graphing contexts and set parameters and values.

To clarify the concept of context, let us examine how it works in the typical English command sentence

Throw the ball over the fence.

In this case the command indicates what is to be thrown and where. A further command, taken in the context of the first, might be,

Throw it back to me.

In the second sentence a new direction is specified, but "it" is still the same ball. If no context has been established, however, the command

Throw it,

is meaningless. The recipient of the command might ask, "Throw what? Where shall I throw it?"

Interactive Graphing relies on a similar context dependency in order to make graphing commands simple and avoid the necessity of frequent repetition. For example, you wish to create a graph (POLY) with a specific curve (SQUARES) in it. After the X and Y data have been entered, you would define the curve:

DEFINE A CURVE NAMED SQUARES AS XDATA, YDATA

This command defines a curve (SQUARES) and establishes it as the context. You then define the graph and its axes with a similar command:

DEFINE A GRAPH NAMED POLY

This command establishes the graph (POLY) as the new context. In order to include the curve in the graph so that they may be displayed together, you have several options, because a context has been established. You can command,

INCLUDE SQUARES IN POLY
or
INCLUDE SQUARE IN IT
or simply
INCLUDE SQUARES

Interactive Graphing knows that you are referring to POLY whether you type in its name or not, because POLY was your last established context.

CONTEXT

At this point you may type

DISPLAY

The graph and the curve are then displayed.

Although context dependency is a valuable asset in attaining efficiency, you should be aware that both artificial and human memories are fallible. Therefore, if you are unsure of the present context, always specify the one you wish to work in.

NOTATION CONVENTIONS

As can be seen in the preceding subsection, Interactive Graphing allows you to type in commands which closely approximate complete English sentences. As long as they begin with command verbs, many varying word orders can usually be used. The program is capable of distinguishing between the essential or key words (operands) in a command and those filler words which can be entered optionally as an aid to clarity. This means that as you increase your familiarity with the command language, you will be able to abbreviate the commands, either partially or completely, to expedite graphing.

Throughout this manual, as commands are illustrated, the following notation conventions apply:

1. A word which must be entered but whose form you yourself choose (e.g., a Partname) is initially capitalized.

2. Key words (operands) in a command are all upper-cased. Words which can be entered optionally, but which the program ignores, are all lower-cased.

   Example:

   DEFINE a CURVE named Squares as Xdata, Ydata

   DEFINE, CURVE — should be entered as shown; however, any of these words may be truncated to the first four characters.

   Squares, Xdata, Ydata — must be entered; you choose the name for these partnames.

   a, named, as — can be entered for sentence clarity, but are ignored by Interactive Graphing.

3. Any operand enclosed in parenthesis is optional, depending on context.

   Example:

   INCLUDE Squares (in GRAPH)

4. A stack of operands indicates that one and only one is required to give the command meaning.

   Example:

   SET AXIS DENSITY
   TRANSFORMATION
   LOCATION

   Choose either DENSITY or TRANSFORMATION or LOCATION.

5. Any punctuation illustrated in a command is required. A blank space can always be substituted for a comma, however.

NOTATION CONVENTIONS

6. Underlines in syntax examples indicate a default condition; that is, what the software assumes unless you specify otherwise.

7. A carriage return, the pressing of the RETURN key, is indicated by ∂.

FILLER WORDS

The following is a list of words which can be entered with no effect on the Interactive Graphing system. The purpose of the words is to help provide English-like sentence capability for the commands. None of these words can be used as a partname.

for	of
as	it
with	a
in	called
at	named
from	mark
to	marks
the	

GETTING HELP

THE HELP COMMAND

Interactive Graphing is designed so that you can generally help yourself if you are unsure about what to do next. If you type

HELP∂

a list of available command verbs appears on the screen. To receive more detail about a specific verb, type a question mark (?) followed by a verb or keyword you want to know about.

Examples:

HELP∂

THE VERBS ARE :

APPEND	BYE	CANCEL	CHANGE	COPY
DEFINE	DELETE	DISPLAY	ENTER	ERASE
EXCLUDE	HDCOPY	HELP	INCLUDE	LIST
POSITION	RECALL	RENAME	SET	STORE

?APPEND

APPEND--THE APPEND VERB IS USED TO ADD DATA TO EXISTING COLUMNS.

FORMAT: APPEND COLUMNNAME (, COLUMNNAME, . . .COLUMNNAME) : VALUE (, VALUE . . VAL

?HELP

HELP--THE HELP VERB IS USED TO GET HELP.

IF I ASK YOU A QUESTION THAT YOU DON'T UNDERSTAND, TYPE A QUESTION MARK AND A CARRIAGE RETURN. I WILL RESPOND WITH SOME HELP. IF YOU WANT CLARIFICATION OF A TERM TYPE A QUESTION MARK FOLLOWED BY THAT TERM AND A CARRIAGE RETURN. IF THAT WORD IS IN MY VOCABULARY, I WILL TELL YOU WHAT IT MEANS TO ME. IF YOU ARE IN COMMAND MODE, YOU CAN USE THE WORD HELP INSTEAD OF A QUESTION MARK.

PROMPT CHARACTERS AND SYSTEM MESSAGES

Whenever the software system expects you to input information, it generates a prompt character. The standard prompt is the greater-than sign (>). When an error has occurred or when you have not input enough information, the system will generate a short message followed by a question mark (?). Enter the requested information. If you are not sure what the short system message means, type

?∂

GETTING HELP

An expanded message will appear on the screen. If the standard prompt (>) appears following a system message, you must enter another command. No action has been taken on the original command. Again, if the short message is not clear, type

?∂

and an expanded message will appear.

NOTE

Some computer systems generate a prompt character of their own in addition to the Interactive Graphing prompt. In such cases the first prompt character is the Interactive Graphing prompt.

UTILITY COMMANDS

BYE

When you wish to end an Interactive Graphing session, type

BYE

ERASE

When you wish the screen to be cleared under program control, type

ERASE

HDCOPY

If your terminal is connected to a Tektronix hardcopy unit, when you type

HDCOPY

a hardcopy of the display will be generated under program control.

USING INTERACTIVE GRAPHING

INTRODUCTION

This section demonstrates the Interactive Graphing commands as they might be used in a sample session. The organization of the section is by command verbs. The verbs are discussed in the following order:

1. ENTER, DEFINE, INCLUDE, DISPLAY
2. POSITION
3. LIST
4. SET
5. APPEND
6. EXCLUDE
7. CHANGE
8. CANCEL, DELETE
9. COPY, RENAME
10. STORE, RECALL

Throughout the section a running example of the command usage is included in the text as an illustration.

INITIALIZATION AND TERMINATION

To begin a session with Interactive Graphing, use the run command for your particular system, followed by IGRAPH and a Carriage Return (∂).

Example:

RUN IGRAPH∂

At this point, if your terminal is a model 4006, 4010, 4012, or 4013, you may continue with your Interactive Graphing session. If you have a terminal model 4014, 4014 with EGM, 4015, or 4015 with EGM and you want to take advantage of their extra features, you must tell the Interactive Graphing Package. Unless you inform the package, it will assume that you have a 4010 terminal. All Tektronix graphics terminals can, however, be operated in a 4010 mode.

USING INTERACTIVE GRAPHING

INTRODUCTION

To specify your terminal, type:

SET TERMINAL MODEL to Value, where Value is one of the following:

1 indicates a 4010, 4012, or 4013 terminal.

2 indicates a 4014 or 4015 terminal.

3 indicates a 4014 or 4015 terminal with EGM.

If you are connected to a transmission line whose baud (transmission) rate is different from the rate specified when the software was implemented, you need to re-specify the rate. Type:

SET TERMINAL BAUD to Value where Value is the new baud rate.

If you wish a different character size from the standard 4014/4015 size, type:

SET TERMINAL FONT to Value where Value is a number from 1 to 4:

1 74 characters/line (standard size).

2 81 characters/line.

3 121 characters/line.

4 133 characters/line.

To end a session with Interactive Graphing, type BYE$_{\partial}$

NOTE

One further Terminal parameter should be mentioned. If you are interested in outputting to a peripheral (for example, a digital plotter), you can specify the peripheral as follows: Type

SET PERIPHERAL to Value where Value is a user-specified value which is passed to a user-written peripheral control routine.

ENTER, DEFINE, INCLUDE, DISPLAY

The four commands discussed under this heading are the minimum number necessary to construct a graph and display it on the screen. ENTER creates a data column and inserts values in the column, generally in X, Y pairs. DEFINE creates and names a graph and a curve. INCLUDE makes the curve a part of the named graph. Finally, DISPLAY is used to display the graph and its curve.

ENTER

The ENTER command can take two basic forms:

1. ENTER Xname:Value(,Value, . . . ,Value)$_{\partial}$
   ENTER Yname:Value(,Value, . . . ,Value)$_{\partial}$

2. ENTER Xname, Yname:Xvalue,Yvalue(Xvalue,Yvalue, . . .Xvalue,Yvalue)

The two methods are almost alike. In (1) you enter the X column first. Specify a name for the X column and all the values which follow are stored as X data. Then enter the Y column in the same way. In (2) you enter both X and Y columns in the same command. Name the X column and the Y column, and the values which follow are stored as X,Y pairs.

Example:

(1) ENTER XDATA: 1, 2, 3, 4, 5$_{\partial}$
    ENTER YDATA: 1, 4, 9, 16, 25$_{\partial}$

(2) ENTER XDATA,YDATA: 1,1 2,4 3,9 4,16 5,25$_{\partial}$

NOTE

In both cases a colon (:) must precede the first data value.

Other capabilities of the ENTER command are discussed in the Advanced Usage section of the manual.

DEFINE

The DEFINE command creates and names a page or a graph or any of their respective subparts. The command takes the following basic forms (a partname consists of up to eight user-supplied, alphanumeric characters):

USING INTERACTIVE GRAPHING

ENTER, DEFINE, INCLUDE, DISPLAY

DEFINE a Part called Partname

CURVE called Partname as Xname, Yname

GRAPH called Partname (with Xaxisname and Yaxisname)

You have the option of naming each axis of the graph, and applications for this feature will be discussed later. If you do not name each axis, Interactive Graphing will supply the axis names for you (Xgraphname and Ygraphname).

NOTE called Partname as "String"

TEXT called Partname as "String"

Examples:

DEFINE a PAGE called P1∂

In this example we needed only to name the page.

DEFINE a CURVE called C1 as XDATA,YDATA∂

In this example we enter the names given to the X and Y columns.

DEFINE a GRAPH called G1∂

In this example we define graph (G1). The X and Y axes are automatically named (XG1 and YG1) for future reference.

DEFINE a NOTE called N1 as 'THIS IS A NOTE' ∂

In this example we give a value to the note (N1). The value is the text string 'THIS IS A NOTE'.

INCLUDE

The INCLUDE command is used to make the named part a member of a larger part. The basic form of the command is the following:

INCLUDE Partname in Partname

Example :

INCLUDE C1 IN G1 (the order of C1 and G1 is immaterial, since the package knows which part is included in which part).

The Interactive Graphing parts and their member relationships are shown in the following chart:

Part	Page	Graph	Text	Curve	Note	Legend	Frame
Sub-parts	Graph Text	Axis Curve Note Legend Frame	None	Column	None	None	None

The INCLUDE command can also be used to position parts. See the subsection Positioning a Part with INCLUDE.

DISPLAY

The DISPLAY command causes the named page or part of a page to be displayed on the terminal screen. It takes the following form:

DISPLAY Partname

Example :

DISPLAY G1∂

All the parts which make up the graph G1 are displayed.

The following parts can be displayed:

PAGE
GRAPH
TEXT
CURVE
NOTE

NOTE

To get a clean display of a graph or a page, it will generally be necessary to clear the screen. You can clear the screen by pressing the PAGE key or by typing ERASE before DISPLAY.

Example: ERASE ; DISPLAY∂

ENTER, DEFINE, INCLUDE, DISPLAY

CREATING A GRAPH

Using the information which has been given so far, let us construct a simple graph, adding detail and complexity to it until the basic capabilities of the four commands have been exhausted. The graph (G1) will consist of a curve (SQUARES) with X and Y columns (XDATA and YDATA).

To initialize Interactive Graphing, type

RUN IGRAPH$_{\partial}$*

Enter the columns in X,Y pairs:

ENTER X1,Y1: 1,1 2,4 3,9 4,16 5,25 6,36 7,49 8,64$_{\partial}$
9,81 10,100$_{\partial}$

NOTE

You can continue entering pairs despite the need for a carriage return. A carriage return is allowed between data points or between X and Y values.

Define the curve:

DEFINE CURVE called SQUARES as X1, Y1$_{\partial}$

Define the graph:

DEFINE a GRAPH called G1$_{\partial}$

Include the curve:

INCLUDE SQUARES (in G1)$_{\partial}$

NOTE

It is not necessary to specify G1 in this command, since G1 is the current context. See Context, section 2.

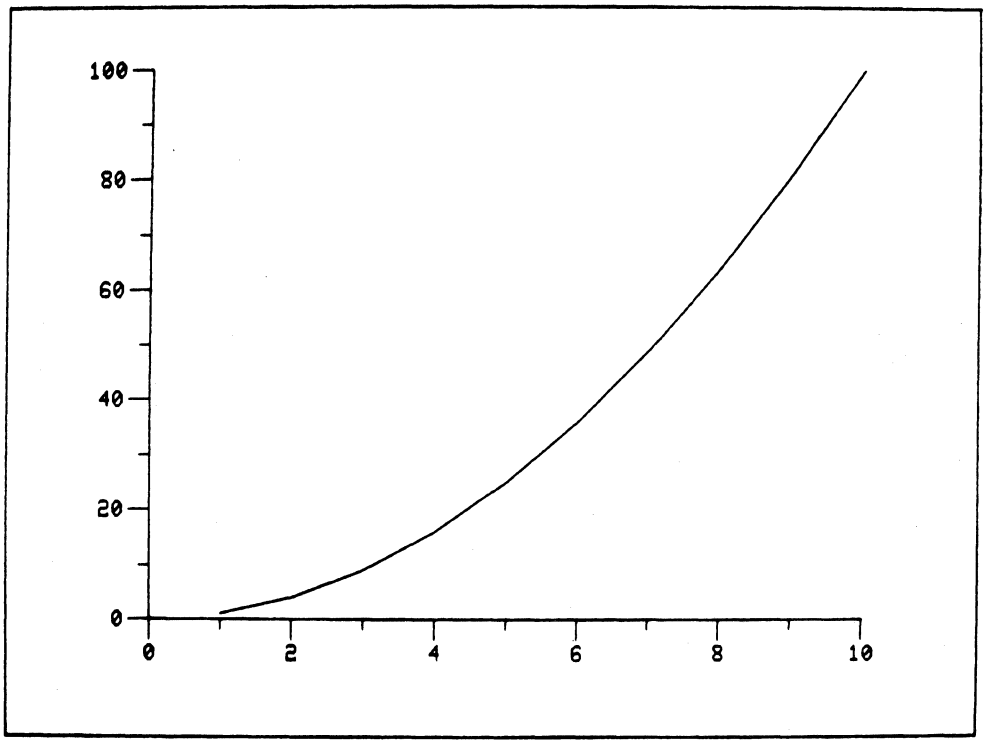
Display the graph and its parts:

ERASE; DISPLAY G1$_{\partial}$

*The command is system dependent. Use the RUN command format for your system.

The following graph is plotted on the screen. A bell sounds when the operation is completed.

Example:



POSITIONING A PART WITH INCLUDE

Whenever a part must have a specific location with reference to another part, it must be included at a given position on the screen. Suppose the already created graph, G1, is to be included in a page. If we type

DEFINE a PAGE P1∂
INCLUDE G1 (in P1)∂

the following request is returned:

NEED POSITION POINT
?

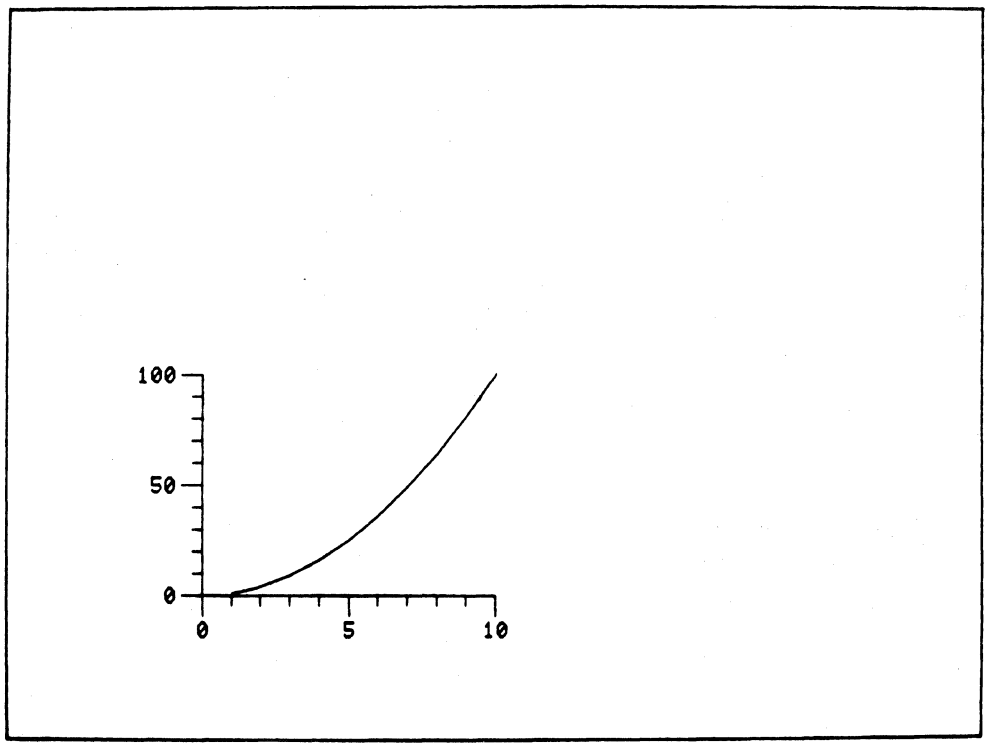
ENTER, DEFINE, INCLUDE, DISPLAY

You must enter the position that the graph is to occupy on the page. The position of the member graph is given as a percentage of the page. A page always occupies the entire screen; therefore, coordinates given as a percentage of the screen are required. These coordinates, measured from the lower left corner of the screen, give the position of the lower left corner and the upper right corner of the graph. For the example, you might type the following:

```
INCLUDE G1 (in P1) at 20,20  50,50
DISPLAY P1
```

The page and all of its members are displayed: The lower left corner of the graph is 20% to the right (X direction) and 20% up (Ydirection); the upper right corner is 50% to the right and 50% up.

Example :



INCLUDING A TEXT

Besides a graph, the only other subpart of a page is a text. Text is any alphanumeric line (character string) and it can be positioned on the page in a way similar to the way a graph is positioned. The location of the text is specified as a set of X,Y coordinates expressed as percentages of the page, that is, relative to the whole screen. However, in this case you may specify that the given coordinates are the start, end, or center of the line of text. Suppose

that in the example page, P1, we desire a line of text as a title of the page. First the text must be defined:

DEFINE a TEXT T1 as 'INTERACTIVE GRAPHING AT WORK'∂

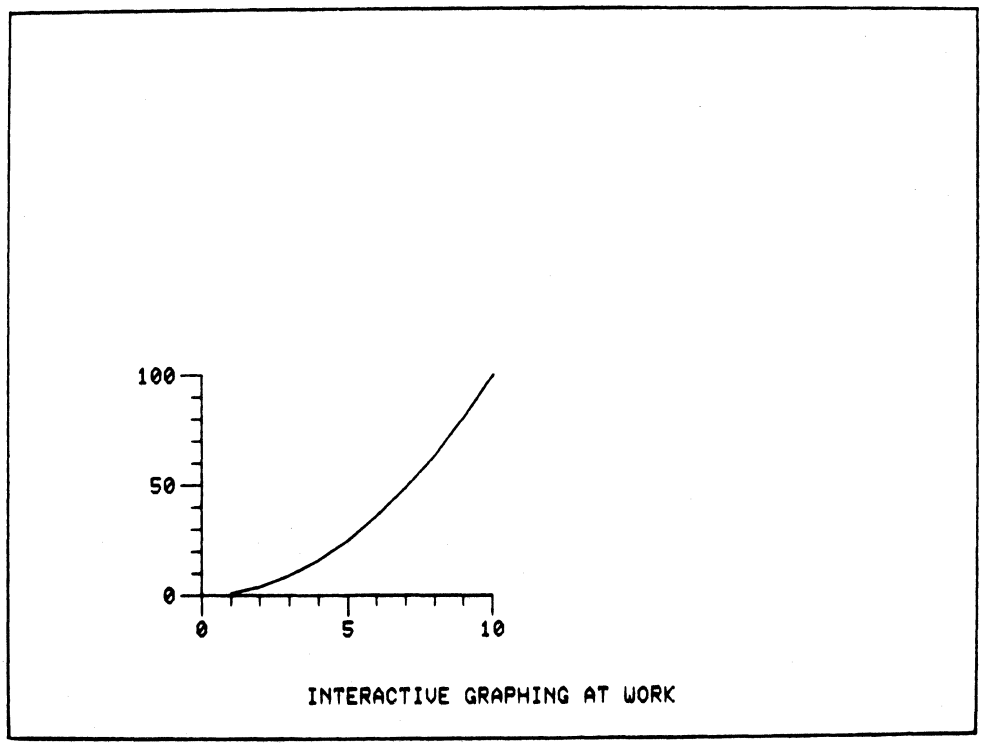
Next, we include the text as the centered title of the page. If neither the start nor the end of the text is specified, the text will automatically be centered on the given coordinates (CENTER is the default or initially assumed, setting). Therefore, we could type either of the following commands for our example:

Either INCLUDE T1 (in P1)* with CENTER at 50,5∂

or INCLUDE T1 (in P1) at 50,5∂
DISPLAY (P1)∂

The page is displayed:

Example :



*This is optional according to context.

ENTER, DEFINE, INCLUDE, DISPLAY

INCLUDING A SECOND CURVE IN A GRAPH

Using the same example page and graph, let us define and include another curve. The curve (Cubes) will use the same X column as the graph already uses; therefore, only the Y column needs to be created. Type,

ENTER YCUBE:1 8 27 64 125 216 343 512 729 1000$_\partial$

Define the new curve:

DEFINE a CURVE called CUBES as X1, YCUBE$_\partial$

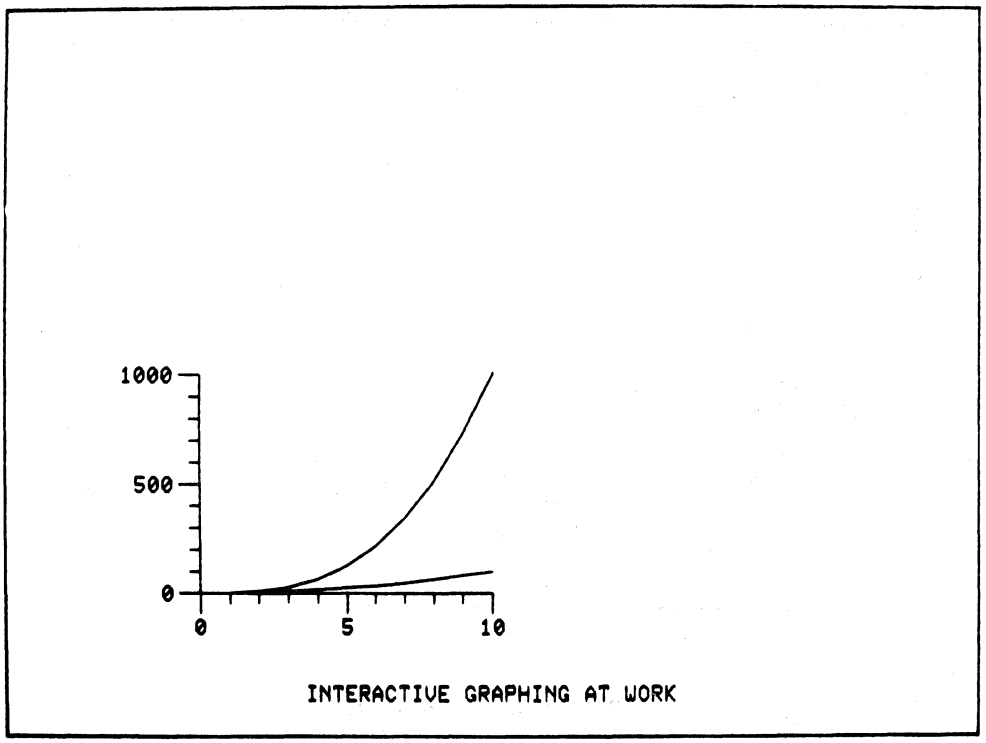
Include the curve in the graph:

INCLUDE CUBES in G1$_\partial$

DISPLAY P1$_\partial$

The page, the graph, and the two curves are displayed. Note that the axes tic marks have been relabeled automatically to accommodate the greater range of points in the new curve.

Example :



POSITIONING NOTES ON A GRAPH WITH INCLUDE

A note is an alphanumeric line which is a member or subpart of a graph. In all other respects it is identical to a text. The most common use for notes is the labeling of the X and Y axes of a graph. Using the example page and graph, let us define and include notes to label the axes. Type,

```
DEFINE a NOTE NX as 'BASE INTEGERS'∂
DEFINE a NOTE NY as 'SQUARES & CUBES'∂
```

We will now include NX in an appropriate position below the X axis of the graph. Because a note is relative to a graph, the positioning coordinates are expressed as percentages relative to the lower left corner of the graph, rather than relative to the screen or page:

```
INCLUDE NX in G1 50,–30∂
```

NOTE

Care must be taken that enough room remains on the screen below the tic mark labels for the note to be printed.

Include NY in an appropriate position to the left of the Y axis. Notice, however, that the note must be positioned vertically:

```
INCLUDE NY in G1 VERTICALLY at –35,50∂
```

NOTE

Care must be taken that enough room remains on the screen to the left of the tic mark label for the note to be printed.

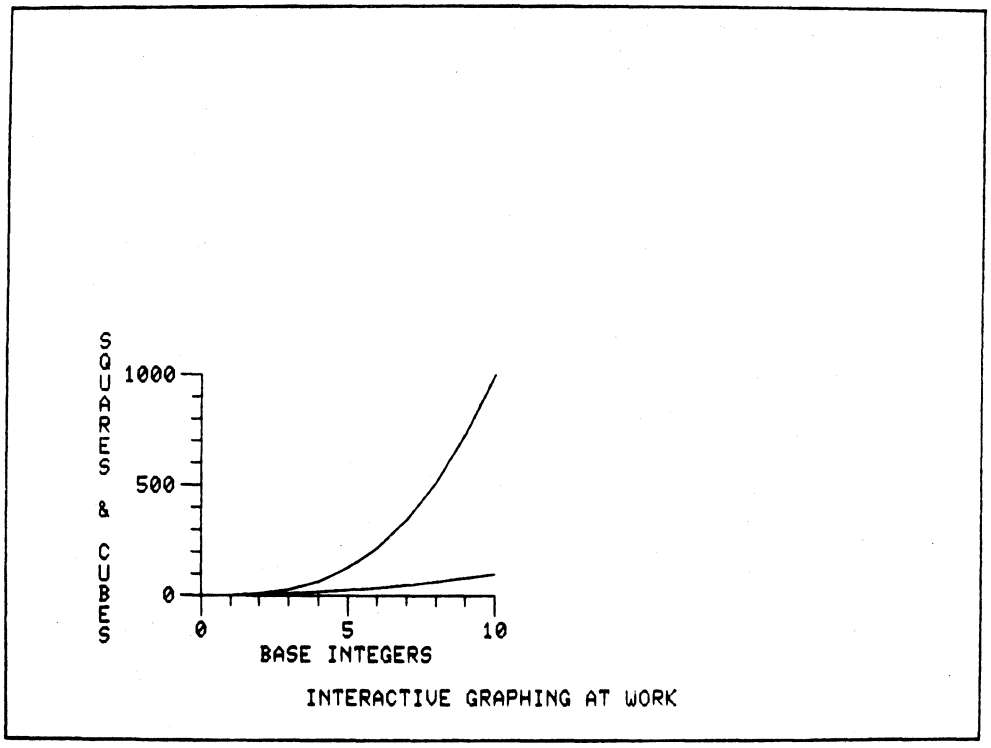
```
DISPLAY P1∂
```

The page and the graph are displayed with the notes in the specified positions:

USING INTERACTIVE GRAPHING

ENTER, DEFINE, INCLUDE, DISPLAY

Example:



The INCLUDE command, as we have seen, can take many forms. A synopsis of these forms is shown below:

INCLUDE Partname in Partname [HORIZONTALLY / VERTICALLY] [START / with CENTER at X,Y / END / at X,Y and X,Y]

where X and Y are expressed as a percentage with relation to a larger, inclusive part. Brackets indicate optional elements, depending on context and what is being included. The underlines indicate the default or initial, setting.

FRAMING A GRAPH

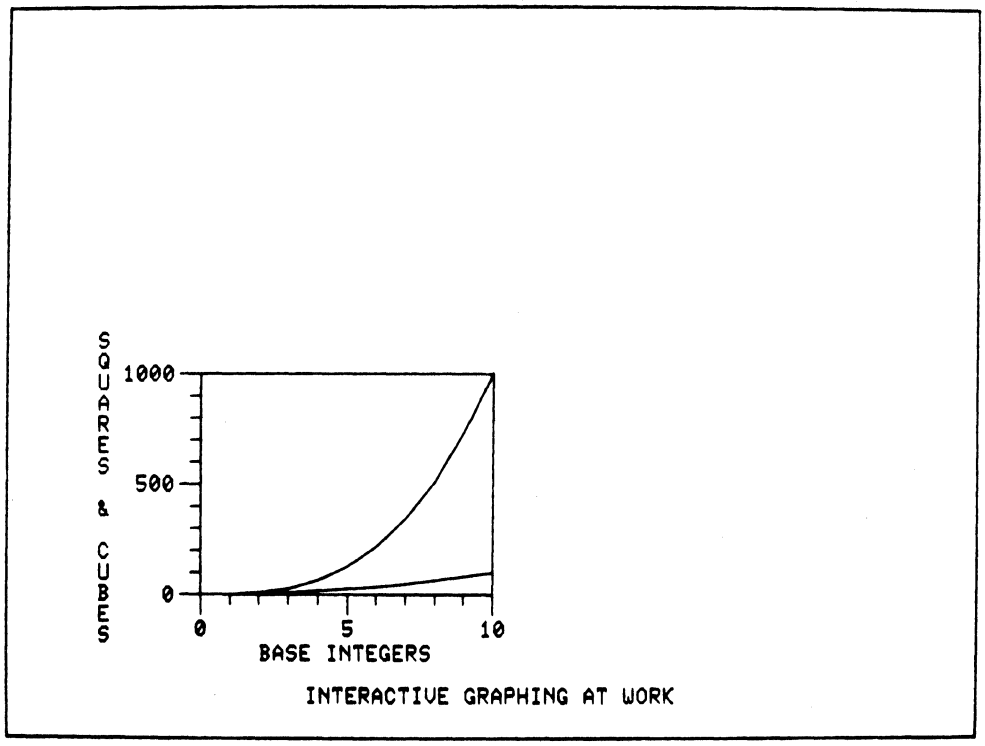
The frame of a graph is simply the completion of a box drawn around the graph. The frame is not named or positioned.

To frame in the example graph, G1, the following sequence would be used:

INCLUDE a FRAME in G1$_\partial$

DISPLAY P1$_\partial$

Example:



DISPLAYING SUB-PARTS

Most of the displays used in the example have been by page, and some have been by graph. You may wish to display only a part of a page or a graph. Every subpart (member) can be displayed by itself. However, the position it occupies on the screen will not necessarily be the one it will hold when included in a page. You may, for example, display a curve before including it in a graph in order to view its shape. In general, it is more valuable to display by page or graph. Axes cannot be displayed alone.

ENTER, DEFINE, INCLUDE, DISPLAY

CONCATENATING COMMANDS

Commands in Interactive Graphing may be concatenated, or joined together, in one line. The commands must be separated by a semicolon (;).

Example:

INCLUDE G1 in P1; DISPLAY P1

POSITION

The POSITION command is used to move an already included part relative to the part of which it is a member. It is very like the INCLUDE command. The forms of the command are shown below:

POSITION Partname [HORIZONTALLY / VERTICALLY] [START / with CENTER at X,Y / END / at X,Y and X,Y]

where X and Y are expressed as a percentage with relation to a larger, inclusive part. Brackets indicate optional elements, depending on context; stacked operands indicate that one and only one must be chosen. The underlines indicate the default, or initial, setting.

Using the example page and graph (P1 and G1) we will move both the text of the page and the graph to demonstrate the use of POSITION. The text (T1) was included at the bottom of the page at percentages coordinates 50,5. Now we wish to center it at the top of the page. Type,

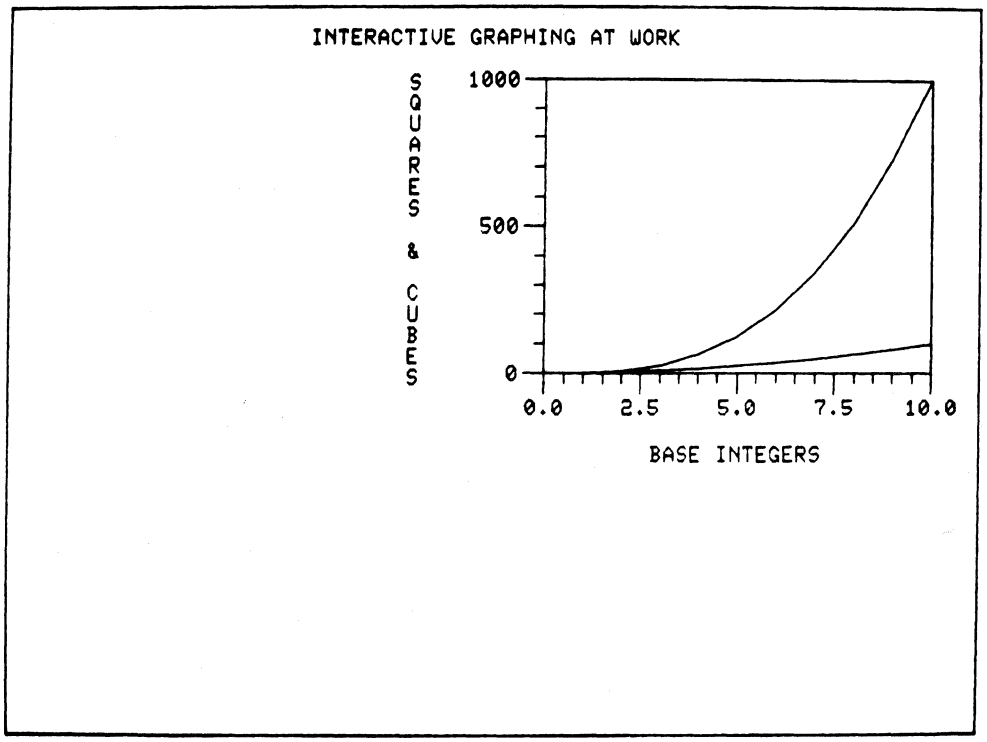
```
POSITION T1 at 50,95
```

The graph was included with lower left corner at 20,20 and upper right corner at 50,50. Now we wish to move the graph to the upper right quadrant of the page. Type,

```
POSITION G1 at 55,50 and 95,90∂
DISPLAY P1∂
```

The page with repositioned text and graph is displayed:

Example:



POSITION

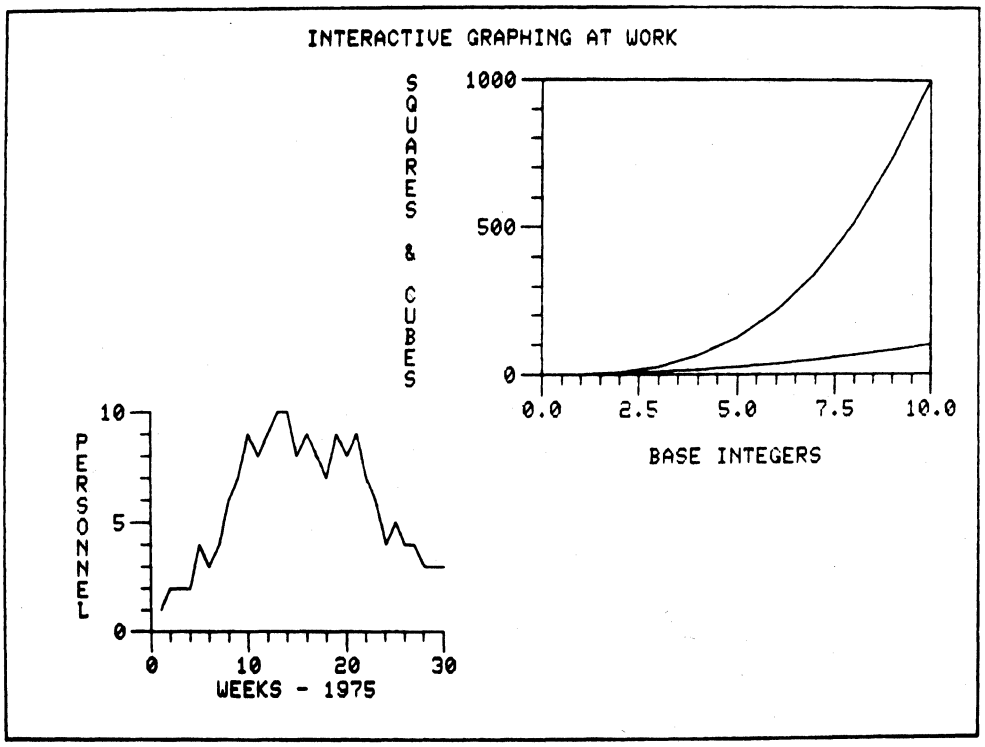
CREATING ANOTHER GRAPH ON A PAGE

Suppose we desire to include another graph on the example page (P1). This graph will show the number of personnel required and the available manpower resources for a project over a thirty week period. The following sequence of commands will create and position the graph:

DEFINE a GRAPH called PROJECT
ENTER TIME, PEOPLE: 1,1 2,2 3,2 4,2 5,4 6,3 7,4 8,6 9,7 10,9∂
11,8 12,9 13,10 14,10 15,8 16,9 17,8 18,7 19,9 20,8 21,9∂
22,7 23,6 24,4 25,5 26,4 27,4 28,3 29,3 30,3∂
DEFINE a CURVE called REQUIRED as TIME,PEOPLE∂
INCLUDE REQUIRED in PROJECT∂
DEFINE a NOTE NTIME as 'WEEKS – 1975'∂
DEFINE a NOTE NPERS as 'PERSONNEL'∂
INCLUDE NTIME in PROJECT at 50,–30∂
INCLUDE NPERS VERTICALLY at –25,50∂
INCLUDE PROJECT in P1 at 15,15 45,45∂
DISPLAY P1∂

The graph Project is displayed.

Example:



POSITION

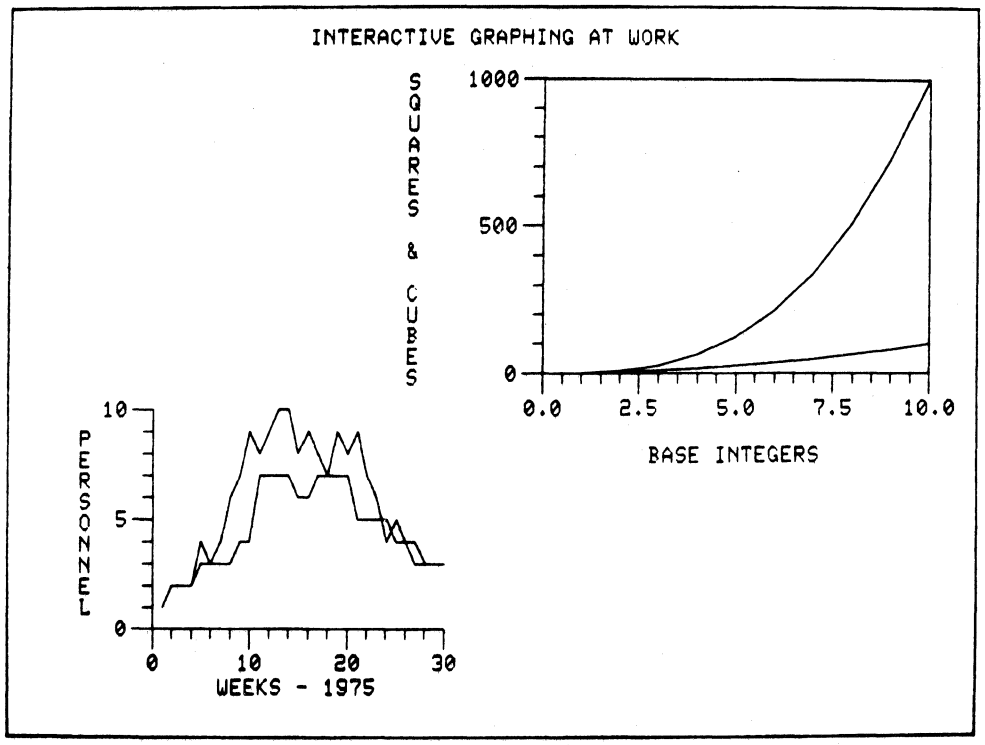
ENTER AVAILABLE: 1 2 2 2 3 3 3 3 4 4 7 7 7 7 6 6 7 7 7 7∂
5 5 5 5 4 4 3 3 3 3∂
DEFINE a CURVE called RESOURCE as TIME, AVAILABLE∂
INCLUDE RESOURCE in PROJECT∂
ERASE; DISPLAY P1∂

NOTE

In the ENTER command above no commas were used between data values. This is because either a comma or a space is sufficient to delimit data values.

The page is displayed showing both graphs and their sub-parts.

Example:



POSITION

LIST

Once a part has been given a name and been defined, it cannot be redefined. It can however, be deleted and defined again. The LIST command allows you to see what partnames have already been defined. You can list all the partnames in use with the following command:

LIST ALL∂

You can list the members of a part by specifying the part:

LIST P1∂

```
          LIST P1
PAGE              -- P1
          GRAPH       -- G1
           55,0,      50.0
           95.0,      90.0
          TEXT        -- T1
                CENTER
           50.0       95.0
          GRAPH       -- PROJECT
           15.0,      15.0
          45.0,       45.0
```

LIST PROJECT∂

```
          LIST PROJECT
GRAPH             --PROJECT
     AXIS           --XPROJECT
     AXIS           --YPROJECT
     CURVE          --REQUIRED
     NOTE           --NTIME
               CENTER
        50.0,    - -30.0
     NOTE           --NPERS
               VERTICAL
                CENTER
      -25.0,     50.0
     CURVE          --RESOURCE
```

POSITION

You can list all the names of a specified part in current use:

LIST ALL PAGE$_{\partial}$

LIST ALL PART$_{\partial}$

The last use of LIST allows you to determine which partname is the current context. However, when the context is in doubt, you should always specify a part by its name to ensure complete accuracy.

SET

The SET command applies only to graphs and their members. The command affects the axes of the graph and the individual curves included in the graph. By means of the SET command you have the capability of adding the touches to graphs that make them individually distinct, truly informative, and visually appealing. The context dependency of Interactive Graphing makes the SET command easy and efficient to use.

DEFINITIONS OF SET PARAMETERS

Set allows you to adjust the values of axis and curve parameters. Premissable parameters are:

AXIS MAJOR TIC FORM
AXIS MINOR TIC FORM
AXIS MAJOR TIC NUMBER
AXIS MINOR TIC NUMBER
AXIS DENSITY
AXIS TRANSFORMATION
AXIS LABEL TYPE
AXIS LABEL WIDTH
AXIS RANGE MINIMUM
AXIS RANGE MAXIMUM
AXIS EXPONENT TYPE
AXIS LOCATION

CURVE LINE FREQUENCY
CURVE LINE STYLE
CURVE LINE SIZE

CURVE SYMBOL STYLE
CURVE SYMBOL FREQUENCY
CURVE SYMBOL SIZE

The relationship of these parameters is shown in the following chart:

PARTS		SET PARAMETERS	
	TERMINAL	MODEL BAUD FONT	
		PERIPHERAL	
GRAPH	AXIS	TIC – – – – – – – – – – – – – – – –	MAJOR: FORM ..... NUMBER
			MINOR: FORM ..... NUMBER
		DENSITY	
		TRANSFORMATION	
		LABEL – – – – – – – – – – – – – – – –	TYPE WIDTH
		RANGE – – – – – – – – – – – – – – – –	MINIMUM MAXIMUM
		EXPONENT	
		LOCATION	
	CURVE	LINE – – – – – –	FREQUENCY STYLE SIZE
		SYMBOL – – – – – – – –	FREQUENCY STYLE SIZE

SET

The value of a parameter is the number to which it is set. For example, if the value of a curve symbol style is 5, the symbol drawn at each data point on the curve is a star.

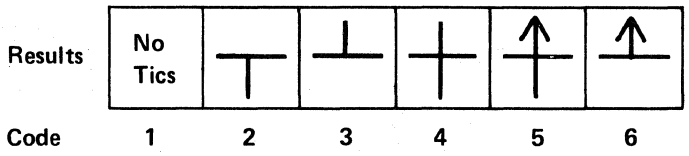
Definitions of the permissable parameters are listed below:

AXIS MAJOR TIC FORM – AXIS MINOR TIC FORM

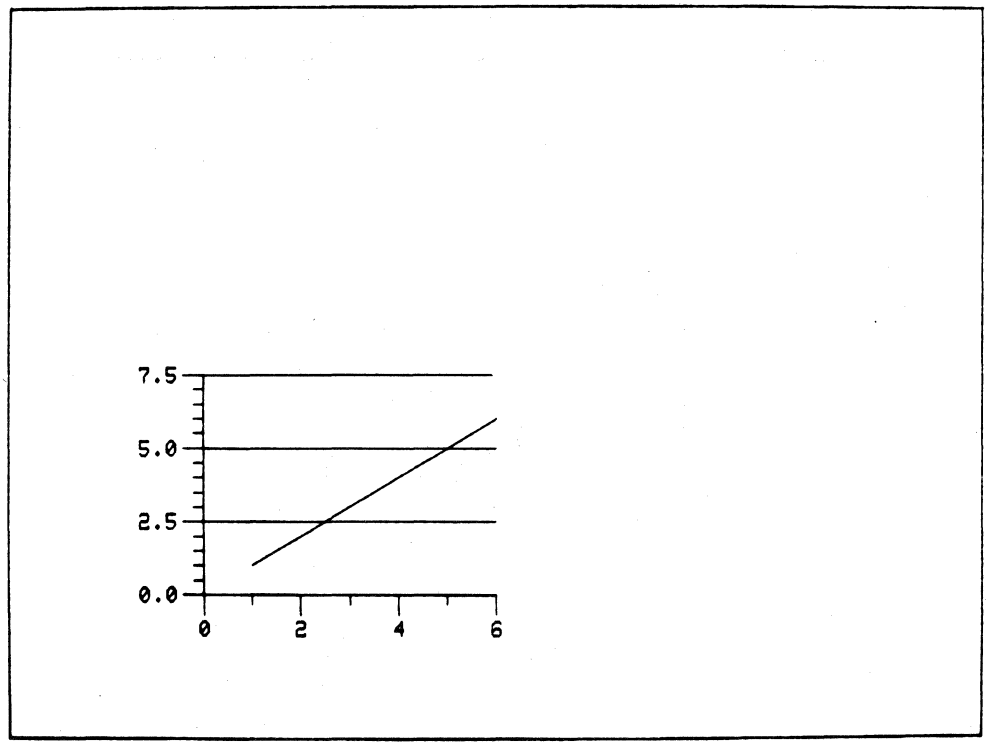
The major or minor tic mark form of either axis may be set to one of the following. The standard major tic form is type 2; the standard minor tic form is also type 2.

TIC MARK FORM

Valid values are numbers 1 through 6 as shown on the chart below. Types 5 and 6 include grid lines.



Example: The major tics on the Y axis are set to 5.



AXIS MAJOR TIC NUMBER – AXIS MINOR TIC NUMBER

By setting this parameter you can specify the exact number of major or minor tic mark intervals for either the X or Y axis or both axes.

AXIS DENSITY

The major tic mark density may be set for either axis according to the following table. Unless you reset it, the tic mark density is 8 for both axes.

TIC MARK DENSITY TABLE

	No Minor Tics	With Minor Tics
Sparse	1	6
↓	2	7
	3	8
	4	9
Dense	5	10

NOTE

If the number of tic marks is set, the density setting will have no effect.

SET

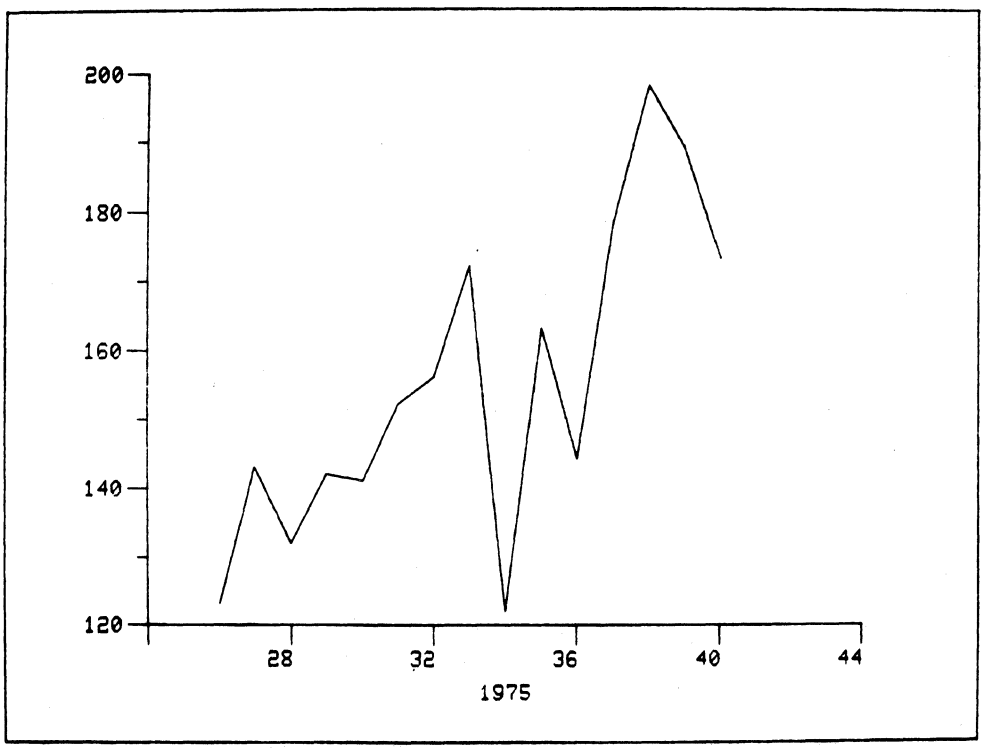
AXIS TRANSFORMATION

Axis transformation determines the scaling represented by each tic mark. The normal transformation is linear for each axis; that is, there is a regular distance represented by each tic mark. However, Interactive Graphing allows you to specify logarithmic scaling for either axis or both axes. A variety of calendar scalings are also available (see the Advanced Usage section of the manual). The following table shows the transformation types.

TRANSFORMATION TYPES

CODE VALUE	TYPE
1	Normal (Linear). Default setting
2 or LOG	Logarithmic (Ratio) Scale
3	Days
4	Weeks
5	Periods
6	Months
7	Quarters
8	Years

Example:



@

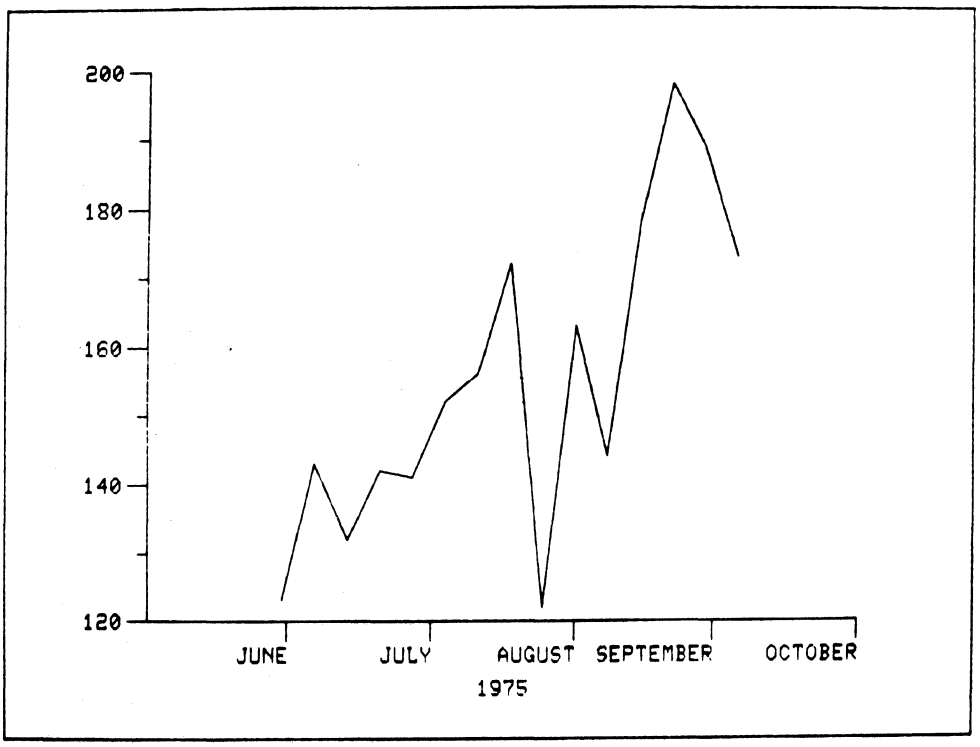
AXIS LABEL TYPE

The label type may be set for tic marks on each axis. It will normally correspond to the transformation type; however, you may wish to vary label type and transformation type, particularly for calendar data. For example, the transformation type may be set to 4 (weeks), but the label type may reflect 6 (months). The label types are shown in the following table:

LABEL TYPES

CODE VALUE	TYPE
0	No Label (Does not delete label in LOG)
1	Matches transformation type. Default setting
2 or LOG	Logarithmic
3	Days
4	Weeks
5	Periods
6	Months
7	Quarters
8	Years

Example :

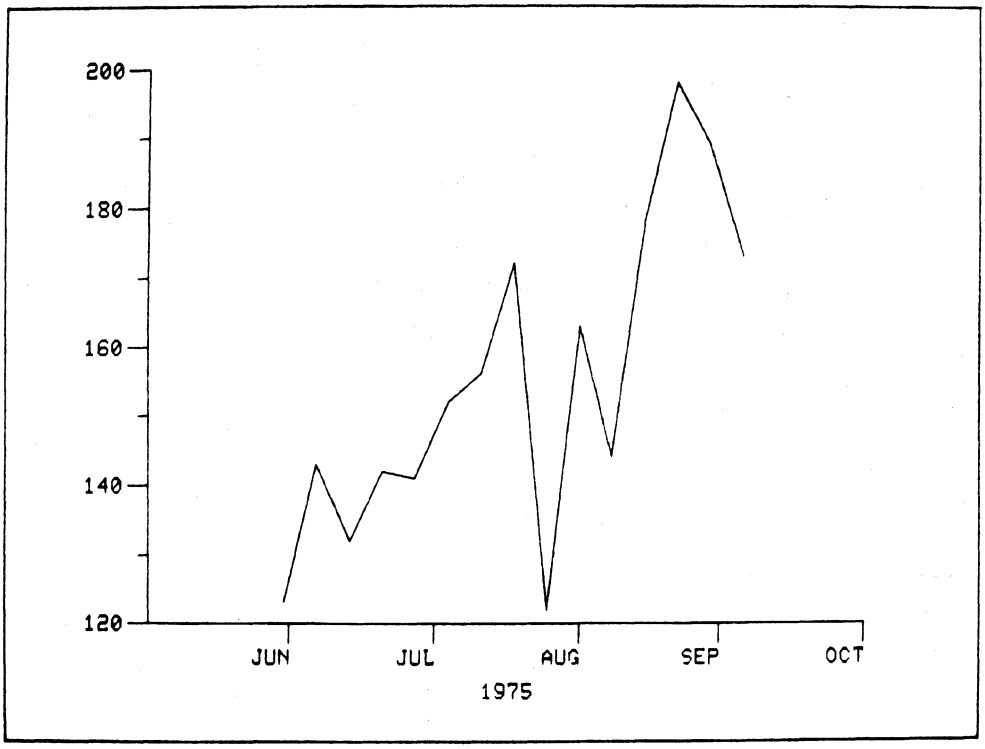


SET

AXIS LABEL WIDTH

You can specify the number of characters that will appear in a tic mark label. For example, suppose the axis label type is set to 6 (months), but you wish to see only the first three letters of each month on the graph (Jan., Feb, Mar, Apr, etc). You would simply set the axis label width to 3.

Example :



AXIS RANGE MINIMUM – AXIS RANGE MAXIMUM

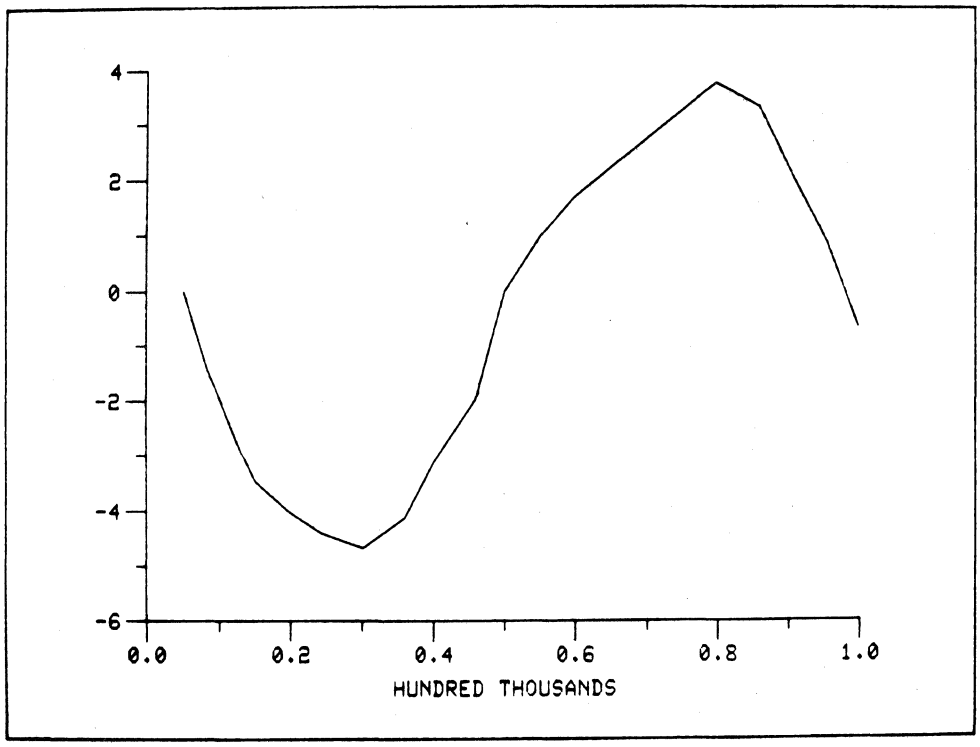
Interactive Graphing automatically sets the ranges, minimum and maximum, for each axis to suit the data range. You can reset the range in order to suit the data or to examine a part of the data (as in a zoom-effect). The minimum and maximum must be set in separate commands. However, the range set by a user may be overridden by Interactive Graphing to ensure an optimum tic interval.

AXIS EXPONENT

Setting the axis exponent type specifies the type of remote exponent label to be used on the appropriate axis. The remote exponent will appear at the center of the X axis set out slightly farther than the individual tic mark labels, or contered above the Y axis. The various types of exponents available and their code values are described in the following table:

VALUE	RESULTING TYPE
1	10^n. The default setting
2	M, MM's, etc. (1 M per thousand)
3	Printed words, e.g., HUNDREDS
4	10. . .0 (1 plus appropriate number of 0)

Example:



If the label type is logarithmic, this variable specifies the form of the major tic mark labels. No remote exponents are formed for logarithmic axes.

SET

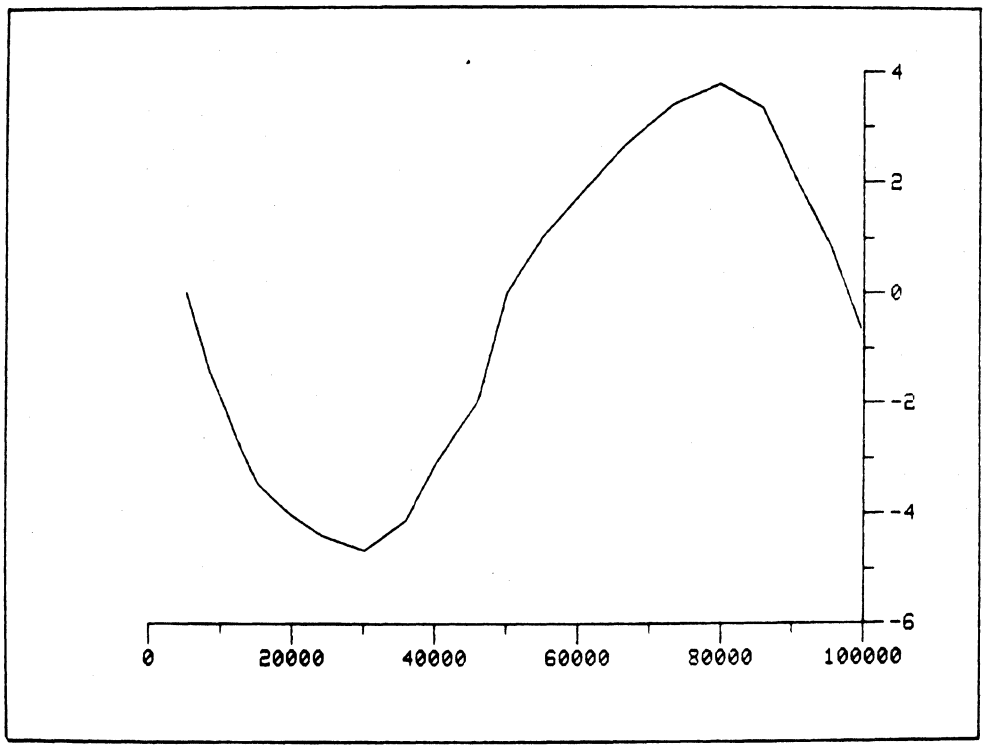
AXIS LOCATION

The axes of a graph are normally located at the left and bottom. You may, however, wish to move either or both of these axes. You can set the axis location to do this. The axis will be relocated according to the specified percentage relative to the size of the graph, right-left, or up-down. All other member parts of the graph will remain constant.

SET Yaxisname LOCATION TO 100

This command moves the Y-axis all the way to the right.

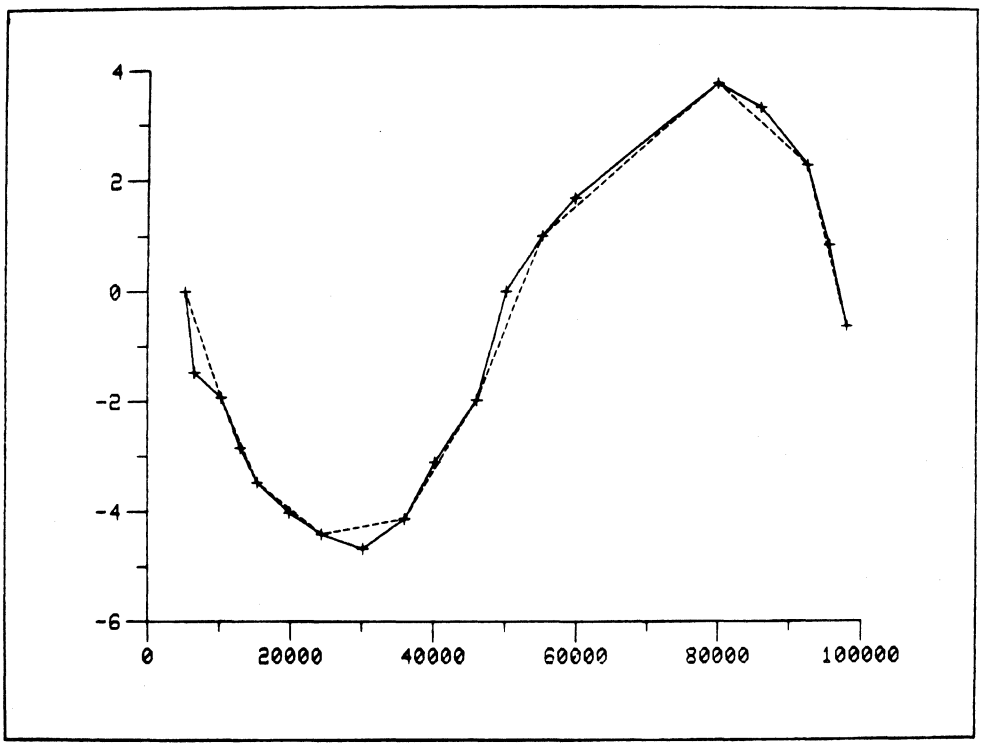
Example:



CURVE LINE FREQUENCY

Normally, a curve consists of lines drawn from each data point to the next; in this case then the curve line frequency is 1. You can, however, set the line frequency so that the lines of the curve are drawn between every other set of data points; in this case the line frequency should be set to 2. A line frequency of 3 causes lines to be drawn between every third set of data points, and so on. This feature may be especially useful for example, when you have a large number of data points.

Example:



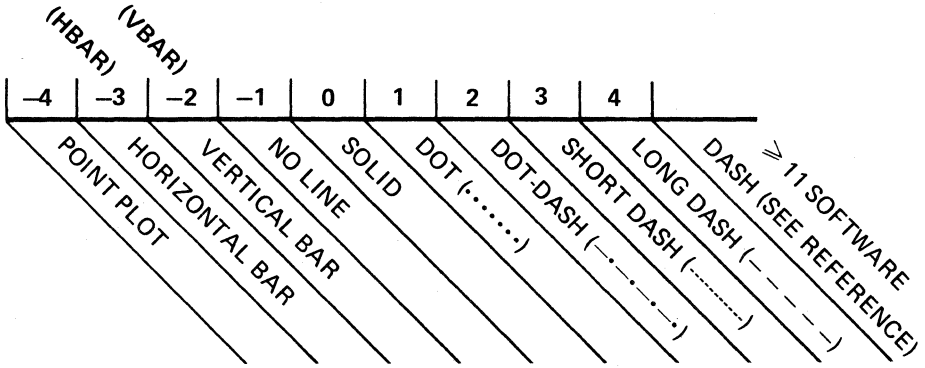
The dotted line indicates curve line frequency of 2.

SET

CURVE LINE STYLE

In order to distinguish between curves on a graph, you can specify a wide range of dashed line types, point plot, or vertical or horizontal bars. See the Reference section for information on both hardware and software dash types. The line style values are shown in the table below:

LINE STYLE SPECIFICATION



An example of the use of VBAR (a vertical bar line style) can be seen in the subsection Using the SET Command.

CURVE LINE SIZE

This parameter applies only to curves with line styles of VBAR or HBAR (that is, either vertical or horizontal bar, respectively). The bars are set to a width such that they touch. You can modify the width, however. Assume that the preset width is 100%. The width can be narrowed by specifying a percentage smaller than 100.

Example:

```
SET CURVE LINE SIZE TO 50
```

CURVE SYMBOL STYLE

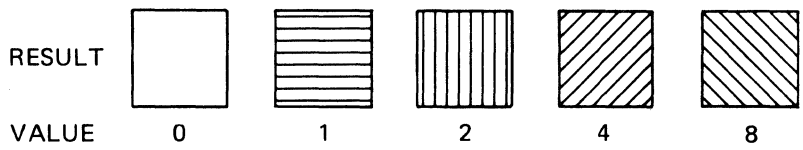
A special symbol may be specified which will be drawn for each data point in a curve. The following tables show each symbol and its respective value.

DATA POINT SYMBOLS

VALUE	SYMBOL	RESULT
0	None	No symbol
1	O	Drawn circle
2	X	Drawn "X"
3	△	Triangle
4	□	Square
5	☆	Star
6	◇	Diamond
7	|	Vertical bar
8	†	Cross
9	↑	Up arrow below point
10	↓	Down arrow above point
11	▽	Reverse Triangle

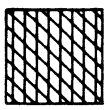
33 → 126 SYMBOL IS CORRESPONDING ASCII CHARACTER

BAR CHART SHADING (SYMBOL STYLE)



OTHERS ARE COMBINATIONS OF THESE UP TO 15

Example:

2 + 8 = 10 = 

SET

CURVE SYMBOL FREQUENCY

Normally, curve symbols are drawn at every data point if they are requested by the user; that is, their frequency is 1. Setting the frequency to 2 causes symbols to be drawn at every other point, setting it to 3 at every third point, and so on.

CURVE SYMBOL SIZE

Setting the size of curve symbols causes a rescaling of the symbol by the specified factor. Normal size is 1. To double the size of a symbol, you would specify a value of 2 for symbol size, and so on. To halve the size, you would use a value of 0.5.

USING THE SET COMMAND

The SET command is easy to use and versatile. Word order (syntax) is highly variable, according to the user's wishes; the only restriction on syntax is that the verb, SET, must appear first. The general form of the command is the following:

SET the Parameter of Partname to Value

Given the general form, the following variations are possible:

1. Word order is independent, excepting only that SET occurs first in the command.

2. Because of context dependency once a partname is established as the working context, it does not need to be reestablished in successive commands until the context changes (see Context, General Information).

Example:

SET THE RANGE MAXIMUM of Y G1 to 10000_{∂}
SET the MAJOR TIC FORM to 5_{∂}
SET the DENSITY to 2_{∂}

Using the example graphs, G1 and Project, developed in the preceding subsections, we will make some useful additions by means of the SET command. First, as a memory refresher, type

DISPLAY G1∂
LIST $G1_{\partial}$
LIST $CUBES_{\partial}$
LIST $SQUARES_{\partial}$

The above commands demonstrate what there is to work with in the first graph, G1. Given this information we will make the following additions:

(a) resetting the Y axis range maximum
(b) Y-axis logarithmic scaling
(c) vertical bars for both curves
(d) a legend

First, set the range:

SET the MAXIMUM RANGE of YG 1 to 10000$_\partial$

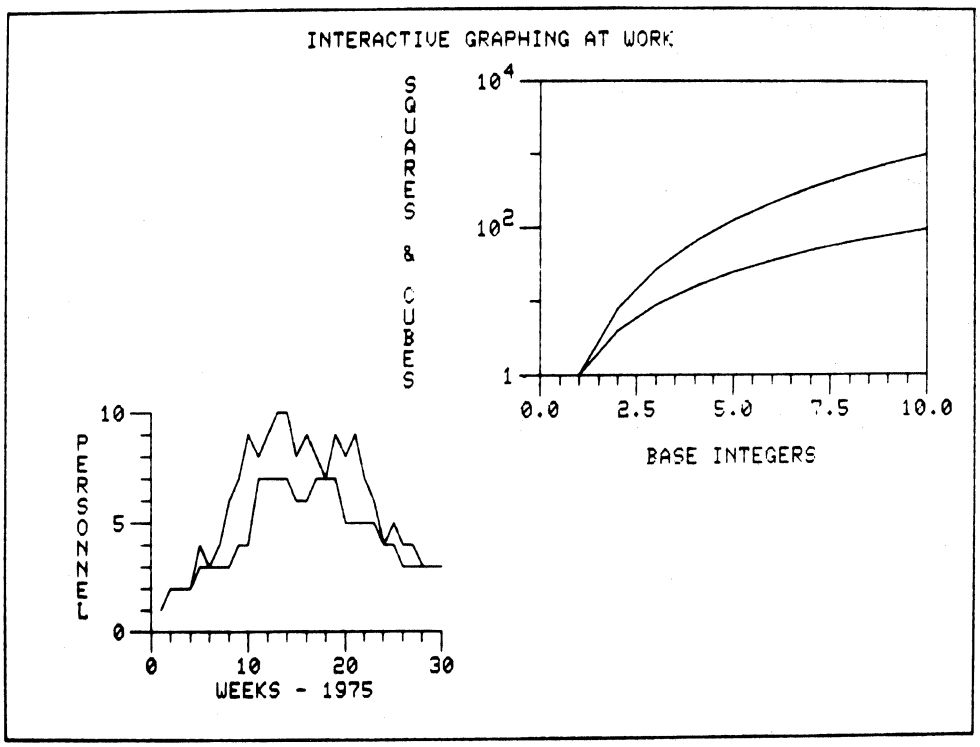
Give the Y-axis logarithmic scaling:

SET the TRANSFORMATION to LOG$_\partial$

NOTE

The Y-axis did not have to be re-specified since it is the current context, as is the graph, G1.

DISPLAY P1$_\partial$



SET

Change both curves to vertical bars:

SET the LINE STYLE of CUBES to VBAR∂
SET the LINE STYLE of SQUARES to VBAR∂

NOTE

Whenever you are unsure of the context in a command, it is wise to specify all of the desired contexts, as in the above example. In the above case the following command would be incorrect:

SET the LINE STYLE OF CUBES to VBAR: SQUARES to VBAR∂

or

SET the LINE STYLE of CUBES and SQUARES to VBAR∂

You need to specify both contexts, CUBES and SQUARES, with individual commands.

Having specified vertical bars for both curves, it would be helpful to distinguish one curve from the other visually. Therefore, let us shade the bars of one of the curves:

SET the SYMBOL STYLE of SQUARES to 8∂

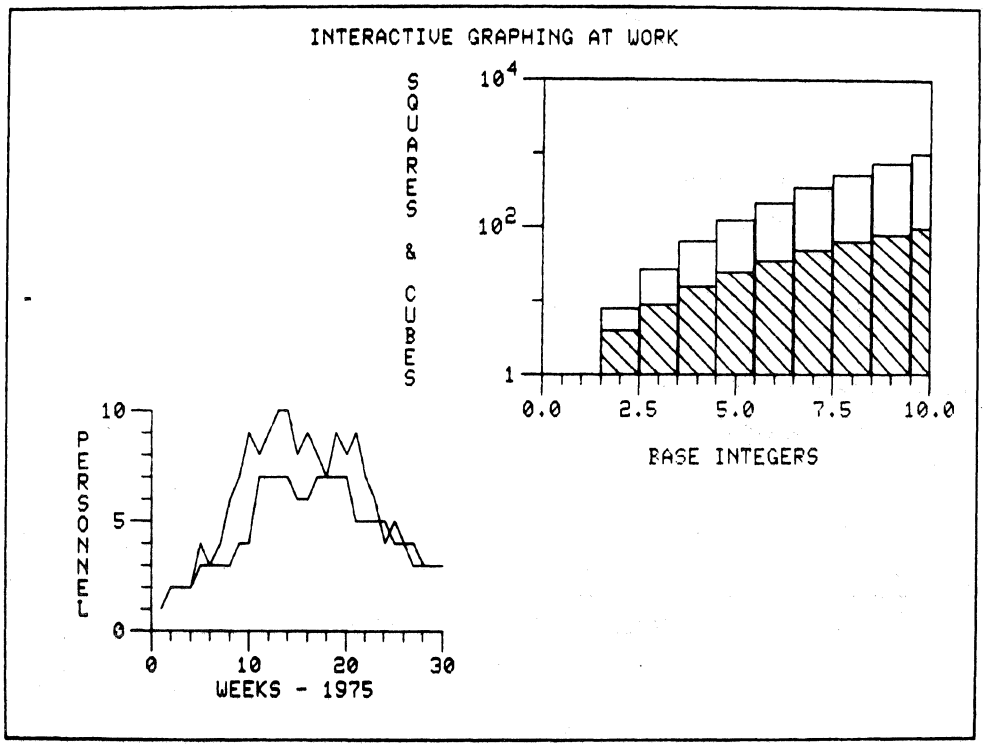
NOTE

Because the context has been established above, the command could have been entered as

SET the SYMBOL STYLE to 12∂

The graph is now displayed as it appears on the page, P1:

DISPLAY P1∂



Now let us turn to the second graph, Project.

For this graph the following parameters will be set:

major tic number
line style
symbol style

The X-axis of Project (XPROJ) has thirty data values; the axis tic mark number can be set to 7 for convenience.

SET the MAJOR TIC NUMBER of XPROJECT to 7_{∂}

SET

The Y-axis (YPROJ) has a range of from 1 to 10 values; therefore, the tic number may be most conveniently set to 10.

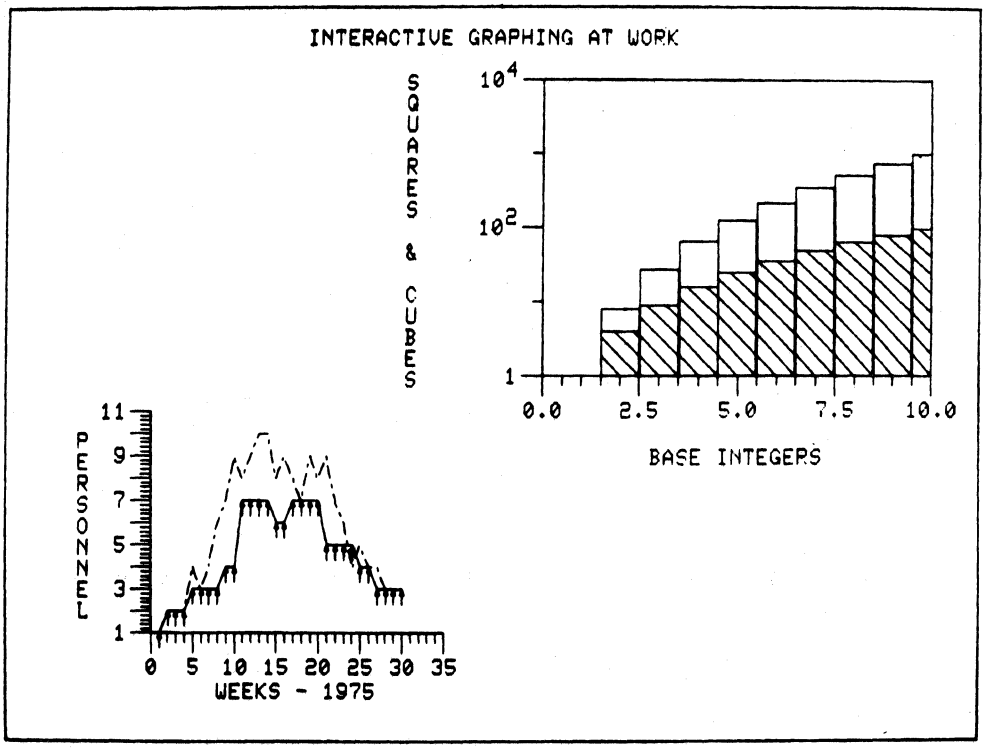
SET the MAJOR TIC NUMBER of YPROJ to 10_{∂}

There are two curves included in the graph, REQUIRED and RESOURCE. It would be useful to be able to distingusih between them visually; therefore, we will give REQUIRED a dashed line.

SET the LINE STYLE of REQUIRED TO 2_{∂}

Now, we will insert an UP arrow symbol at each data point on the other curve.

SET the SYMBOL STYLE of RESOURCE TO 9_{∂}



INCLUDING A LEGEND IN A GRAPH

A Legend consists of a table associated with a graph which identifies the line styles of the curves in the graph. A Legend must be included or positioned relative to its associated graph. The Legend shows the line and symbol styles of each curve and the first eight characters of the curve name. In the example graph, G1, there are two curves to be identified with a Legend.

INCLUDE a LEGEND in G1 with END at −70,90∂

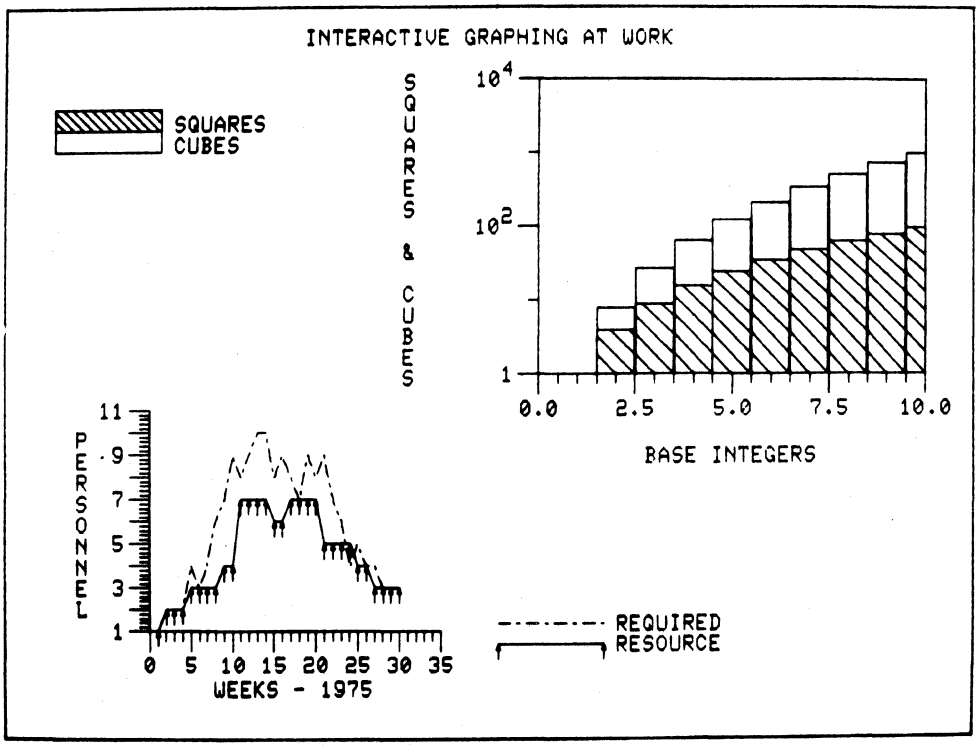
where −70 is the percentage relative to G1 in the X direction, and 90 is the percentage relative to G1 in the Y direction.

In the example graph, Project, a Legend would also be valuable.

INCLUDE a LEGEND in PROJECT with START at 120,10∂

where 120 is the percentage relative to Project in the X direction, and 10 is the percentage relative to Project in the Y direction.

DISPLAY P1∂



APPEND

The APPEND command is used to add values to the end of an already existing curve. The general form of the command is as follows:

APPEND Xname(,Yname, . . .Zname):Xvalue(,Yvalue, . . .Zvalue)

Like the ENTER command APPEND requires a colon (:) before the first data value.

The Xname, Yname, and Zname are, if used, the same as the column names that were used to load the curve. As with the ENTER command APPEND may be used to add data to an individual data column or to add data in X,Y pairs.

Using the example graph, G1, let us add two more data points to each of the two curves:

APPEND X1,Y1:11,121 12,144∂

APPEND YCUBE:1331 1728∂

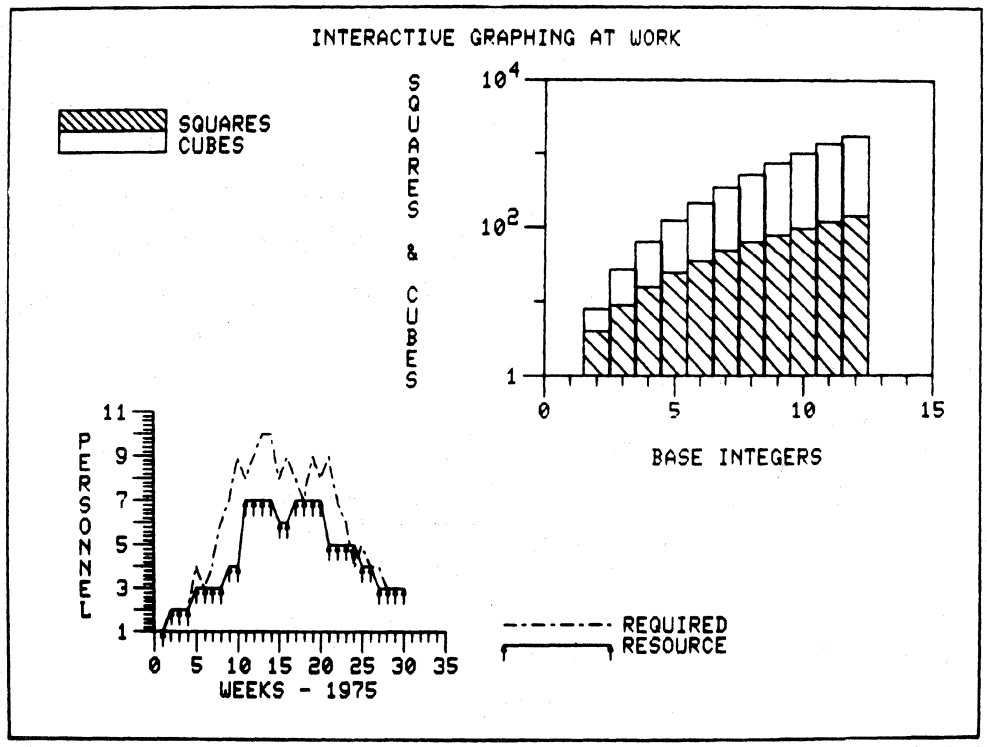
NOTE

Remember that the two curves which make up G1 consist of three columns, two Ynames plotted against one Xname. The APPEND command above could, therefore, have been entered as the following:

APPEND X1, Y1, YCUBE:11,121,1331 12,144,1728∂

The graph can now be displayed with the two added data points:

ERASE: DISPLAY



@

EXCLUDE

The EXCLUDE command is the opposite of INCLUDE. It causes the named sub-part to be excluded from membership in its associated part. The excluded sub-part, however, still exists. It may be included in any appropriate part. In the case of the graph, G1, for example, we might wish to exclude the curve, Cubes. First, we will change the line style of the curves in the graph to a solid line. Then we exclude both the curve, Cubes, and the legend of the graph.

```
SET the LINE STYLE of SQUARES to 0
EXCLUDE CUBES
EXCLUDE LEGEND
ERASE; DISPLAY
```

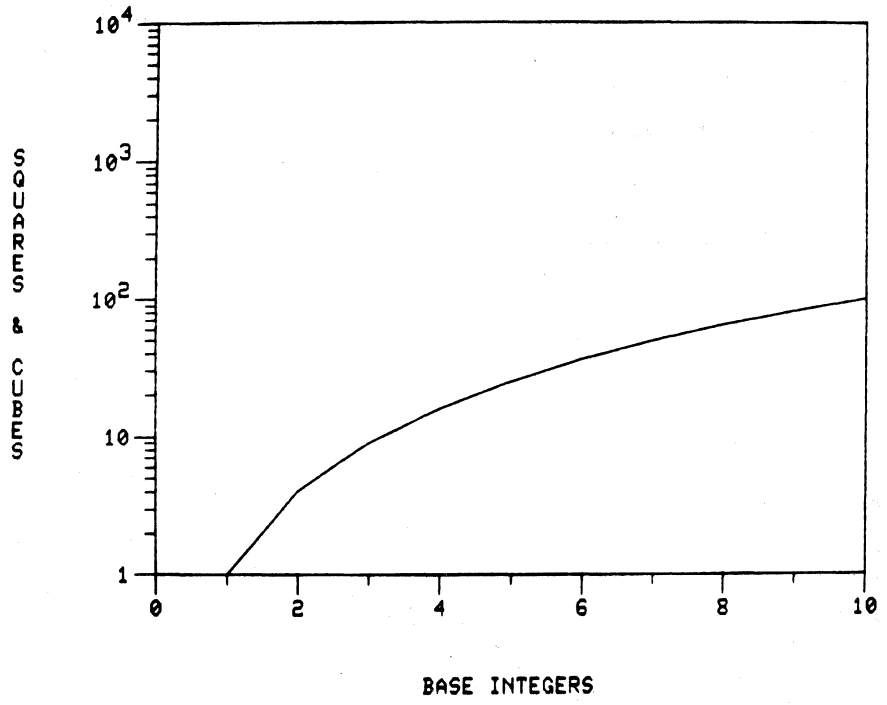
NOTE

Because of the context established, we did not have to type,

EXCLUDE CUBES from G1

or

EXCLUDE LEGEND from G1



The general form of the EXCLUDE command is the following:

EXCLUDE Partname from Partname

CHANGE

The CHANGE command can be used for altering, inserting, or removing data points in the specified curve.

NOTE

For CHANGE to work correctly, the graph containing the curve to be changed must be the most recently displayed graph.

The general form of the command is the following:

CHANGE X,Y in Curvename to: X_1,Y_1

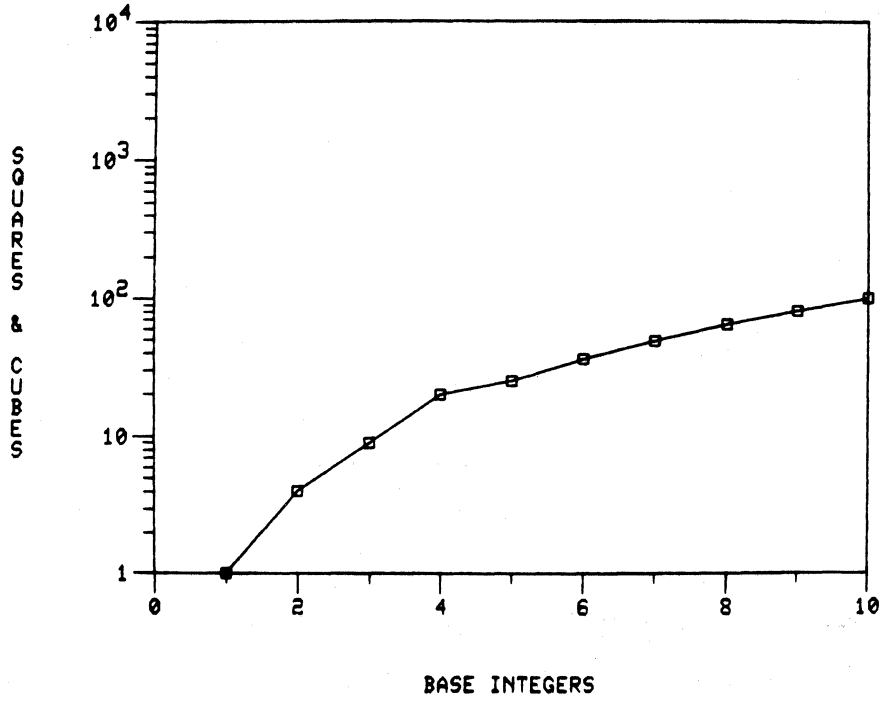
ALTERING DATA POINTS

The CHANGE command can alter one or more data points. To do this using the example graph, G1, we will change the point (4,16) to (4,20).

CHANGE 4,16 in SQUARES to: 4,20

The colon (:) must be included in the change command. Its position in the command is immediately before the first new data point. In the displayed graph below, the curve symbol style has been set to 4 so that small squares appear at each data point.

ERASE; DISPLAY

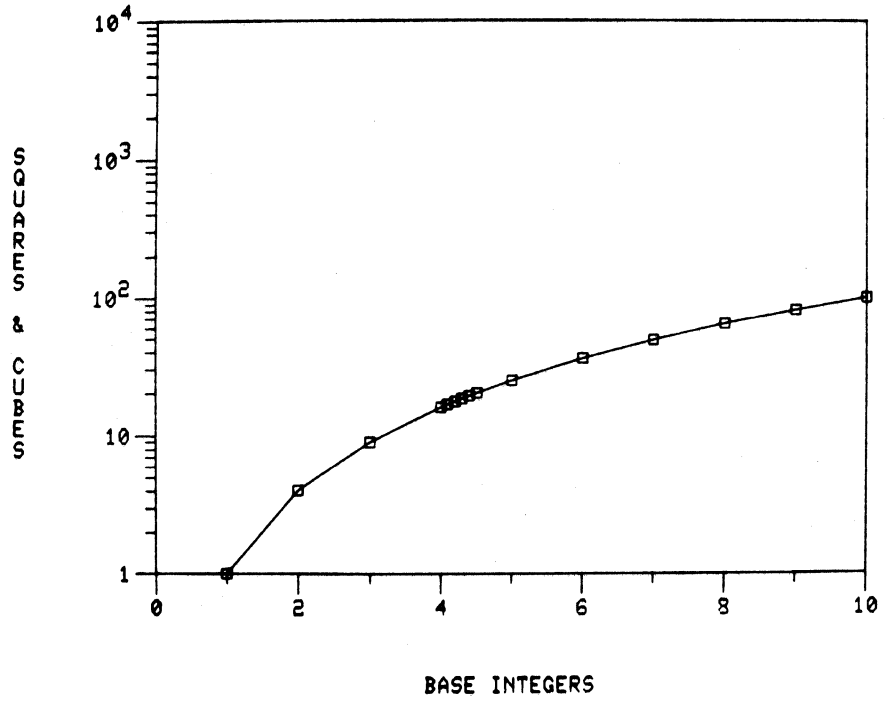


CHANGE

INSERTING DATA POINTS

CHANGE can be used for inserting a series of data points in a curve. The same general form of the command as used above applies. For example,

```
CHANGE 4,20 (in SQUARES) to: 4,16 4.1,16.81 4.2,17.64 4.3,18.49
4.4,19.36 4.5,20.25
ERASE; DISPLAY
```



CHANGE

REMOVING DATA POINTS

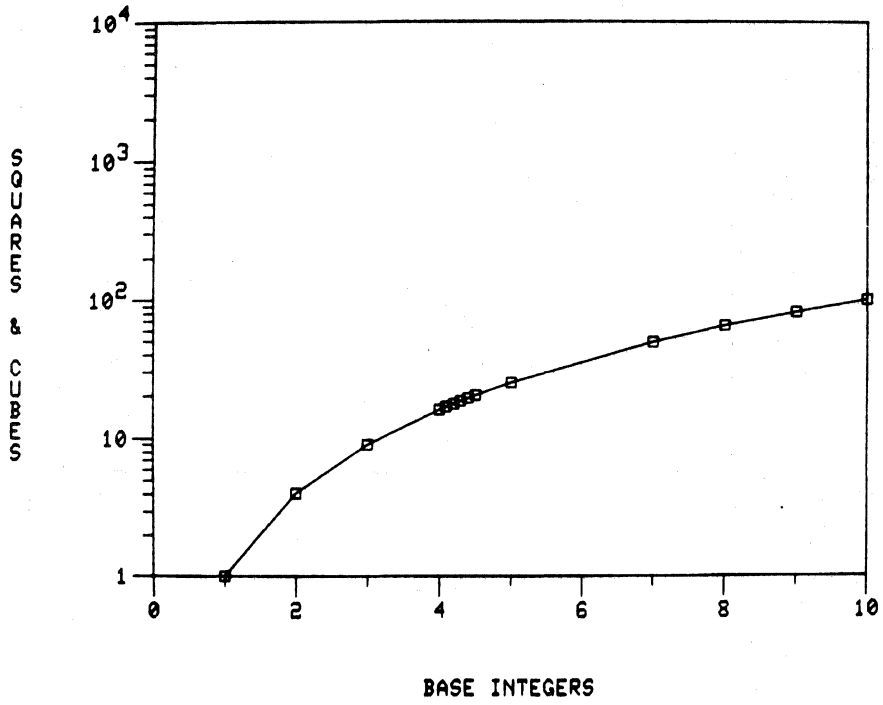
CHANGE can also be used to remove a data point. The command must be used once for each data point to be removed. For example,

CHANGE 6,36 (in SQUARES)

NOTE

When a data point is removed from the curve, no colon (:) is necessary.

ERASE;DISPLAY



CANCEL, DELETE

The CANCEL command allows you to delete all the information entered since the last command was executed. The command consists of the single word, entered anywhere in a command sentence. All of the information preceding the discard is ignored, and all of the information following the discard (if there is any) is executed following the sentence ending carriage return. For example,

SET the DENSITY of XG1 to 2 CANCEL DEFINE a LEGEND at L1$_{\partial}$

In the above command the SET command is ignored, and the DEFINE command is executed.

NOTE

The CANCEL command works as described above with the following exception: When used to cancel an ENTER, APPEND, or CHANGE command, CANCEL will be ineffective after you type the colon. For example,

ENTER XDATA,YDATA: 2,24 3,48 CANCEL

The command to enter the data was not cancelled.

DELETE

This command is the opposite of DEFINE. It causes the named part to be deleted entirely. The only way to use the part again thereafter is to redefine it. For example,

DELETE XDATA

The part XDATA no longer exists. However, references to XDATA are not removed. Suppose a curve, SALES, includes XDATA and YDATA as its data columns. If you were to display SALES, an error message would be displayed instead, since XDATA no longer exists. You would then redefine XDATA. This is a useful feature when you wish to plot a new set of points but do not want to go to the trouble of renaming the data columns.

COPY, RENAME

The COPY command is used to create a duplicate part with a new name from an already existing part. The new part has the same sub-parts associated with it as the old part has. The general form of the command is the following:

COPY Oldpartname to Newpartname

For example, if we wish to create a new page, a duplicate of P1, we would type,

COPY P1 to P2∂

Both pages now exist with different names.

The user may wish to duplicate or modify a graph and include both graphs on the same page:

COPY PROJECT to PROGRESS∂
INCLUDE PROGRESS in P2 at 20,60 and 40,90∂

The following example demonstrates that when a part is copied to a new part, all the sub-parts associated with it remain the same:

```
      LIST G1
GRAPH       -- G1
      AXIS          -- XG1
      AXIS          -- YG 1
      CURVE         -- SQUARES
      NOTE          -- NX
            CENTER
        50.0,   -30.0
      NOTE          -- NY
            VERTICAL
             CENTER
        -35.0,      50.0
                      -- FRAME
      COPY G1 TO NEWG1

      LIST NEWG1
GRAPH       -- NEWG1
      AXIS        -- XG1
      AXIS        -- YG1
      CURVE       -- SQUARES
      NOTE        -- NX
```

```
        CENTER
    50.0,       -30.0
NOTE            -- NY
        VERTICAL
          CENTER
   -35.0,       50.0
                -- FRAME
```

RENAME

The RENAME command is used to change the name of an existing named part and change all references to that named part to correspond. The general form of the command is as follows:

RENAME Oldpartname to Newpartname

Example:

RENAME YDATA3 to YCUBES

NOTE

RENAME is useful for changing the name of a part in order to avoid a name conflict when a page is recalled.

The following example demonstrates how RENAME changes all references to a part to correspond with its new name:

```
      LIST ALL COLUMN
COLUMN          -- X1
COLUMN          -- Y1
COLUMN          -- YCUBE
COLUMN          -- TIME
COLUMN          -- PEOPLE
COLUMN          -- AVAILABLE
      LIST SQUARES
CURVE           -- SQUARES
  COLUMN            -- X1
  COLUMN            -- Y1
 SYMBOL         STYLE
              4.00000000
RENAME Y1 TO YSQUARES
```

COPY, RENAME

```
      LIST ALL COLUMN
COLUMN          -- X1
COLUMN          -- YSQUARES
COLUMN          -- YCUBE
COLUMN          -- TIME
COLUMN          -- PEOPLE
COLUMN          --AVAILABLE
      LIST SQUARES
CURVE           -- SQUARES
  COLUMN            -- X1
  COLUMN            -- YSQUARES
 SYMBOL         STYLE
              4.00000000
```

STORE, RECALL

The STORE command is used to write the named page to a named system file. All information to be stored must have been included previously in a page. The general form of the command is as follows:

STORE Pagename (,Pagename . . .) into FILE Filename

ALL into FILE Filename

NOTE

Filenames are limited to a length of four characters.

Example:

```
   STORE P1 FILE MAN
FILE              -- MAN
FILE ALREADY EXISTS
ADD ON OR REPLACE
? REPLACE

PAGE STORED
  PAGE            --P1
```

STORE ALL into FILE Filename$_{\partial}$

This command stores copies of all pages into the referenced file.

NOTE

All information in internal storage is lost at the end of each on-line session unless it can be stored in a file for later use. For some computer systems, a list of keywords available for file names must be developed at implementation.

RECALL

The RECALL command is used to retrieve the named page from a bulk storage file. The general form of the command is as follows:

RECALL Pagename (,Pagename,Pagename. . .) from FILE Filename

Example:

RECALL P1 from FILE F3$_{\partial}$

STORE, RECALL

If you wish only to know which pages exist in a file, type

RECALL FILE Filename

This causes a directory of pages in the named file to be printed on the screen. However, no pages are read into memory.

Example:

```
        RECALL FILE MAN
FILE          -- MAN
  FILE      DIRECTORY
  PAGE        -- P1
```

ADVANCED USAGE

INTRODUCTION

This section of the manual contains information about more advanced techniques of Interactive Graphing than have been discussed so far. It infers that you have already read the section headed Using Interactive Graphing and are generally familiar with the commands used to create, edit, and store graphic data.

The techniques and examples in this section are not intended to be comprehensive. In working with Interactive Graphing, you will undoubtedly discover ways of using the software for your own graphing applications that are not covered here. What this section is designed to do is to give you the tools needed for experimentation on your own.

Topics covered under Advanced Usage are:

- Missing Data Points
- Short Column Forms (Short form and Calendar form)
- Graphic Input
- Command/Data Files
- Multiple Axes
- A Sample Bar Graph

ADVANCED USAGE

MISSING DATA POINTS

Often collection of data is intermittent. That is, the data are not available over part of its range. Therefore, it may be advantageous to indicate the discontinuity by a corresponding break in the data curve of a graph. Such a break can be thought of as consisting of "missing" data points. As an example, suppose that payroll data for a plant is accumulated monthly and that the plant is closed during July for vacations. A curve indicating payroll information would be connected between each of the intervals through June. The line would then be continuous beginning with the month of August. The break would indicate the unavailable data for July.

To indicate a missing data value when you are entering or appending data, type two adjacent commas.

Example:

ENTER SALES: 100,110,200,,300,100,,,500

A missing value is indicated between 200 and 300 and two missing values between 100 and 500.

NOTE

(a) Although commas and spaces are often interchangeable in Interactive Graphing, only commas can be used for missing data points.

(b) Do not begin an input line with a missing data point.

Likewise, to indicate a missing data value at the beginning of an entered or appended series, type two commas immediately after the colon which introduces the series.

Example:

APPEND SALES:,,150, 110, 200

A missing value is indicated before 150

When using a graphic crosshair for input (see the Graphic Input section), you can indicate a missing data point by pressing the M key to trigger the crosshair.

For an application of missing data points, see example 2 under Plotting Data with Calendar Intervals, later in this section.

SHORT COLUMN FORMS

PLOTTING DATA AT REGULAR INTERVALS

For many graphing applications you may find that your data are gathered at regular intervals (for example, every two minutes, once a month, every foot, every liter). The data gathered in this way comprise only one data column, and this column can be plotted against an axis which consists of regular intervals. This kind of plot is sometimes called a Y-only data plot, because you are interested only in the Y-column data values. All you need to do to create the X column is to specify a starting point (the point at which data gathering begins) and a regular interval at which each Y value occurs. Of course, you could easily plot irregular X values against regular Y values in the same manner.

To avoid the necessity of generating a simple list of numbers to form a data column, you can use a variation of the standard column form called the Short form. With the Short form, only three values must be specified. They are: the number of points to be plotted, the value of the beginning point, and the increment between each successive point. A Short column form cannot be altered by the CHANGE or the APPEND command. To alter such a column, delete it and then re-enter it.

The Short column form is used with the ENTER command. The column for which the Short form is used must have a name, in the same way as a standard column must. Use of the Short column form is not only easier to set up, but it also saves computer memory space. The general form of the command is as follows:

ENTER Columnname SHORT number of points, beginning value, increment

Example:

ENTER ALTITUDE SHORT 20,0,100

For this example the column, Altitude, consists of 20 points. Its beginning point is 0. There is an increment of 100 between points. A colon should not be used when generating a Short form column. See the example under Graphic Input.

PLOTTING DATA WITH CALENDAR INTERVALS

One of the most common uses for graphing is for data plotted over a period of time (years, quarters, months, periods, weeks, or days). Interactive Graphing offers you the ability to set up a short form column for calendar data. Graphs plotted using the Calendar form are automatically labeled to match the calendar interval specified. Also, if you wish, you have the capability of specifying a calendar label that does not correspond to the way in which the data was gathered. For example, suppose your data were gathered on a daily basis over the course of a year. You might want to plot this daily data with labels that reflected weeks or months. You can do this simply by changing the axis label type.

SHORT COLUMN FORMS

NOTE

Calendar label types can only be used with Calendar data columns.

The Calendar column form works in very much the same way as the Short form works. It also uses the ENTER command. And, like Short form columns, Calendar columns cannot be altered with the CHANGE or APPEND command. To alter such a column, delete it and then re-enter it.

The general command for a Calendar column is the following:

ENTER Columnname CALENDAR number of periods, number of periods per year, beginning year, beginning period

INTERVAL	INTERVALS/YEAR
year	1
quarter	4
month	12
period	13
week	52
day	365

Example 1:

ENTER YR7576 CALENDAR 18,12,1975,7

In this example the column, YR7576, plots data over an 18 month (12 months per year) time frame, beginning in 1975 with the seventh month (July).

Notice that the colon which normally is used in the ENTER command has been omitted here. A colon should not be used when generating a Calendar column.

Example 2 – Stock Values

The graph in this example plots the price fluctuations of a stock during the month of October, 1975. The data are plotted on a daily basis. Each data point (represented by a box) shows the closing price of the stock beginning each week on Monday afternoon and concluding Friday afternoon. Missing data points, discussed earlier in this section, indicate the inactive market on Saturdays and Sundays throughout the month as well as the Columbus Day holiday.

To set up the calendar axis for this plot, we need four pieces of information:

(1) The number of intervals in October (31).

(2) The number of such intervals per year. Since we are plotting on a daily basis, we know that the number is 365.

(3) The beginning year (1975).

(4) The beginning period. The first of October is the 274th day of 1975.

Therefore, the command to generate the calendar axis would be like the following:

```
ENTER TIME CALENDAR 31,365,1975,274
```

The other axis can now be entered:

```
ENTER PRICE: 29.25,29.125,29.50,,,29.625,29.75,29.875,30.0,30.0,,,,31.25,
31.0,31.5,32.0,,,31.0,31.5,31.75,32.25,32.75,,,33.5,33.75,33.75,31.75,29.875
```

Now we can define the curve and the graph and include the curve:

```
DEFINE CURVE STOCK AS TIME,PRICE;DEFINE GRAPH MARKET;
INCL STOCK
```

NOTE

In this last line there are some things of interest:

(a) We have concatenated three commands.

(b) Many of the filler words have been omitted.

(c) The command INCLUDE has been abbreviated to its minimum first four characters.

As you become more familiar with Interactive Graphing, you may wish to apply the above techniques for reducing the time it takes to generate a graph.

Before the curve is displayed, it would be nice to arrange the minor tic marks so that they represent one per day:

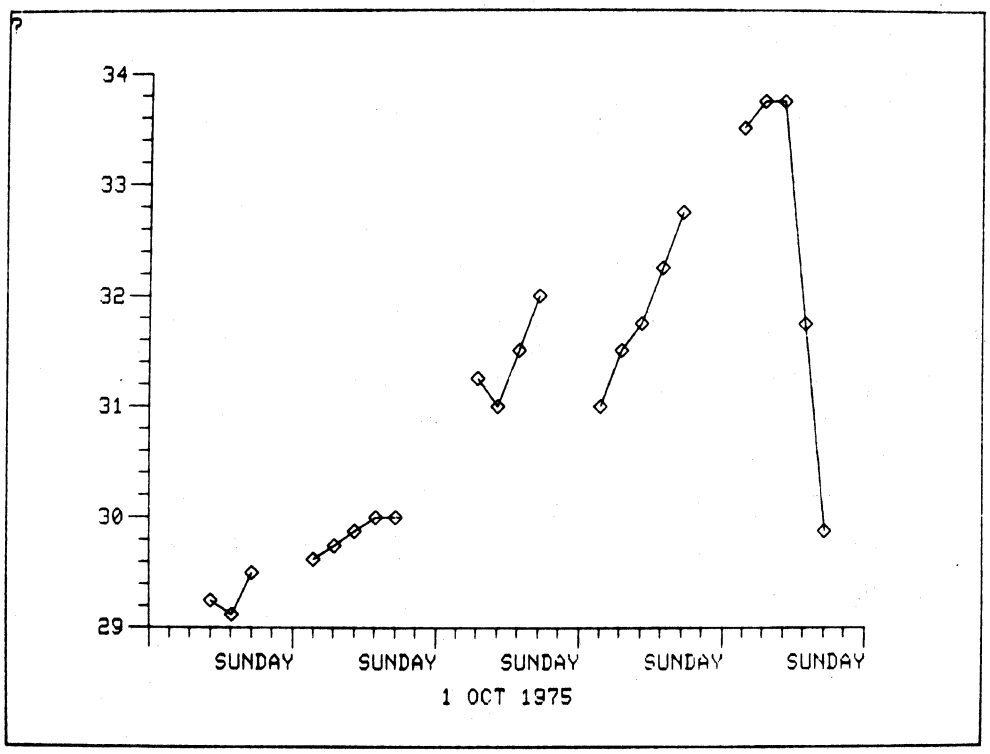
```
SET MINOR TIC NUMBER OF XMARKET TO 7
```

SHORT COLUMN FORMS

We can also represent each data point by a box:

```
SET STOCK SYMBOL STYLE 4
DISPLAY
```

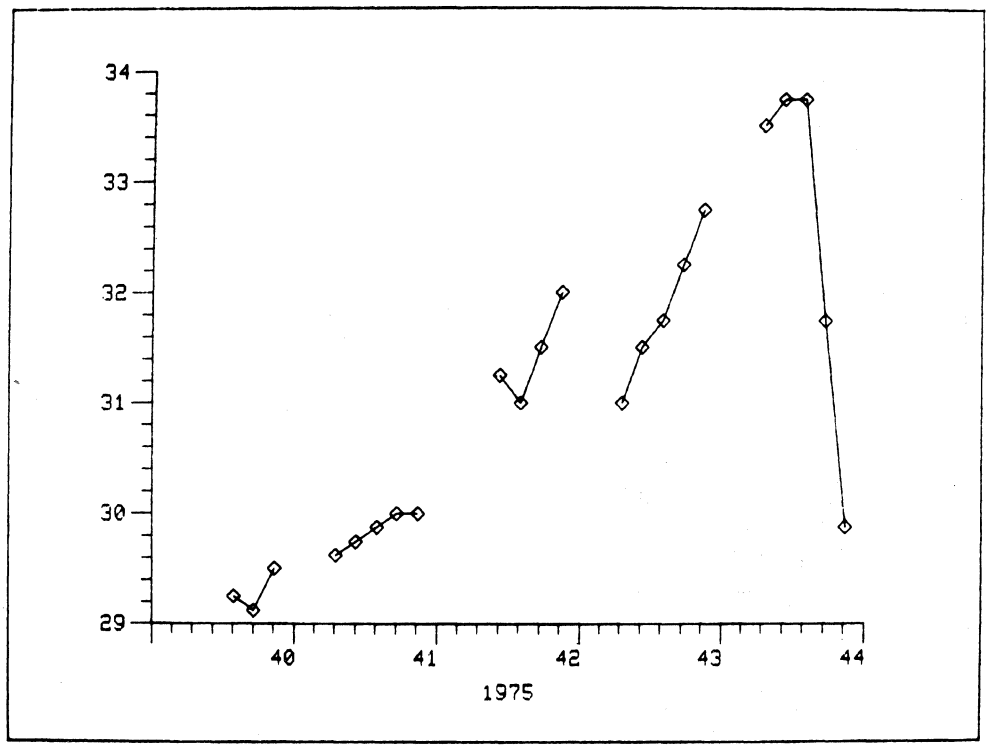
FIGURE 4-1



By changing the label type of the X-axis, we can display the graph with labels representing weeks instead of days:

```
SET XMARKET LABEL TYPE 4
DISPLAY MARKET
```

FIGURE 4-2

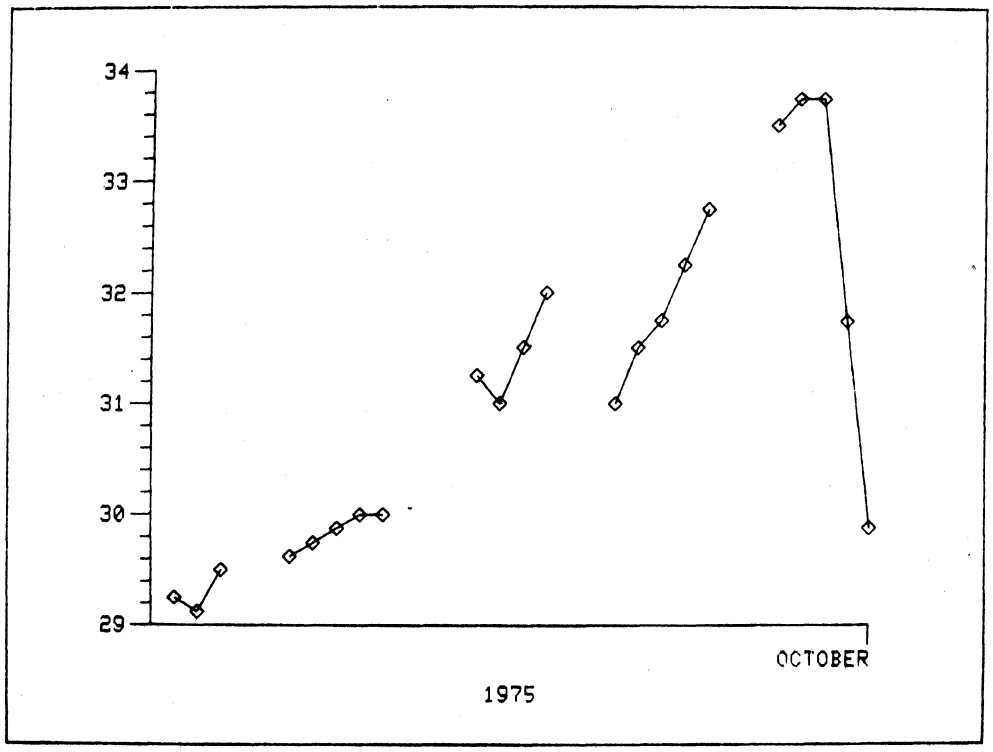


SHORT COLUMN FORMS

Or the label type can be set to monthly basis:

```
SET XMARKET LABEL TYPE 6
DISPLAY MARKET
```

FIGURE 4-3



GRAPHIC INPUT

After a graph or a curve has once been displayed, the terminal graphic input crosshair can be used to indicate position points on the screen and to input data coordinate pairs in conjunction with the ENTER, APPEND, and CHANGE commands.

NOTE

All Tektronix graphics terminals are equipped with the crosshair cursor except the 4006 terminal and the E4010 terminal.

To activate the graphic crosshair, type a pound sign (#) followed by a carriage return. The crosshair will appear on the screen. Position the crosshair at the desired location and press a keyboard key (with the exception of the #, RETURN, LF, or M). The crosshair will disappear and reappear. A second position pair may be entered by positioning the crosshair and again pressing a key. When no more coordinate pairs are required, press the # key. The crosshair will then disappear and the regular prompt will appear.

Some uses of the graphic crosshair are discussed below.

POSITIONING A GRAPH

To use the crosshair to position a graph on a page, observe the following sequence. Type

POSITION Graphname on Pagename at #

When the crosshair appears, position it where you wish to place the lower left corner of the graph. Press a keyboard key. When the crosshair appears, position it where you wish to place the upper right corner of the graph. Press a keyboard key. Dispense with the crosshair by pressing #. When the standard prompt appears, you can display the page. The graph will be positioned where you wanted it.

ENTERING OR APPENDING TO A PAIR OF COLUMNS

You can use the graphic crosshair to enter points of a curve or to append points to an already existing curve. First, display the graph which includes the curve.

NOTE

The columns of the curve must both be standard form rather than Short or Calendar form.

GRAPHIC INPUT

When you have displayed the graph, type

ENTER Xcolumn,Ycolumn:#∂

or

APPEND Xcolumn,Ycolumn:#∂

The graphic crosshair will appear. Position it on the graph at the first desired point and press a keyboard key (with the exception of #, RETURN, LF, or M). When the crosshair reappears position it over the next point, and press a key. Continue entering points as desired. Use of the M key will indicate a missing data point. After all points have been entered, press #. The crosshair will disappear. When the standard prompt appears, you can display the graph with the newly entered or appended curve.

CHANGING A PAIR OF COLUMNS

You can use the graphic crosshair to change points in an already existing curve. First, display the graph which includes the curve.

NOTE

The columns of this curve must both be standard form rather than Short or Calendar form.

When you have displayed the graph, type

CHANGE Curvename #∂

When the crosshair appears, position it over the point to be changed and press the colon (:) key. No other key will properly indicate the point. When the crosshair reappears, you can enter a replacement point or points, pressing a keyboard key after each entry (you are not restricted to the colon for their entries). The use of the M key will indicate a missing data point. When all entries have been made, press #. When the standard prompt appears, you can display the graph with the changed point(s).

EXAMPLE – MODIFICATION OF A CURVE

The graph in figure 4-4 was generated using the Short form axis entry described earlier in this section. The graph shows air temperature changes at regular altitude intervals (100 meters). The sequence of commands used to generate the graph is also shown. Note the following information about the commands:

(a) Many of the commands are concatenated together in one line.

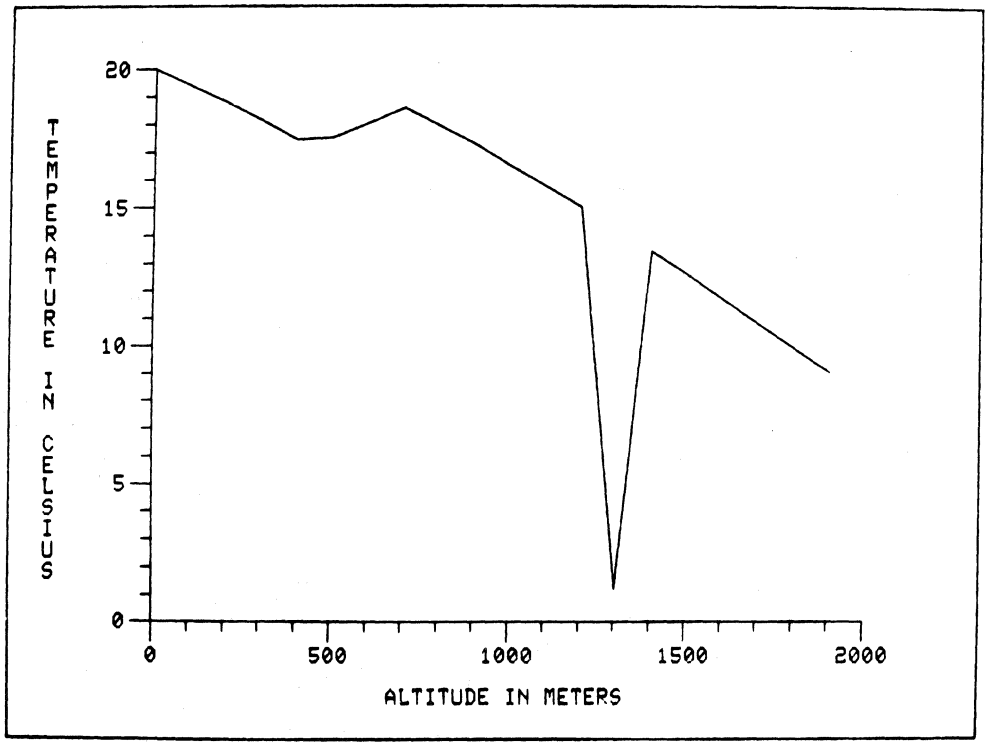
(b) Most filler words have been eliminated from the commands.

(c) Many of the commands have been abbreviated to the minimum of their first four characters.

When you have become familiar enough with Interactive Graphing, the techniques listed above will be useful in cutting down the time necessary for you to generate a graph.

```
ENTER ALTITUDE SHORT 20 0 100
ENTER TEMP:20,19,99,18.78,18.11,17.44,17.50,18.0,18.58,17.91,17.24,
16.50,15.769,15.034,1.225,13.41,12.608,11.71,10.82,9.939,9.049
DEFINE CURVE AIRTEMP ALTITUDE,TEMP
DEFINE GRAPH ATMOS;INCL AIRTEMP;DEFINE NOTE ALT 'ALTITUDE IN METERS'
DEFINE NOTE TMP 'TEMPERATURE IN CELSIUS'
INCL TMP VERT AT -15,50; INCL ALT 50,-15; DISP ATMOS
```

FIGURE 4-4



GRAPHIC INPUT

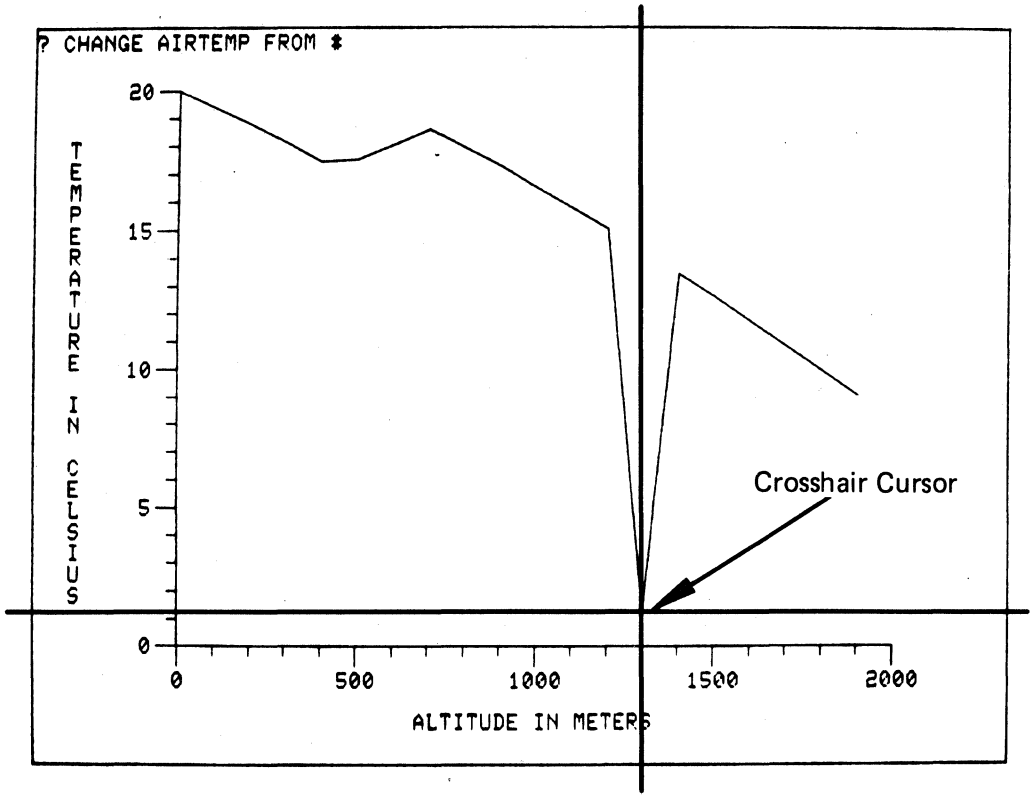
Another interesting thing about this graph is that there has obviously been a mistake made in the entry of the temperature at 1300 meters. This error could easily be corrected using the CHANGE command, since we could find the coordinates of the incorrect entry without much trouble. However, to illustrate the use of the crosshair cursor, we will use a graphic input sequence to modify the curve:

Type

CHANGE AIRTEMP FROM #∂

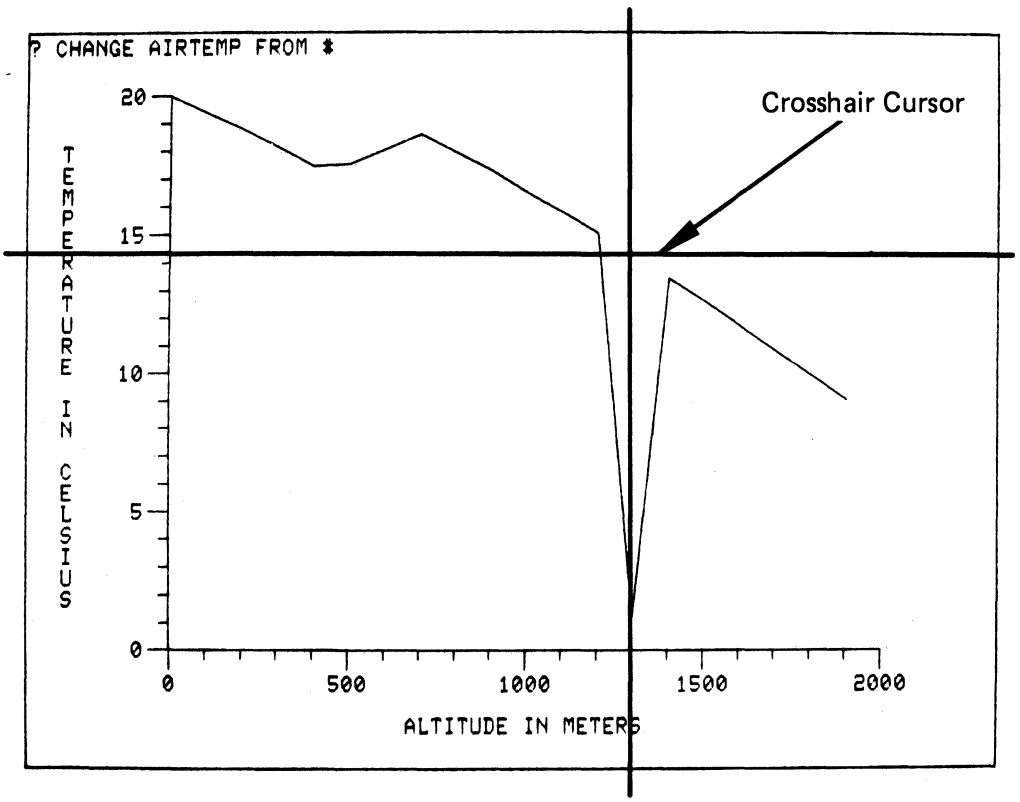
When the crosshair appears, move it directly over the point to be changed as shown in figure 4-5. Press the colon (:) key to trigger the input of the point to be changed.

FIGURE 4-5



Now move the crosshair over the correct point on the graph as shown in figure 4-6. Press any keyboard key with the exception of #, RETURN, LF, or M.

FIGURE 4-6



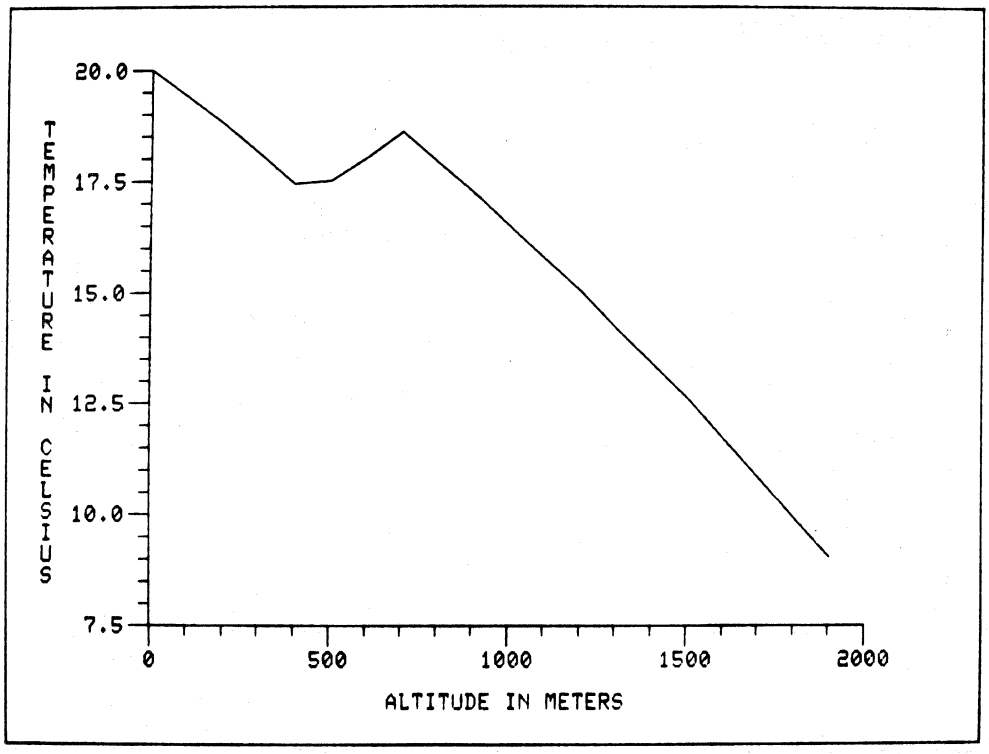
To remove the graphic crosshair from the screen, press the # key. Now re-display the graph,

DISP ATMOS

GRAPHIC INPUT

The corrected curve and graph appear in figure 4-7.

FIGURE 4-7



COMMAND/DATA FILES

In the discussion of the STORE and RECALL commands we dealt with the use of files for the storage of named pages. The present section deals with a different type of file: the command/data file. Interactive Graphing allows its input source to be changed interactively according to your needs. Normally, the source of command input and data input will be your terminal keyboard; however, you might want input to come from an independently created character file on a system storage device, such as a disk or magnetic tape. Data can be stored in such a file using your computer system's text editor or a user-written application program. Commands can be stored using the text editor. In contrast to the STORE/RECALL type of file, the command/data file can neither be created nor modified by the Interactive Graphing system.

Interactive Graphing can, however, read a command/data file and use its content as an input alternative to keyboard input. Any character file composed of legitimate keyboard data can be read. The name of the file must not exceed a length of four characters,* and no line in the file can be longer than 72 characters. Interactive Graphing commands can be mixed with data in the same file, and, depending on implementation, one file can call another.

To change input from the keyboard to a stored character file, type

!FILE Filename∂

Whenever the Interactive Graphing system encounters this sequence, it transfers control to the named file for input. One obvious restriction for this sequence is that it must be the last thing typed in a command.

Example:

Suppose you wish to enter numeric data stored in a file to fill two data columns. The file, STUF, has numeric paired data in it. To enter this data, type

ENTER XDATA,YDATA: !FILE STUF∂

The data will be read from the file and used to form columns named XDATA and YDATA.

NOTE

Any non-numeric data within the file, STUF, will automatically terminate the ENTER or APPEND command. If you want to enter or append from a file containing non-numeric characters, use your system text editor to remove these characters first.

*On some computer systems (e.g., IBM/TSO) a list of keywords available for file names must be developed at implementation.

COMMAND/DATA FILES

COMMENTS WITHIN FILES

We have stated above that alphabetic characters are not allowed in data files which are used with the ENTER or APPEND commands. This is not strictly true. It is possible to include non-numeric characters if they are part of a comment. The signal for a comment in a command/data file is the greater-than (>) sign, the same sign as is used for the standard prompt character. If you insert a > sign in any line of a file, the remainder of the line is treated as a comment and is ignored as Interactive Graphing input.

For example, the first line of a data file may be a title which tells you the purpose of the file:

```
>THIS IS DATA FILE NUMBER 1:  SALES AND OVERHEAD
```

The rest of the file can be sprinkled with such information as you find useful for editing the file:

```
1.  48.3  21.6 >THESE COLUMNS INDICATE SALES AND OVERHEAD
```

Later, when you wish to use the information in the file as input to Interactive Graphing, the comments like the one above will be ignored.

EXAMPLE – CREATING A GRAPH FROM A COMMAND FILE

This example will show how a command file can be used in conjunction with graphic input to create a graph. Comments are used in the example to show you what values to input from the keyboard.

The file, which can be created with your computer system's text editor, consists of two parts. In the first part you set up an empty graph, specifying the range of the axes. In the second part of the file, the graphic crosshair is brought up on the displayed empty graph. Using the crosshair as described in the Graphic Input section, you can specify the points of the graph. When you remove the crosshair by pressing the # key, the screen is erased and the curve and graph displayed. The file used is listed below:

```
>COMMAND FILE TO START A GRAPH FROM SCRATCH USING GIN
DEFINE GRAPH G
ENTER X,Y;
DEFINE CURVE C X,Y
INCLUDE C IN G
SET RANGE MIN OF XG>TYPE MINIMUM X VALUE
SET RANGE MAX OF XG>TYPE MAXIMUM X VALUE
SET RANGE MIN OF YG>TYPE MINIMUM Y VALUE
SET RANGE MAX OF YG>TYPE MAXIMUM Y VALUE
ERASE;DISPLAY G
```

```
APPEND X,Y:#
ERASE;DISPLAY
*END OF FILE*
```

The use of this file is shown as follows:

Call in the file for Interactive Graphing to read.

```
!FILE DRWR          (The name of the file is arbitrarily chosen.)
```

The file is read in. When the software encounters a missing value, an error message is generated, and the line containing the error is printed out in full (including the comment which describes the necessary user input).

```
SET RANGE MIN OF XG>TYPE MINIMUM X VALUE>
TO WHAT VALUE
?                              (Enter the value.)
```

The next line which generates an error message is printed.

```
SET RANGE MAX OF XG>TYPE MAXIMUM X VALUE>
TO WHAT VALUE
?                              (Enter the value.)
```

When all the necessary values have been entered, the screen is erased and an empty graph displayed. Use the graphic crosshair to input the points of the curve you want. When you press the # key, the graph and curve are displayed.

INPUT FROM THE KEYBOARD IN A FILE

In the last example we saw how a command file can be created which can be made to pause for keyboard input and then resume execution. The method used in the example was to withhold necessary command input. This forced an error message which could be responded to by entering the requested value. Such a method for allowing keyboard input can be useful.

However, sometimes you may want keyboard input when it is not desirable to generate an error message. You can call for one line from the keyboard at any time by inserting the following signal in your file:

```
!!∂
```

As soon as this signal is read by Interactive Graphing, the terminal bell rings. Nothing will appear on the screen. Execution will pause awaiting a line of keyboard input followed by a carriage return. In fact, one useful application for the !! signal is to cause a pause in the execution of a file so that you can examine the screen. When you want execution to begin again, simply press the RETURN key.

COMMAND/DATA FILES

Another feature of the !! is that it can be followed by a comment. That means that you can document the nature of the desired keyboard input line. For example:

```
!!>ENTER THE NAME OF THE STORAGE FILE∂
```

When this line is typed out on the screen, whoever is using the file will know what needs to be entered.

EXAMPLE – UPDATING A FILE

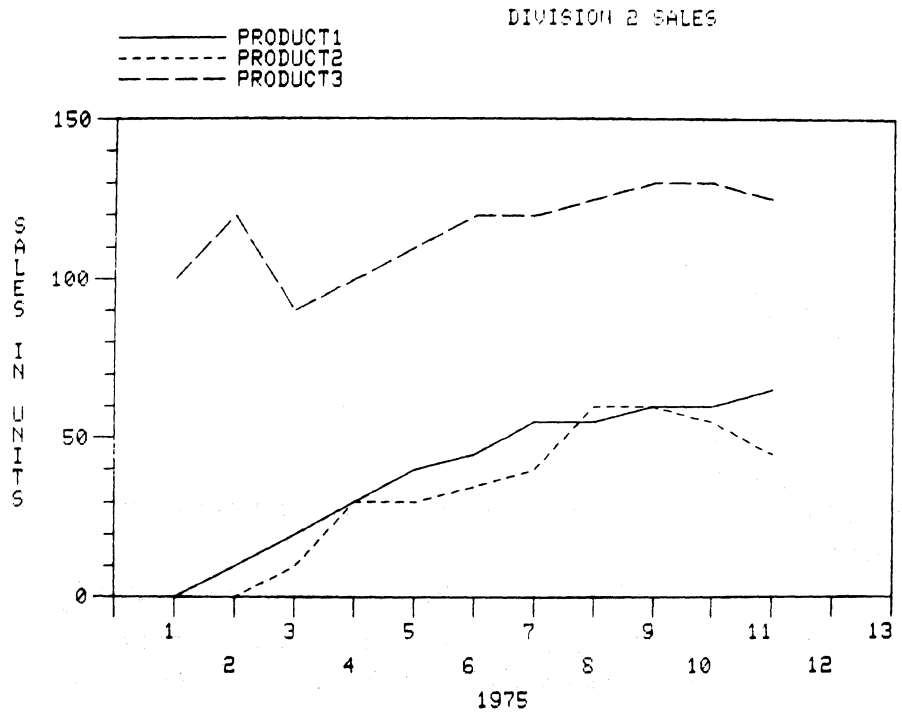
This example shows how a command file can be used to recall an Interactive Graphing file for updating and return it to storage. The command file used in the example is shown below:

```
UPDATE FILE FOR PERIOD UPDATE OF SALES GRAPHS
RECALL PAGE1 FROM FILE SALE
APPEND        )TYPE PROD1: (AMOUNT)
APPEND        )TYPE PROD2: (AMOUNT)
APPEND        )TYPE PROD3: (AMOUNT)
ERASE
DISPLAY PAGE1
)THIS NEXT LINE REQUEST ONE LINE OF INPUT FROM THE TERMINAL
)IT WILL EFFECT A PAUSE UNTIL THE OPERATOR TYPES RETURN
!!
ERASE;
!!)TO PREVENT UPDATE TYPE BYE – ELSE JUST RETURN
STORE PAGE1 IN FILE SALE
BYE
* END OF FILE *
```

This example command file first recalls an already existing file named SALE from storage. The page named Page1 contains a graph with three data columns which are to be updated. The command file contains commands which will append data to the three columns: PROD1, PROD2, and PROD3. The information to be appended is requested by the comment inserted in each command line. This comment is printed out when an error is experienced.

After the appended data is entered. The screen is erased and the graph with the newly appended information is displayed. At this point an !! inserted in the command file halts execution of the file so that the screen can be examined and possibly hardcopied.

FIGURE 4-8



After the RETURN key is pressed, the following message is printed:

TO PREVENT UPDATE TYPE BYE – ELSE JUST RETURN

This message is the comment following another !! signal. In order to have the file automatically updated, press RETURN. The command file then exits you from Interactive Graphing.

SKIPPING DATA COLUMNS IN A FILE

Some application programs produce output containing many columns of data. Possibly only a few of these columns can be useful for graphing. Or, the Interactive Graphing system may not have enough internal storage to read in all of the columns of data. Unwanted columns can be discarded as the values are read from the file. To do this, substitute the word SKIP for the name of the column in the ENTER or APPEND command.

COMMAND/DATA FILES

Example:

A file named DAT1 contains four columns:

```
)THIS IS A DATA FILE
1.   31.50   17   2300
2.   32.50   19   1300
3.   31.75   18   1400
4.   29.75   17   1500
5.   34.50   16   1400
6.   37.75   12   1300
7.   40.10   11   2100
8.   41.20   10   1450
9.   42.30    9   1750
10.  46.10   18   2110
```

Suppose you do not need the first and third columns. The correct columns are entered using the following command:

```
ENTER SKIP, PRICE,SKIP,VOLUME:  !FILE DAT1∂
```

The columns PRICE and VOLUME are filled with data from the second and fourth columns in DAT1. The first and third columns of DAT1 are skipped over, as can be seen if the columns are listed.

```
)LIST PRICE,VOLUME
VOLUMN          -- PRICE
      31.50
      32.50
      31.75
      29.75
      34.00
       6.00
       7.00
       8.00
       9.00
      10.00
```

Example (cont)

```
COLUMN          --VOLUME
                2300.00
                1300.00
                1400.00
                1500.00
                1400.00
                1300.00
                2100.00
                1450.00
                1750.00
                2110.00
)LIST ALL
COLUMN      -- PRICE
COLUMN      -- VOLUME
```

ADVANCED USAGE

MULTIPLE AXES

Although a typical graph consists of one or more curves on a data space defined by a horizontal and a vertical axis, occasionally data plotted on two different graphs can be combined so that the two graphs share one axis. The resulting graph will then have three axes, one shared and two not shared.

To produce such a composite or "multiple axis" graph, follow these general steps:

1. Make sure that the graphs to be combined share one axis in common.

2. Include the graphs in exactly the same position on a page.

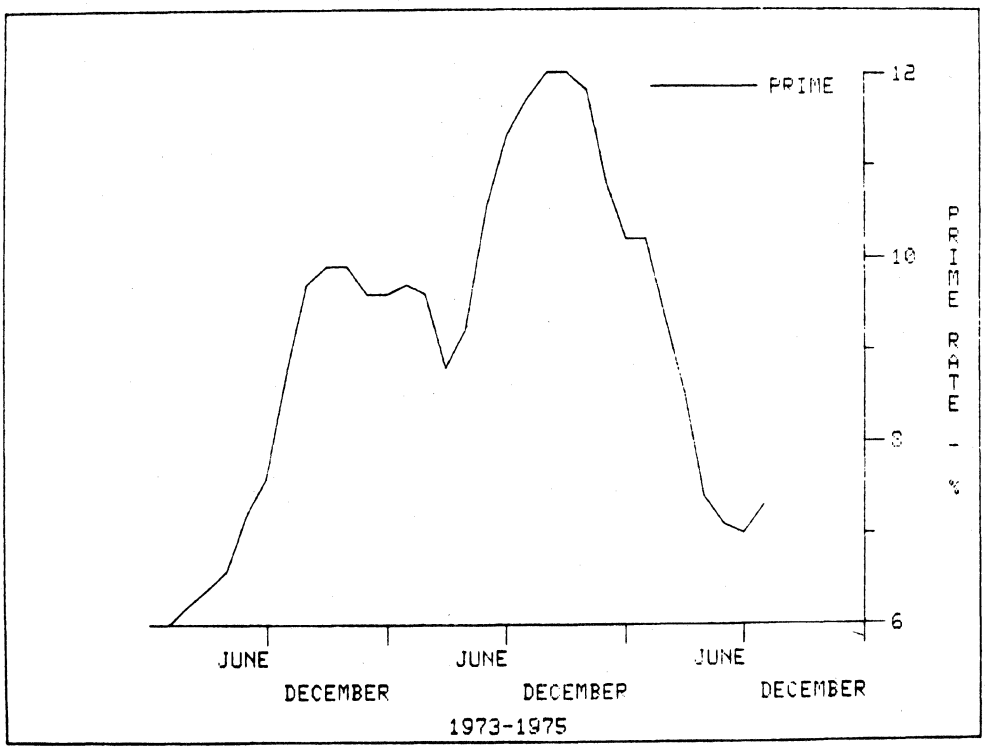
3. Offset the axes by means of the SET Axis LOCATION command.

4. Position notes and legends in relation to the new axis locations.

Example:

This example results in the same graph that was shown on the first page of this manual (figure 1-1). The graph is composed of two graphs superimposed. The first graph, PRIMEGR, is shown below in figure 4-9.

FIGURE 4-9



MULTIPLE AXES

As you will note, the vertical axis of the graph has already been moved with the SET Axis LOCATION command. Each axis in a graph has a name, although you do not have to name it. Interactive Graphing names the axes for you, unless you name them yourself. The default names are Xgraphname and Ygraphname. Therefore, the axis names for PRIMEGR are XPRIMEGR and YPRIMEGR. The command used to move the Y-axis of PRIMEGR all the way to the right was:

```
SET YPRIMEGR LOCATION  100
```

Assuming that we already have defined a curve called BORROW and have entered values for its data columns, the next steps are the following:

We define a new graph, naming the axes. The X-axis will be shared with PRIMEGR, so we will use the same name. The Y-axis name is arbitrarily chosen.

```
DEFINE GRAPH BOROGR AS XPRIMEGR,YBOROGR
INCLUDE CURVE BORROW
```

Now we define a new page and include both our graphs, PRIMEGR and BOROGR, at the same location:

```
DEFINE PAGE INTEREST;INCLUDE PRIMEGR 15,15  85,85;
INCL BOROGR 15,15  85,85
```

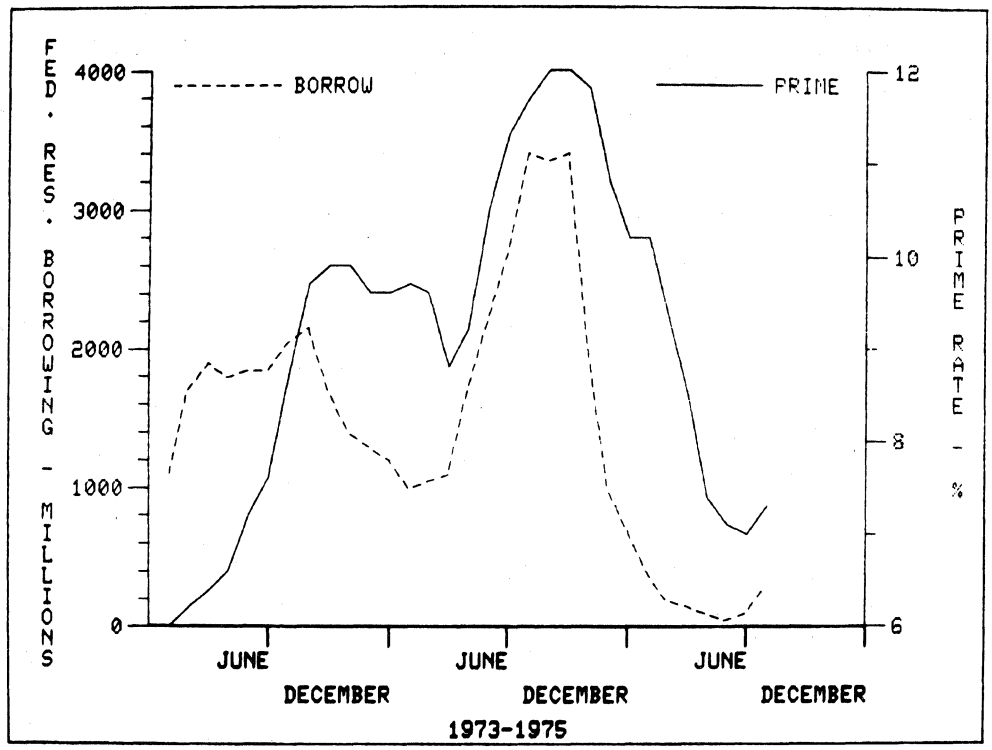
Now, to tidy up the graph, we will include the already defined note, NOTEBORO, as well as including a legend for the curve BORROW:

```
INCLUDE NOTEBORO VERTICAL AT –15,50; INC LEGEND BOROGR
```

The result is the same as the graph on page 1-1.

MULTIPLE AXES

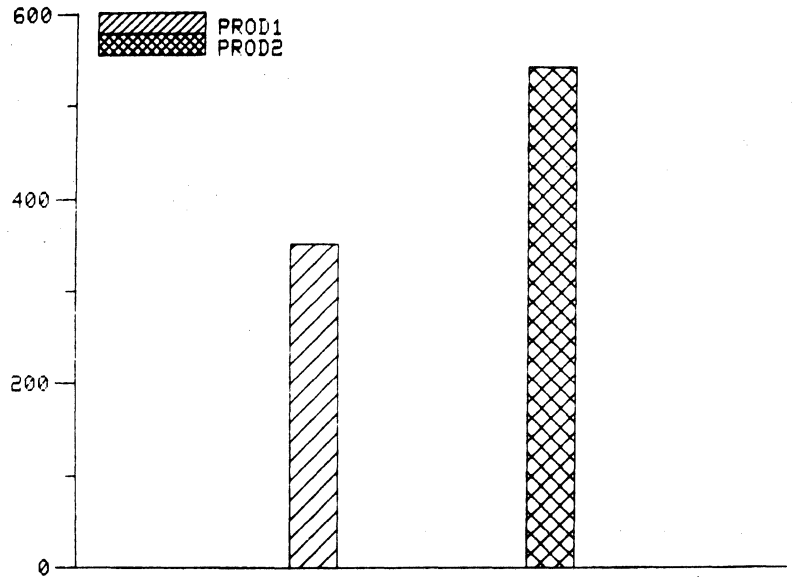
FIGURE 4-10



A SAMPLE BAR GRAPH

This example shows you how to create a graph like the one in figure 4-11. There are a couple of interesting things about this graph. First of all, it consists of two bars, yet each bar represents one whole curve. Second, although the graph obviously has an X-axis, the data is only being plotted against the Y-axis: The X-axis has no calibration at all.

FIGURE 4-11



This is a very common kind of graph, and the following technique can produce one like it:

1. Define two curves.

2. For each curve enter one data point. The X value of this point is determined by the spacing you want between bars along the X-axis. The Y value of the point is the top of the bar along the Y-axis.

3. Set a vertical bar line style for each curve.

4. Set a different symbol style for each bar.

5. Clear the X-axis of tics and labels by setting the label type and major tic form.

A SAMPLE BAR GRAPH

Naturally, it would be easy to adapt this procedure to produce a graph with horizontal bars and/or with multiple curves.

The actual sequence of Interactive Graphing commands is shown below.

```
ENTER X1,Y1:1,350
DEFINE CURVE PROD1 X1,Y1
ENTER X2,Y2:2,540
DEFINE CURVE PROD2 X2,Y2
DEFINE GRAPH PRODUCTS;INCLUDE PROD1;INCLUDE PROD2
SET LINE STYLE PROD1 TO VBAR; SET LINE STYLE PROD2 TO VBAR
SET SYMBOL STYLE PROD1 TO 4;SET SYMBOL STYLE PROD2 TO 12
SET RANGE MIN OF XPRODUCT TO ø; SET MAX TO 3
INCLUDE LEGEND START 3,100
SET RANGE MIN YPRODUCT TO 0
SET LABEL TYPE XPRODUCT 0;SET MAJOR TIC FORM TO 1
DISPLAY PRODUCTS
```

PART DEFINITIONS

Page

A page is a display of a number of associated sub-parts all on the screen at the same time. It may be made up of any convenient number of graphs and any number of text strings positioned anywhere on the screen. One command will result in construction of all of the page. A page may be saved on external (disc or tape) storage and retrieved from it.

Graph

A graph is a display part or construct which consists of a scaled two dimensional data space defined by an X-axis and a Y-axis and also contains one or more data curves. It may optionally include a legend and any number of text notations. Changing the position of a graph on a page will correspondingly change the position of all the parts of the graph.

Curve

A curve consists of the definition of two columns or vectors and optional information describing its appearance as it is plotted on a graph (for example, the style of line used to connect the data points, or the symbol to be plotted at the data point).

Axis

An axis includes a set of real (virtual) space plotting limits which define one dimension of a plotting space. These limits are set automatically but may be reset with the SET command. Optionally, an axis may include information altering the display of the axis tic marks and labels from default values.

Column

A list of data points associated with one axis of a graph.

Note

A note consists of a text string with an indication of horizontal or vertical position, a justification/centering indicator, and a position relative to the graph origin (the screen lower left corner). Position is expressed as a percentage of the length of the axis in that direction. For example, an appropriate placing of a centered title would be at coordinates (50,110).

Legend

A legend is a list of the curve names in the graph, each preceded by its associated line and/or symbol. It can be positioned in the same way as a note.

Text

A text consists of an alphanumeric character string, length parameter, justification indicator, horizontal/vertical indicator, and a screen coordinate position expressed as percentages of the entire page. Since a text string is a part of a page, not part of a graph, its position remains fixed on the screen. This is the major distinction between text and note.

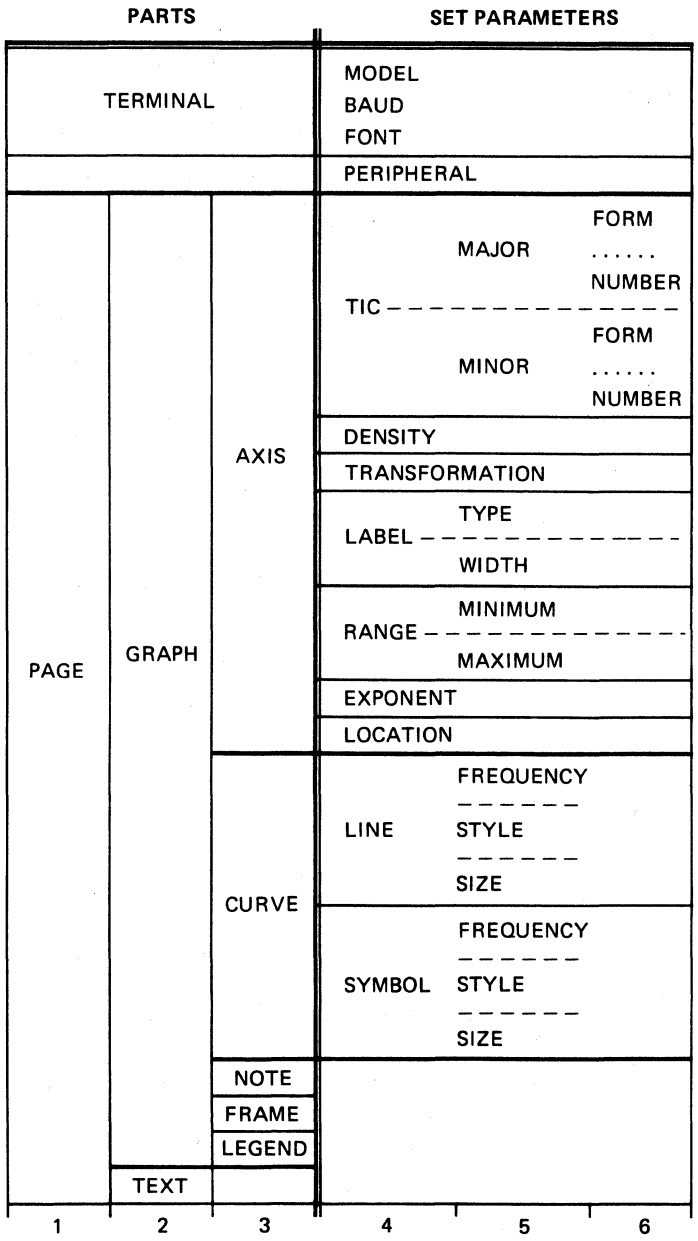
@

PART DEFINITIONS

Frame — A frame is a box drawn around the specified graph. A frame cannot be repositioned.

Partname — A partname is simply a user-given name of one specific part. For example, a specific page may be named SALES. SALES is the partname of a specific page.

The following chart illustrates the left to right hierarchy of parts, their subparts, and the SET parameters:



COMMAND VERBS

APPEND

Use – To add additional values to an existing data column. This facilitates updating periodic data.
General Form – APPEND Columnname (,Columnname . . .Columnname): Value (,Value . . .Value);

Example: APPEND X1, Y1:6,36,7,49,8,64;

Meaning – Appends the numeric values to the named data lists. The data lists are assumed to exist already. The sequential loading operation works in the same way as with ENTER.

BYE

Use – To end an Interactive Graphing session.

CHANGE

Use – To alter points (coordinate pairs) of a curve.

General Form – CHANGE (Curvename) X,Y to: $X_1,Y_1 \ldots X_n,Y_n$)

Example: CHANGE C1 20,30 to: 25,32

Meaning – Finds and deletes data point 20,30 in the curve, C1, and inserts in its place 25,32.

Example: CHANGE C1 20,30

Meaning – Deletes data point 20,30 from the curve, C1.

Example: CHANGE C1 20,30 to: 20,30 25,35 30,40;

Meaning – Deletes the data point 20,30 from C1 and inserts at that position the points 20,30 25,35 30,40.

COPY

Use – To reproduce a part with a new name. Can be used in place of a Define.

General Form – COPY Oldpartname to Newpartname

Example: COPY P1 to P2

Meaning – Creates a definition of a page named P2 which has the same member parts as does P1. Type of part is taken from the old part type.

DEFINE

Use – To create and name a page, graph, or part of a page or graph. It can be used to locate a part relative to the part of which it is a member.

General Form – DEFINE Part Partname (as X axisname, Y axisname*)

Example: DEFINE NOTE N1 as 'THIS IS A NOTE'

Meaning – Creates a Note definition named N1 containing the string: THIS IS A NOTE.

Example: DEFINE GRAPH G1, XG1, YG1

Meaning – Creates a graph definition named G1, with X axis named XG1 and Y axis named YG1.

*Axisnames apply to graphs only.

COMMAND VERBS

CANCEL

Use – To delete all input since the end of the last successful command sentence.

General Form – CANCEL (no operands)

Example: SET the TIC FORM to CANCEL DISPLAY

Meaning: The SET command sentence would not be executed. The DISPLAY command would be executed.

NOTE

The CANCEL command can be used effectively to discard the ENTER, APPEND, and CHANGE commands only before the colon (:) is typed.

DISPLAY

Use – Initiate the display of the named page or part of a page.

General Form – DISPLAY (Partname)
(Part)

Example: DISPLAY PAGE

Meaning – Display all the parts which make up the 'current' page, "current" being the page most recently displayed in part or whole.

Example: DISPLAY G1

Meaning – Display the part named G1 if it is a displayable part (named data columns, axes, legends, and frames by themselves are not displayable). If G1 is the name of a graph all of that graph would be displayed including Curves, Legend, and Notes, if any.

ERASE

Use – To erase the terminal screen.

EXCLUDE

Use – To remove a given subpart from a part specification – inverse of INCLUDE (does not delete the given subpart).

General Form – EXCLUDE partname from Part
Partname

Example: EXCLUDE G1 from P1

Meaning – Remove G1 graph from page P1.

Restriction – Sub-part to be excluded must currently be an existing member of the specified part (e.g., G1 must be a member of P1).

HDCOPY — Use – To generate a hardcopy of the display (if your terminal is connected to an optional hardcopy unit).

HELP — Use – To obtain the next more detailed level of assistance information.

General Form –

HELP or ?Partname / ?Parameter name / ?Verb

Example: HELP

Action – Prints list of available verbs.

?SET

Action – Prints list of parts for which parameters may be set and general form of set verb sentence.

Example: ? CURVE

NOTE

When an error message appears, you can cause an expanded message to be displayed if you type a question mark (?).

INCLUDE — Use – When appropriate, makes the named part a sub-part and locates it.

General Form –

INCLUDE Partname in Partname [HORIZONTALLY / VERTICALLY] [START / with CENTER at X,Y / END / at X,Y and X,Y]

NOTE

Where coordinate pairs are (X,Y), input is typed as percentages of the length of the part in which the sub-part is included.

Example: INCLUDE C1 in G1

INCLUDE C1 in Graph

Meaning – A curve named C1 is included as a subpart of the graph G1.

Note: Position information for a CURVE is meaningless.

COMMAND VERBS

INCLUDE (cont)

Part	Page	Graph	Text	Curve	Note	Legend	Frame
Member	Graph		None		None	None	None
	TEXT	Curve					
		Note					
		Legend					
		Frame					

LIST

Use – Print on terminal screen the members and/or parameters of the named part, or all part names of given part.

General Form – LIST ALL
Part
Partname

Example: LIST P1

Meaning – Lists all members (graphs and text) of page P1.

Example: LIST PAGE

Meaning – Lists the current page.

Example: LIST ALL PAGE

Meaning – Lists all page names in internal storage.

ENTER

Use – To create one or more columns and load them with data values.

General Form – ENTER Dataname (,Dataname,. . .Dataname): Value (,Value, . . .Value);

Example: ENTER X1,Y1: 2,4,3,9,4,16,5,25;

Meaning – A column named Z1 and a column named Y1 are created in the data structure. The first value (2) is put into X1, the next (4) into Y1, the next in (3) in X1, continuing pairs and over line boundaries until any alphabetic character occurs or until the end of sentence character (;) is encountered.

ENTER (cont)

Example: ENTER X1,X2,,Y1 2,4,.01,8,3,9,.02,27;

Meaning – As above, the columns, X1,X2, and Y1 are created. The first number is placed in X1, the second in X2, the third (.01) however, is in the place held by a null operand and is discarded, the fourth is placed in Y1, continuing to the end. This facilitates selective loading from existing files.

Example: ENTER XDATA CALENDAR, 6, 12, 1975, 7

Meaning – This loads a calendar form in which 6 is the number of periods; 12 is the number of periods per year; 1975 is the beginning year; and 7 is the starting period (i.e., July).

Example: ENTER XDATA SHORT 6,1,2

Meaning – This loads the linear short form in which 6 is the number of points to be plotted; 1 is the first value; and 2 is the increment. The values plotted would be 1, 3, 5, 7, 9, and 11.

POSITION

Use – To define the location of parts relative to the part of which they are a sub-part. (e.g., the position of a note relative to the graph of which it is a part.)

General Form –

Position Partname [HORIZONTALLY / VERTICALLY] [START / with CENTER at X,Y / END / at X,Y and X,Y]

NOTE

Where coordinate pairs are (X,Y) input is typed as percentages of the length of the part in which the sub-part is included.

Example: POSITION N1 with END at 100, 110 (a note)

Meaning – This positions the note string. The right end is even with the right side of the graph and vertically 110% to the height of the graph from the bottom of the graph (10% of the height of the graph above it).

COMMAND VERBS

RECALL

Use — Read page from named bulk store.

General Form — RECALL Pagename (,Pagename . . .) on FILE Filename
ALL

Meaning — If a file is recalled without a specified Pagename, a directory of pages in the file will be listed; no pages are read in.

NOTE

If any page or part of a page has a name which already exists in main memory, the page is not read. An error message appears showing the conflict. The part(s) in question should be renamed or removed before the RECALL command is attempted again.

DELETE

Use — To remove a given named part. (Opposite from DEFINE.)

General Form — DELETE Partname (,Partname . . .)

Example: DELETE G1

Meaning — Delete the part named G1.

SET

Use — Set value of parameters of parts of a page.

General Form —

	(Part)		(Part)	
SET		. . .		(Parameter)Parameter Value
	(Partname)		(Partname)	

Example: SET the SYMBOL STYLE of the CURVE to 3.
SET CURV2 SYMBOL STYLE 3.

Complete Context

SET	Page:	P1	(from context)
	GRAPH:	G1	(from context)
	CURVE:	CURV2	
	Parameter:	Symbol Style	
	Value:	3	

STORE

Use — Write the named page to bulk storage in a named file (a four character name).

General Form — STORE Pagename (,Pagename . . .) into FILE Filename
ALL

STORE (cont)

Example: STORE P1 into FIL3

Meaning – Stores a copy of a page named P1 into bulk storage file named FIL3. P1 is not removed from internal storage.

Example: STORE ALL

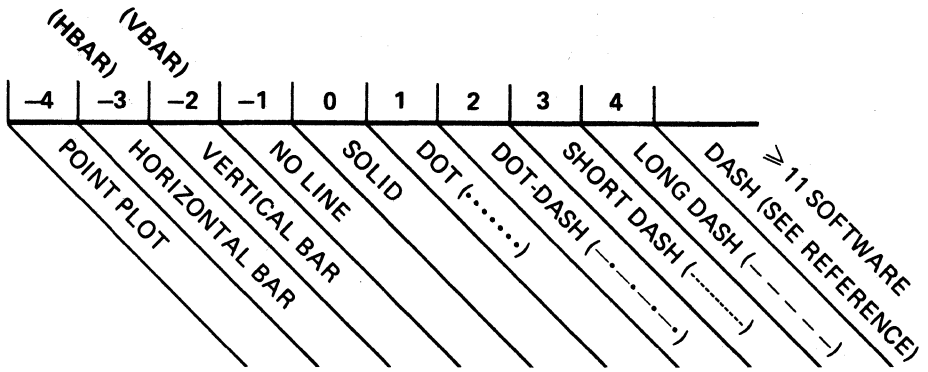
Meaning – Stores all pages in memory into the named FILE.

NOTE

If the named file already exists on the FILE in storage, the question REPLACE OR ADD? is printed. Type either REPLACE (rewrites the file with new page(s) only) or ADD (appends the new page(s) to the old).

SET PARAMETER VALUES

LINE STYLE SPECIFICATION



Software dashed lines may be specified with a concatenation of any of the following code numbers:

Code Value	Meaning
1	5 raster units visible
2	5 raster units invisible
3	10 raster units visible
4	10 raster units invisible
5	25 raster units visible
6	25 raster units invisible
7	50 raster units visible
8	50 raster units invisible
9	alternating visible and invisible segments

Example: 5212 causes a dash dot line (—·—·—·)

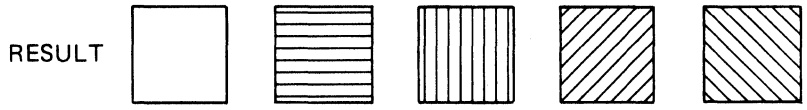
DATA POINT SYMBOLS

VALUE	SYMBOL	RESULT
0	None	No symbol
1	O	Drawn circle
2	X	Drawn "X"
3	△	Triangle
4	□	Square
5	☆	Star
6	◇	Diamond
7	|	Vertical bar
8	†	Cross
9	↑	Up arrow below point
10	↓	Down arrow above point
11	▽	Reverse Triangle

33 → 126 SYMBOL IS CORRESPONDING ASCII CHARACTER

SET PARAMETER VALUES

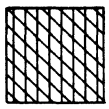
BAR CHART SHADING (SYMBOL STYLE)



OTHERS ARE COMBINATIONS OF THESE UP TO 15

Example:

2 + 8 = 10 =



LABEL TYPES

CODE VALUE	LABEL TYPE
0	No Label
1	Matches transformation type. Default setting
2 or LOG	Logarithmic
3	Days
4	Weeks
5	Periods
6	Months
7	Quarters
8	Years

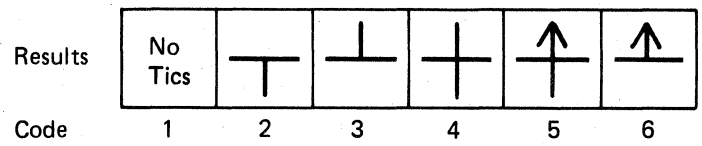
TRANSFORMATION TYPES

CODE VALUE	TRANSFORMATION
1	Normal (Linear). Default setting
2 or LOG	Logarithmic (Ratio) Scale
3	Days
4	Weeks
5	Periods
6	Months
7	Quarters
8	Years

SET PARAMETER VALUES

TIC MARK FORM

Valid values are numbers 1 through 6 as shown on the chart below. Types 5 and 6 include grid lines. Type 2 is the default setting for both major and minor tics.



TIC MAPK DENSITY TABLE

	No Minor Tics	With Minor Tics
Sparse ↓ Dense	1	6
	2	7
	3	8
	4	9
	5	10

NOTE

Density setting is inoperative if the number of tics has also been set.

EXPONENT TYPE

Value	Resulting Type
1	10^{n}. The default setting
2	M, MM's, etc. (1 M per thousand)
3	Printed words, i.e., HUNDREDS
4	10. . .0 (1 plut appropriate number of 0's)

If the label type is logarithmic, this variable specifies the form of the major tic mark labels. No remote exponents are formed for logarithmic axes.

INDEX