AN ABSTRACT OF THE THESIS OF

CHARLES ROSS WATSON for the degree Doctor of Philosophy
(Name) (Degree)

in General Science presented on December 14, 1973
(Major Department)

Title: AN INTERACTIVE COMPUTER GRAPHICS SYSTEM APPLIED TO THE LIFE SCIENCES

Redacted for privacy

Abstract Approved: ______________________
(Larry C. Hunter)

Graph production by computer requires a blending of human effort in scale selection with computer precision in datum point placement. Existing techniques are designed primarily for computer scientists; the average scientific investigator, educator, or businessman tends to avoid them for one or more of the following reasons: expense, complexity, or excessive turnaround. This dissertation describes development and application of a graph preparation system designed for unsophisticated users which is moderate in cost, easy to use, and provides immediate interactive response.

It is written in BASIC on a Univac 1108 and utilizes low cost Tektronix graphics terminals. Through menu selection, the user defines scale factors, datum point indicators, axis type (linear or logarithmic), and adds optional functions which may be displayed superimposed on the data. Immediate display of the results to encourage user creativity in modification of parameters is an important feature of the system.

Step-wise implementation, which maximized user input to system design, is described. Extensive modification of commercially obtained plot driving routines allowed expansion of program features in accord with user requests.

Four detailed examples illustrate the use of the system by biologists and ecologists in communication of results and in analysis of data. A complete program listing and user's manual are appended.

AN INTERACTIVE COMPUTER GRAPHICS
SYSTEM APPLIED TO THE LIFE SCIENCES

by

Charles Ross Watson

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Commencement June 1974

APPROVED:

Redacted for privacy

Professor of Computer Science

Redacted for privacy

Chairman of Department of General Science

Redacted for privacy

Dean of Graduate School

Date thesis is presented December 14, 1973

Typed by Mary Lou Lomon and Linda Tipton for

Charles R. Watson

ACKNOWLEDGEMENTS

This work was sponsored under Contract AT(45-1)-1830 between the U.S. Atomic Energy Commission and Battelle Memorial Institute. It was performed at the Pacific Northwest Laboratory, Richland, Washington.

The following people at BNW provided guidance and encouragement to the project: Lee Eberhardt, Ecosystems Department, John Thomas and Dick Olson, Biology Department, and Rich Hooper, Systems Department.

Lee Rogers, Bea McClanahan, John Hedlund, and Burt Cushing, all of BNW, gave permission for inclusion of portions of their current research.

The staff of the Oregon State University Computer Center, especially its director, Larry Hunter, provided facilities and encouragement at various stages of project definition and development.

Finally, the patience and understanding of my wife, Barbara, is greatly acknowledged.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

AN INTERACTIVE COMPUTER GRAPHICS SYSTEM APPLIED TO THE LIFE SCIENCES

I. INTRODUCTION

Graphs are an important mode of communication in science, education and business. They may be prepared manually or with the aid of mechanical and electronic devices. This report describes development and application of an interactive computer graphics system based on low cost, storage type, cathode ray tube (CRT), computer terminals. It is designed primarily for use by persons unfamiliar with computer techniques. It allows them to prepare linear or logarithmic graphs by answering a series of questions about their data and the desired scale factors. While still seated at the terminal, they may view the resulting graph, modify the data and/or the scale factors, view the new graph, make further modifications, and so on, until satisfied with the results. A hard copy generating device produces output as shown in Figure 1.

The system, called EZPLOT, does not replace existing computer techniques for displaying data, rather it compliments them. These techniques are compared through discussion of their advantages and disadvantages in the second chapter of this dissertation. Some give excellent results

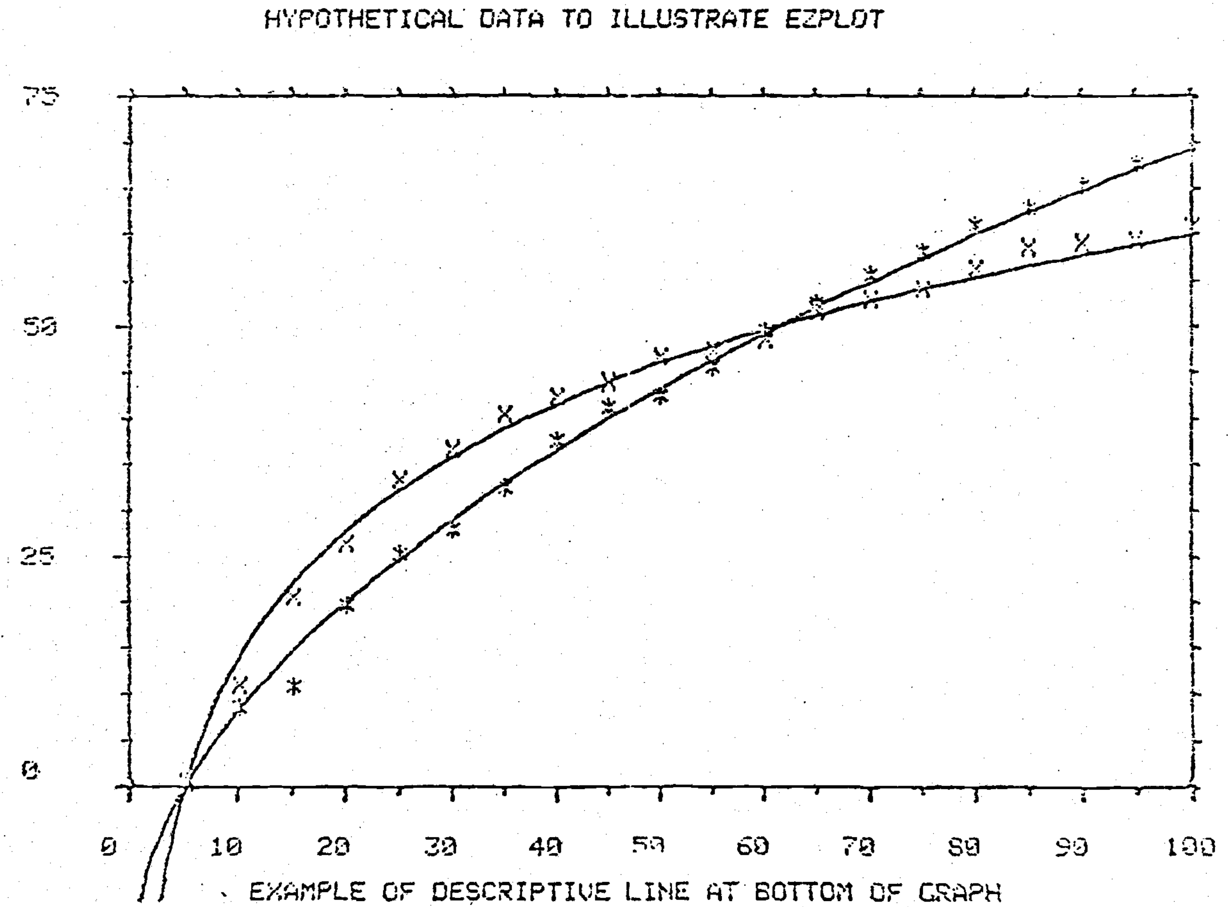


FIGURE 1. Graph of Hypothetical Data With Superimposed Functions as Drawn by EZPLOT

but are either costly or slow, others are quick and inexpensive but give relatively crude results. EZPLOT falls in the middle of this spectrum of graph drawing computer techniques. Its main advantage is immediacy, its primary disadvantages are the fixed physical size of the output and the premium cost generally charged for interactive computer usage.

Throughout the design phase of system development, the paramount concern was to make EZPLOT responsive to the needs of persons who understand graph drawing but are uninterested in computer technology. Four elements of system design, discussed in Chapter Three, were carefully considered prior to implementation. These four, hardware environment, software environment, techniques of interaction with the user, and reasonable goals for the system, determined the character of EZPLOT.

The fourth chapter of the dissertation deals with implementation, which progressed step-wise as new groups of goals were added to operational ones. This type of implementation insured that the system fit the needs of the users. New features were added in response to user requests rather than as a result of the computer scientist's idea of what the user might want.

The system is used by researchers in the Biology and Ecosystem Departments of Battelle Northwest. The fifth chapter deals with examples of statistical analysis of their data in which EZPLOT made a significant contribution to understanding of experimental results.

The essential configuration for operation of this system is: 1) a computer, 2) a CRT graphics terminal, 3) a hard copy generator (optional), 4) availability of BASIC computer language, and 5) the program EZPLOT. The implementation described at Battelle Northwest uses the following specific items:

- Computer - Univac 1108 (operated by Computer Sciences Corp.)
- Operating System - CSCX
- Language - BASIC (enhanced version by CSC)
- Terminal - Tektronix 4002 or 4010
- Hard Copy Unit - Tektronix 4610 .

EZPLOT could be installed on another computer with similar terminals; however, variations in BASIC compilers could require program modifications which would limit the flexibility of the system.

II. BACKGROUND

Scientific investigators are accustomed to presenting their findings in the form of a graph. People in other disciplines also make liberal use of graphic techniques for presenting data. Experts on the art of technical report writing assume that their readers are familiar with and frequently use graphs.

> Graphs are so useful -- and so common -- that definition is almost pedantry: everyone is familiar with them and knows quite well what they are. They are particularly the instrument for representing pictorially all kinds of numerical data. . . . graphs are useful in a multiple variety of technical, industrial and governmental communications -- so useful, indeed, for so many purposes that every kind of report writer employs them and every kind of reader is accustomed to them (9, p. 192).

It is true that graphs are useful and ubiquitous; however, at the risk of being pedantic, a dictionary definition is inserted here for clarity.

> GRAPH (graf), n. 1. A drawing that exhibits a relationship, often functional, between two sets of numbers as a set of points having coordinates determined by the relationship. 2. Any pictorial device, as a pie chart or bar graph, used to display numerical relationships. Also called "chart". 3. A representation of a quantity, as of a complex number, by a geometric object such as a point in a plane. (1, p. 574)

The first and more common meaning is, of course, the type of graph that EZPLOT produces.

Although these definitions stress the ability of a graph to make clear some relationship that has been determined, there are two reasons for preparing a graph of one's data: 1) to communicate, and 2) to analyze.

> The primary purpose of the graph is to present numerical data in visual form. With the growth of its use in numerous fields of endeavor, the functions of the graph have multiplied. It serves as a means of presenting visually tables of statistics in a simple, readable, and interesting form. The graph also makes clear undiscernible facts, such as correlations, which might be overlooked in tabulated data. It facilitates the presentation of facts for comparative purposes, and in many instances the graph indicates significant facts not obviously apparent in numerical form.
>
> The most useful purpose of the graph is to save time and effort in analyzing statistics and tables. . . . In addition, it permits the drawing of logical conclusions on the basis of the data depicted (4, p. 3).

This dual purpose for drawing graphs is usually overlooked in discussions of graphing techniques. The usefulness of a graph as an indicator of "significant facts not obviously apparent" (4, p. 3) is limited by the time and effort required to prepare each graph. For example, if one has ten sets of similar data, there is a strong temptation to graph one set carefully and assume that the relationship is similar for the other nine. Or, if a set of data is graphed on linear paper and there is no apparent reason for re-drawing the same graph on logarithmic or semi-logarithmic paper, it will probably not be re-drawn.

For these reasons, instructions on graph preparation tacitly assume that the person preparing the graph intends to use the graph as an aid in communication rather than as an aid in analysis. This assumption implies that graphs intended for communication should follow guidelines for effective communication and that graphs for analysis need not be as carefully prepared. With the help of computers to do the drudgery, it is possible to prepare precise and easily understood graphs for both purposes.

The limitations and advantages of computerized approaches to graphing must be viewed in light of the manual procedure.

MANUAL GRAPH PREPARATION

The manual technique for preparing a graph is a two-step process. The first phase is creative; it involves human judgement and imagination in selecting the appropriate axes and scale factors to produce a pleasing graph. The second phase is mechanical or routine: data symbols are placed at the proper coordinates, axes are labeled, titles are supplied, etc. Instructions for graph preparation usually contain a list similar to the following:

1. Select the type of coordinate paper.
2. Determine the variables for ordinate and abscissa.
3. Determine the scale units.
4. Locate the axes and mark the scale values in pencil.
5. Plot the points representing the data.
6. Draw the curve. If the curve is to strike an average among the plotted points, a trial curve should be drawn in pencil. If the curve consists of a broken line, as is the case with discontinuous data, the curve need not be drawn until the graph is traced in ink.
7. Label the axes directly in ink.
8. Letter the titles, notes, and so on. The title should be lettered on a trial sheet that can be used as a guide for lettering directly in ink on the graph.
9. Check the work and complete the diagram by tracing the curve in ink (18, p. 529).

The two phases are clear: steps 1, 2, and 3 are the only ones that require judgement, the others are more mechanical.

For small sets of data, the manual procedure is convenient and simple. The person preparing the graph probably has a supply of various types of graph paper at his desk and has an intuitive feel about selecting the paper and preparing the graph. Problems appear when he finds that his choice of scale factors emphasize the wrong aspect of his data. Should he re-draw the graph, or are there more pressing things to do with his time? Another problem arises when a colleague asks the casual question: "I wonder what that would look like on semi-log paper?" These demands on a person's time are compounded if the data set is large.

The time required to produce such a graph is dependent on the two steps. The time for the first step, the creative process of choosing scales, may be considered constant. The time for the mechanical portion depends on the number of points. As the number of data points increases, the anticipated time to complete the graph increases. At some number of data points the investigator will decide that it is not worthwhile to draw the graph by hand.

There are several options available when the anticipated graphing time exceeds that which one is willing to commit: 1) the technician may be employed to prepare the graph; 2) the apparent number of data points may be reduced by plotting representative points such as every fifth or tenth one; and 3) the data may be summarized and appropriate averages plotted. These techniques may be useful if the purpose of the graph is communication. However, if the purpose is analysis, these techniques may cause the investigator to overlook some fact not obviously apparent in summarized or tabular form.

The value of manual graph preparation is two-fold: the person primarily associated with the data is directly involved and the cost is low. There are three major disadvantages: first, the time required becomes large as the size of the data set increases; second, it involves drudgery; and third, it is usually necessary to re-draw a polished copy for publication.

COMPUTER AIDED GRAPH PREPARATION

There are several viable alternatives to manual graph preparation. They all apply the computer to the mechanical portion of the process. Most of them complicate the creative aspects of graph preparation.

None of the existing computer aided graph producing techniques is able to replace the first phase of the graphing procedure, the human judgement as to scale factors. To do that would require that this phase of the process be reduced to an algorithm. Several attempts have been made to do this by formulating rules and guidelines (8, 4). However, more realistic authors contend that there is no universal way of choosing axes and scale factors because each graph is unique (20, 30).

In 1943 the American Society of Mechanical Engineers attempted to standardize graphing techniques through the publication: "American Standard for Engineering and Scientific Graphs for Publication". This standard is quite specific and detailed as to the choice of scale factors, etc. For example:

(d) For arithmetic scales, the scale numbers shown on the graph and space between coordinate rulings should preferably correspond to 1, 2 or 5 units of measurement, multiplied or divided by 1, 10, 100, etc. (3, p. 14).

In 1959, this body published revised standards (1) which abandoned detailed codification in favor of a series of examples of good and poor graphs. Human judgement is recognized as a key factor in successful presentation of graphs.

Production of graphs with a computer will always involve an interaction between the user and the computer. The utility of computer aided graphing procedures should be judged by three criteria: convenience, speed, and quality. The method of determining and modifying scale factors should be convenient; it should approximate the manual procedure. The results should be available quickly; the computer should draw a graph faster than the user can. The quality should be excellent; there should be no reason to re-copy a computer-produced graph for publication.

No computer aided graphing system meets all these goals. Each system is deficient in one or more of these areas. That means that the user must choose the system that will meet his particular need. The system which prepares graphs of publication quality is probably not the system of choice for analyzing a set of data.

For purposes of comparison, the major systems are condensed into four types: mechanical (either incremental or vector) plotters, conventional on-line printers, traditional cathode ray tube terminals, and storage tube CRT devices.

MECHANICAL PLOTTERS[1]

This category includes devices such as the Cal Comp flat bed and drum plotters which are designed specifically to produce hard copy computer controlled graphic output by mechanically positioning a writing instrument on some type of paper. They excel at preparing high resolution drawings for blueprints, printed circuit layouts, etc.

Several steps are required to cause a mechanical plotter to prepare a graph. The data are translated into commands which position the plotter pen by a program operating in the main computer. These commands are transferred to the plotter, usually as a file on magnetic tape, although some mechanical plotters operate on-line. The plotter then follows the commands to produce the graph. An

[1] For a comprehensive machine by machine review of this type of plotter see: Auerbach on Digital Plotters and Image Digitizers; Auerbach Publishers, Princeton, 1972.

example, produced by a Cal Comp plotter, is shown in Figure 2.

This process is rather difficult for the user because of the time delay (turnaround times of several hours to a day or more are common) and because most plot preparation programs require more than casual familiarity with computer programming techniques. Because of the high quality of the output, many people accept the inconvenience of obtaining the services of a computer programmer whenever a series of graphs is needed.

ON-LINE PRINTERS

Line printers and teletypes are used as graph printing devices when low cost and rapid turnaround is important. Although such graphs lack precision and clarity as shown in Figure 3, they are sometimes helpful in analysis of data.

Any computer with printed output can be programmed to produce such graphs at the same cost as normal output. This type of graph is frequently incorporated in the output of routine data analysis programs in which the axes and scale factors remain fixed. Some computer installations provide interactive user-oriented graph producing packages

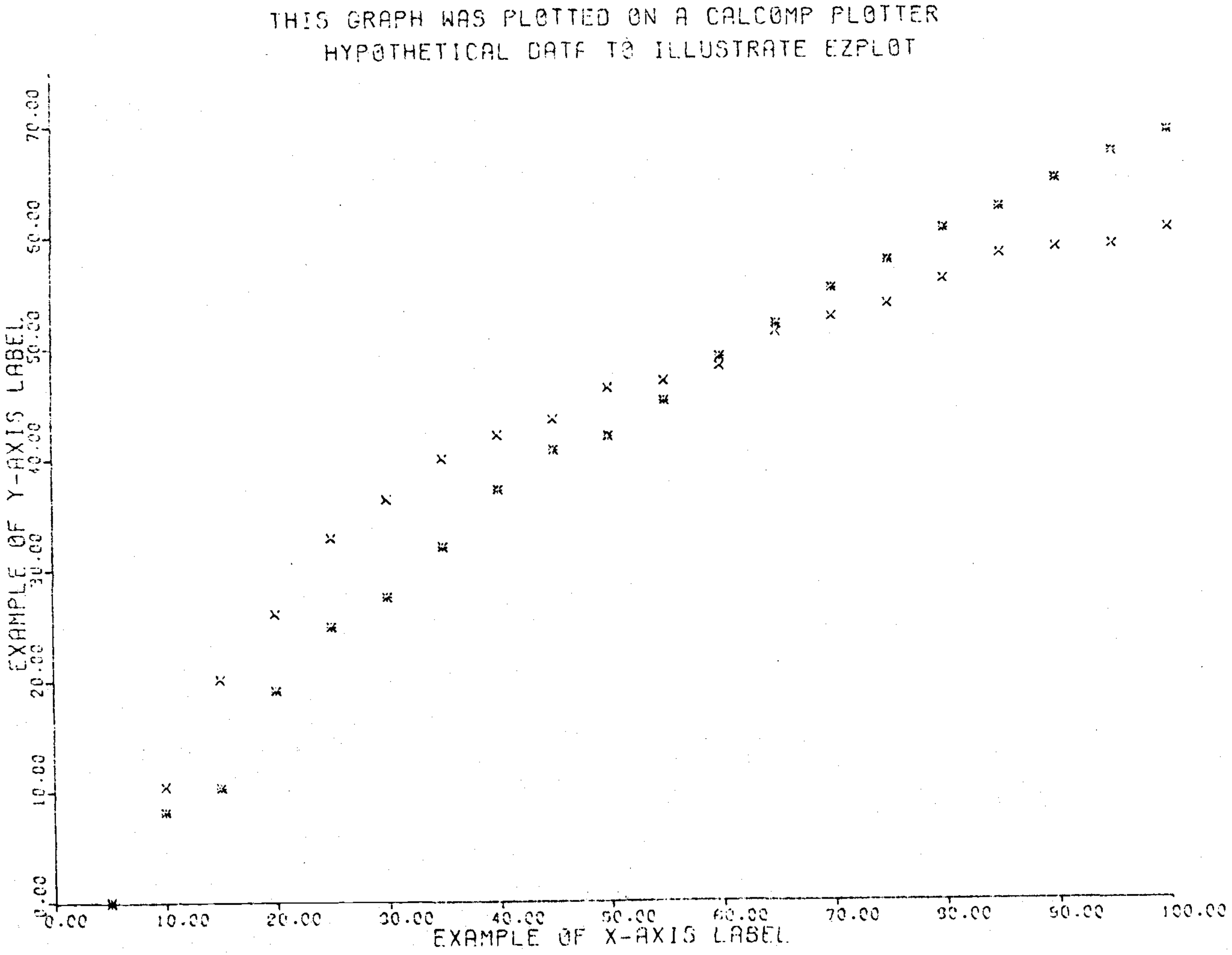


FIGURE 2. Graph of hypothetical data drawn by a mechanical (CalComp) plotter.

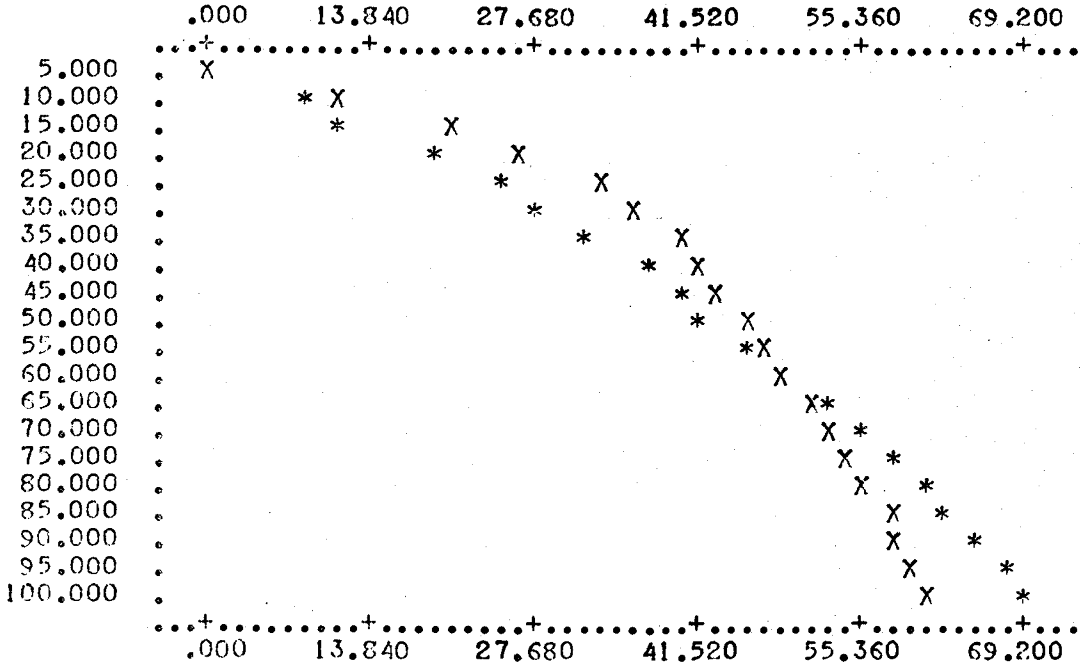


FIGURE 3. Graph of hypothetical data printed by a teletype.

in conjunction with time sharing operation of teletype terminals.[2]

Graph production on this type of device requires different logic from that outlined for manual graph production. Since the paper moves one line at a time in only one direction, the graph must be printed in that sequence. Thus, the computer program must store at least a portion of the

[2] Three such plot routines are described in: CSCX BASIC Library, (Computer Sciences Corp., Los Angeles, 1969). They are: Scatter, Line-Plot, and Histogram. These subroutines of ***STATPK (discussed on pp. E00004-00.123-00 through E00004-00.125-00) are easy to use, but the user has no control over scale factors. Figure 3 was made with Line-Plot.

graph and scan it for printing. The programmer is focused to think in terms which are unfamiliar to the person accustomed to preparing graphs manually.

A new type of line printer, classified as nonimpact devices, operates by drawing the characters rather than printing them by impact (37). Although these devices require special paper and cannot produce multiple copies, they are quiet, inexpensive and fast. In large computer installations they tend to be used as remote terminals. They are often the only printing device attached to minicomputers.

Several of these printers have true graphic capability. If the printer can draw letters, it should also draw lines. The graphic capabilities of this class of printers are presently underexploited. The logic required to draw a graph suffers from the same handicap as with impact line printers. Storage of a display may tax the capacity of a minicomputer memory. Given the proper software, this type of printer could make a valuable contribution to the art of data analysis.

CATHODE RAY TUBE (REFRESHED)

The term computer graphics is usually associated with the type of CRT found in television receivers. Because of the short (millisecond) persistence of the phosphor on the screen, the display must be re-written several times each second. In return, the pattern on the screen may be caused to change dynamically. This requires that the computer store the contents of the entire screen which may consist of several hundred thousand raster positions. The usual practice is to operate the graphics terminal through a computer dedicated to that purpose.

The initial cost of a CRT terminal and associated computer is high ($100,000 or more) and the operating cost is high because of the requirement that the computer exclusively service the graphics terminal. Consequently, this system is seldom used for the pedestrian task of graphing data. The primary use of refreshed CRT terminals is in computer aided design. In this capacity, effective use has been made of the computer's ability to rotate displays, add or delete portions on command, recall diagrams of component parts from a data bank and assemble the parts on the screen (15, 22, 23).

The usual method of obtaining a hard copy of the display on the screen is either through photography or by directing the display file to an off-line mechanical plotter.

CATHODE RAY TUBE (STORAGE TUBE)

This relatively new type of device differs from the conventional CRT in that the image is viewable for long periods of time (an hour or more if necessary) (31, 35). Graphic output may be written a dot at a time; however, the entire screen must be erased if the display is to be changed. In exchange for the loss of ability to dynamically alter the display, this system drastically reduces the cost of CRT graphics. The initial cost is moderate (under $10,000), and the operating cost is comparable to that of other time sharing terminals.

A hard copy device is available which produces good quality copies on photosensitive paper. The figures in this dissertation were obtained from such a device.

Software is available which is similar to that for off-line mechanical plotters (32). Through calls to FORTRAN sub-routines the user's program creates a series of commands which position the electron beam on the screen. Because the process is much quicker than that described for off-line mechanical plotters, storage tube CRT terminals are often used as preview devices. The user can adjust parameters until he is satisfied with the display, then cause the same display to be plotted mechanically.

COMPARISON OF COMPUTER GRAPHING SYSTEMS

The characteristics of the four systems are summarized in Table I.

TABLE I. CHARACTERISTICS OF COMPUTER AIDED GRAPH PRODUCING SYSTEMS

Type of System	Mechanical Plotter (Off-Line)	On-Line Printer Or TTY	CRT (Refreshed)	CRT (Storage Tube)
Convenience	Poor	Fair	Excellent	Good
Mode of operation	Batch	Either	Interactive	Interactive
Turnaround time	Hours	Minutes	Seconds	Minutes
Hard copy quality	Excellent	Fair to good	None*	Good*
Initial cost	High	Low	Very high	Moderate
Operating cost	Moderate	Low	High	Moderate

*These systems can be used to preview displays to be drawn off-line on an incremental plotter.

The choice of a computer aided graph drawing system depends on the purpose of the graph and the needs of the user. Mechanical plotters are preferred when there is ample time and high quality output is important. On-line printers, which combine low cost with speed, are useful adjuncts to data analysis. Investigators with access to traditional computer graphics terminals will continue to insist on the convenience of dynamic interaction with the display as long as they can afford the cost of operating such devices. Display systems built around storage tube

CRT's offer a compromise, combining the convenience of a visual display with moderate cost.

A general purpose graph-drawing system should produce graphs good enough for effective communication at a cost (of time and money) low enough to promote its use in analysis of data. With this principle in mind, a strategy was developed to effectively use a Tektronix 4010 storage tube CRT terminal as an interactive graph drawing tool.

III. STRATEGY

The effectiveness of a computer application depends on the balance with which the resources of computer technology are applied to the problem at hand. The result must be both cost effective and user effective. Three strategic aspects of program development were considered in designing EZPLOT:

- Computer environment
- Programming language
- Techniques of interactive programs.

Analysis of these factors and discussions with a group of scientific investigators who expressed interest in using computer aided graph drawing resulted in a set of specific goals for EZPLOT which satisfy many user needs at a minimum cost.

COMPUTER ENVIRONMENT

Understanding of the hardware configuration, shown in Figure 4, is necessary before software strategies can be discussed.

The Univac 1108 is located at a computer center operated by Computer Sciences Corporation several miles from the Tektronix 4010 terminal. They are connected by a voice-grade telephone line through a standard acoustical coupler. This isolation frees the user from consideration

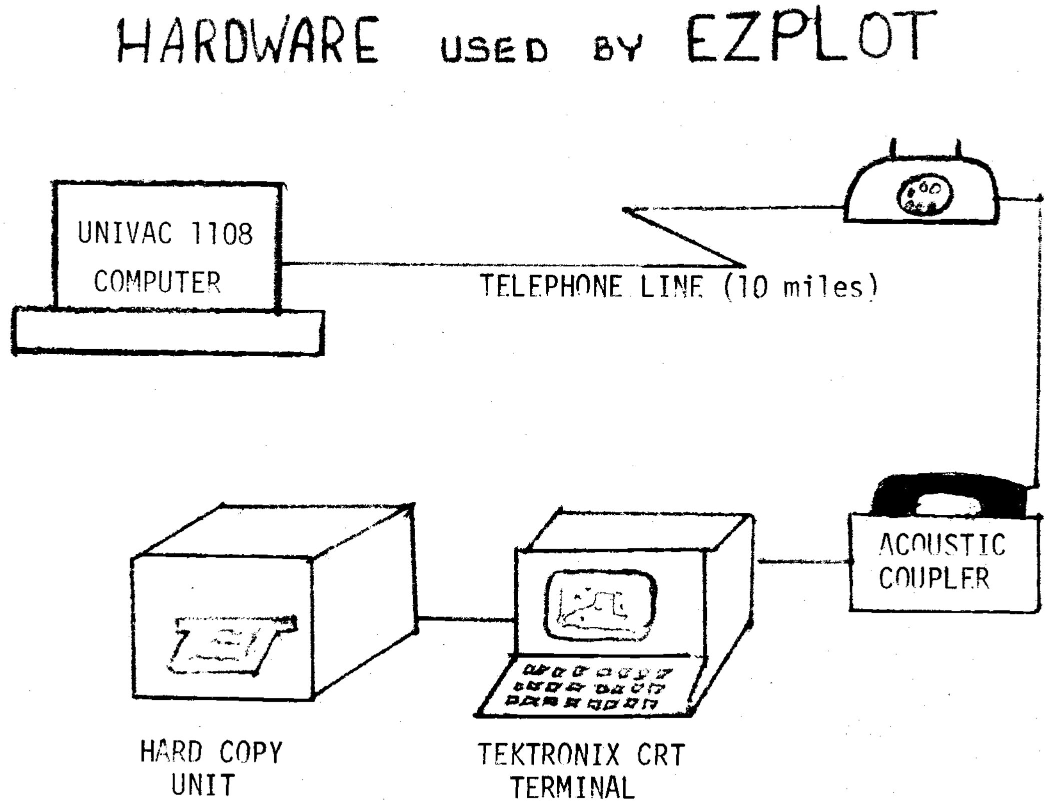


FIGURE 4. Relationship between hardware components used by EZPLOT

of physical aspects of the 1108 and allows him to concentrate on the particular problem at hand. The main disadvantage of the separation of users from computer operations is that users have little ability to influence the operating system design of the main computer. As will be shown later, the way in which the 1108 interfaces with the Tektronix 4010 placed interesting constraints on EZPLOT.

The graphic terminal and hard copy unit are described elsewhere (33, 34) and will be only briefly discussed here. The Tektronix 4010 alphanumeric-graphic terminal has a direct view storage CRT (visible portion 19 by 14.3 centimeters) for computer output and a typewriter keyboard for user input. The visible portion of the screen is composed of 780 by 1023 raster dots, these are individually addressable for graphic output. For alphanumeric output, the terminal contains logic to display characters with a 5 x 7 matrix of raster units, 72 of these characters fill a line and 35 lines fill the screen.

It is easy to draw characters of various sizes and orientations through the graphics mode of the terminal. However, it takes much longer to display a character this way than through the alphanumeric mode. The first strategic decision was to restrict all alphanumeric display to the fixed size characters generated automatically by the terminal. This limits output flexibility but saves execution time.

The hard copy unit, a Tektronix 4610, is located adjacent to the terminal. Copies are available in a few seconds at the users command. Since copy making temporarily inhibits the input/output ability of the terminal, special consideration is made to allow pauses in program execution for static display that the user may wish to copy. In EZPLOT, this condition is signaled by the appearance of a question mark on the screen.

The Univac 1108 is committed to large scale data processing, involving long runs updating several magnetic tapes. In addition, it must serve the needs of a scientific community interested in computational ability. This dual purpose is accomplished through the operating system.

The computer is essentially divided in two; there are two operating systems sharing the core. Users at remote terminals have access to either. The batch mode system, CRJE (Conventional Remote Job Entry), has access to 196,608 words (36 bits per word) of memory plus drum and tape storage with turnaround time depending on workload and priority. The time sharing mode, CSCX (12), gives the user access to a fixed size work area of 32,000 words. There are powerful editing routines available in this system as well as the interactive language, BASIC. Turnaround time under CSCX is immediate because the user has access to a specific physical portion of the computer rather than sharing it with others. In exchange for rapid turnaround, the user must sacrifice ability to use magnetic tapes and is limited in the size of his program and data arrays. In addition, there is a premium cost attached to this mode of operation.

In conjunction with the acquisition of the Tektronix terminal, a group of FORTRAN packages for the production of graphic output was secured (32). These are very flexible and an interactive program similar to EZPLOT could easily be built around one of them. However, FORTRAN programs may run only on CRJE in our computer environment. A survey of potential users indicated that waiting, often several hours, for results was unacceptable. A major strategic decision was to write the general purpose graph drawing program in BASIC.

BASIC

BASIC (Beginner's All Purpose Symbolic Instruction Code) is a language designed for novice computer programmers operating in a time sharing environment (16, 29). In contrast to batch oriented languages such as FORTRAN or COBOL, BASIC anticipates and encourages programmer/computer interaction during program development and debugging (6, 28).

The emphasis of the language is on simplified input and output. For example: there is no distinction between fixed and floating point numbers, format statements are not needed, and arithmetic statements may appear anywhere one would expect to use them (i.e., PRINT X *(A+B) causes the results Y of the computation Y = X *(A+B) to be printed).

BASIC was designed as a flexible interactive language. When it was developed at Dartmouth it was anticipated that the typical BASIC program would be small and that larger programs would be written in another language such as FORTRAN. Therefore, the designers of the language, in optimizing for novice user convenience, adopted restrictions which compound the task of developing large BASIC programs.

Mnemonic naming of variables cannot be used to improve program clarity because variable names are restricted to one letter, optionally followed by one digit, while array names may only be single letters. This naming convention limits the number of variables to 286 and the number of arrays to 26.

Conditional branching is allowed, but conditional replacement statements are not; so that logical flow of the program is interrupted by IF (something) GO TO (line number) and GO TO (line number) statements.

Program documentation through the use of REMARK statements is a feature of the language. However, because REMARK statements are executable in the sense that FORTRAN CONTINUE statements are, they contribute to the storage requirements for the program. Programmers who anticipate that their program and data will approach the maximum allowable size are forced to minimize the use of REMARK statements.

In spite of these difficulties, which only affect the programmer, BASIC is a good language for EZPLOT because it minimizes user frustration. The flexibility of input format inherent in BASIC allows novice users to enter data conveniently. Most BASIC compilers allow for user correction of input errors during execution without re-running the entire program.

BASIC AT CSC

A serious limitation of BASIC is the lack of uniformity among its implementations (21, 26). Each compiler writer has added modifications to the original version. Enhancements to BASIC in the CSC version (11) are used throughout EZPLOT. It may be difficult to transfer it to another computer which has a different version of the language.

Some of the enhancements in the CSC implementation include:

- Ability to handle file input/output
- The verb: GET, which allows keyboard input of string variables without enclosure in quotation marks.
- ON (expression) GO TO (line number list) with drop-through on out of range expression
- PRINT USING an image or an implied image
- String manipulation statements: LEN, and STR.

The CSC version features an incremental compiler; each line of code is interpreted at run time. This feature allows debugging at run time; it is possible to interrupt the computer, change some instructions, and continue execution without changing the contents of data arrays. EZPLOT uses this feature by inviting the user to interrupt execution to add statements which compute functions the user wants to display.

Another advantage of this type of compilation is ability to easily recover form error conditions. If bad input is detected (for example, if the user enters two decimal points with a number) a message appears, execution stops, and the user is allowed to either exit or re-enter the value after giving the BASIC command CONTINUE. Values entered successfully prior to that point are preserved.

TECHNIQUES FOR INTERACTIVE PROGRAMS

The ordinary problems of a computer programmer are compounded when he writes programs for interactive use. Not only must the program conform to the requirements of the compiler and produce correct results, but it must also interface with the most unpredictable of phenomena, human beings.

The programmer must rely on the human user to supply critical program parameters in addition to data to be manipulated. He must devise tests of these input parameters to avoid out-of-range conditions within his program. In addition, he must make the user comfortable by avoiding delay in response and by asking understandable questions.

The development of EZPLOT occupied several months, much of which was devoted to analysis of user difficulties. The results of this attention to user complaints are summarized in the form of flexible guidelines which are applicable to all interactive programs.

- FOLLOW LOGIC OF THE MANUAL PROCEDURE. The user is familiar with a logical way of doing things. The interactive program should follow this pattern at the expense of efficiency. For example, EZPLOT asks the user questions about the data, the type of axis, the scale of the x-axis, the scale of the y-axis, and then about the titles. This follows the pattern of the manual procedure for preparing graphs.

- KEEP QUESTIONS SHORT. Do not bore the user with lengthy questions, remember that he may repeat the sequence of questions several times at each session. Short abrupt statements are used as questions in EZPLOT. "What is the form of your input?" is reduced to "Input type?".
- PROVIDE DETAILED QUESTIONS IF REQUESTED. Neophyte users will not always understand the condensed questions. These users are the exception ; therefore, a detailed form of the question should be available on request. This is accomplished in EZPLOT by the convention that a response of the number 0 to any question will cause a detailed question to appear.
- STANDARDIZE THE RESPONSE EXPECTED FROM THE USER. Do not expect a numeric answer one time and an alphabetic one the next. The user is more comfort-

able when he can anticipate the type of response required. Questions in EZPLOT either refer to a menu of options or ask for data input. The user always enters a single number, followed by a carriage return, in response to a menu-type question.

- KEEP MEANING OF RESPONSES CONSISTENT. Let the pattern of the responses be consistent. If 1 indicates a positive response to one question, do not have it mean a negative response to the next. In general, the following pattern in used in EZPLOT: 0 = print the menu for this question, 1 = a positive response, 2 = a negative response, and 3, 4, 5, etc., are used if there are more than two items on the menu.
- BE FLEXIBLE. The users will discover ambiguities in the questions no matter how carefully they are prepared. More than half of the questions in EZPLOT have been re-written after analysis of user difficulties.

GOALS OF EZPLOT

The final element of strategy was to adopt reasonable goals for the program. The major aim was to develop a program that worked; embellishments could come later. Therefore, the goals are rather modest. Many things which are technically possible, such as rotating the display, "zooming in" on a portion of the graph, splitting the screen to display more than one graph, and using the graphic input thumb wheels to manipulate data elements, were rejected as unnecessary and undesirable complications to the program.

The second major decision was to limit EZPLOT to general cases. This was the reason for rejection of special axes such as a time scale in days or months, preselected scales (i.e., 100 by 100 or 10 by 10), and other ideas suggested by users. Data reduction, regression analysis, and other statistical manipulations are available in other programs so they are omitted from EZPLOT. Finally, data input to EZPLOT is restricted to a keyboard entry routine or from a user prepared BASIC file. Editing capabilities to extensively rearrange data files are not the province of EZPLOT; they are available through CSCX.

The initial goals were:

- Keyboard data entry
- Several sets of data on one graph
- Linear scales
- Data modification
- Titles and numeric scale identification.

Soon some requests by users were added to make EZPLOT more general purpose:

- Logarithmic scales
- Functions as well as data
- Data input from computer file
- Dynamic modification of parameters to re-display graph.

User suggestions which may be added in the future include:

- Split screen display of two sets of axes
- Interchange of x and y axis
- Revised data format to allow several dependent variables for each independent variable
- Interface with output of existing BASIC regression programs.

IV. IMPLEMENTATION

To make the goals of EZPLOT a reality, a program was written combining three operational elements:

- User activated decision points
- Conversational routines in which the user supplies information
- Internal routines with no user participation.

The intermingling of these program elements is illustrated in the general flow chart presented as Figure 5. On this abbreviated chart, diamonds represent questions asked at decision points within the program. These seven questions require that the user select options from a menu. The sequence of questions depends on previous responses.

The hexagons on the chart represent conversational routines. The number of questions asked in each is a function of the responses and the data. The rectangles represent internal routines. Many of these also contain logic not shown on the chart which can alter the sequence of questions presented to the user.

Two appendices included in this dissertation are for readers interested in details of program operation. A computer programmer may wish to compare the flow chart with the text of EZPLOT found in Appendix A. Potential users will be more interested in the EZPLOT Users Manual found in Appendix B.

EZPLOT LOGIC

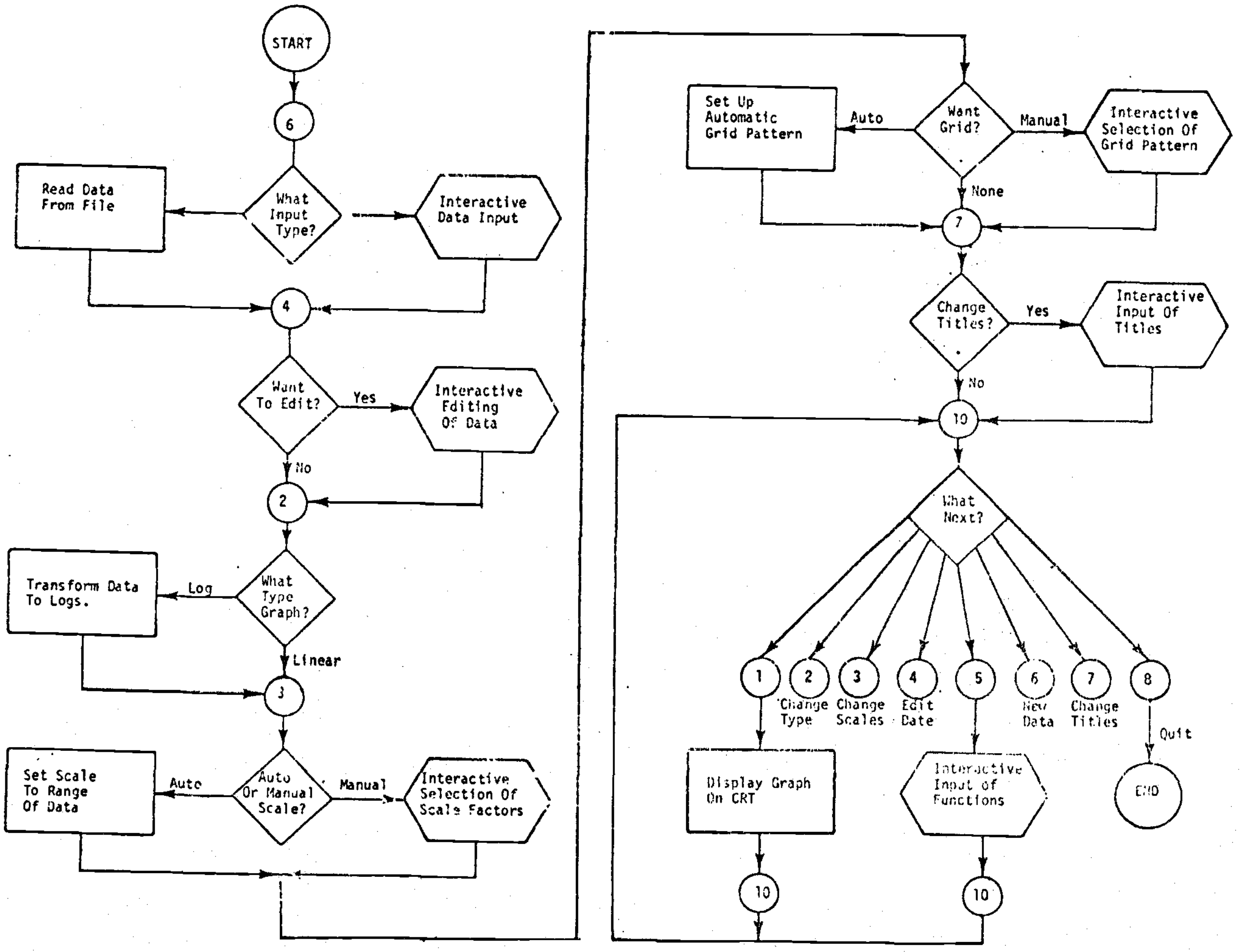


FIGURE 5. Flow chart of main logic of EZPLOT.

As indicated on the flow chart, the pivotal point of the program is the question WHAT NEXT? This question divides operation of EZPLOT into two phases: 1) the user interacts with the program to supply data, scale factors, titles, and other parameters, then 2) the computer draws a two dimensional graph on the CRT screen. The user may cycle through these phases until he is satisfied with the resulting display.

Implementation of the program will be discussed in terms of these two phases. The second, in which the graph is generated, will be treated first. The interactive phase is more straightforward and will be discussed later.

TEKPLOT SUBROUTINES

The graphics portion of EZPLOT was derived from a set of FORTRAN subroutines distributed by the terminal manufacturer (32). These plot driving TEKPLOT subroutines are designed to be machine independent; they are written in FORTRAN II and avoid the use of COMMON or LABELED COMMON. This reduces the problems of installing TEKPLOT on different computers, but requires the use of several subroutines devoted to keeping track of scale factors, rotation factors, and other parameters. These terminal status values are not available to the main program. Only two subroutines need be written to adapt TEKPLOT to a new computer: one, CHIN, to define character transmission between the terminal and the computer, and the other, CHOUT, to translate in the reverse direction.

After installing TEKPLOT on the UNIVAC 1108, Computer Sciences personnel coverted a major portion of it to BASIC, calling the resulting graphics package **PLT[5]. This series of 29 subroutines provides the full range of graphics output supported by TEKPLOT, although hardware restrictions do not allow use of graphics input. Thus, the cross hair cursor on the terminal is unavailable to EZPLOT.

BASIC subroutines differ from FORTRAN subroutines; there is no parameter list associated with the call statement. The technique for passing data to a GO SUB in **PLT is to place the data in dummy variables used by the subroutine.

BASIC subroutines are not independent and all variables are essentially in COMMON. This means that the terminal status values are available to a BASIC program using **PLT whereas they are not in FORTRAN using TEKPLOT.

[5] Unpublished information from Larry Gannon, Computer Sciences Corp., Richland, Washington.

The large size of **PLT (approximately 1200 BASIC statements) limits size of the main program. Early in development of the interactive portion of EZPLOT this maximum was reached, the program plus **PLT exceeded the fixed portion of computer memory allowed each user. Serious consideration was given to splitting the graphics program into two parts, one to prepare the data, the other to obtain the scale factors and draw the graph. This proved to be too complicated for novice users.

Therefore, the graphics support package, **PLT, was reduced to eight subroutines consisting of about 250 BASIC statements. This was achieved by discarding those subroutines not necessary to support the goals of EZPLOT. Features which were discarded outright include:

- the software character generator and it's 400 data elements
- the pause (not needed in BASIC)
- all references to point or incremental mode graphics
- the axis generator, replaced by an EZPLOT subroutine
- the grid generator, replaced by an EZPLOT subroutine.

Elimination of subroutines associated with these features also allowed reduction in size of those remaining by removing linkages to the deleted subroutines. Some of the bookkeeping subroutines, TEK001 through TEK007 were eliminated, others were modified. The most often called subroutine, TPLOT, was replaced with a much smaller version tailored only to the needs of EZPLOT.

The eight graphics support subroutines derived from **PLT and their line numbers in EZPLOT are:

1100 BEGIN, which sets transmission rate and initializes bookkeeping.

1200 ERASE, which transmits the erase code and delays processing until erase is complete.

1500 VECTOR, which sets graphic mode.

2100 SCALE, which updates scale factor bookkeeping.

2300 WINDOW, which edits window limits for legality and updates bookkeeping.

4500 CHOUT, which transmits a coded character to the terminal.

4400 TEK002, which plots current x,y points

3900 TEK003, which partitions x,y points into bits to pass to terminal.

These could be further reduced if necessary. However, since they represent less than 20% of EZPLOT, they are no longer a major hindrance to expansion of the program.

EZPLOT SUBROUTINES

There are eight graphics support subroutines included in EZPLOT. These serve as links between the interactive portion of the program and the modified plot driver, **PLT.

The main EZPLOT subroutine, called MY OWN TPLOT, is located at line 9000. The 15 lines of this subroutine replace approximately 100 lines of TPLOT. The major saving is the result elimination of software plot symbol generation. TPLOT and **PLT provide eight optional symbols, such as dots, arrows, circles, or boxes. In EZPLOT, all plot symbols are hardware generated keyboard characters, with the exception of the diamond which is drawn by vector mode techniques. The task of MY OWN TPLOT is then reduced to sending TEK002 the coordinates of the next location on the screen and a flag to indicate whether or not the vector is to be visible.

The ABSOLUTE SCALE subroutine is located at line 4240. It sends the proper values to the SCALE and WINDOW subroutines to set the scale so that screen raster location corresponds to program logical location. EZPLOT operates in absolute scale while displaying the top and bottom captions.

The USER'S SCALE subroutine starts on line 4350. It sets the scale so that translation from user's units to raster locations is automatic. EZPLOT operates in user's scale while drawing linear graphs and modifies the scale before drawing log plots.

Considerable attention was given to the EZPLOT axis drawing routines. The **PLT ones were unacceptable because they represented the philosophy that the user must choose standard size axes just as if he were selecting graph paper from a drawer. Flexible axis definition is an essential concept of EZPLOT. The AXIS subroutine at line 5340 and the auxiliary LOG X TIC and LOG Y TIC subroutines at lines 9700 and 10000 cause a neatly labeled box to be drawn defining whatever size "graph paper" the user specifies.

The outline around the graph is drawn in this order: First the top and bottom horizontal borders (including tic marks and numeric labels where appropriate) are drawn, then the left and right vertical ones. This method reduces the execution time by eliminating redundant calculation of the tic mark location, as shown in Figure 6, a flow chart of the horizontal axis portion of the AXIS subroutine. Note that drawing visible grid lines is a simple matter of interrogating a switch variable between drawing of the lower and upper tic marks.

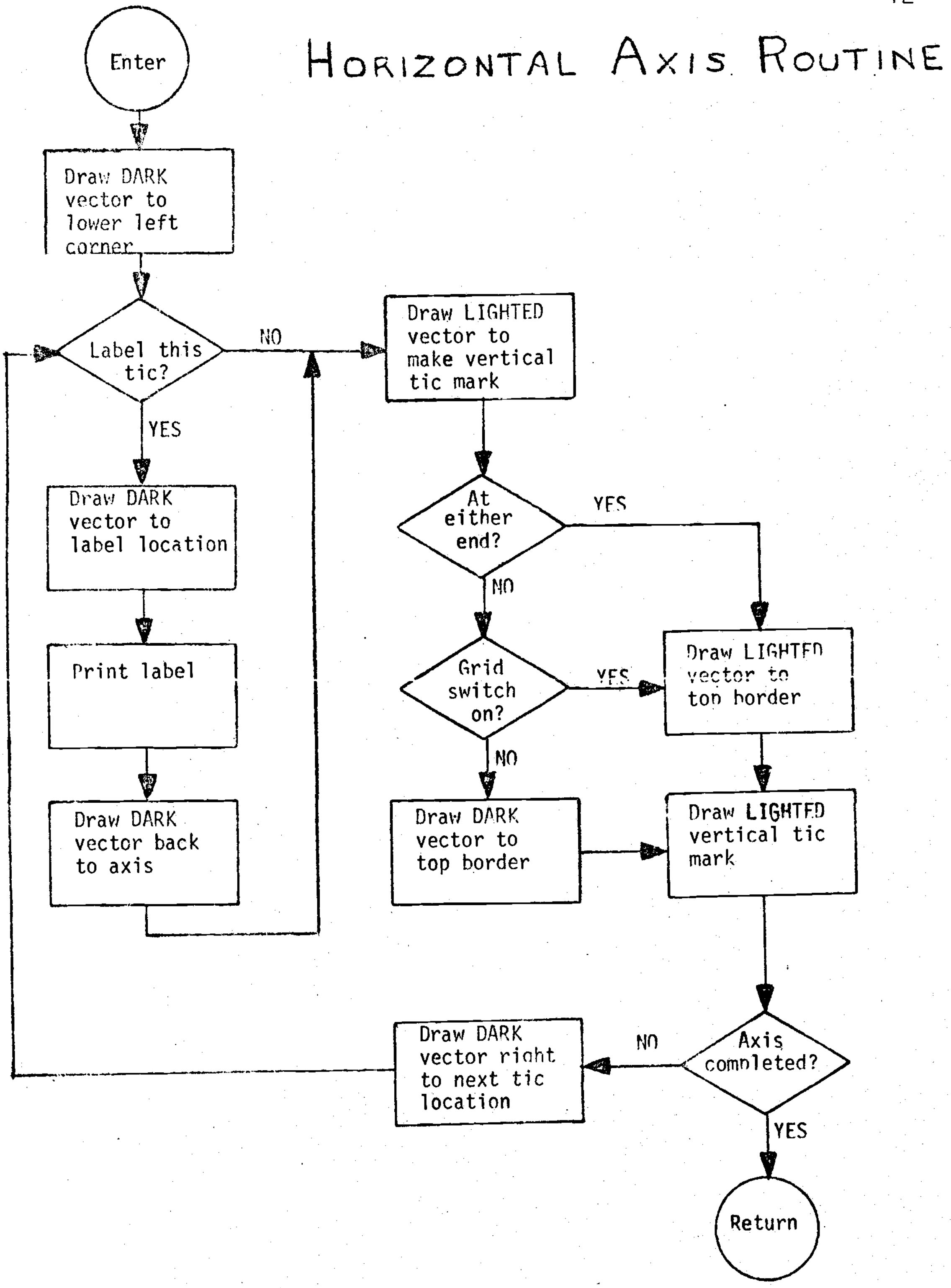

HORIZONTAL AXIS ROUTINE
Enter
Draw DARK vector to lower left corner
Label this tic?
NO
YES
Draw LIGHTED vector to make vertical tic mark
Draw DARK vector to label location
Print label
Draw DARK vector back to axis
At either end?
YES
NO
Grid switch on?
YES
NO
Draw LIGHTED vector to ton horder
Draw DARK vector to top border
Draw LIGHTED vertical tic mark
Axis comnleted?
NO
YES
Draw DARK vector rioht to next tic location
Return

FIGURE 6. Logic of horizontal axis drawing portion of EZPLOT.

An alternate axis drawing scheme which progressed around the box counter-clockwise proved to be slower and did not allow a simple method of adding a grid to the display.

Calculation of tic mark location in the AXIS subroutine is a simple mathematical procedure. If the user wants a total of 12 tic marks, the location of the next tic to the right of the present one is 1/12th of the range of the axis added to the x-coordinate of the present location. Because of the way the user defines tic mark location on a log axis, this method cannot be used in that case. Rather than clutter up the AXIS subroutine with branching logic to treat log axes differently, two other subroutines are used.

The axis on a log scale is drawn by first "tricking" the AXIS subroutine into thinking that the user wants only one tic mark. This produces an axis properly labeled at either end with no intervening tic marks. A call to LOG X (or Y) TIC adds the tic marks and labels at the locations designated by the user.

The two remaining EZPLOT graphics subroutines, VERTICAL GRID and HORIZONTAL GRID, at lines 4470 and 4630, are called if the user elects to design his own grid scheme. They are quite simple because the user has supplied the program with the specific location of each grid line.

EZPLOT LOGIC

The logical flow of the interactive portion of EZPLOT is shown in Figure 5. Three areas are worthy of discussion here: two because of the difficulty they caused, and one because it was so easy to implement due to the nature of BASIC as enhanced by CSC.

The two difficult areas are associated with the transformation to logarithmic scaling. First there was the problem of labeling the log axis, second, the problem of transforming data to the new scale. Equally spaced tic marks are necessary on a linear plot, but how does one calculate their position on a log scale? If the linear axis runs from 10 to 20 with tic marks at 10, 12, 14, 16, 18 and 20, a simple log transformation will suffice. However, if the linear axis runs from 0 to 10, with tics at 0, 2, 4, 6, 8 and 10, a log transformation is impossible because the log of 0 (or a negative value) is undefined in BASIC.

Even with properly defined end points of the axis, computational techniques for spacing the tic marks are complicated. For example, the axis may run from 75 to 125; that is, it may include portions of two decades, or it may include a full decade and a portion of two others, i.e., 75 to 1500. The programming challenge of such a problem is great.

This challenge was not accepted; log axis tic mark location is decided by the user of EZPLOT, not by the program. Rather than deciding how far apart he wants the marks, as he does for linear axis, the user must explicitly define each tic mark location on a log axis. Thus, the programming problem is solved by asking the user to supply the logic rather than attempting to do it for him.

That solution could not be applied to the other problem associated with log scaling, that of negative numbers. Suppose a user has a data file with 100 data pairs of which a few have zero or negative values for one or both of the coordinates. He obtains a linear graph and decides to transform one of the axes to logs. It is simple enough to have the program skip the transformation and display of illegal points, but what information should be given the user?

Several ideas were tried in the development stages of EZPLOT. At first the user was told: YOU CANNOT PLOT NEGATIVE VALUES ON A LOG SCALE, PLEASE EDIT YOUR DATA FILE. The users objected because they had to use the slow (in a real time sense) interactive editing routine to clean up the data.

The second version listed the data as follows:

```
YOU CANNOT PLOT THE FOLLOWING ON A LOG SCALE
1,0
20,-0.5
-1,4
-3,-0.003367
DO YOU WANT THE PLOT ANYWAY?
```

Experience with this version indicated that the user always wanted the plot anyway and was annoyed at having to answer what to him was a redundant question.

In the final version the data is listed thus:

```
WILL SKIP 1,0
WILL SKIP 20,-0.5
WILL SKIP -1,4
WILL SKIP -3,-0.003367 .
```

These messages appear between the time the user requests a log scale and the time he defines the scale factors. This allows him to change his mind when WHAT NEXT? is asked before the graph is drawn if he realizes that his graph is meaningless without the negative data.

The third problem in developing the logic of EZPLOT was how to add functions to the graph. Implementation of this was planned in two steps. First a routine for graphic display of the function was written, then would come a routine to translate the user's input into a function.

The graphic display of a function was accomplished by simply dividing the range of the horizontal axis into 100 increments and evaluating the vertical component of the function at each one of those points. To debug this routine. a test function was written into the program. Further testing required program interruption for modification of the function. This technique worked well and is a simple way for the average user to input his functions.

Thus, rather than using a translation routine to determine the function the user wants, EZPLOT allows him to interrupt execution and replace dummy functions in the program with BASIC statements.

EXAMPLE OF USE

The use of EZPLOT is best explained through an example. Assume that you are investigating some phenomenon such as growth rate of plants or animals, strength of metals, or decay of radioactivity. Further assume that you are comparing the behavior of two cases (two plants, two metals, or two radionuclides), you have summarized your results as shown in Table II, and wish to use EZPLOT to produce graphs of different aspects of this data.

You complete the procedure to activate communication between the terminal and the computer, then enter the interactive phase of EZPLOT.

In the keyboard data entry portion of the program, you type the data as a series of x-y pairs identified by a plot symbol (one of the keyboard characters; in this case, an asterisk (*) for one case and the letter x for the other). The program then displays the data as shown in Figure 7. At this point you may correct mistakes, insert forgotten points or change plot symbols, until you are satisfied that the computer has stored your data correctly.

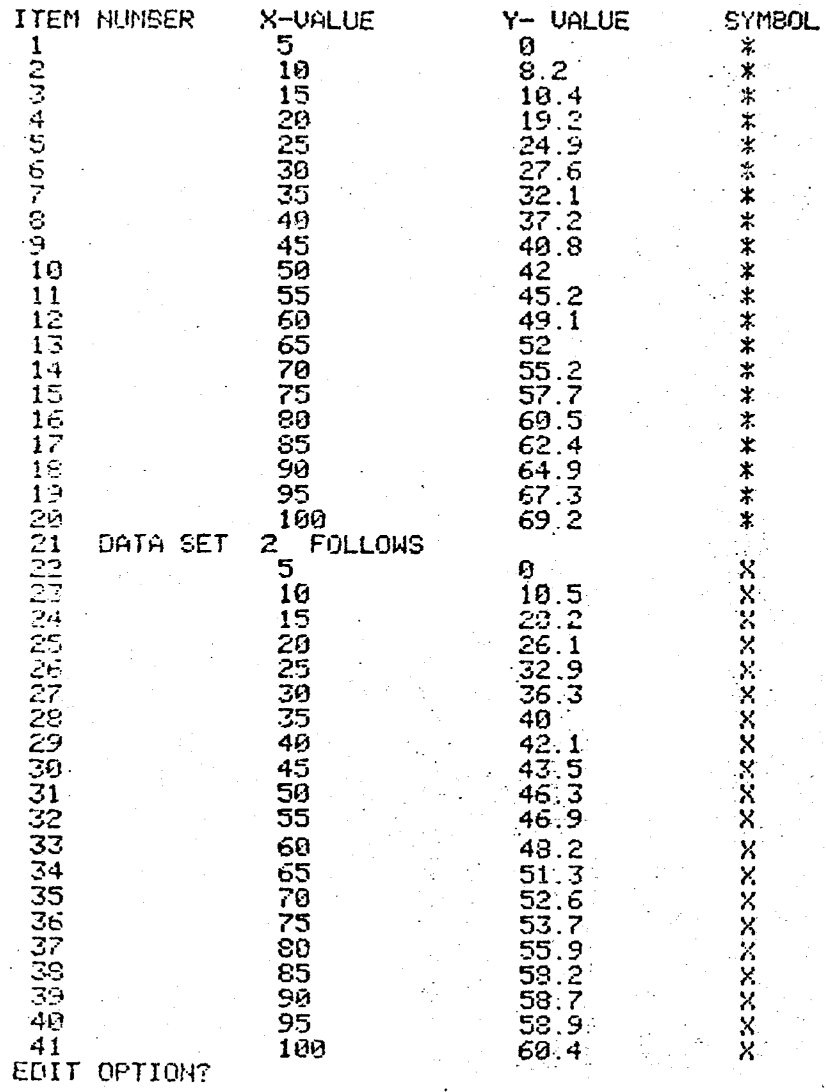
To obtain a graph quickly, you choose the default options, allowing the computer to determine scale factors to accommodate the range of the data. After supplying the top and bottom descriptive lines to the program, you request and obtain the display shown in Figure 8.

The results are accurate but not pleasing to the eye. Since you intend to use this graph in subsequent discussions with your associates, you elect to re-enter the interactive phase of EZPLOT and choose scale factors more appropriate to

TABLE II

HYPOTHETICAL DATA TO ILLUSTRATE EZPLOT

INDEPENDENT VARIABLE	DEPENDENT VARIABLES	
	CASE 1	CASE 2
5	0	0
10	8.2	10.5
15	10.4	20.2
20	19.2	26.1
25	24.9	32.9
30	27.6	36.3
35	32.1	40.0
40	37.2	42.1
45	40.8	43.5
50	42.0	46.3
55	45.2	46.9
60	49.1	48.2
65	52.0	51.3
70	55.2	52.6
75	57.7	53.7
80	60.5	55.9
85	62.4	58.2
90	64.9	58.7
95	67.3	58.9
100	69.2	60.4



```
ITEM NUNBER     X-VALUE        Y- VALUE      SYMBOL
 1              5              0              *
 2              10             8.2            *
 3              15             10.4           *
 4              20             19.2           *
 5              25             24.9           *
 6              30             27.6           *
 7              35             32.1           *
 8              40             37.2           *
 9              45             40.8           *
 10             50             42             *
 11             55             45.2           *
 12             60             49.1           *
 13             65             52             *
 14             70             55.2           *
 15             75             57.7           *
 16             80             60.5           *
 17             85             62.4           *
 18             90             64.9           *
 19             95             67.3           *
 20             100            69.2           *
 21   DATA SET  2  FOLLOWS
 22             5              0              X
 23             10             10.5           X
 24             15             23.2           X
 25             20             26.1           X
 26             25             32.9           X
 27             30             36.3           X
 28             35             40             X
 29             40             42.1           X
 30             45             43.5           X
 31             50             46.3           X
 32             55             46.9           X
 33             60             48.2           X
 34             65             51.3           X
 35             70             52.6           X
 36             75             53.7           X
 37             80             55.9           X
 38             85             58.2           X
 39             90             58.7           X
 40             95             58.9           X
 41             100            60.4           X
EDIT OPTION?
```

FIGURE 7. Display of data file for EZPLOT example.

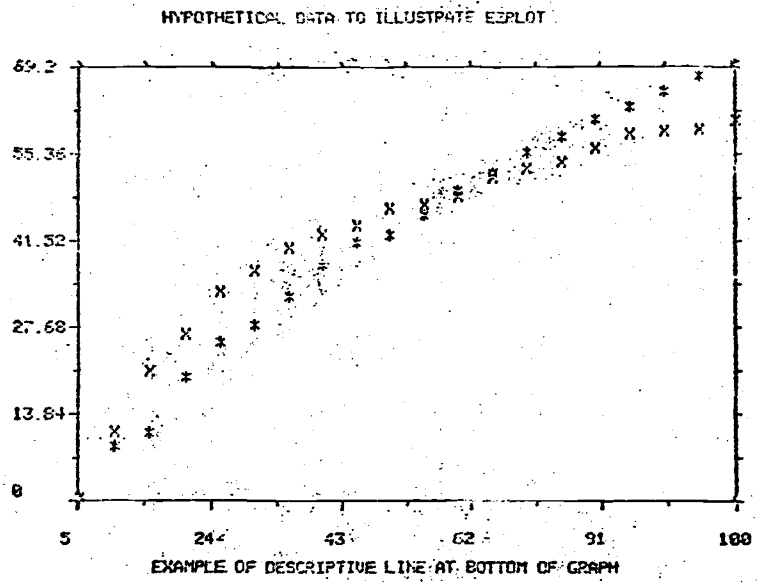


FIGURE 8. Graph of hypothetical data using automatic scaling option.

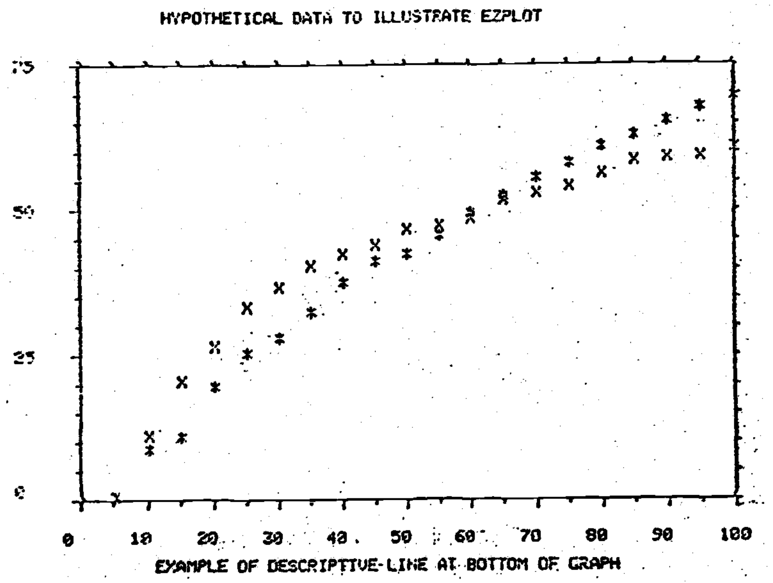


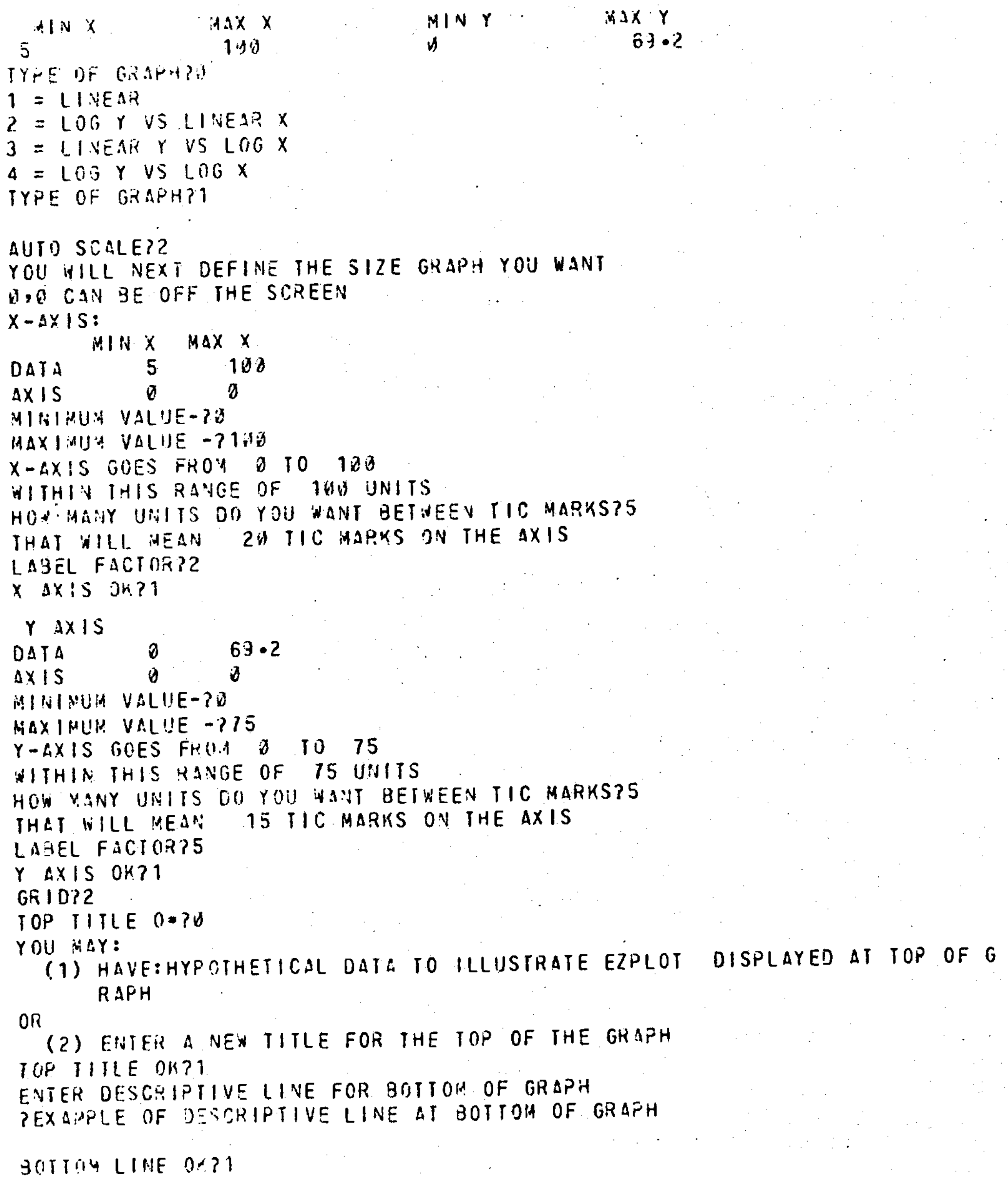
FIGURE 9. Graph of hypothetical data obtained by using manual scaling routine.

the data. The results shown in Figure 9 are much more pleasing. The dialogue leading to Figure 9 is shown in Figure 10.

You re-enter the interactive phase again because you have previously obtained some functions which fit the data reasonably well. To include these in the display you have to formulate the equations in BASIC, which is almost as natural as ordinary algebraic notation. Figure 11 shows the data with the two functions superimposed.

Since one of the functions contained an exponential expression, you know that a plot on semi-log paper would yield a straight line. To illustrate this aspect of your investigation, you select the appropriate semi-log option of EZPLOT. Again, you choose to dictate the choice of scale factors because you want the plot to be easily understood by your colleagues. The resulting semi-log plot, Figure 12, illustrates that EZPLOT quickly and accurately transforms the data, re-plotting it as it were on a new type of graph paper.

This example, which included relatively slow keyboard input of 40 pairs of points and output of four graphs, was completed within 30 minutes. This time may be reduced if the data are prepared previously on a data file stored in



```
  MIN X          MAX X               MIN Y          MAX Y
 5               100                 0                69•2
TYPE OF GRAPH?0
1 = LINEAR
2 = LOG Y VS LINEAR X
3 = LINEAR Y VS LOG X
4 = LOG Y VS LOG X
TYPE OF GRAPH?1

AUTO SCALE?2
YOU WILL NEXT DEFINE THE SIZE GRAPH YOU WANT
0,0 CAN BE OFF THE SCREEN
X-AXIS:
      MIN X  MAX X
DATA      5     100
AXIS      0     0
MINIMUM VALUE-?0
MAXIMUM VALUE -?100
X-AXIS GOES FROM  0 TO  100
WITHIN THIS RANGE OF  100 UNITS
HOW MANY UNITS DO YOU WANT BETWEEN TIC MARKS?5
THAT WILL MEAN   20 TIC MARKS ON THE AXIS
LABEL FACTOR?2
X AXIS OK?1

 Y AXIS
DATA      0     69•2
AXIS      0     0
MINIMUM VALUE-?0
MAXIMUM VALUE -?75
Y-AXIS GOES FROM  0  TO  75
WITHIN THIS RANGE OF  75 UNITS
HOW MANY UNITS DO YOU WANT BETWEEN TIC MARKS?5
THAT WILL MEAN   15 TIC MARKS ON THE AXIS
LABEL FACTOR?5
Y AXIS OK?1
GRID?2
TOP TITLE 0•?0
YOU MAY:
  (1) HAVE:HYPOTHETICAL DATA TO ILLUSTRATE EZPLOT  DISPLAYED AT TOP OF G
      RAPH
OR
  (2) ENTER A NEW TITLE FOR THE TOP OF THE GRAPH
TOP TITLE OK?1
ENTER DESCRIPTIVE LINE FOR BOTTOM OF GRAPH
?EXAMPLE OF DESCRIPTIVE LINE AT BOTTOM OF GRAPH

BOTTOM LINE OK?1
```

FIGURE 10. Interactions between EZPLOT and user which preceeded display of Figure 9.

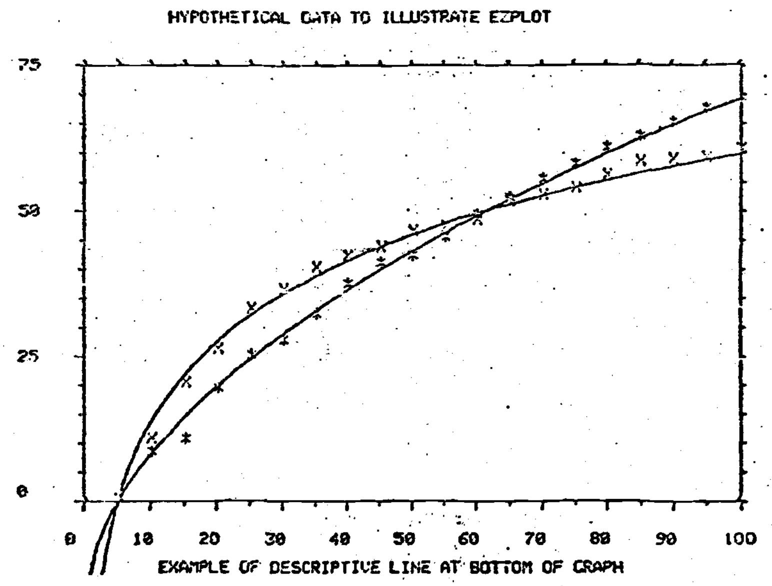


FIGURE 11. Graph of hypothetical data with superimposed functions.

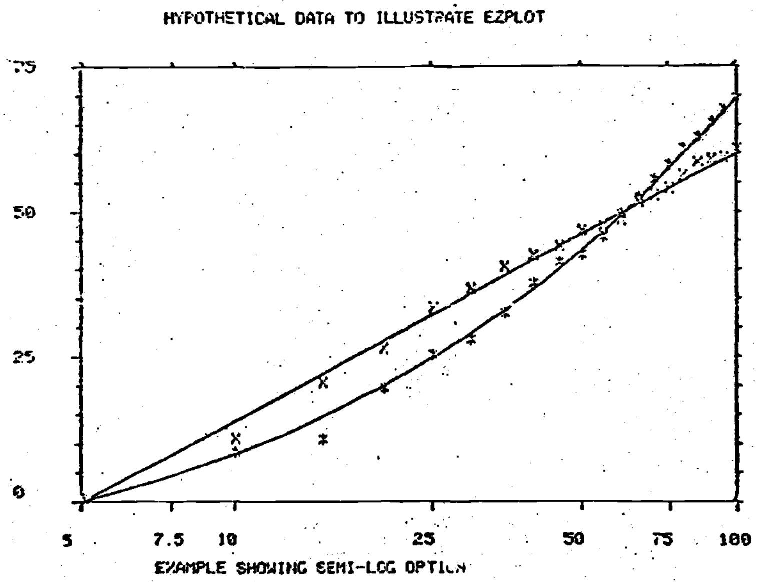


FIGURE 12. Graph of hypothetical data with superimposed functions on semi-log axes.

the computer. Investigators with little training in computer technology are currently using EZPLOT in the manner described in the foregoing example.

CONVERSION CONSIDERATIONS

Implementation of EZPLOT on a Univac 1108 operating under the CSCX operating system in Richland, Washington, has been described. It is appropriate to discuss the possibility of using it on other computers.

EZPLOT has been successfully tested on a similar computer configuration located in Los Angeles[6]. This time sharing computer service is available nation wide through telephone lines. Users with access to it will experience little difficulty installing and operating EZPLOT.

Users who do not have access to this system may consider converting EZPLOT to run on their own computer system. The success of such conversion will depend on whether their computer supports BASIC and, if it does, on whether their BASIC is interpreted or compiled.

If BASIC is available, it must support graphic output to the Tektronix terminal. This means that there must be some method of character string manipulation so that

[6] INFONET, an expanded version of CSCX, uses several UNIVAC 1108 computers. It is available through Computer Sciences Corporation, Los Angeles, California.

appropriate commands may be sent to the terminal. Some BASIC implementations do not provide this capability. Another problem is that some computers do not allow BASIC files as large as EZPLOT. In that case, some features of the program will have to be discarded to reduce its size. A minor consideration is differences in internal representation of keyboard characters among computers which may require conversion of portions of the program.

Assuming that EZPLOT is installed on the new computer, user convenience in operation of it will depend on whether BASIC is interpreted or compiled. Much of the user enthusiasm toward EZPLOT in Richland depends on the interpretive nature of CSCX-BASIC. This allows the user to interrupt program execution without loss of previously entered data. The interruption feature is essential to the simple way that the user inputs functions to be plotted with his data.

If the new BASIC is compiled rather than interpreted, EZPLOT will be much less convenient to use. If functions are desired, they will have to be inserted into the program prior to execution. If display of them indicates modifications are necessary, the user will have to stop the program, modify the functions, then re-initiate the program including re-entry of data and scale parameters.

If the new computer does not support BASIC, or if the new BASIC will not support graphic terminal output, EZPLOT will have to be converted to another language. Since the logical flow of the program has been thoroughly tested, language conversion would be relatively straight-forward. However, since most other languages, such as FORTRAN, are compiled, it will be inconvenient for the user to input and modify functions.

User convenience would suffer in another way because the new language would probably require more computer sophistication on the part of the user. For example, most FORTRAN versions do not allow the extent of free format input supported by BASIC. Further, it may be more difficult for the user to correctly input his functions if he has to write them in another language.

As an alternative to having the user input his functions directly into the program, a converted EZPLOT could employ some form of formula translator to construct the appropriate functions based on user responses. This would be an extensive modification to what is presently a relatively simple program.

In conclusion, EZPLOT could be converted to either a compiled version of BASIC or to another language such as FORTRAN. Both conversions would be much less convenient for the user. EZPLOT was designed to take advantage of an interpreted BASIC language, and from a user's standpoint, it should remain in that configuration.

V. APPLICATIONS

Two types of investigators utilize EZPLOT at Battelle Northwest. One group consists of those who interact with the program to produce graphs of their data. They were trained by the author in the use of both the CRT terminal and EZPLOT, and were given a copy of the Users Manual (Appendix B). Further contact with them is limited to complaints about hardware or software failures. Therefore, discussion of their applications of EZPLOT is not included in this dissertation.

Other investigators prefer to involve the author or another statistician in the analysis of their data. To them, EZPLOT appears as a tool of the statistician. Several examples of this type of application are presented here. In each case, the investigator has given permission that his results in graphical form be included in this dissertation along with a brief description of his experiment.

ANT ACTIVITY

In a study of the behavior of Harvester Ants (Pogonomyrmex occidentalis), Lee Rogers made a series of intensive observations at one ant colony for three days in the summer of 1971 (27). He counted the number of ants leaving and returning to the colony, recording the tally at five minute intervals. Other variables such as air and soil temperature, quantity of solar radiation, and relative humidity were also recorded throughout the day.

Later a statistician was asked to help incorporate these observations into a simulation of an ecosystem. An equation was needed which would relate the number of active ants to soil temperature; that is, at a given soil temperature how many ants could be expected to be outside the colony? No restrictions were placed on the type of equation to be fit to the data because there was no evidence that this aspect of ant behavior followed any particular model. A search of the literature led Lee to speculate that the relationship would be linear if indeed a relationship could be demonstrated.

The observations were keypunched and the two parameters of immediate interest, soil temperature and ant activity, were placed on a computer file. The statistician used EZPLOT to produce a preliminary plot of this data file. This graph, Figure 13, showed that there was no ant activity when the soil surface temperature was below 25 or above 50°C. Other than that, no particular pattern was evident.

In cases such as this, when the investigator gives no indication that the data should fit a particular model, the polynomial model is often applied. It is a simple

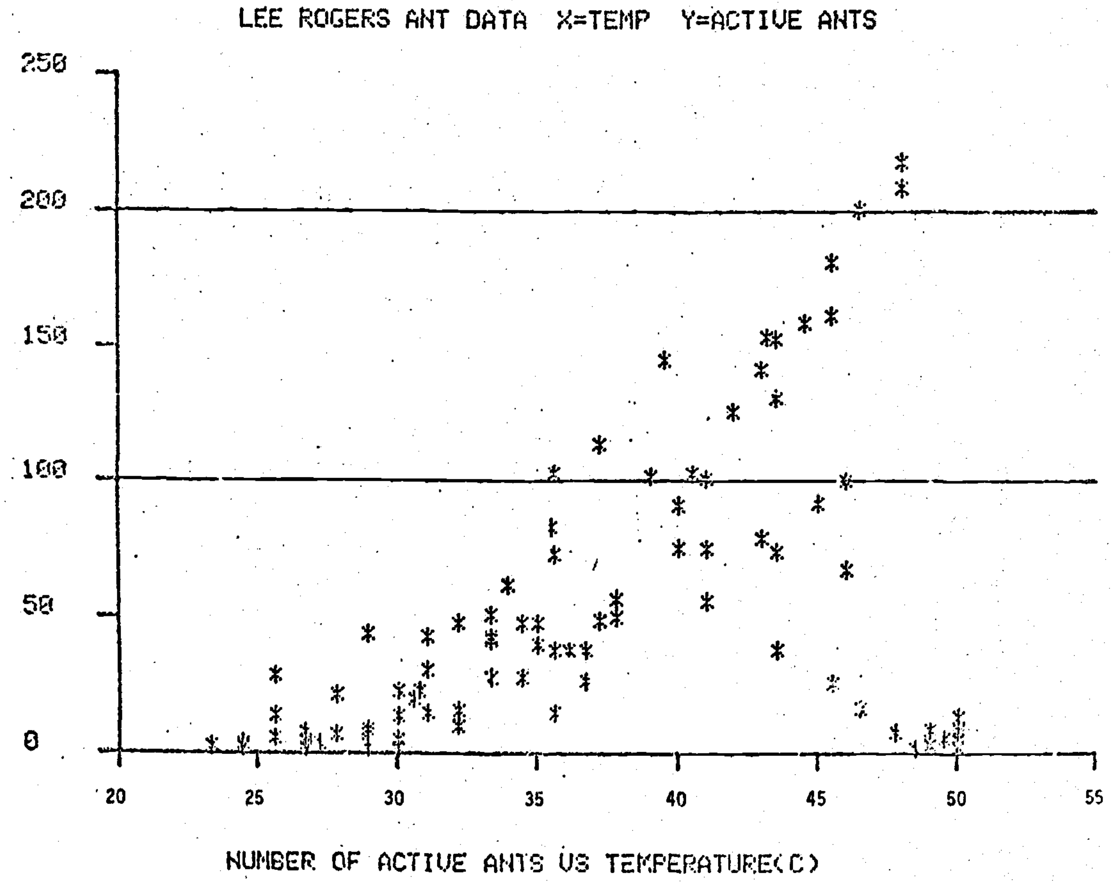


FIGURE 13. Number of active ants at a given soil surface temperature, composite of 3 days of observation.

regression process to compute the parameters of an equation of the form:

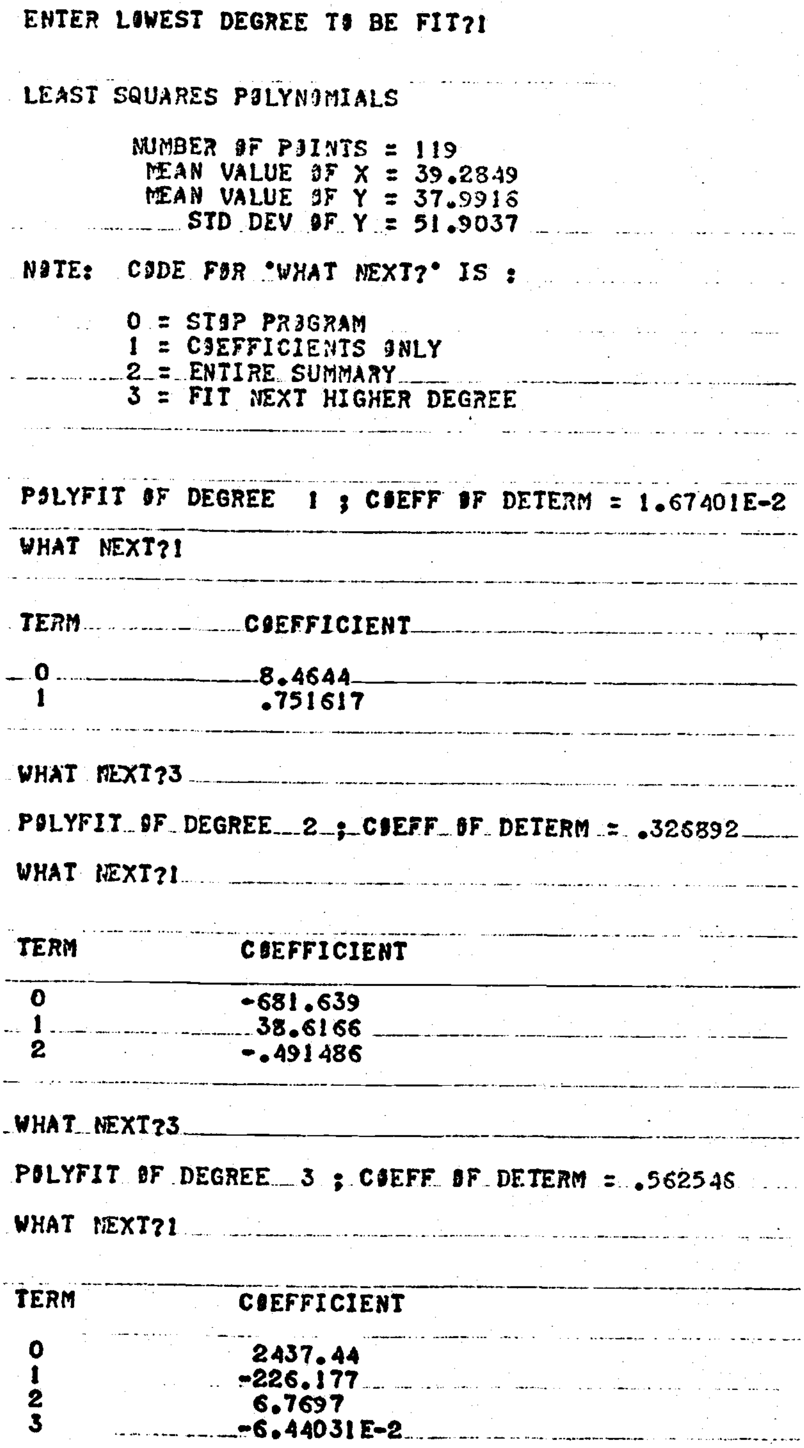
$$y = A + Bx + Cx^2 + Dx^3 + \text{etc.}$$

which give the closest fit to the data. Other models are more difficult to apply because they require that the statistician supply partial derivitives and estimates of the parameters.

The data file was used in conjunction with an interactive polynomial curve fitting routine (10, program no. 03-1823). This illustrates the concept of keeping EZPLOT a graph drawing program only. The ability to perform interactive curve fitting already existed, there was no need to incorporate it into the plot program. Some of the results of the polynomial fitting are shown in Figure 14 and 15.

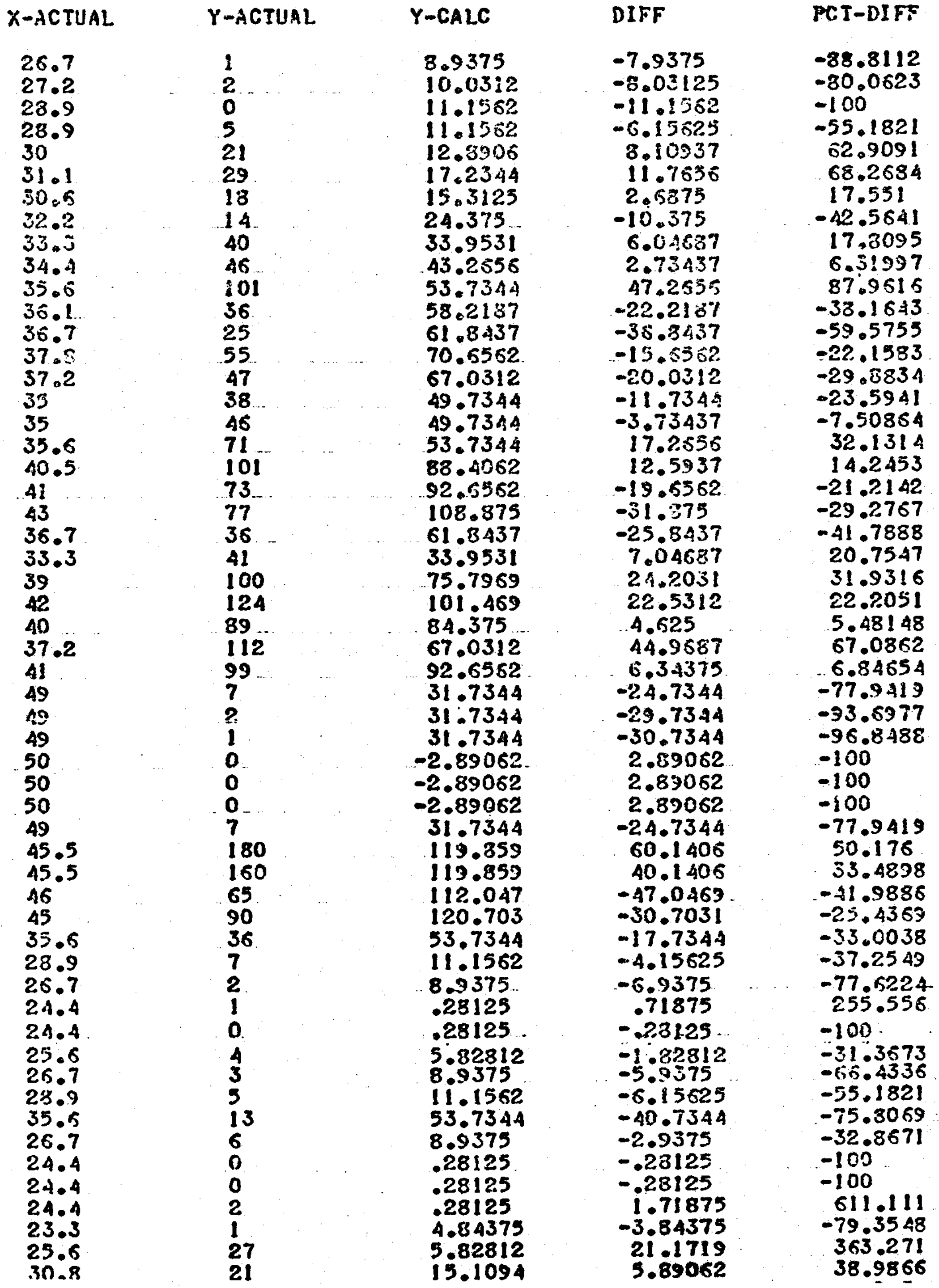
None of the resulting equations fit the data very well. Polynomial equations of order 1 through 8 were tried, but the coefficient of determination was in no case greater than 0.6 (1.0 indicates exact correspondence between the equation and the points). Clearly a conference between the statistician and the investigator was necessary to find some aspect about the data which would allow better curve fitting.

Without EZPLOT, the statistician would have shown



```
ENTER LØWEST DEGREE TØ BE FIT?1

LEAST SQUARES PØLYNØMIALS

        NUMBER ØF PØINTS = 119
         MEAN VALUE ØF X = 39.2849
         MEAN VALUE ØF Y = 37.9916
            STD DEV ØF Y = 51.9037

NØTE:  CØDE FØR 'WHAT NEXT?' IS :

       0 = STØP PRØGRAM
       1 = CØEFFICIENTS ØNLY
       2 = ENTIRE SUMMARY
       3 = FIT NEXT HIGHER DEGREE


PØLYFIT ØF DEGREE  1 ; CØEFF ØF DETERM = 1.67401E-2

WHAT NEXT?1

TERM            CØEFFICIENT

 0                8.4644
 1                .751617

WHAT NEXT?3

PØLYFIT ØF DEGREE  2 ; CØEFF ØF DETERM = .326892

WHAT NEXT?1

TERM            CØEFFICIENT

 0               -681.639
 1                38.6166
 2               -.491486

WHAT NEXT?3

PØLYFIT ØF DEGREE  3 ; CØEFF ØF DETERM = .562546

WHAT NEXT?1

TERM            CØEFFICIENT

 0                2437.44
 1               -226.177
 2                6.7697
 3               -6.44031E-2
```

FIGURE 14. Printout of a portion of the interactive polynomial regression analysis of ant activity data.



```
X-ACTUAL      Y-ACTUAL      Y-CALC       DIFF         PCT-DIFF

 26.7          1             8.9375      -7.9375      -88.8112
 27.2          2            10.0312      -8.03125     -80.0623
 28.9          0            11.1562      -11.1562     -100
 28.9          5            11.1562      -6.15625     -55.1821
 30            21           12.8906       8.10937      62.9091
 31.1          29           17.2344       11.7656      68.2684
 30.6          18           15.3125       2.6875       17.551
 32.2          14           24.375       -10.375      -42.5641
 33.3          40           33.9531       6.04687      17.8095
 34.4          46           43.2656       2.73437      6.31997
 35.6          101          53.7344       47.2656      87.9616
 36.1          36           58.2187      -22.2187     -38.1643
 36.7          25           61.8437      -36.8437     -59.5755
 37.8          55           70.6562      -15.6562     -22.1583
 37.2          47           67.0312      -20.0312     -29.8834
 35            38           49.7344      -11.7344     -23.5941
 35            46           49.7344      -3.73437     -7.50864
 35.6          71           53.7344       17.2656      32.1314
 40.5          101          88.4062       12.5937      14.2453
 41            73           92.6562      -19.6562     -21.2142
 43            77           108.875      -31.375      -29.2767
 36.7          36           61.8437      -25.8437     -41.7888
 33.3          41           33.9531       7.04687      20.7547
 39            100          75.7969       24.2031      31.9316
 42            124          101.469       22.5312      22.2051
 40            89           84.375        4.625        5.48148
 37.2          112          67.0312       44.9687      67.0862
 41            99           92.6562       6.34375      6.84654
 49            7            31.7344      -24.7344     -77.9419
 49            2            31.7344      -29.7344     -93.6977
 49            1            31.7344      -30.7344     -96.8488
 50            0           -2.89062       2.89062     -100
 50            0           -2.89062       2.89062     -100
 50            0           -2.89062       2.89062     -100
 49            7            31.7344      -24.7344     -77.9419
 45.5          180          119.859       60.1406      50.176
 45.5          160          119.859       40.1406      33.4898
 46            65           112.047      -47.0469     -41.9886
 45            90           120.703      -30.7031     -25.4369
 35.6          36           53.7344      -17.7344     -33.0038
 28.9          7            11.1562      -4.15625     -37.2549
 26.7          2            8.9375       -6.9375      -77.6224
 24.4          1            .28125        .71875       255.556
 24.4          0            .28125       -.28125      -100
 25.6          4            5.82812      -1.82812     -31.3673
 26.7          3            8.9375       -5.9375      -66.4336
 28.9          5            11.1562      -6.15625     -55.1821
 35.6          13           53.7344      -40.7344     -75.8069
 26.7          6            8.9375       -2.9375      -32.8671
 24.4          0            .28125       -.28125      -100
 24.4          0            .28125       -.28125      -100
 24.4          2            .28125        1.71875      611.111
 23.3          1            4.84375      -3.84375     -79.3548
 25.6          27           5.82812       21.1719      363.271
 30.8          21           15.1094       5.89062      38.9866
```

FIGURE 15. Detailed printout of portion of third degree polynomial fit to ant activity data.

Figures 14 and 15 to the investigator. It is difficult to use such tabular data to explain why the computer (a device equated to God by many non-mathematical investigators) could not fit the data. However, the investigator was shown Figure 16 in which the third and fourth degree polynomials, which came closest to fitting the data, are shown superimposed on the 119 data points.

Lee Rogers saw clearly from Figure 16 that the problem was in the erratic number of ants observed when the temperature was between 40 and 50 degrees. Examination of the complete set of observations showed that one of the neglected variables, time of day, was needed to fully understand the observations.

A general pattern was apparent when the observations were displayed chronologically. The ants started emerging from the colony in mid-morning when the soil temperature rose to 25 or more degrees. As the soil warmed, the number of active ants increased until the temperature reached 40 degrees. Between 11 AM and noon, as the soil temperature rose from 40 degrees to 50 degrees, the number of active ants decreased. After noon, there was a period when no ants were active. The ants remained in the colony until the soil temperature started to drop. When the soil surface was cool enough a

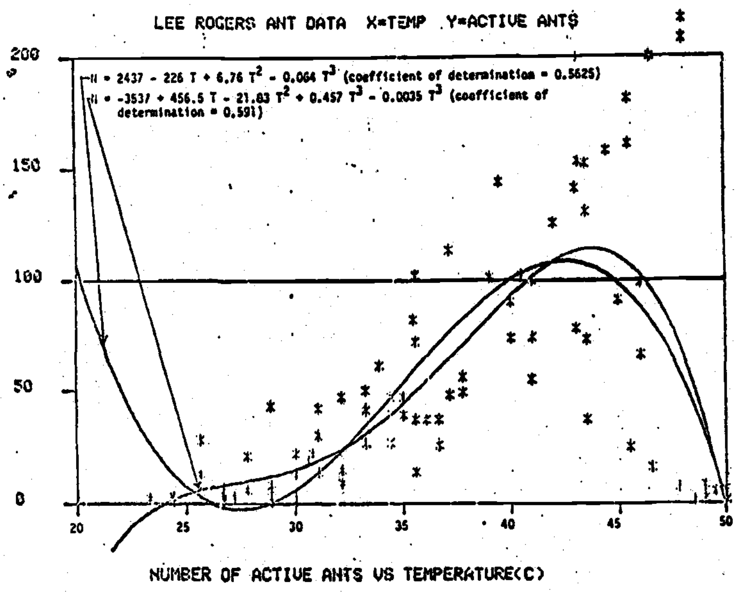


FIGURE 16. Number of active ants at given temperature with 3rd and 4th degree polynomial fits superimposed.

large number of ants emerged from the colony, then the number of active ants steadily decreased as the afternoon progressed.

This suggested that two equations were needed to relate ant activity to soil temperature, one for morning, and another for the afternoon. The data file was split into two portions, and the polynomial regression analysis was repeated. The resulting curves shown in Figure 17 and 18 fit the observations much better.

The success of this approach suggested that the linear model may also fit the data. Figures 19 and 20 show results of this approach. The linear regressions fit almost as well as the third and fourth degree polynomials and have the additional advantage of simplicity.

The ability to graph and re-graph the large number of observations made it much easier for the statistician to communicate with the investigator. The goodness of fit of the various regression equations was readily apparent to the investigator when presented in graphic form. He, in turn, was able to show the analysis of his observations to his co-workers in simple graphic form.

This analysis required two sessions at the computer terminal. First the interactive curve fitting routine was

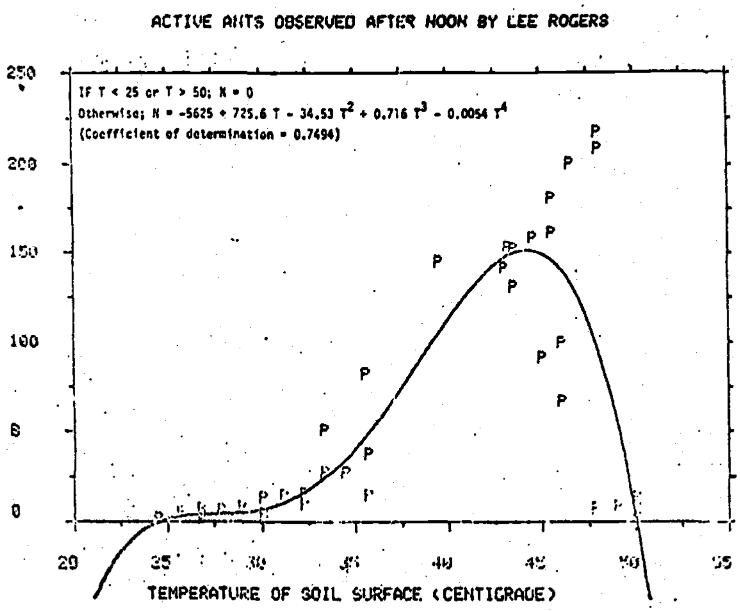


FIGURE 17. Number of active ants observed after noon at a given temperature, with polynomial regression fit superimposed.

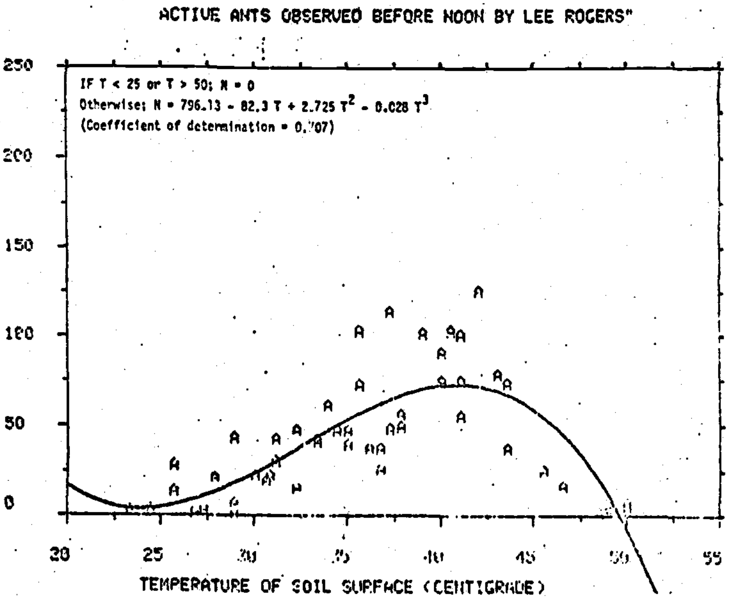


FIGURE 18. Number of active ants observed before noon at a given temperature, with polynomial regression fit superimposed.

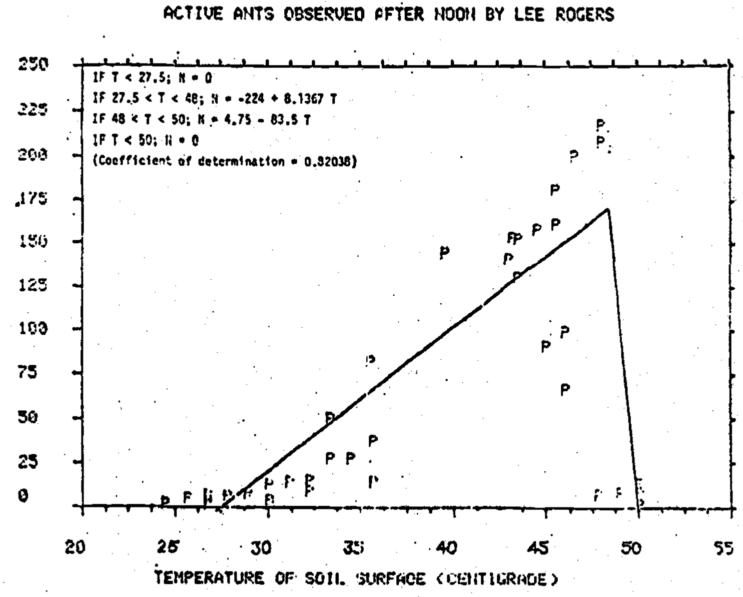


FIGURE 19. Number of active ants observed after noon at a given temperature, with linear model superimposed.

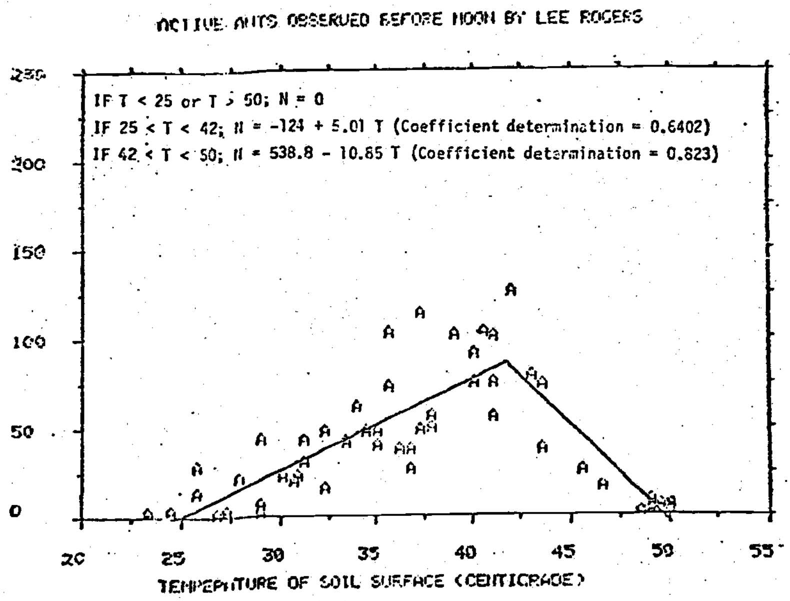


FIGURE 20. Number of active ants observed before noon at a given temperature, with linear model superimposed.

used and Figures 13 and 16 were prepared with EZPLOT. The elapsed time for this session was approximately 1½ hours. After these results were shown to the investigator, another computer session was planned. Dividing the data into morning and afternoon observations was done by manually separating the cards and transfering the two decks to new computer files.

The polynomial and linear curve fitting programs were run and graphs of the results prepared in about one hour. The savings of time and increase in communication ability due to EZPLOT are clearly evident in the analysis of Lee Roger's ant activity observations.

EINSTEINIUM 253 EXCRETION ANALYSIS

This application is another illustration of the simplification in communication between the statistician and investigator which results from the ability to automatically produce graphs of large data sets. In this case, the statistician wanted to show the effect of several data reduction techniques on the outcome of the regression analysis requested by the investigator. Neither the statistician nor the investigators would have plotted all 189 data points by hand, but analysis of such a graph, drawn by EZPLOT, allowed both of them to quickly understand ramifications of the analysis.

The analysis involved a portion of a study of the toxicity of the trans-plutonium nuclide einsteinium-253, which is a pure alpha emitter. This material is being produced in anticipation of future use in radioisotope power sources, often called nuclear batteries, to be encapsulated and implanted in recipients of artificial hearts. Einsteinium-253 is an ideal candidate for this use because the short range of alpha particles limits the potential hazard to man to situations in which the encapsulation fails and the material is deposited in some radiosensitive organ of the body. Since this man-made nuclide has been available only in very small quantities, little information is available regarding its fate after it enters the body.

B. J. McClanahan and H. A. Ragan are investigating the metabolism of einsteinium-253 in a representative large laboratory animal species, miniature swine (25). A group of weanlings were injected with 3 micro Curies/kg of einsteinium-253 and held for observation. One important parameter, excretion rate, was monitored by collecting 24 hour urine and feces samples daily for the first ten days after injection and at frequent intervals thereafter.

Radiochemical analysis of the samples enabled the investigators to calculate the percent of injected dose excreted each day.

A graph of the mean percent excreted each day was prepared manually. This graph, on semi-log paper, showed the classic two compartment exponential excretion pattern (12). This is interpreted by assuming that two modes of metabolism contribute to the total excretion pattern. A portion of the injected material is quickly excreted via the urine; it is assumed that it passes directly from the blood to the kidney. Another portion is excreted over a long time period; it is assumed that this material is transfered from the blood to some intermediate site where it is retained and slowly released back to the blood from which part of it goes to the kidney.

If the two compartment model accounts for the observed einsteinium-253 excretion pattern, the parameters A and C of the equation:

$$X = A\ e^{-Bt} + C\ e^{-Dt} \text{ (where } t = \text{time)}$$

indicate the proportion of the injected material deposited in each compartment, while B and D indicate the half-time of residence in the compartment. The statistician was asked to evaluate these parameters for these observations.

The data from the individual pigs were transferred to a computer file and the mean percent excreted each day was re-evaluated to eliminate the possibility of mathematical error in the previous calculations. Graphs of the data and the summary statistics are shown in Figures 21 and 22.

The statistician used several non-linear regression techniques to evaluate the fit of the model to the means of the observations and to various weightings of the means. Using EZPLOT, he displayed these evaluations of the model superimposed on the 189 data points. This helped him explain the effect of these statistical techniques to the investigators. They were able to help refine the curve fitting by identifying questionable data points which were contributing to the large variance on a few days.

Two of the graphs used in this analysis are shown as Figures 23 and 24. The first semi-log plot was used in a discussion of the problems associated with extrapolating beyond the duration of the experiment. Needless to say, converting a manual plot from a scale of 0 to 70 days into one from 0 to 200 days and displaying two functions accurately is too difficult to demand serious consideration just for

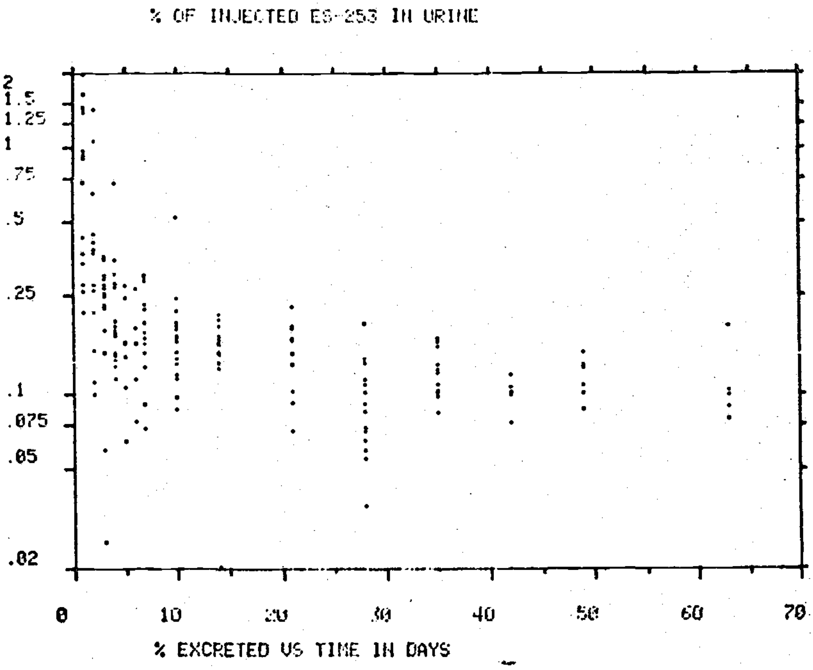


FIGURE 21. Percent of injected einsteinium-253 excreted in urine of miniature swine.

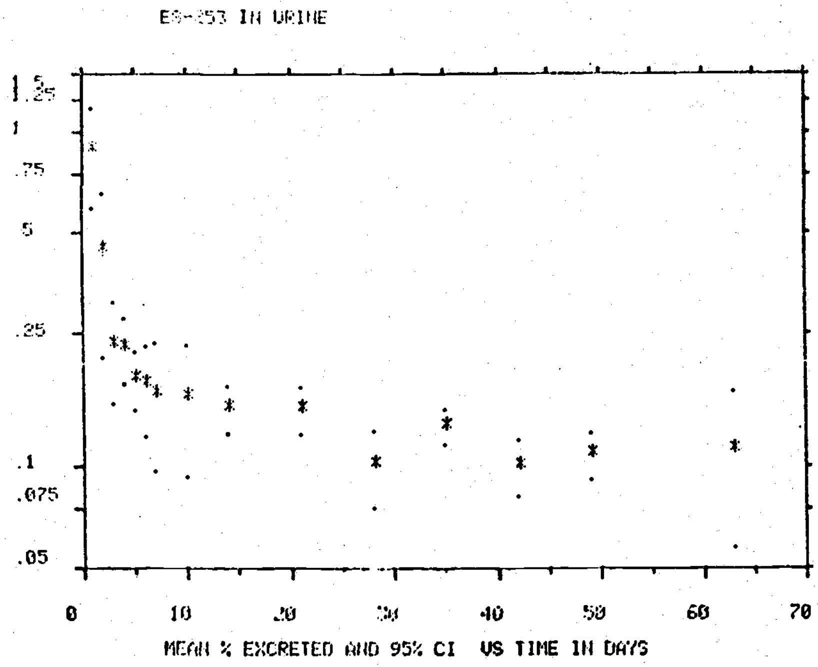


FIGURE 22. Mean and 95% confidence interval of percent of injected einsteinium-253 excreted in urine.

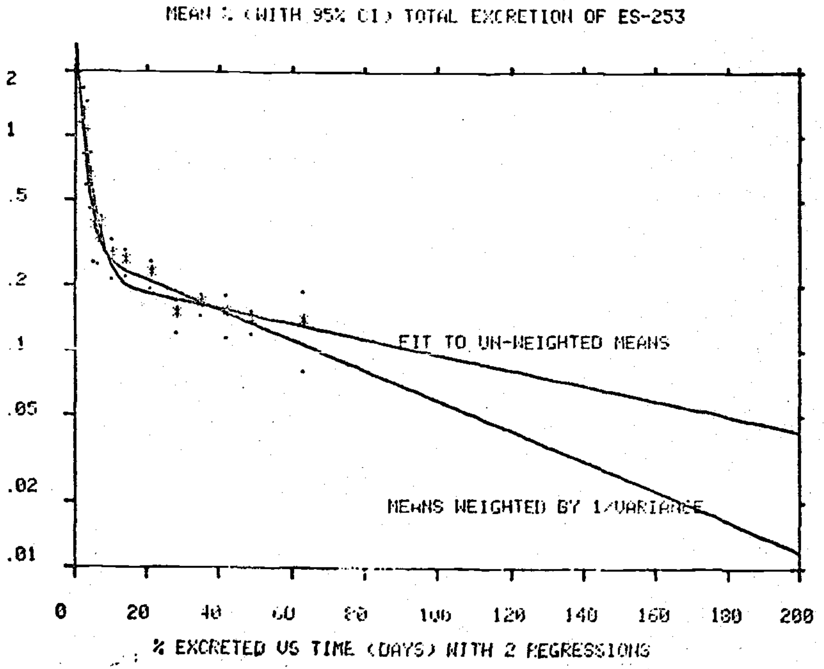


FIGURE 23. Mean percent einsteinium-253 excreted with two nonlinear regression lines shown. Horizontal scale extended to 200 days.

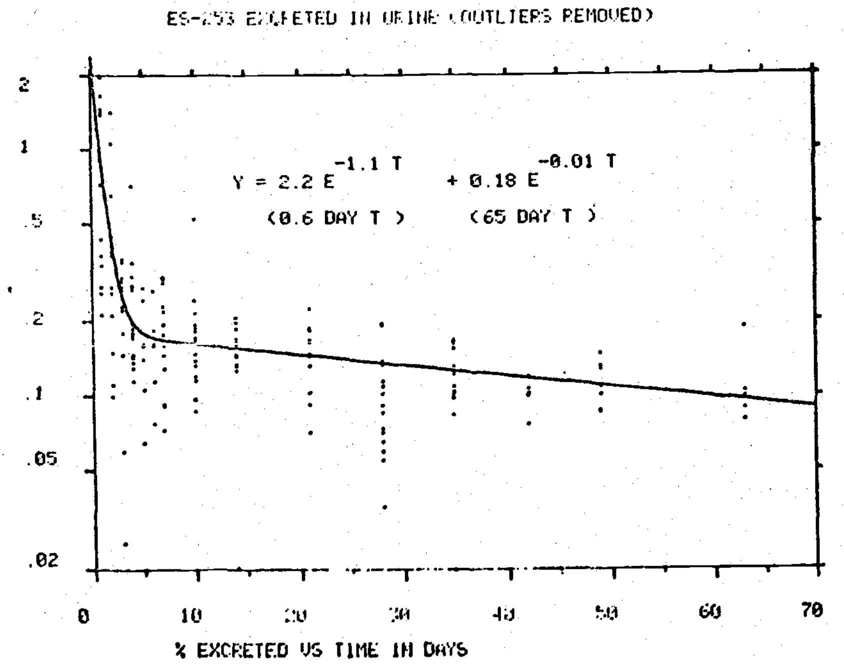


FIGURE 24. Mean percent einsteinium-253 excreted with outliers removed, illustrating rounded regression function.

the sake of a discussion. However, it is so easy for the computer to adjust the scale that the statistician didn't hesitate to specify this additional plot.

The second graph, Figure 24, shows the refined fit to the urine data. It appears to fit the data well, and will be used in publication of the experimental results.

PROBIT ANALYSIS

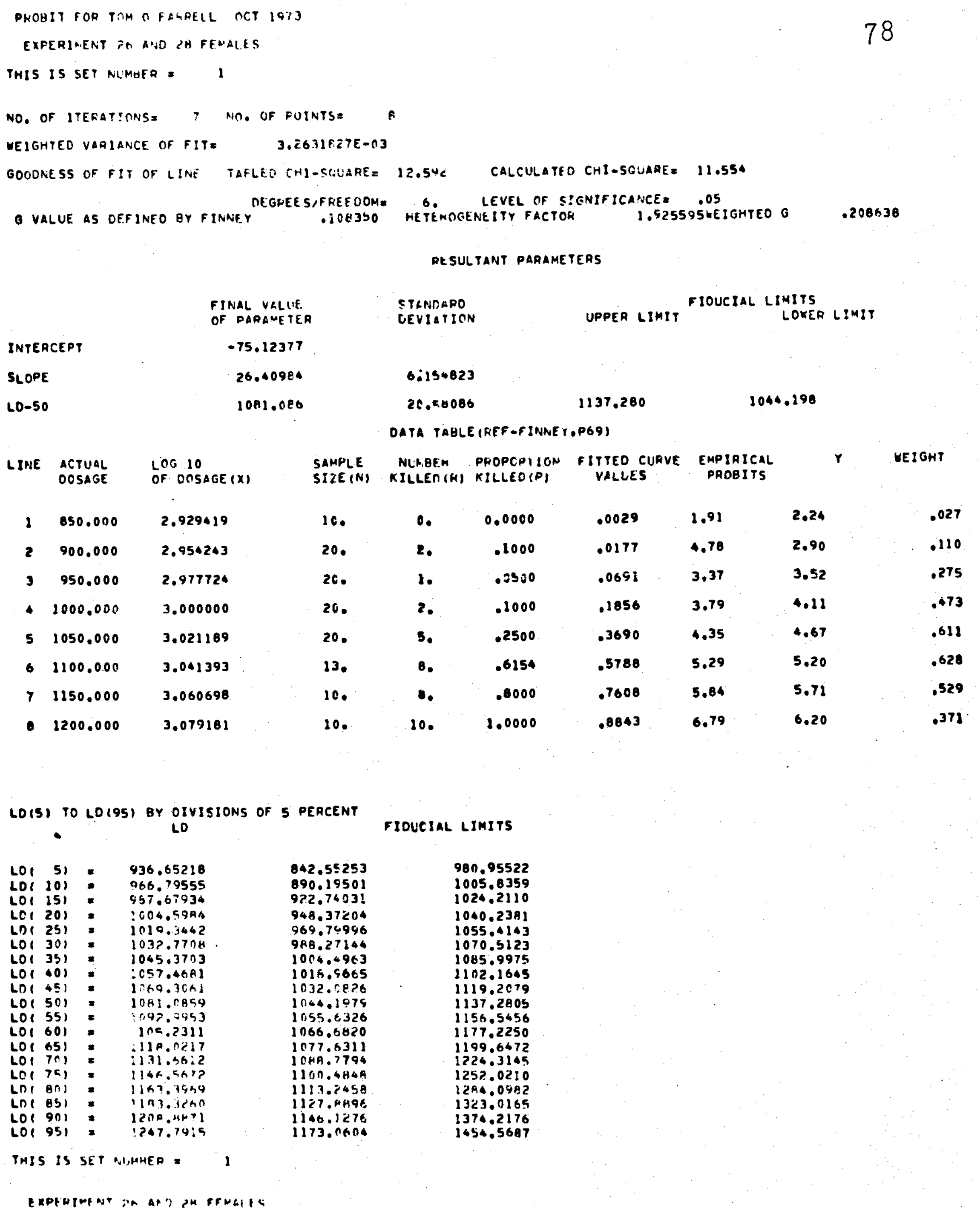
EZPLOT was initially used in this application to help a particular investigator understand the implications of a computer printout of an analysis of his data. Response by this and other investigators was so favorable that this use of graphic output to supplement numerical results is now routine.

Probit analysis (14) is a technique for interpreting the results of bioassay experiments in which groups of organisms are exposed to various concentrations of toxicant for an arbitrary time. The aim of a bioassay is to relate dose of toxicant to some observable end point such as death of the organism. Probit analysis provides a measure of the median lethal dose, that is, the concentration of toxicant necessary to kill half of the test population in a given time, called LD_{50}.

As with any statistical data reduction technique, probit analysis also determines the error associated with the predicted LD_{50}. The technique is tedious and is best done by a computer program. Output from one such program is shown in Figure 25. Most of the information in the figure is of interest only to the statistician because it helps him understand the validity of the predicted LD_{50}. The investigator is interested in the LD_{50} value and in some measure of the confidence in this value.

A problem in communication developed when one investigator asked the statistician how to use the values printed by the computer. The investigator found the LD_{50} value for the experiment, and assumed that he should summarize his experiment by reporting the LD_{50} and the standard deviation associated with that estimate. The statistician knew that the proper measure of confidence in the estimate is the upper and lower 95% fiducial limits, but he was unable to explain to the investigator's satisfaction why this was true.

The table of estimated concentrations needed to kill given percentages of the population, printed at the bottom of Figure 25, also includes fiducial limits for each estimate. These estimates along with the observed data were plotted as shown in Figure 26.



```
 PROBIT FOR TOM O FARRELL  OCT 1973

  EXPERIMENT 26 AND 28 FEMALES

THIS IS SET NUMBER =      1


NO. OF ITERATIONS=    7   NO. OF POINTS=      8

WEIGHTED VARIANCE OF FIT=        3.2631827E-03

GOODNESS OF FIT OF LINE   TABLED CHI-SQUARE=  12.592     CALCULATED CHI-SQUARE=  11.554

                              DEGREES/FREEDOM=    6.     LEVEL OF SIGNIFICANCE=   .05
 G VALUE AS DEFINED BY FINNEY          .108350   HETEROGENEITY FACTOR        1.925595WEIGHTED G       .208638

                                                 RESULTANT PARAMETERS


                          FINAL VALUE          STANDARD                             FIDUCIAL LIMITS
                          OF PARAMETER         DEVIATION             UPPER LIMIT              LOWER LIMIT

INTERCEPT                  -75.12377

SLOPE                       26.40984           6.154823

LD-50                       1081.086           20.58086            1137.280          1044.198
                                              DATA TABLE(REF-FINNEY,P69)

LINE  ACTUAL       LOG 10              SAMPLE   NUMBER   PROPORTION  FITTED CURVE  EMPIRICAL           Y          WEIGHT
      DOSAGE       OF DOSAGE(X)        SIZE(N)  KILLED(R) KILLED(P)    VALUES       PROBITS


  1   850.000      2.929419              10.       0.      0.0000       .0029       1.91        2.24             .027

  2   900.000      2.954243              20.       2.       .1000       .0177       4.78        2.90             .110

  3   950.000      2.977724              20.       1.       .0500       .0691       3.37        3.52             .275

  4  1000.000      3.000000              20.       2.       .1000       .1856       3.79        4.11             .473

  5  1050.000      3.021189              20.       5.       .2500       .3690       4.35        4.67             .611

  6  1100.000      3.041393              13.       8.       .6154       .5788       5.29        5.20             .628

  7  1150.000      3.060698              10.       8.       .8000       .7608       5.84        5.71             .529

  8  1200.000      3.079181              10.      10.      1.0000       .8843       6.79        6.20             .371



LD(5) TO LD(95) BY DIVISIONS OF 5 PERCENT
                         LD                                 FIDUCIAL LIMITS


LD(  5)   =    936.65218           842.55253          980.95522
LD( 10)   =    966.79555           890.19501          1005.8359
LD( 15)   =    987.67934           922.74031          1024.2110
LD( 20)   =    1004.5984           948.37204          1040.2381
LD( 25)   =    1019.3442           969.79996          1055.4143
LD( 30)   =    1032.7708           988.27144          1070.5123
LD( 35)   =    1045.3703           1004.4963          1085.9975
LD( 40)   =    1057.4681           1016.9665          1102.1645
LD( 45)   =    1069.3061           1032.0826          1119.2079
LD( 50)   =    1081.0859           1044.1979          1137.2805
LD( 55)   =    1092.9953           1055.6326          1156.5456
LD( 60)   =    1105.2311           1066.6820          1177.2250
LD( 65)   =    1118.0217           1077.6311          1199.6472
LD( 70)   =    1131.6612           1088.7794          1224.3145
LD( 75)   =    1146.5672           1100.4848          1252.0210
LD( 80)   =    1163.3969           1113.2458          1284.0982
LD( 85)   =    1183.3260           1127.8896          1323.0165
LD( 90)   =    1208.8871           1146.1276          1374.2176
LD( 95)   =    1247.7915           1173.0604          1454.5687

 THIS IS SET NUMBER =      1

   EXPERIMENT 26 AND 28 FEMALES
```

FIGURE 25. Output of probit analysis computer program.

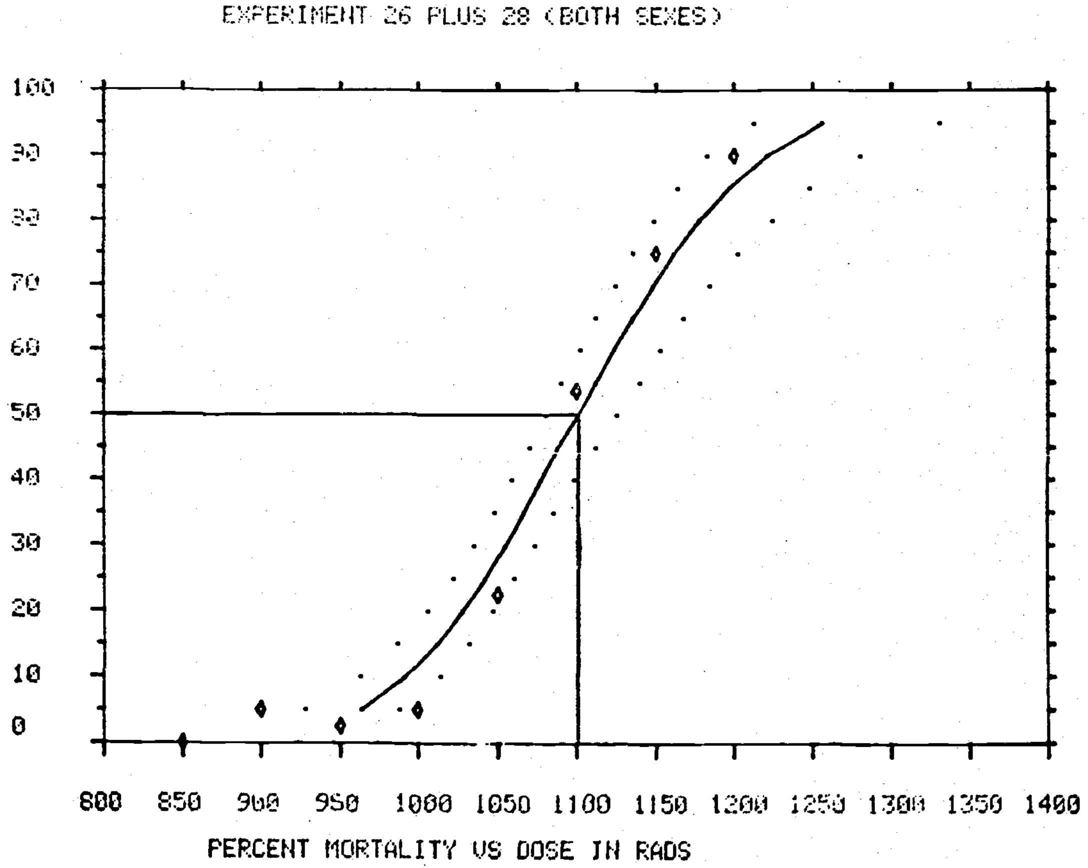


FIGURE 26. Graph of output of probit analysis program with partially erased grid lines used to highlight the LD_{50} point.

The investigator was able to be more at ease with the fiducial limit concept after seeing this graph. The non-symetrical nature of the limits is apparent in the figure. It is also clear how the closeness of the predicted survival curve to the observed data is reflected in the fiducial limits.

However, the investigator was more impressed with the ability to convey so much information about his experiment in a single graph. Previously he had published tables of LD_{50} values relating his investigations. Now he plans to publish graphs similar to Figure 26.

ZINC-65 IN ALGAE

The previous examples illustrate the use of computer generated graphs as a means of communication. This application of EZPLOT demonstrates that graphs often reveal relationships not obviously apparent in tabular data. The statistician and the investigator were led to reevaluate both the purpose and the procedure of the experiment after analyzing the graphs produced by EZPLOT.

Zinc-65 was formerly released in large quantities into the Columbia River from the nuclear reactors on the Hanford Project. The movement of this or other radionuclides through the aquatic food chain is a potential hazard to people who eat fish from the river. Burt Cushing is investigating one aspect of this problem, the kinetics of zinc-65 in river bottom phytoplankton (13).

Algae colonies are allowed to form on glass plates submerged in the Columbia River. The plates are brought into the laboratory and attached to Geiger-Muller tubes in a trough of running water pumped from the river. Ordinarily, the water passes through the trough once, but it can be made to re-circulate. Reactor operation is simulated by placing the trough in the re-circulate mode and adding a small quantity of zinc-65 to the water. Some of the radionuclide adheres to the algae, some is incorporated into the structure of growing algae, and most of it remains in the water. After a given time, removal of the reactors is simulated by disposing of the contaminated water and introducing fresh water.

Count rates from the G-M tubes were automatically recorded every 30 minutes. The statistician[7] was asked to analyze these data in order to relate decline in radioactivity after addition of fresh water to the growth rate and life span of members of the algae colony. A computer

[7] John Thomas was the statistician in this case. The author worked with him to produce the graphs.

file of the data was constructed for use with an interactive non-linear regression analysis program. Two models were proposed, one based on exponential growth of the colony, the other based on a power function.

The interactive regression programs require that the user supply reasonable estimates of the parameters of the model. The statistician needed a graphic representation of the data in order to make these estimates. To save time, he manually plotted representative data by choosing every tenth point. After adjusting his estimates until the regression program converged, he used EZPLOT to draw the functions superimposed on all the data values, shown in Figure 27.

His attention was immediately drawn to the apparent sinusodial pattern in the data about the exponential fit. This was unexpected and would have gone undetected if only manual data display techniques were available. The cause of the pattern could be either measurement error or some unexpected behavior of the algae.

A conference with the investigator confirmed the automatic nature of the recording apparatus. Since the period of the apparent sine wave was approximately 24 hours, the lighting conditions in the laboratory became suspect. A check with the building manager confirmed that the fluores-

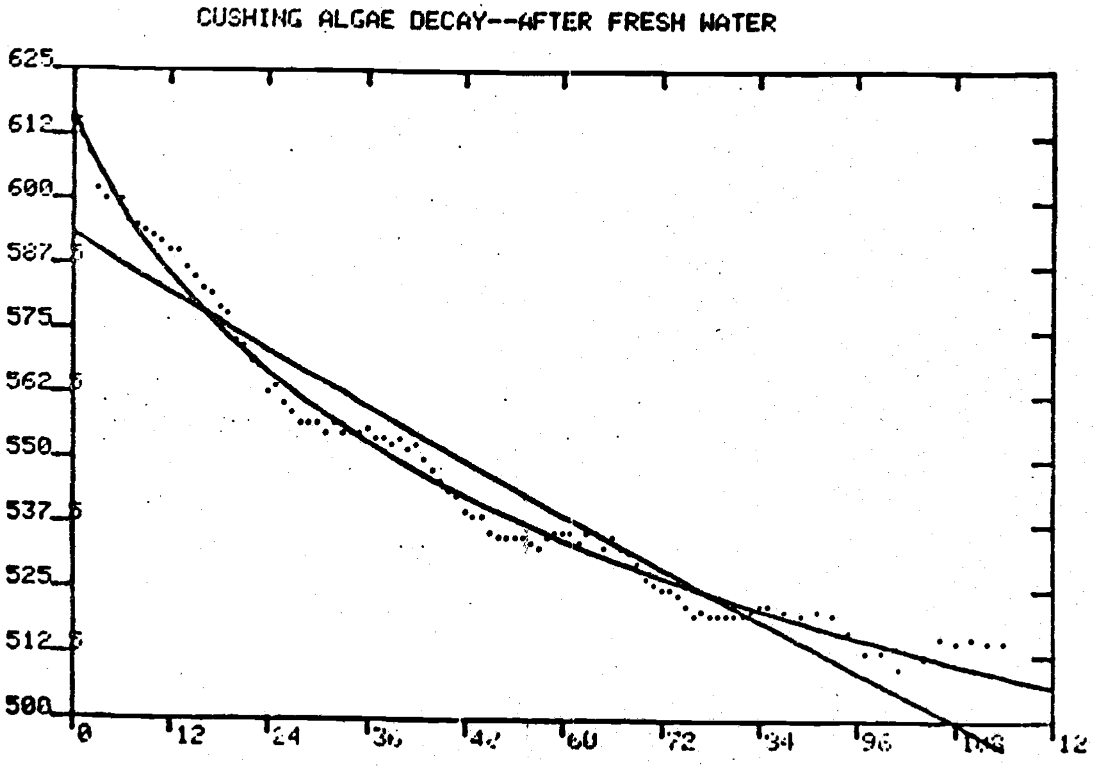


FIGURE 27. Radioactivity observed in algae colony after fresh water replaced zinc-65 labeled water. Results of two nonlinear models are superimposed on the data.

cent lights were on at all times as requested by the investigator. There emerged from that conference the strong suspicion that the algae "remember" the diurnal pattern of natural light in the river. Circadian rhythm has not been reported for this type of algae, but is observed in other plant species (23).

The apparent fluxuation in algae growth was further analyzed by using the regression analysis program to fit the residuals from the exponential fit of the radioactivity observations to a sine function model. The results, plotted in Figure 28, indicate that the period exhibited by the algae colony is not constant, but becomes longer as time progresses. This may be due to a loss of "memory" after the colony is removed from natural lighting conditions. It may also be due to dilution of the colony by growth of new members in the laboratory.

These findings prompted re-evaluation of the experimental technique. In addition, they changed the direction of the investigation. Further tests are planned to demonstrate the reproducibility of the apparent sinusodial pattern of radioactivity. A test is envisioned with an artifical algae colony (a partial roll of toilet paper or a swatch of cloth) to observe the count rate pattern under

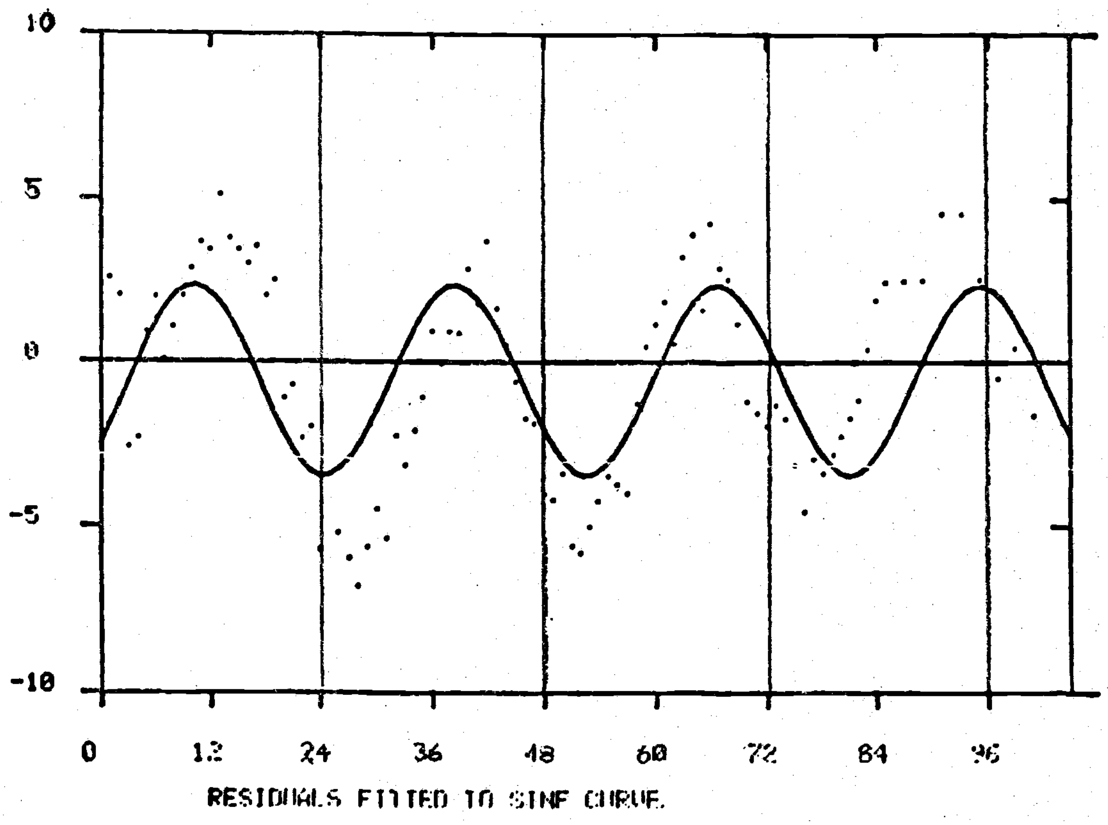


FIGURE 28. Residuals from the modeling of zinc-65 activity in algae after freshwater added, fit to a sine function. This illustrates the circadian rhythm discovered in the algae.

conditions of adsorption only. Experiments are also planned to test the effect of variation in the time between removal from the river and introduction of the radionuclide.

In general, the investigation now focuses on the kinetics of algae growth more than it does on the kinetics of zinc-65 in the system.

VI. SUMMARY AND CONCLUSIONS

Graphs were discussed as instruments of communication within the scientific community and as tools for revealing relationships not readily apparent in tabular data. Manual methods of graph preparation are tedious. None of the computer aided techniques examined were entirely satisfactory to the researcher who is not also a computer programmer. Such users tend to avoid computerized methods of graph drawing because they are either expensive, complicated to initiate, or involve lengthy turnaround times.

A relatively new type of graphics display, utilizing storage type CRT's, was shown to offer acceptable compromises between output quality and user convenience. Other systems using such terminals have been described (7, 15, 17, 22) which successfully meet the needs of more sophisticated users at reasonable cost. Strategy for development of an unsophisticated, interactive, computer graphics system using these terminals was discussed. User convenience was the main element in that strategy. The system was designed for investigators in fields such as biology and ecology who have neither the inclination nor the time to become sophisticated in computer techniques.

Hardware for the system consists of: Tektronix 4002 or 4010 graphic terminals and optional hard copy generators, connected to a Univac 1108 computer running under the CSCX operating system. Software was written in BASIC because of the immediate response available under this operating system and because BASIC is very forgiving to unsophisticated users. Prior to software implementation, careful consideration was given to techniques of interactive programming which would maximize user convenience. The menu picking technique was adopted with display of the menu only at the user's request.

Goals for the system were developed through close contact with a group of potential users. These goals appear rather modest in light of hardware capabilities, because emphasis was placed on usable output within a reasonable implementation time rather than on exotic results after several man-years of development. These goals were implemented a few at a time so that user experience could influence program development.

Implementation of the system required extensive modification of the plot driving package TEKPLOT to meet the program size limitation imposed by CSCX. Interface with this minimum plot driving package required development of routines to generate the type of graph format requested by the users. Finally, the interactive portion of the program was modified several times in response to user feedback.

Four applications of EZPLOT were discussed in detail to show the improvement in communication between investigators achieved through easily obtained graphs. These applications also illustrate the importance of a creative human effort in the pictorialization of output from data analysis.

The main conclusions which may be drawn from this work are:

- Inclusion of user input in the design phase and establishing reasonable goals, which were implemented a few at a time, were responsible for the fairly rapid implementation of the system.
- Allowing the user to supply logic at run time, as in the case of labeling of logarithmic axes, not only reduced program size and execution time, it also met with user enthusiasm. Never underestimate the user!
- The ability of the user to see his graph immediately encourages his creativity by inviting modification of scale parameters "just to see how it would look this way".

- Restricting EZPLOT to the task of graph preparation rather than including data analysis (i.e., regression) has not been a hardship to users. It encouraged development of a reliable general purpose tool rather than a graphic appendage to an analysis package.
- This system of interactive computer graphics has provided a group of investigators who are not interested in computers with a valuable and useful research tool.

BIBLIOGRAPHY

1. American Heritage Publishing Co. The American Heritage Dictionary of the English Language. Boston: American Heritage Publishing Co. and Houghton Mifflin, 1973.

2. American Standards Association. American Standard Illustrations for Publication and Projection. New York: American Society of Mechanical Engineers, publication number ASA Y15.1-1959, 1959.

3. American Standards Association. American Standard for Engineering and Scientific Graphs for Publication. New York: American Society of Mechanical Engineers, publication number ASA Z15.3-1943, 1943.

4. Arkin, H., and Colton, R. R. Graphs - How to Make and Use Them. New York: Harper & Brothers, 1936.

5. Auerbach Publishers. Auerbach on Digital Plotters and Image Digitizers, Princeton, New Jersey: Auerbach Publishers, 1972.

6. Barnett, Eugene H. Programming Time-Shared Computers in BASIC, New York: Wiley-Interscience, 1972.

7. Brown, Daniel J. An Interactive Graphical Analysis Information Display System. Unpublished M.S. dissertation, University of Washington, 1973.

8. Cameron, A. J. A Guide to Graphs. Oxford: Pergamon Press, 1968.

9. Comer, D. B. and Spillman, R. R., Modern Technical and Industrial Reports. New York: G. P. Putnam & Sons, 1962.

10. Computer Sciences Corp. CSCX BASIC Library. Los Angeles: Computer Sciences Corp. 1969.

11. Computer Sciences Corp. CSCX BASIC Reference. Los Angeles: Computer Sciences Corp., 1970.

12. Computer Sciences Corp. CSCX System Summary. Los Angeles: Computer Sciences Corp., Document number

E00005-00-00, 1970.

13. Cushing, C. E., and Watson, D. G., "Decline of Radioactivity in Biota Following Shutdown of Hanford Reactors." Pacific Northwest Laboratory Annual Report for 1972 to the USAEC Division of Biological and Environmental Research Volume 1: Life Sciences, Part 2: Ecological Studies., Washington, D.C.: US Atomic Energy Commission publication number BNWL-1750 PT2 UC 48. 1973. pp. 6.16-6.18.

14. Finney, D. J. Probit Analysis. 3rd edition. London: Cambridge University Press, 1971.

15. Green, R. Elliot., and Parslow, R. D. eds. Computer Graphics in Management, Case Studies of Industrial Applications. Princeton, New Jersey: Auerbach Publishers, 1970.

16. Kemeny, John G., and Kurtz, Thomas E., BASIC Programming. 2nd ed. London: Wiley, 1971.

17. Lind, Richard Oscar. Data Structure, Software and Biomedical Application of an Interactive Graphics System Using a Small Computer. Unpublished Ph.D. dissertation. Northwestern University, 1972.

18. Luzadder, William J., Basic Graphics. 2nd ed. Englewood Cliffs, New Jersey: Prentice Hall, 1968.

19. McAfee, John G., "Problems in Evaluating the Radiation Dose for Radionuclides Excreted by the Kidneys." Symposium on Medical Radionuclides: Radiation Dose and Effects. Edited by Roger J. Cloitier, C. Lowell Edwards, and Walter S. Snyder. Oak Ridge, Tenn.: US Atomic Energy Commission and US Public Health Dept. publication number CONF-691212, 1969. pp. 271-294.

20. Meyers, Cecil H., Handbook of Basic Graphs, A Modern Approach. Belmont, California: Dickenson Publishing Co., 1970.

21. Ogdin, Jerry L., "The Case Against BASIC." Datamation September 1, 1971, pp. 34-41.

22. Parslow, R. D., and Green, R. E., Computer Graphics in Medical Research and Hospital Administration, London: Plenum Press, 1971.

23. Parslow, R. D., and Green, R. Elliot. eds. Advanced Computer Graphics - Economics, Techniques, and Applications. London: Plenum Press, 1971.

24. Pittendrigh, C. S., "Circadian Rhythms and Circadian Organization of Living Systems." Cold Springs Harbor Symposium on Quantitative Biology, V. 25, 1960. pp. 159-184.

25. Ragan, Harvey A., McClanahan, B. J., and Hackett, Patricia A., "Einsteinium Toxicity and Metabolism in Miniature Swine", Pacific Northwest Laboratory Annual Report for 1972 to the USAEC Division of Biological and Environmental Research, Volume 1: Life Sciences, Part 1: Biological Sciences. Edited by Roy C. Thompson. Washington D.C.: US Atomic Energy Commission publication number BNWL-1750 PT1 UC 48. 1973.

26. Richardson, Leroy Charles. Specification Techniques for Interactive Computer Systems. Unpublished Ph.D. dissertation, Carnegie-Mellon University, 1972.

27. Rogers, Lee E., The Ecological Effects of the Western Harvester Ant (Pogonomyrmex occidentalis) in the Shortgrass Plains Ecosystem. Technical Report number 206-Grassland Biome. U.S. International Biological Program, December 1972.

28. Sammet, Jean E., Programming Languages: History and Fundamentals. New York: Prentice Hall, 1969.

29. Sharpe, William F. BASIC - An Introduction to Computer Programming Using the BASIC Language. . New York: The Free Press, 1967.

30. Shilov, G. E., How to Construct Graphs. Translated From Russian by Jerome Kristian and Daniel A. Levine. Boston: D. C. Heath, 1963.

31. Stotz, Robert H., "A New Display Terminal". Computer Design, April, 1968, pp. 80-86.

32. Tektronix, Inc. Plot-10 FORTRAN (TEKPLOT) Users Manual, Beaverton, Oregon: Tektronix Inc., Document number 062-1475-00, 1971.

33. Tektronix Inc., Tektronix 4010 and 4010-1 Users Manual. Beaverton, Oregon: Tektronix Inc., Document number 070-1225-00, 1071.

34. Tektronix Inc., Tektronix 4610/4610-1 Hard Copy Users Manual. Beaverton, Oregon: Tektronix Inc., Document number 070-1299-00, 1971.

35. Van Dam, A., and Michener, J. C. "Storage Tube Graphics, A Comparison of Terminals". Advanced Computer Graphics - Economics, Techniques, and Applications. Edited by R. D. Parslow and R. Elliot Green. London: Plenum Press, 1971, pp. 851-863.

36. Wallis, Peter J. L., "Univac's 1100 Series - A Basis for Continuing Enhancements". Data Processing Magazine, April, 1970, pp. 46-52.

37. Zaphiropoulos, Renn., "Nonimpact Printers". Datamation, May, 1973, pp. 71-76.

APPENDICES

APPENDIX A

This appendix contains a brief survey of the BASIC statements used in EZPLOT, a key to array names and their function in the program, a key to variable names and their function in the program, a detailed key to the array named P showing the effect of possible values stored in that array, and finally, it contains a listing of the program.

The program was not written as a work of art, but as a functional research tool. Do not be distracted by poor programming practices hastily appended to existing logic. The program listing reflects one stage of a graphics system which is still being refined.

A brief survey of BASIC statements used in EZPLOT:

Statement	Fortran Equivalent	Description
' (apostrophe)	None	Remainder of instruction line is remark
DIM	DIMENSION	Declares maximum bounds of arrays and lengths of string variables
END	END	Terminates program execution
FILES	None	Closes or opens files
FOR	DO	Initiates DO loop, must be terminated by a NEXT statement
GOSUB	CALL	Transfers program control to a subroutine
GOTO	GO TO	Transfers program control to a statement other than the next sequential one
IF	IF ( ) GO TO	Tests logical expression and transfers a TRUE condition
: (image)	FORMAT	Provides a format for a single line, used with PRINT USING or OUTPUT USING
INPUT	READ	Causes "?" to appear at terminal then accepts data
NEXT	None (Continue)	Terminator of DO loop
ON (N) GO TO ,,,	GO TO (,,,) N	Transfers control according to integer value of an expression
OUTPUT	None	Prints line at terminal
OUTPUT USING	WRITE (6,101) Text	Prints formated lines

Statement	Fortran Equivalent	Description
PRINT		Same as OUTPUT
PRINT USING		Same as OUTPUT USING
REM	C	Adds remark to program listing
RETURN	RETURN	Returns program control from a subroutine
<	.LT.	Less than
=	.EQ.	Equal to
>	.GT.	Greater than
<>	.NE.	Not equal
<=	.LE.	Less than or equal to
>=	.GE.	Greater than or equal to

Since array and variable names are limited to a single letter or a letter and a number by BASIC, the following key to variable names is necessary for understanding of the program listing.

ARRAY	DIMENSION	FUNCTION IN EZPLOT
A	200	Storage of x coordinate of data points after possible internal transformation by EZPLOT
B	200	Storage of y coordinates of transformed data.
C	200	0 = Ith point may be plotted on log axis.
P	20	Storage of user responses to menu-type questions.
Q	20	Storage of internal switches and some user responses to non-menu type questions.
R	100*	Used by TEKPLOT subroutines (line numbers 11000 through 45160)

ARRAY	DIMENSION	FUNCTION IN EZPLOT
S	100*	Used by TEKPLOT subroutines.
T	100*	Used by TEKPLOT subroutines.
X	200	Storage of x-coordinate input data.
Y	200	Storage of y-coordinate input data.
Z$	200	String array to store plotting symbols.
B$	55	String array to store 55 character bottom line.
M$	55	String array to store 55 character line displayed at the top of the completed graph.

*These arrays could be greatly reduced if necessary.

VARIABLE RELATED TO X-AXIS	VARIABLE RELATED TO Y-AXIS	FUNCTION IN EZPLOT
F	F	F = number of x-p pairs to be plotted.
A1	B1	Minimum axis value in user's frame of reference.
A2	B2	Maximum axis value in user's frame of reference.
A3	B3	Range of axis in user's terms (ie. A3 = A2-A1)
A4	B4	Scale factor (nr. of raster units/user's unit)
A5	B5	Screen location of minimum axis value.
A6	B6	Number of units between tic marks.
A8	B8	Margin factor (nr. of user's units viewable beyond the defined plot area, set at 1/75 of the x-axis range and 1/50 of the y-axis range)
A9	B9	Number of tic marks on axis.

VARIABLE RELATED TO X-AXIS	VARIABLE RELATED TO Y-AXIS	FUNCTION IN EZPLOT
Cl	Dl	Minimum axis value as possibly transformed (if axis is linear, Cl = Al; if log, Cl = log Al).
C2	D2	Maximum axis values as possibly transformed.
C3	D3	Range of axis as possibly transformed.
C6	C8	Minimum data value.
C7	C9	Maximum data value.

USES OF THE P ARRAY IN EZPLOT

VARIABLE	CONTENTS	MEANING
P(1)		INPUT TYPE?
	1	File in users library
	2	Keyboard data entry
	3	File and quick linear plot requested
	4	No data, functions only
P(2)		CORRECT FILE?
	1	Yes
	2	No
P(3)		LIST AND?OR EDIT DATA?
	1	Yes
	2	No
P(4)		TYPE OF GRAPH?
	1	Linear
	2	Log Y vs Linear X
	3	Linear Y vs Log X
	4	Log Y vs Log X

VARIABLE	CONTENTS	MEANING
P(5)		AUTO SCALE?
	1	Yes
	2	No
P(6)		GRID?
	1	Yes, grid line at every tic mark
	2	No grid at all
	3	User will supply grid locations
P(7)		TOP TITLE OK?
	1	Yes
	2	No
P(8)		BOTTOM LINE OK?
	1	Yes
	2	No
P(9)		WHAT NEXT?
	1	Plot data
	2	Change type of graph
	3	Change Scale
	4	Edit data
	5	Add functions
	6	Get new set of data
	7	Quit
P(10)		X-AXIS OK?
	1	Yes
	2	No
P(11)		Y-AXIS OK?
	1	Yes
	2	No

VARIABLE	CONTENTS	MEANING
P(12)		EDIT OPTION?
	1	Display data
	2	Change 1 or more items
	3	Insert items
	4	Switch a pair of items
	5	Delete an item
	6	Change Plot symbol
	7	Data ok
P(15)		INPUT EXPERT?
	1	Yes
	2	No

```
*EZPLT     10:30     09/02/73

100 DIM R(100),T(100),S(100)
110 DIM Z(200)$(1),X(200),Y(200)
120 DIM A(200),B(200),C(200)
130 DIM P(20)
140 DIM M$(55) ,B$(55)
150 DIM Q(20)
160 L1 = 0.43429448
170 A$ = "CRWTMP"
180 GØSUB 11000       'CALL BEGIN
190 GØSUB 12000       'ERASE
200 PRINT "EZPLØT VERSIØN 2  5/3/73"
210 PRINT "FØR DETAILS ABØUT ANY QUESTIØN ANSWER 0, ØTHERWISE"
220 PRINT "ANSWER ALL QUESTIØNS WITH SINGLE DIGIT 1-9"
225 Q(2) = D9 = 0
230 PRINT "INPUT TYPE";
240 INPUT P(1)
250 ØN P(1) GØTØ 330,7000,320,310
270 PRINT "     (1) FILE IN USERS LIBRARY"
280 PRINT "     (2) WANT TØ ENTER DATA FRØM KEYBØARD"
290 PRINT "     (3) FILE AND WANT QUICK LØØK AT LINEAR GRAPH"
295 PRINT "     (4) NØ DATA, WANT TØ DRAW FUNCTIØNS ØNLY"
300 GØTØ 230
310 F = 1
311 X(1) = Y(1) = 1
312 X(2) = Y(2) = 999
313 GØTØ 800
320 P(4) = P(5) = 1
330 PRINT "WHAT NAME DID YØU GIVE IT";
340 INPUT A$
350 FILES A$
360 GET #1,M$
370 PRINT M$
380 GØTØ 430
390 PRINT "IS THIS:"
400 PRINT "    (1)  THE PRØPER FILE"
410 PRINT "ØR"
420 PRINT "    (2)  SHALL WE TRY ANØTHER FILE NAME"
430 PRINT "CØRRECT FILE";
440 INPUT P(2)
450 ØN P(2) GØTØ 460,230
455 GØTØ 390
460 IF P(1) = 3 GØTØ 470
461 PRINT " LIST AND/ØR EDIT DATA";
462 INPUT P(3)
463 ØN P(3) GØTØ 470,470
464 PRINT "1 = YES, 2 = NØ"
465 GØTØ 461
470 C6 = C8 = +999999
480 C7 = C9 = -999999
490 F = 0
500 FØR I = 1 TØ 200
510 IF Q(2) = 1 GØTØ 530
515 IF P(1) = 2 GØTØ 530
520 INPUT #1,X(I),Y(I),Z(I)$
530 IF X(I) = 999 GØTØ 641
540 IF X(I) = 888 GØTØ 630
550 IF X(I) >C6 GØTØ 570
```

```
560 C6 = C1 = X(I)
570 IF X(I) <C7 GØTØ 590
580 C7 = C2 = X(I)
590 IF Y(I) > C8 GØTØ 610
600 C8 = D1 = Y(I)
610 IF Y(I) < C9 GØTØ 630
620 C9 = D2 = Y(I)
630 F = F + 1
635 C(I) = 0
640 NEXT I
641 IF P(3) <>1 GØTØ 650
642 P(1) = P(3) = 2
643 GØTØ 7990
650 IF Q(2) <>1 GØTØ 670
660 ØN P(9) GØTØ 3120,800,1270
670 IF P(1) = 3 GØTØ 820
680 PRINT USING "NØW HAVE  ### ITEMS ",F
690 PRINT "  MIN X          MAX X            MIN Y          MAX Y"
700 PRINT C6,C7,C8,C9
710 FILES
800 PRINT " TYPE ØF GRAPH";
805 Q(17) = Q(18) = 0
810 INPUT P(4)
820 ØN P(4) GØTØ 1060,1060,920,920
830 PRINT "1 = LINEAR"
840 PRINT "2 = LØG Y VS LINEAR X"
850 PRINT "3 = LINEAR Y VS LØG X"
860 PRINT "4 = LØG Y VS LØG X"
870 GØTØ 800
920 REM  LØG ØF X VALUES
930 FØR I = 1 TØ F
940 IF X(I) <> 888 GØTØ 970
950 A(I) = X(I)
960 GØTØ 1040
970 IF X(I) = 999 GØTØ 1040
980 IF X(I) >0.0 GØTØ 1030
990 PRINT "WILL SKIP:";
1000 PRINT X(I),Y(I)
1010 C(I) = 1
1020 GØTØ 1040
1030 A(I) = L1 * LØG (X(I))
1040 NEXT I
1050 GØTØ 1090
1060 FØR I = 1 TØ F
1070 A(I) = X(I)
1080 NEXT I
1090 ØN P(4) GØTØ 1240,1100,1240,1100
1100 REM LØG ØF Y VALUES
1110 FØR I = 1 TØ F
1120 IF Y(I) <> 888 GØTØ 1150
1130 B(I) = Y(I)
1140 GØTØ 1220
1150 IF Y(I) = 999 GØTØ 1220
1160 IF Y(I) > 0 GØTØ 1210
1170 C(I) = 1
1180 PRINT "WILL SKIP:  ";
1190 PRINT X(I),Y(I)
1200 GØTØ 1220
1210 B(I) = L1 * LØG (Y(I))
1220 NEXT I
1230 GØTØ 1270
1240 FØR I = 1 TØ F
1250 B(I) = Y(I)
1260 NEXT I
1265 IF P(1) = 3 GØTØ 1420
1269 REM ØN NEG LØG AXIS. FØRCE USER INTØ MANUAL SCALE
```

```
1270 ØN P(4) GØTØ 1279,1271,1273,1271
1271 IF C8<0 GØTØ 1275
1272 IF P(4) <>4 GØTØ 1279
1273 IF C6<0 GØTØ 1275
1274 GØTØ 1279
1275 P(5) = 2
1276 GØTØ 1750
1279 PRINT "AUTØ SCALE";
1280 INPUT P(5)
1290 ØN P(5) GØTØ 1420,1750
1300 PRINT "1 = AUTØMATIC SCALE TØ FIT DATA"
1310 PRINT "2 = USER WILL SUPPLY SCALE FACTØRS"
1320 GØTØ 1270
1420 REM AUTØ SCALE
1422 A1 = C6
1424 A2 = C7
1426 B1 = C8
1428 B2 = C9
1470 ØN P(4) GØTØ 1500,1500,1480,1480
1480 C1 = L1 * LØG(A1)
1490 C2 = L1 * LØG(A2)
1500 C3 = C2 - C1
1505 A3 = A2 - A1
1510 A6 = A3 /10
1560 ØN P(4) GØTØ 1590,1570,1590,1570
1570 D1 = L1 * LØG(B1)
1580 D2 = L1 * LØG(B2)
1590 D3 = D2 -D1
1595 B3 = B2 - B1
1600 B6 = B3/10
1610 Q(7) = Q(8) = 2
1620 A9 = B9 = 10
1630 REM ALL LINEAR SCALES CØME HERE
1640 A4 = 900/C3
1650 A5 = 100 -(C1*A4)
1660 B4 = 600/D3
1670 B5 = 100 -(D1*B4)
1680 A8 = C3/75
1690 B8 = D3/50
1700 GØTØ 2610
1710 REM MANUAL SCALE FACTØRS
1750 PRINT "YØU WILL NEXT DEFINE THE SIZE GRAPH YØU WANT"
1760 PRINT "0,0 CAN BE ØFF THE SCREEN"
1850 PRINT "X-AXIS:"
1852 PRINT "       MIN X   MAX X"
1854 PRINT "DATA       ";
1856 PRINT C6,C7
1857 PRINT "AXIS       ";
1858 PRINT A1,A2
1880 PRINT USING 1890;
1890 :MINIMUM VALUE-
1900 INPUT A1
1910 PRINT USING 1920;
1920 :MAXIMUM VALUE -
1930 INPUT A2
1940 IF A2<A1 GØTØ 1850
1942 ØN P(4) GØTØ 1950,1950,1944,1944
1944 IF A1 > 0 GØTØ 1950
1946 PRINT "CANNØT HAVE NEGATIVE VALUES ØN LØG SCALE"
1948 GØTØ 1850
1950 A3 = C3 = A2 - A1
1960 C1 = A1
1970 C2 = A2
1980 PRINT " X-AXIS GØES FRØM ";
1990 PRINT A1;
2000 PRINT "TØ ":
```

```
2010 PRINT A2
2011 ØN P(4) GØTØ 2029,2029,2012,2012
2012 A6 = A3
2013 A9 = 1
2014 PRINT "IN ADDITIØN TØ THESE 2 ENDPØINTS"
2015 PRINT "HØW MANY ØTHER TIC MARKS ARE WANTED (MAX 10)";
2016 INPUT Q(17)
2017 FØR I = 1 TØ Q(17)
2018 PRINT "X VALUE FØR ";
2019 PRINT I;
2020 INPUT L(I)
2021 NEXT I
2022 Q(7) = 1
2023 GØTØ 2091
2029 PRINT "WITHIN THIS RANGE ØF ";
2030 PRINT A3;
2040 PRINT "UNITS"
2050 PRINT "HØW MANY UNITS DØ YØU WANT BETWEEN TIC MARKS";
2060 INPUT A6
2070 B9 = A3/A6
2080 Q(3) = 1
2090 GØTØ 2380
2091 PRINT "X AXIS ØK";
2092 INPUT P(10)
2093 ØN P(10) GØTØ 2100,1850
2094 PRINT "1 = YES, 2= NØ"
2095 GØTØ 2091
2100 Q(3) = 0
2105 ØN P(4) GØTØ 2110,2110,2140,2140
2110 Q(7) = Q(9)
2120 A9 = B9
2130 GØTØ 2170
2140 C1 = L1 * LØG(A1)
2150 C2 = L1 * LØG (A2)
2160 C3 = C2 - C1
2170 GØSUB 12000
2172 PRINT " Y AXIS"
2174 PRINT "DATA      ";
2176 PRINT C8,C9
2177 PRINT "AXIS      ";
2178 PRINT B1,B2
2200 PRINT USING 1890;
2210 INPUT B1
2220 PRINT USING 1920;
2230 INPUT B2
2240 IF B2<B1 GØTØ 2170
2242 ØN P(4) GØTØ 2250,2244,2250,2244
2244 IF B1 > 0 GØTØ 2250
2246 PRINT "CANNØT HAVE NEGATIVE VALUES ØN LØG SCALE"
2248 GØTØ 2170
2250 B3 = D3 = B2 -B1
2260 D1 = B1
2270 D2 = B2
2280 PRINT "Y-AXIS GØES FRØM ";
2290 PRINT B1;
2300 PRINT " TØ ";
2310 PRINT B2
2311 ØN P(4) GØ TØ 2329,2312,2329,2312
2312 B6 = B3
2313 B9 = 1
2314 PRINT "IN ADDITIØN TØ THESE 2 ENDPØINTS, "
2315 PRINT "HØW MANY ØTHER TIC MARKS ARE WANTED (MAX 10)";
2316 INPUT Q(18)
2317 FØR I = 1 TØ Q(18)
2318 PRINT " Y VALUE FØR TIC ";
2319 PRINT I:
```

```
2320 INPUT M(I)
2321 NEXT I
2322 Q(8) = 1
2324 GØTØ 2520
2329 PRINT "WITHIN THIS RANGE ØF ";
2330 PRINT B3;
2340 PRINT "UNITS"
2350 PRINT "HØW MANY UNITS DØ YØU WANT BETWEEN TIC MARKS";
2360 INPUT B6
2370 B9 = B3/B6
2380 PRINT USING "THAT WILL MEAN #### TIC MARKS ØN THE AXIS",B9
2390 IF B9 <1 GØTØ 2530
2400 IF B9 >51 GØTØ 2530
2410 GØTØ 2460
2420 PRINT " 1 = EVERY TIC WILL BE LABELED"
2430 PRINT " 2 = EVERY ØTHER TIC WILL BE LABELED"
2440 PRINT " 3 = EVERY THIRD TIC LABELED"
2450 PRINT " ETC."
2460 PRINT "LABEL FACTØR";
2470 INPUT Q(9)
2480 IF Q(9) <1 GØTØ 2420
2490 IF Q(9) > B9 GØTØ 2420
2500 IF Q(3)=1 GØTØ 2091
2510 Q(8) = Q(9)
2520 PRINT "Y AXIS ØK";
2521 INPUT P(11)
2522 ØN P(11) GØTØ 2560,2170
2523 PRINT "1 = YES, 2 = NØ"
2524 GØTØ 2520
2530 PRINT "TØØ FEW ØR TØØ MANY"
2540 IF Q(3) = 1 GØTØ 2050
2550 GØTØ 2350
2560 ØN P(4) GØTØ 1640,2570,1640,2570
2570 D1 = L1 * LØG(B1)
2580 D2 = L1 * LØG(B2)
2590 D3 = D2 - D1
2600 GØTØ 1640
2610 REM
2620 IF P(1) = 3 GØTØ 2970
2621 PRINT "GRID";
2622 INPUT P(6)
2623 ØN P(6) GØTØ 2630,2630,2790
2624 PRINT "1 = AUTØMATIC GRID AT EACH TIC"
2625 PRINT "2 = NØ GRID AT ALL"
2626 PRINT "3 = USER WILL SUPPLY GRID LØCATIØNS"
2627 GØTØ 2621
2630 PRINT "TØP TITLE ØK";
2640 INPUT P(7)
2650 ØN P(7) GØTØ 2760,2740
2660 PRINT "YØU MAY:"
2670 PRINT "  (1) HAVE:";
2680 PRINT M$;
2690 PRINT "  DISPLAYED AT TØP ØF GRAPH"
2700 PRINT "ØR"
2710 PRINT "  (2) ENTER A NEW TITLE FØR THE TØP ØF THE GRAPH"
2720 GØTØ 2630
2740 PRINT "ENTER TITLE FØR THIS DATA (MAX 55 CHARACTERS)"
2750 GET M$
2755 GØTØ 2630
2760 IF P(6) = 1 GØTØ 2950
2770 Q(6) = 0
2780 GØTØ 2960
2790 PRINT "HØW MANY  VERTICAL GRID LINES (0=NØNE,MAX IS 10)";
2800 INPUT Q(4)
2810 IF Q(4)<1 GØTØ 2860
2815 IF Q(4) > 10 GØTØ 2860
```

```
2820 FØR I = 1 TØ Q(4)
2830 PRINT USING "WHAT X VALUE FØR GRID LINE ##",I;
2840 INPUT K(I)
2850 NEXT I
2860 PRINT "HØW MANY  HØRIZØNTAL GRID LINES (0=NØNE, MAX IS 10)";
2870 INPUT Q(5)
2880 IF Q(5) <1 GØTØ 2960
2885 IF Q(5) > 10 GØTØ 2960
2890 FØR I = 1 TØ Q(5)
2900 PRINT USING "WHAT Y VALUE FØR GRID LINE ##",I;
2910 INPUT J(I)
2920 NEXT I
2930 GØTØ 2630
2940 REM----SET UP AUTØ GRID
2950 Q(6) = 1
2960 IF Q(2) = 1 GØTØ 2990
2970 PRINT "ENTER DESCRIPTIVE LINE FØR BØTTØM ØF GRAPH"
2980 GET B$
2990 PRINT "BØTTØM LINE ØK";
2992 INPUT P(8)
2994 ØN P(8) GØTØ 3090,2970
3000 PRINT "ARE YØU:"
3010 PRINT "(1) SATISFIED WITH: ";
3020 PRINT B$
3030 PRINT "    ØR DØ YØU"
3040 PRINT "(2) WANT A DIFFERENT BØTTØM LINE"
3050 GØTØ 2990
3090 Q(1) = 1
3100 IF Q(2) =1 GØTØ 4180
3110 IF P(1) = 4 GØTØ 5160
3120 Q(2) = 1
3145 REM------------PLØT GRAPH HERE-----------------------
3150 GØSUB 12000    'PAGE
3160 GØSUB 4350       'USERS SCALE
3170 GØSUB 15000      'VECTØR MØDE
3180 GØSUB 5340     'DRAW AXIS
3181 IF Q(17) = 0 GØTØ 3184
3182 GØSUB 9700    'LØG X TIC MARKS IF ANY
3184 IF Q(18) = 0 GØTØ 3190
3186 GØSUB 10000   'LØG Y TIC MARKS IF ANY
3190 IF Q(4) < 1 GØTØ 3210
3200 GØSUB 4470    'VERTICAL GRID
3210 IF Q(5)<1 GØTØ 3230
3220 GØSUB 4630    'HØRIZØNTAL GRID
3225 IF P(1) = 4 GØTØ 3870
3230 FØR I = 1 TØ F
3240 IF Z(I)$ = " " GØTØ 3860
3250 IF C(I) = 1 GØTØ 3860
3260 J = 0
3270 IF Z(I)$ = "-" GØTØ 3330
3280 J = 0
3290 IF Z(I)$ = "D" GØTØ 3340
3300 IF Z(I)$ = "." GØTØ 3340
3310 K = 2
3320 GØTØ 3350
3330 J = 1
3340 K = 1
3350 IF A(I) <>888 GØTØ 3420
3360 IF B(I) <> 888 GØTØ 3420
3370 X1  = A(I+1)
3380 X2 =  B(I+1)
3390 X3 = 0
3400 GØSUB 9000
3410 GØTØ 3860
3420 X1  = A(I)
3430 X2 = B(I)
```

```
3440 IF I >1 GØTØ 3470
3450 X3 = 0
3460 GØ TØ 3490
3470 X3 = J
3490 GØSUB 9000
3500 IF K<>2 GØTØ 3580
3510 REM ADJUST KEYBØARD SYMBØLS TØ CENTER PRØPERLY
3520 X1 = A(I) -3/A4
3530 X2 = B(I) -4/B4
3540 X3 = 0
3550 GØSUB 9000   'DARK TØ DATA PØSITIØN
3560 PRINT Z(I)$
3570 GØTØ 3860
3580 IF Z(I)$ = "D" GØTØ 3650
3590 X1 = A(I)
3600 X2 = B(I)
3610 X3 = 1
3630 GØSUB 9000
3640 GØTØ 3860
3650 REM DRAW DIAMØND CENTERED ØN ACTUAL DATA PØSITIØN
3660 X1 = A(I) + 4/A4
3670 X2 = B(I)
3680 X3 =  0
3690 GØSUB 9000
3700 X1 = A(I)
3710 X2 = B(I) -6/B4
3720 X3 = 1
3730 GØSUB 9000
3740 X1 = A(I) - 4/A4
3750 X2 = B(I)
3760 X3 = 1
3770 GØSUB 9000
3780 X1 = A(I)
3790 X2 = B(I) + 6/B4
3800 X3 = 1
3810 GØSUB 9000
3820 X1 = A(I) +4/A4
3830 X2 = B(I)
3840 X3 = 1
3850 GØSUB 9000
3860 NEXT I
3870 IF D9 <> 9 GØTØ 3890
3880 GØSUB 4810
3890 FØR I = 1 TØ 5
3900 X1 = 7
3910 GØSUB 45000   'RING BELL
3920 X1 = 0
3930 GØSUB 45000
3940 NEXT I
3950 GØSUB 4240   'ABSØLUTE SCALE
3960 X1 = 200
3970 X2 = 780
3980 X3 =  0
3990 GØSUB 9000
3992 X1 = 200
3993 X2 = 780
3994 X3 = 0
4010 X1 = 200
4020 X2 = X3 =  0
4030 GØSUB 9000
4040 ØUTPUT B$
4041 X1 = 200
4042 X2 = 760
4043 X3 = 0
4044 GØSUB 9000
4045 ØUTPUT M$
```

```
4050 X1 = X3 =  0
4060 X1 = X2 = X3 = 0
4070 GØSUB 9000    'GØTØ UPPER LEFT
4071 X1 = X3 = 0
4072 X2 = 780
4073 GØSUB 9000
4080 ØUTPUT "?";
4090 GET C$
4100 GØSUB 12000      'PAGE
4105 P(1) = 0
4110 GØTØ 4180
4120 PRINT "  1 = PLØT DATA AS IT IS"
4130 PRINT "  2 = CHANGE TYPE ØF GRAPH"
4140 PRINT "  3 = CHANGE SCALE"
4150 PRINT "  4 = EDIT DATA"
4160 PRINT "  5 = ADD FUNCTIØNS"
4165 PRINT "  6 = GET NEW DATA FILE"
4170 PRINT "  7 = QUIT"
4180 PRINT "WHAT NEXT";
4190 INPUT P(9)
4210 ØN P(9) GØTØ 3120,470,470,7990,5160,225,4220,4120
4211 GØTØ 4120
4220 END
4230 REM ......................
4240 REM ..........  SUBRØUTINE TØ SET SCALE TØ ABSØLUTE
4250 X1 = X2 = 1
4260 X3 =  0
4270 GØSUB 21000
4280 X1 = 0
4290 X2 = 0
4300 X3 = 1020
4310 X4 = 780
4320 GØSUB 23000
4330 RETURN
4340 REM .....................
4350 REM .........  SUBRØUTINE TØ SET SCALE TØ USERS CHØICE
4360 X1 = A4
4370 X2 = B4
4380 X3 = A5
4390 X4 = B5
4400 GØSUB 21000
4410 X1 = C1 - 20*A8
4420 X2 = D1 - 20*B8
4430 X3 = C2 + 10*A8
4440 X4 = D2 + 10*B8
4450 GØSUB 23000
4460 RETURN
4470 REM ----- SUBRØUTINE FØR VERTICAL GRID LINES
4480 FØR I = 1 TØ Q(4)
4490 ØN P(4) GØTØ 4520,4520,4500,4500
4500 X1 = G = L1 * LØG(K(I))
4510 GØTØ 4530
4520 X1 = G = K(I)
4530 X2 = B1
4540 X3 =  0
4550 GØSUB 9000
4560 X1 = G
4570 X2 = B2
4580 X3 = 1
4600 GØSUB 9000
4610 NEXT I
4620 RETURN
4630 REM ----- SUBRØUTINE FØR HØRIZØNTAL GRID LINES
4640 FØR I = 1 TØ Q(5)
4650 X1 = A1
4660 ØN P(4) GØTØ 4690,4670,4690,4670
```

```
4670 X2 = G = L1 * LØG(J(I))
4680 GØTØ 4700
4690 X2 = G = J(I)
4700 X3 =  0
4710 GØSUB 9000
4720 X1 = A2
4730 X2 = G
4740 X3 = 1
4760 GØSUB 9000
4770 NEXT I
4780 RETURN
4790 REM ......................
4800 REM ......................
4810 REM SUBRØUTINE FØR USER FUNCTIØNS (MAX 5)
4820 D8 = Y
4830 D7 = X
4840 FØR J = 1 TØ D6
4850 X1 = C1
4860 X2 = D1
4870 X3 =  0
4880 GØSUB 9000    'DARK TØ ØRIGIN
4890 FØR I = A1 TØ A2 STEP A3/100
4900  X = I
4910 IF I<>0 GØTØ 4930
4920 X = 0.0000001
4930 ØN J GØTØ 4950,4970,4990,5010,5030
4940 REM USER XTATEMENTS NEXT
4950 Y = X
4960 GØTØ 5040
4970 Y = X - .5
4980 GØTØ 5040
4990 Y = D1
5000 GØTØ 5040
5010 Y = D1
5020 GØTØ 5040
5030 Y = D1
5040 ØN P(4) GØTØ 5045,5045,5050,5050
5045 X1 = X
5047 GØTØ 5060
5050 X1 = L1 * LØG(X)
5060 ØN P(4) GØTØ 5065,5070,5065,5070
5065 X2 = Y
5067 GØTØ 5080
5070 X2 = L1 * LØG(Y)
5080 X3 = 1
5081 IF I <>C1 GØTØ 5100
5082 X3 = 0
5100 GØSUB 9000
5110 NEXT I
5120 NEXT J
5130 Y = D8
5140 X = D7
5150 RETURN
5160 PRINT "YØU MUST ENTER YØUR FUNCTIØN AS A BASIC STATEMENT"
5170 D9 = 9
5180 PRINT "ENTER NUMBER ØF FUNCTIØNS YØU WISH TØ DISPLAY (LIMIT 5)";
5190 INPUT D6
5200 ØN D6 GØTØ 5220,5220,5220,5220,5220
5210 GØTØ 5180
5220 PRINT "INSERT FUNCTIØNS AT 4950,4970,49690,5010,5030"
5250 PRINT "EXAMPLE: 4950 Y = SIN(Y)"
5260 PRINT "TØ SEE THE RESULTS, TYPE ""GØ TØ 9600"""
5270 PRINT "HIT CØNTRØL-SHIFT-K NØW!!"
5280 GET A$
5290 GØTØ 4180
5300 GØTØ 2960
```

```
5340 REM -----AXIS SUBRØUTINE
5345 GØSUB 15000
5350 REM HØRIZØNTAL TIC MARKS  TØP AND BØTTØM
5360 Q(9) = 0
5370 FØR I = 0 TØ A9
5380  G = G1 = A1 + A6*I
5390 ØN P(4) GØTØ 5410,5410,5400,5400
5400 G = L1 * LØG (G)
5410 IF Q(9)<>0 GØTØ 5468
5420 X1 = G - 3*A8
5430 X2 = D1 - 5*B8
5440 X3 = 0
5450 GØSUB 9000
5460 PRINT G1
5464 X2 = D1 - B8
5466 GØTØ 5470
5468 X2 = D1 - B8/2
5470 Q(9) = Q(9) + 1
5480 IF Q(9) <> Q(7) GØTØ 5500
5490 Q(9) = 0
5500 X1 = G
5520 X3 = 0
5530 GØSUB 9000
5540 X1 = G
5550 X2 = D1
5560 X3 = 1
5570 GØSUB 9000
5580 X1 = G
5590 X2 = D2
5600 IF I = 0 GØTØ 5640
5610 IF I = A9 GØTØ 5640
5620 X3 = Q(6)
5630 GØTØ 5650
5640 X3 = 1
5650 GØSUB 9000
5660 X1 = G
5670 X2 = D2 + B8/2
5680 X3 = 1
5690 GØSUB 9000
5700 NEXT I
5710 REM Y AXIS TIC MARKS AND ØPTIØNAL GRID
5715 GØSUB 15000
5720 Q(9) = 0
5730 FØR I = 0 TØ B9
5740 G = G1 = B1 + B6*I
5770 ØN P(4) GØTØ 5790,5780,5790,5780
5780 G = L1 * LØG(G)
5790 IF Q(9) <> 0 GØTØ 5848
5800 X1 = C1 - 10*A8
5810 X2 = G -B8/2
5820 X3 = 0
5830 GØSUB 9000
5840 PRINT G1
5844 X1 = C1 - A8
5846 GØTØ 5850
5848 X1 = C1 - A8/2
5850 Q(9) = Q(9) + 1
5860 IF Q(9) <> Q(8) GØTØ 5890
5870 Q(9) = 0
5890 X2 = G
5900 X3 = 0
5910 GØSUB 9000
5920 X1 = C1
5930 X2 = G
5940 X3 = 1
5950 GØSUB 9000
```

```
5960 X1 = C2
5970 X2 = G
5980 IF I = 0 GØTØ 6020
5990 IF I = B9 GØTØ 6020
6000 X3 = Q(6)
6010 GØTØ 6030
6020 X3 = 1
6030 GØSUB 9000
6040 X1 = C2 + A8/2
6050 X2 = G
6060 X3 = 1
6070 GØSUB 9000
6080 NEXT I
6090 RETURN
7000 REM ------ SUBRØUTINE TØ INPUT DATA FRØM KEYBØARD
7010 K = 1
7020 GØTØ 7120
7040 PRINT "IF YØU CHØØSE TØ CØNNECT YØUR DØTS WITH A LINE,"
7050 PRINT "REMEMBER THAT THEY WILL BE CØNNECTED IN THE ØRDER"
7060 PRINT "IN WHICH YØU ENTER THEM. ENTER 1 IF YØU WANT THEM CØNNECTED"
     ;
7070 PRINT " 2 IF NØT";
7080 INPUT K1
7090 IF K1 = 2 GØTØ 7110
7100 H$ = "-"
7110 GØTØ 7300
7120 PRINT
7121 PRINT "ENTER DESCRIPTIVE LINE FØR THIS DATA"
7122 GET M$
7125 PRINT "INPUT EXPERT";
7126 INPUT P(15)
7127 IF P(15) =1 GØTØ 7250
7130 PRINT "DATA WILL BE ENTERED AS X,Y PAIRS"
7140 PRINT "EACH SET ØF DATA PØINTS MUST BE PRECEEDED BY A PLØT SYMBØL"
7150 PRINT "TØ HAVE DATA PØINTS CØNNECTED ENTER KEYBØARD PERIØD AS PLØT
     SYMBØL"
7160 PRINT "THE KEYBØARD ""D"" WILL RESULT IN A SMALL DIAMØND "
7180 PRINT "ØTHER KEYBØARD CHARACTERS WILL BE CENTERED AT THE DATA"
7190 PRINT "LØCATIØN, BUT MAY PRINT ØFFSET BECAUSE ØF THEIR SHAPE"
7200 PRINT "I SUGGEST YØU USE SYMETRICAL KEYBØARD CHARACTERS"
7210 PRINT "SUCH AS:  Ø,*,O,@,X,+,#,  AND Q."
7220 PRINT "MAXIMUM NUMBER ØF PØINTS IS 200"
7230 PRINT "WHEN LAST ITEM IS ENTERED, ENTER 999,999 TØ SIGNAL END ØF DA
     TA"
7240 PRINT "WHEN YØU WISH TØ CHANGE PLØT SYMBØL ENTER 888,888"
7250 PRINT USING "ENTER ØNE KEYBØARD CHARACTER TØ BE USED WITH DATA SET
     ## ",K;
7260 INPUT H$
7270 IF H$ <> "-" GØTØ 7290
7280 H$ = "."
7290 IF H$ = "." GØTØ 7040
7300 PRINT
7310 PRINT USING "ENTER SEVERAL X,Y PAIRS FØR DATA SET ## ",K
7320 FØR I = 1 TØ 200
7330 MAT INPUT H(1,2)
7340
7350 X(I) = H(1,1)
7360 Y(I) = H(1,2)
7370 IF X(I) = 888 GØ TØ 7410
7380 Z(I)$ = H$
7390 IF X(I) = 999 GØ TØ 7560
7400 GØ TØ 7570
7410 K = K + 1
7420 Z(I)$ = "!"
7430 PRINT USING "ENTER NEW SYMBØL FØR DATA SET ## ",K;
```

```
7440 INPUT H$
7450 IF H$ = "-" GØTØ 7470
7460 GØTØ 7480
7470 H$ = "."
7480 IF H$ = "." GØTØ 7510
7490 PRINT USING "ENTER MØRE PAIRS FØR DATA SET ## ",K
7500 GØTØ 7570
7510 PRINT "DØ YØU WANT THESE DØTS CØNNECTED WITH A LINE (1) ØR NØT(2)";
7520 INPUT K1
7530 IF K1 = 2 GØTØ 7490
7540 H$ = "-"
7550 GØTØ 7490
7560 IF Y(I) = 999 GØ TØ 7590
7570 NEXT I
7580 PRINT "YØU HAVE EXCEEDED 200 DATA PØINTS, WILL PLØT WHAT YØU HAVE E
   NTERED"
7590 REM
7600 F =I
7610 G = 1
7640 REM ------RØUTINE TØ EDIT DATA
7650 GØTØ 7990
7660 GØSUB 12000    'PAGE
7670 PRINT "  YØUR DATA IS:"
7680 PRINT
7690 PRINT "ITEM NUMBER    X-VALUE          Y- VALUE      SYMBØL"
7700 G = 1
7710 FØR I = 1 TØ F
7720 IF X(I) = 888 GØ TØ 7740
7730 GØTØ 7800
7740 PRINT I;
7750 PRINT " DATA SET ";
7760 G = G + 1
7770 PRINT G;
7780 PRINT " FØLLØWS"
7790 GØTØ 7900
7800 PRINT I,X(I),Y(I),Z(I)$
7810 IF I = 32 GØTØ 7870
7820 IF I = 66 GØTØ 7870
7830 IF I = 100 GØTØ 7870
7840 IF I = 134 GØTØ 7870
7850 IF I = 168 GØTØ 7870
7860 GØTØ 7900
7870 PRINT "?";
7880 GET B$
7890 GØSUB 12000    'PAGE
7900 NEXT I
7910 GØTØ 7990
7920 PRINT "ØPTIØNS     REASØNS"
7930 PRINT "   1          DISPLAY DATA"
7940 PRINT "   2          CHANGE 1 ØR MØRE ITEMS"
7950 PRINT "   3          INSERT ITEMS"
7960 PRINT "   4          SWITCH A PAIR ØF ITEMS"
7970 PRINT "   5          DELETE AN ITEM"
7980 PRINT "   6          CHANGE PLØT SYMBØL FØR A SET"
7985 PRINT "   7          DATA ØK, PRØCEED"
7990 PRINT "EDIT ØPTIØN";
8000 INPUT P(12)
8020 ØN P(12) GØTØ 7660,8030,8340,8450,8740,8630,8025,7920
8025 IF P(9) <> 4 GØTØ 470
8026 PRINT "YØU MAY HAVE CHANGED MIN ØR MAX ØF DATA, SUGGEST YØU RESCALE -
"

8027 GØTØ 4180
8030 PRINT "ENTER NUMBER ØF ITEM TØ CHANGE";
8040 INPUT K1
8050 PRINT USING "NEW X FØR ITEM ###",K1;
8060 INPUT X(K1)
```

```
8070 PRINT USING "NEW Y FØR ITEM ###",K1;
8080 INPUT Y(K1)
8090 GØTØ 7990
8100 REM
8340 PRINT "WHAT ITEM # PRECEEDS THE ITEM YØU WISH TØ INSERT";
8350 INPUT K3
8360 FØR I = F TØ K3 + 1 STEP -1
8370 X(I+1 ) = X(I)
8380 Y(I+1) = Y(I)
8390 Z(I+1)$ = Z(I)$
8400 NEXT I
8410 F = F + 1
8420 PRINT "CØMPUTER HAS RENUMBERED YØUR ITEMS"
8425 IF P(12) = 5 GØTØ 7990
8430 K1 = K3 +1
8440 GØTØ 8050
8450 PRINT "WHAT ITEMS DØ YØU WISH SWITCHED"
8460 PRINT "FIRST ITEM";
8470 INPUT K4
8480 PRINT "SECØND ITEM";
8490 INPUT K5
8500 D = X(K4)
8510 X(K4) = X(K5)
8520 X(K5) = D
8530 E = Y(K4)
8540 Y(K4) = Y(K5)
8550 Y(K5) = E
8560 G$ = Z(K4)$
8570 Z(K4)$ = Z(K5)$
8580 Z(K5)$ = G$
8590 : ITEM ## IS NØW ITEM ##
8600 PRINT USING 8590,K4,K5
8610 PRINT USING 8590,K5,K4
8620 GØTØ 7990
8630 PRINT "SYMBØL CHANGE"
8640 PRINT "FIRST ITEM NUMBER";
8650 INPUT K7
8660 PRINT "LAST ITEM NUMBER ";
8670 INPUT K8
8680 PRINT "ENTER NEW SYMBØL";
8690 INPUT H$
8700 FØR I = K7 TØ K8
8710 Z(I)$ = H$
8720 NEXT I
8730 GØTØ 7990
8740 PRINT "WHAT ITEM # TØ DELETE";
8750 INPUT K6
8760 FØR I = K6 +1 TØ F
8770 X(I-1) = X(I)
8780 Y(I-1) = Y(I)
8790 Z(I-1)$ = Z(I)$
8800 NEXT I
8810 F = F -1
8820 GØTØ 8420
9000 REM  --------- MY ØWN TPLØT SUBRØUTINE  ----------
9010 X4 = 0
9020 T(58) = X1
9030 T(59) = X2
9040 T(60) = X3
9050 T(61) = X4
9060 S0 = T(61)
9070 R(2) = T(58) * R(6) + R(8)
9080 R(3) = T(59) * R(7)+R(9)
9090 S1 = R(1)
9100 IF T(60) = 0 GØTØ 9140
9110 X1 = 0
```

```
9120 GØSUB 44000
9130 GØTØ 9300
9140 GØSUB 15000
9150 S0=0
9300 RETURN
9600 GØTØ 4180
9700 REM X AXIS LØG TIS MARKS
9710 FØR I = 1 TØ Q(17)
9720 X0 = L1 * LØG(L(I))
9730 X1 = X0 - 3*A8
9740 X2 = D1 - 5*B8
9750 X3 = 0
9760 GØSUB 9000
9770 PRINT L(I)
9780 X1 = X0
9790 X2 = D1 - B8
9799 X3 = 0
9800 GØSUB 9000
9810 X1 = X0
9820 X2 = D1
9830 X3 = 1
9840 GØSUB 9000
9850 X1 = X0
9860 X2 = D2
9870 X3 = 0
9880 GØSUB 9000
9890 X1 = X0
9899 X2 = D2 + B8/2
9900 X3 = 1
9910 GØSUB 9000
9920 NEXT I
9940 RETURN
10000 REM----- Y AXIS TIC MARKS IF LØG --------
10010 FØR I = 1 TØ Q(18)
10020 X0 = L1 * LØG(M(I))
10030 X1 = C1 - 10*A8
10040 X2 = X0
10050 X3 = 0
10060 GØSUB 9000
10070 PRINT M(I)
10080 X1 = C1 - A8
10090 X2 = X0
10100 X3 = 0
10110 GØSUB 9000
10120 X1 = C1
10123 X2 = X0
10130 X3 = 1
10140 GØSUB 9000
10150 X1 = C2
10160 X2 = X0
10170 X3 = 0
10180 GØSUB 9000
10190 X1 = C2 + A8/2
10199 X2 = X0
10200 X3 = 1
10210 GØSUB 9000
10220 NEXT I
10230 RETURN
11000 REM ------- BEGIN -------------------------------------
11015 T(1)=300
11020 FØR T=1 TØ 22
11030 R(T)=0.
11040 NEXT T
11050 R(6)=1.
11060 R(7)=1.
11070 R(14)=1023.
```

```
11080 R(15)=761.
11090 R(17)=1024.
11100 R(18)=762.
11110 R(22)=1.
11120 GØSUB 15000
11140 REM  CØMPUTE CHARACTERS PER .5 SECØND
11150 REM  AT SPECIFIED BAUD RATE
11160 R(23)=T(1)/16.
11170 RETURN
11180 REM --------------------------------------------------------------
12000 REM          ERASE
12005 X1=27
12006 GØSUB45000
12007 X1=12
12008 GØSUB45000
12010 X1=24
12020 GØSUB 45000
12030 REM  ØUTPUT A CANCEL CØDE
12040 T=R(23)
12050 REM  FETCH CHARACTERS PER .5 SECØND
12060 FØR T1=1 TØ T+1
12070 X1=0
12080 GØSUB 45000
12090 NEXT T1
12100 REM  ØUTPUT .5 SECØNDS WØRTH ØF NULLS
12120 RETURN
12130 REM --------------------------------------------------------------
15000 REM          VECTØR
15010 X1=31
15020 GØSUB 45000
15030 REM  ØUTPUT A US FIRST AS INSURANCE.
15040 IF R(10)=0 GØTØ 15080
15050 X1=25
15060 GØSUB 45000
15070 REM  ØUTPUT AN EM
15080 R(1)=1
15090 REM  SET THE R ARRAY TØ VECTØR STATUS
15100 X1=1
15110 GØSUB 44000
15120 REM  DARK VECTØR TØ THE CURRENT PØSITIØN.
15130 RETURN
15140 REM --------------------------------------------------------------
21000 REM             SCALE
21010 S9=3
21020 T(3)=X1
21030 T(4)=X2
21040 T(5)=X3
21050 T(6)=X4
21060 R(6)=T(S9)
21070 R(7)=T(S9+1)
21080 R(8)=T(S9+2)
21090 R(9)=T(S9+3)
21100 RETURN
21110 REM --------------------------------------------------------------
23000 REM             WINDØW
23010 S=10
23020 T(10)=X1
23030 T(11)=X2
23040 T(12)=X3
23050 T(13)=X4
23060 FØR T=1 TØ 4
23070 R(T+30)=R(T+5)
23080 NEXT T
23090 REM  FETCH CURRENT SCALING FACTØRS AND ØRIGIN.
23100 R(35)=T(S)*R(31)
23110 R(36)=T(S+1)*R(32)
```

```
23120 R(37)=T(S+2)*R(31)
23130 R(38)=T(S+3)*R(32)
23140 REM  SCALE THE WINDØW CØØRDINATES.
23320 T1=1
23330 FØR T2=1 TØ 2
23340 FØR T3=3 TØ 4
23350 R(T1+34)=R(T1+34)+R(T3+30)
23360 T1=T1+1
23370 NEXT T3
23380 NEXT T2
23390 REM  ADD IN THE ØRIGIN
23400 REM  NØW TEST THAT MINIMUM VALUES ARE LESS THAN MAXIMUM VALUES.
23410 FØR T1=1 TØ 2
23420 T2=T1+2
23430 IF R(T1+34)<=R(T2+34) GØTØ 23480
23440 T3=R(T2+34)
23450 R(T2+34)=R(T1+34)
23460 R(T1+34)=T3
23470 REM  EXCHANGE MIN AND MAX.
23480 IF R(T1+34)>=0 GØTØ 23510
23490 R(T1+34)=0
23500 REM  MAKE SURE MIN VALUE >= 0
23510 NEXT T1
23520 IF R(37)<=1023.  GØTØ 23550
23530 R(37)=1023.
23540 REM  MAKE SURE XMAX <= 1023
23550 IF R(38)<=761. GØTØ 23590
23560 R(38)=761.
23570 REM  MAKE SURE YMAX <= 761
23580 REM  NØW RECØMPUTE THE ØLD ØFF-SCREEN FLAG
23590 T1=R(4)
23600 T2=R(5)
23610 REM  FETCH LAST (X,Y) AND RECØMPUTE THE FLAG.
23620 T3=0
23630 IF T1>=R(35)  GØTØ 23650
23640 T3=1
23650 IF R(37)>=T1  GØTØ 23670
23660 T3=2
23670 IF T2>=R(36)  GØTØ 23690
23680 T3=T3+4
23690 IF R(38)>=T2  GØTØ 23710
23700 T3=T3+8
23710 R(19)=T3
23720 REM  SAVE THE ØFF-SCREEN FLAG.
23730 FØR T1=1 TØ 4
23740 T2=11+T1
23750 R(T2)=R(T1+34)
23760 NEXT T1
23770 REM  STØRE WINDØW LIMITS BACK TØ R ARRAY.
23780 RETURN
23790 REM ---------------------------------------------------------
39000 REM            TEK003
39010 R9=67
39020 T(67)=X1
39030 T(68)=X2
39040 T(69)=X3
39050 IF T(R9+2) =0  GØTØ 39090
39060 R(51)=1000
39070 R(52)=1000
39080 IF T(R9+2)<0 GØTØ 39270
39090 R(53)=INT(T(R9)/32)
39100 R(54)=INT(T(R9+1)/32)
39110 R(55)=T(R9)-R(53)*32+64
39120 R(56)=T(R9+1)-R(54)*32+96
39130 IF R(54)=R(52)   GØTØ 39180
39140 R(52)=R(54)
```

```
39150 R(54)=R(54)+32
39160 X1=R(54)
39170 GØSUB 45000
39180 X1=R(56)
39190 GØSUB 45000
39200 IF R(53)=R(51)   GØTØ 39250
39210 R(51)=R(53)
39220 R(53)=R(53)+32
39230 X1=R(53)
39240 GØSUB 45000
39250 X1=R(55)
39260 GØSUB 45000
39270 RETURN
39280 REM -------------------------------------------------------
44000 REM              TEK002
44010 R=66
44020 T(66)=X1
44030 R1=R(2)
44040 R2=R(3)
44050 R3=R(16)
44100 R0=0
44110 R4=R(4)
44120 R5=R(5)
44130 REM  FETCH CURRENT WINDØW LIMITS
44140 FØR R6=1 TØ 4
44150 R(R6+30)=R(R6+11)
44160 NEXT R6
44170 IF R1>=R(31)  GØTØ 44200
44180 R0=1
44190 GØTØ 44220
44200 IF R(33)>=R1  GØTØ 44220
44210 R0=2
44220 IF R2>=R(32)  GØTØ 44250
44230 R0=R0+4
44240 GØ TØ 44270
44250 IF R(34)>=R2  GØTØ 44270
44260 R0=R0+8
44270 R7=INT(R(19))
44280 IF R0+R7=0   GØTØ 44860
44290 R(35)=INT(R0/4)
44300 R(36)=INT(R7/4)
44310 R(37)=R0-R(35)*4
44320 R(38)=R7-R(36)*4
44330 IF R0=0    GØTØ 44390
44340 IF R7=0    GØTØ 44390
44350 IF R(35)*R(36)=0  GØTØ 44370
44360 IF R(35)-R(36)=0  GØTØ 44950
44370 IF R(37)*R(38)=0  GØTØ 44390
44380 IF R(37)-R(38)=0  GØTØ 44950
44390 IF R(1)<=1        GØTØ 44420
44400 IF R0=0   GØTØ 44860
44410 GØTØ 44950
44420 R8=R1-R4
44430 IF R8=0   GØTØ 44450
44440 R(81)=(R2-R5)/R8
44450 IF R7=0   GØTØ 44680
44460 IF R(36)=0      GØTØ 44540
44470 R6=2*R(36)
44480 R3=R(R6+30)
44490 R6=R4
44500 IF R8=0   GØTØ 44520
44510 R6=R6+(R3-R5)/R(81)
44520 IF R(33)<R6  GØTØ 44540
44530 IF R6>=R(31)  GØTØ 44600
44540 IF R(38)<=0  GØTØ 44950
44550 R6=2*R(38)-1
```

```
44560 R6=R(R6+30)
44570 R3=R5+R(81)*(R6-R4)
44580 IF R(34)<R3  GØTØ 44950
44590 IF R3<R(32)  GØTØ 44950
44600 R4=R6
44610 R5=R3
44620 X1=29
44630 GØSUB 45000
44640 X1=INT(R4)
44650 X2=INT(R5)
44660 X3=T(R)
44670 GØSUB 39000
44680 IF R0=0        GØTØ 44860
44690 IF R(35)=0     GØTØ 44770
44700 R3=2*R(35)
44710 R3=R(R3+30)
44720 R6=R4
44730 IF R8=0        GØTØ 44750
44740 R6=R6+(R3-R5)/R(81)
44750 IF R(33)<R6  GØTØ 44770
44760 IF R6>=R(31) GØTØ 44830
44770 IF R(37)<=0 GØTØ 44950
44780 R6=2*R(37)-1
44790 R6=R(R6+30)
44800 R3=R5+R(81)*(R6-R4)
44810 IF R(34)<R3  GØTØ 44950
44820 IF R3<R(32)  GØTØ 44950
44830 X1=INT(R6)
44840 X2=INT(R3)
44850 GØTØ 44880
44860 X1=INT(R1)
44870 X2=INT(R2)
44880 IF T(R)=0      GØTØ 44930
44890 R(39)=X1
44900 X1=29
44910 GØSUB 45000
44920 X1=R(39)
44930 X3=T(R)
44940 GØSUB 39000
44950 R(4)=R1
44960 R(5)=R2
44970 R(19)=R0
44980 RETURN
44990 REM -----------------------------------------------------------
45000 REM        CHØUT
45010 R(90)=0
45020 R(91)=1
45030 R(92)=T(63)=X1
45040 R(93)=INT(R(92)/8)
45050 R(90)=R(90)+(R(92)-R(93)*8)*R(91)
45060 IF R(93)=0 GØTØ 45100
45070 R(92)=R(93)
45080 R(91)=R(91)*10
45090 GØTØ 45040
45100 ' AT THIS PØINT R(90)=TØ AN ØCTAL NUMBER TØ PLACE
45110 ' IN THE ØUTPUT STRING T$
45120 T$="@@@000"
45130 S$=STR(R(90))
45140 T$(8-LEN(S$))=S$(2,LEN(S$)-1)
45150 PRINT T$
45160 RETURN
READY
```

APPENDIX B

This appendix contains the EZPLOT Users Manual which is currently in use by research scientists at Battelle-Northwest.

BNWL-MA-85

EZPLOT

USERS MANUAL

by

C. R. Watson
Statistics Section
Systems Department

September 1973

Battelle
Pacific Northwest Laboratories
Richland, Washington

PREFACE

The best manual for EZPLOT is experience. The program is interactive and easy to use. Sit down at the terminal and make some graphs. You will learn about features that I haven't mentioned in this report; you may learn about features I didn't know I had built into the program.

I suggest you use this manual in two ways: first, read it through to get a general idea of how EZPLOT is used; second, keep it handy as a reference while you are using the program. I find that I forget details about the various options and their results. A quick look in the manual can save quite a bit of computer time.

Feel free to make suggestions about both the program and this users manual; they are dynamic research aids. I hope to continue improving EZPLOT to make it more responsive to your needs. Suggestions of new features and ways of improving old ones are always welcomed. EZPLOT is designed for you, not for me, so let me know how well it is meeting that goal.

Chuck Watson

LIST OF FIGURES

INTRODUCTION

This manual explains how to use EZPLOT, a BASIC computer program available on the UNIVAC 1108, to prepare graphs of your data. The program operates through Tektronix 4010 or 4002 cathode ray tube (CRT) computer terminals. Figure 1 (reduced in size for this publication), illustrates the quality of the graphs and demonstrates the optional feature of including functions as well as data. EZPLOT allowed the user to modify scale parameters and obtain the desired graph in a few minutes.

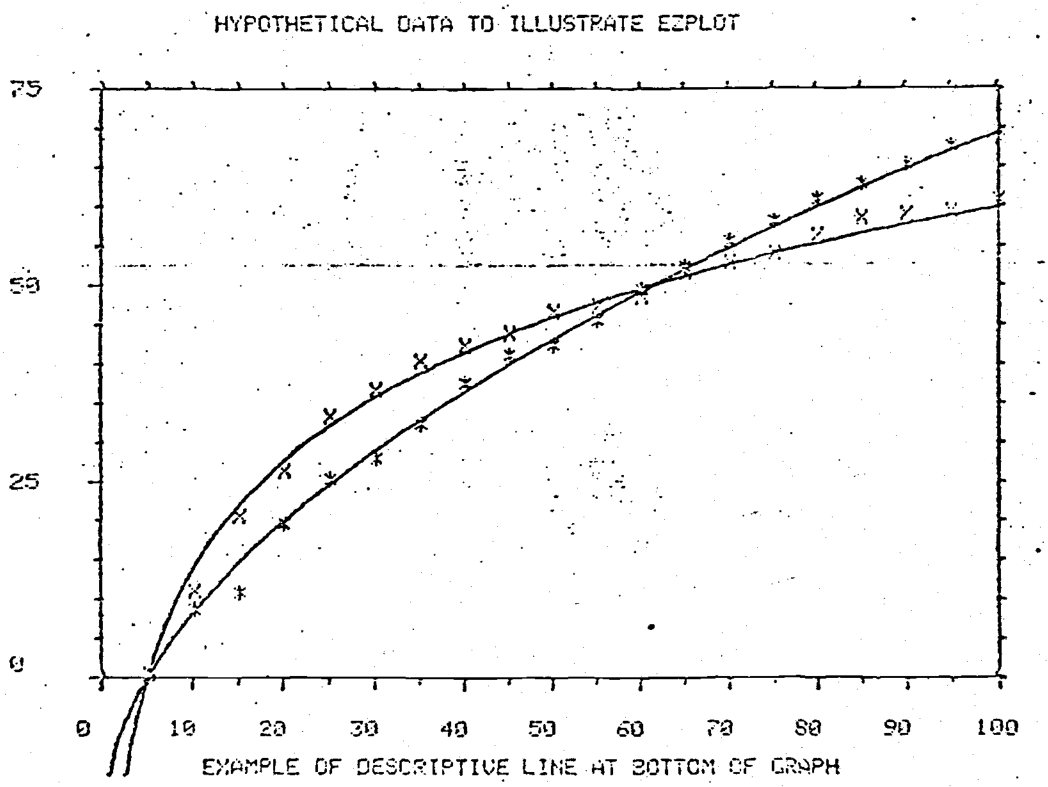


FIGURE 1. CRT Display after User's Interactions with EZPLOT to Define Linear Graph and Add Two Functions.

EZPLOT is oriented toward the needs of scientific investigators. It serves two functions: it helps communicate experimental results through visual display and it aids in the analysis of data through scale and axis transformation. The main features and limitations of the program are:

- Dual modes

  Data is easily entered via the terminal keyboard. Data files, stored in the user's BASIC library, may also be used. In both cases, the total number of pairs of points cannot exceed 200.

- Data editing routine

  A simple procedure is included for editing data by deleting points, adding them, or changing values.

- Multiple data sets

  Several sets of data may be plotted on a single set of axes, each identified by a different keyboard character.

- Transformation of axes

  Either axis may be linear or logarithmic. An axis may be redefined and the data re-plotted within minutes.

- Descriptive titles

  Two alphanumeric lines (limited to 55 characters each) are displayed, one at the top and the other at the bottom of the graph.

- Optional grid

  The user may select to have a grid superimposed on the graph.

- Functions

  Up to 5 functions may be superimposed on the display. Functions such as $y = 5\ x^2 + 3x + 2$ or $y = 27\ e^{-1.4x} + 17.5x$ which the user can express as BASIC statements are acceptable.

- Default mode

  The user may elect to have the program select the scale factors for a quick display of the data.

TEKTRONIX TERMINALS

Tektronix CRT terminals are very similar to teletype terminals with a CRT display added. Details of the operation of specific models will be found in a manual located near the terminal. Some of the unusual keyboard features of the 4010 are illustrated in Figure 2.

Most Tektronix terminals have hard copy units attached. A copy of the display may be obtained by pressing the "MAKE COPY" button on either the keyboard or the hard copy unit. The copier should be allowed several minutes of warm-up before making a copy. Copy quality depends in part on the thermal history of the silver activated paper. It may also be altered by exposure to heat or strong light immediately after the copy emerges from the unit. A good procedure is to make Xerox reproductions as soon as practical after each session of terminal usage.

To increase the useful life of the CRT screen, the Tektronix terminal will automatically dim the display if there are no keyboard entries in a 90 second period. To re-activate the intensity of display without entering extraneous data to the computer, press the SHIFT key.

Information to be sent to the computer is stored in the terminal until the RETURN key is depressed. If a mistake is made in entering a line of data, there are two methods of correcting it. Pressing the CONTROL and X keys at the same time causes the entire line to be disregarded. Pressing the SHIFT and O keys causes a backspace so that the last character may be re-entered.

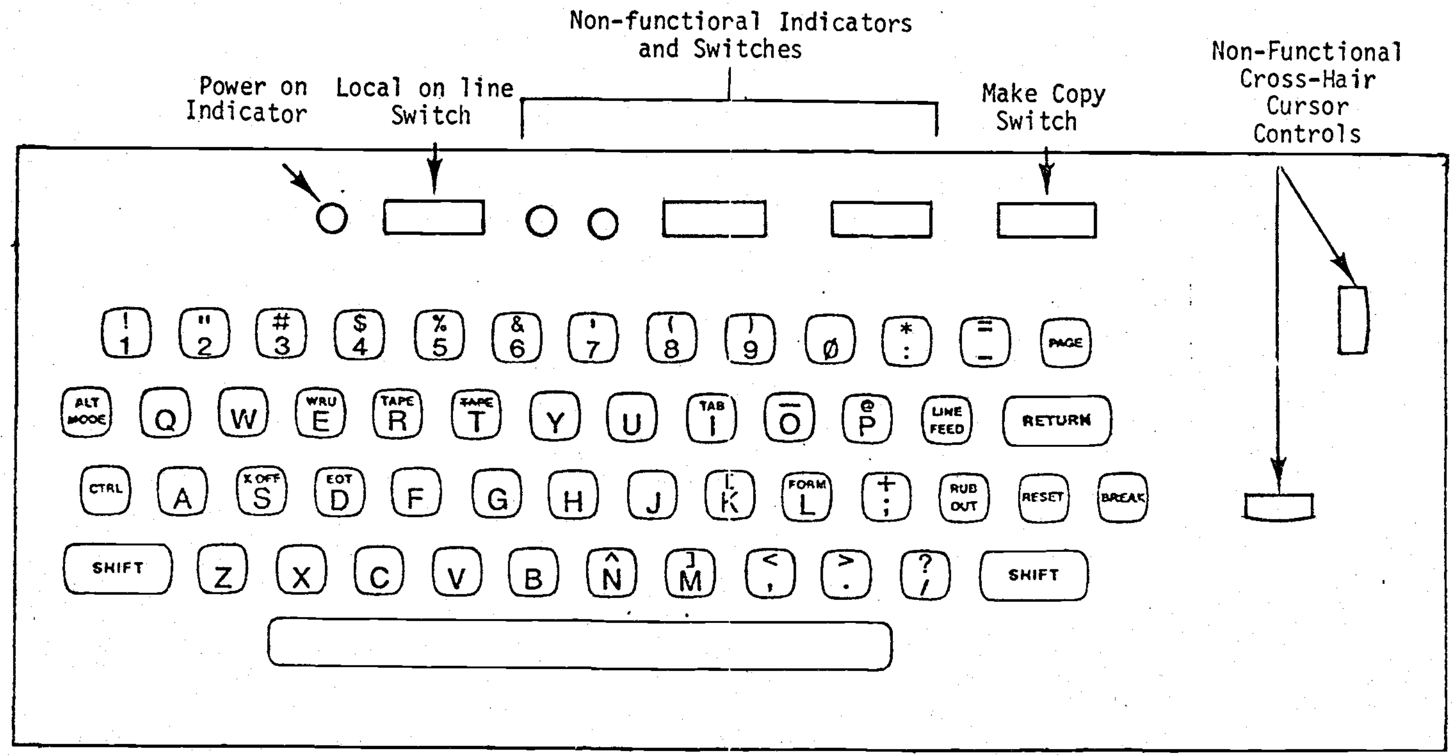

Power on Indicator
Local on line Switch
Non-functional Indicators and Switches
Make Copy Switch
Non-Functional Cross-Hair Cursor Controls
! 1 " 2 # 3 $ 4 % 5 & 6 ' 7 (8) 9 Ø * : = – PAGE
ALT MODE Q W WRU E TAPE R TAPE T Y U TAB I O @ P LINE FEED RETURN
CTRL A X OFF S EOT D F G H J [K FORM L + ; RUB OUT RESET BREAK
SHIFT Z X C V B ^ N] M < , > . ? / SHIFT

FIGURE 2. Features of the Tektronix 4010 Keyboard

(Reprinted by Permission of Tektronix, Inc. All Rights Reserved.)

SIGN-ON PROCEDURE

To activate EZPLOT, the following steps should be followed:

1. Turn on power to terminal, hard copy unit, and acoustic coupler (terminal power switch is located on the pedestal near your right knee).

2. Clear the screen with the PAGE key.

3. Dial the number of the UNIVAC 1108. This phone number depends on the particular terminal location and should be posted near the phone.

4. When you hear the computer whistle, place the handset on the cradle of the acoustic coupler.

5. A greeting from the UNIVAC should appear at the upper left corner of the screen:

```
ON AT 14:15:59  30 SEP 73  SUNDAY     RLG83J   067
ASSISTANCE:   RUN    **NEWS
USER ID,PASSWORD,PROJECT  ID—
```

6. Respond with: your USER ID (CUB #), your PASSWORD (if any), and TO. The addition of the T and zero to the normal sign on procedure tells the computer you are using a CRT. UNINTELLIGIBLE OUTPUT WILL RESULT if you omit these two letters.

7. The computer will respond with:

```
SYSTEM —
```

8. Respond by pressing three keys at once:

```
CONTROL   SHIFT   D
```

9. The computer will respond with:

```
VERSION 14 OCT 71  20:07
READY
```

10. Enter: RUN *EZPLOT; then hit RETURN.

INTERACTION WITH EZPLOT

Successful completion of the sign-on procedure will result in a flash as the screen erases itself. The following greeting will appear at the upper left corner of the screen:

```
EZPLOT VERSION 2  5/3/73
FOP DETAILS ABOUT ANY QUESTION ANSWER 0, OTHERWISE
ANSWER ALL QUESTIONS WITH SINGLE DIGIT 1-9
INPUT TYPE?
```

Don't panic! This greeting serves three functions, telling you that:

- You have activated the correct BASIC program - EZPLOT.
- The primary mode of operation is numeric user responses. Zero will cause the display of the menu of options associated with a given question.
- The first question is: INPUT TYPE?

If you are familiar with EZPLOT you may have memorized each list of options or you may operate with this manual in hand. If you know the option you want, simply enter the option number and hit the RETURN key. If you are unsure of the option, entering zero will display the choices. If you are unsure what the options mean, the section of this manual, "Explanation of Questions", provides details about each choice.

The program operates by asking you a series of questions about your data. Then, it draws the graph you have specified. You may then change some of the specifications and obtain a new graph. The program asks these questions in a logical order, much as you would ask yourself questions before preparing a graph by traditional manual methods.

The first series of questions deal with defining the data that you wish to plot. Input may come from a file you have specially stored in the computer or you may enter the data points through the terminal keyboard.

The next logical thing to do is define the type of graph paper on which you wish to plot the data. Rather than choosing from various sized linear and logarithmic paper, you are allowed to define the way in which the computer treats the CRT screen.

You are also given great latitude in the choice of scale units for the two axes. Each will be given a number of "tic" marks according to your specifications. These marks will be clearly labeled on output. You may wish to see the graph with or without a grid superimposed on the data, so the program will ask for your preference.

The final information needed to prepare your graph is descriptive labels for both the top and bottom of the graph.

Once you have selected all these parameters, the program will erase the screen and draw the graph. After giving you a chance to make a hard copy of the results, the program will allow you to re-enter the interactive phase of establishing parameters. This allows completion of several graphs of the same data.

EXPLANATION OF QUESTIONS

Each of the major questions asked by EZPLOT is associated with a list of logical options. The question, the options, and advice about the choices are presented on the following pages. It is a good procedure to have these pages available while using EZPLOT.

Some of the questions lead to conversational routines. These are (hopefully) self-explanatory and will not be discussed in detail here.

The sequence of questions depends on your responses. The order in which they appear in the following pages is typical of a session in which the user elects to use all of the features of EZPLOT. User response must always terminate with the RETURN key.

Users who are also BASIC programmers will appreciate the inclusion of program line numbers with each question. These line numbers may facilitate error recovery and/or allow a programmer to modify EZPLOT. They are not needed in normal usage.

QUESTION: INPUT TYPE?

LINE NUMBER: 230

MENU:

```
INPUT TYPE?0

  (1)  FILE IN USERS LIBRARY
  (2)  WANT TO ENTER DATA FROM KEYBOARD
  (3)  FILE AND WANT QUICK LOOK AT LINEAR GRAPH
  (4)  NO DATA, WANT TO DRAW FUNCTIONS ONLY

INPUT TYPE?
```

ADVICE:

- Option 1 - This implies that you have prepared your data in the proper form and have saved it in a BASIC library file. Figure 3 illustrates the proper data format.
- Option 2 - This implies that you have the data in hand and are ready to enter it over the terminal. Be prepared to enter X,Y pairs rather than a series of X values and then a series of Y values. Remember that numbers must be separated by either a space or a comma.
- Option 3 - This provides a quick look at the data previously stored in the user's library and is very useful in early stages of data analysis.
- Option 4 - To display functions only, you must also enter scale factors in a later question. Therefore, be sure that you choose a scale which will include the values of your function(s).

```
100  HYPOTHETICAL DATA TO ILLUSTRATE EZPLOT
110  5 0 *
120  10 8.2 *                First line must be a title.
130  15 10.4 *
140  20 19.2 *
150  25 24.9 *
160  30 27.6 *
170  35 32.1 *
180  40 37.2 *               Separate elements with either
190  45,40.8,*               a blank or a comma.
200  50,42,*
210  55,45.2,*
220  60,49.1,*
230  65,52,*
240  70,55.2,*               Plot symbol - good ones are:
250  75,57.7,*                 *, X, O, ., I, + .
260  80,60.5,*               The dash (-) is a special plot
270  85,62.4,*               symbol, resulting in connected
280  90,64.9,*               data points.
290  95,67.3,*
300  100,69.2,*
310  888,888,*               Convention to indicate new
320  5,0,X                   set of data follows.
330  10,10.5,X
340  15,20.2,X
350  20,26.1,X
360  25,32.9,X               It is a good idea to order your
370  30,36.3,X               data on X if you plan to plot
380  35,40,X                 with the data connected.
390  40,42.1,X
400  45,43.5,X
410  50 46.3,X
420  55,46.9,X
430  60,48.2,X
440  65,51.3,X
450  70,52.6,X
460  75,53.7,X
470  80,55.9,X
480  85,58.2,X
490  90,58.7,X
500  95,58.9,X
510  100,60.4,X
520  999,999,X               Must end with 999,999.
                 Plot symbols
              y values
         x values
line numbers
```

FIGURE 3. Data File for Use by EZPLOT

QUESTION: LIST AND/OR EDIT DATA?

LINE NUMBER: 461

MENU:

```
LIST AND/OR EDIT DATA?0
1 = YES, 2 = NO
LIST AND/OR EDIT DATA?
```

ADVICE:

- Option 1 - This will lead you to the editing routine which is described later.
- Option 2 - If you skip editing or listing data at this point and you later see errors in the plot of the data, you may return to this question as one of the options to a later question: WHAT NEXT?
- Remember - Respond with a number, either 1 or 2. A response of YES or NO results in a computer detected error condition.

QUESTION: EDIT OPTION?

LINE NUMBER: 7990

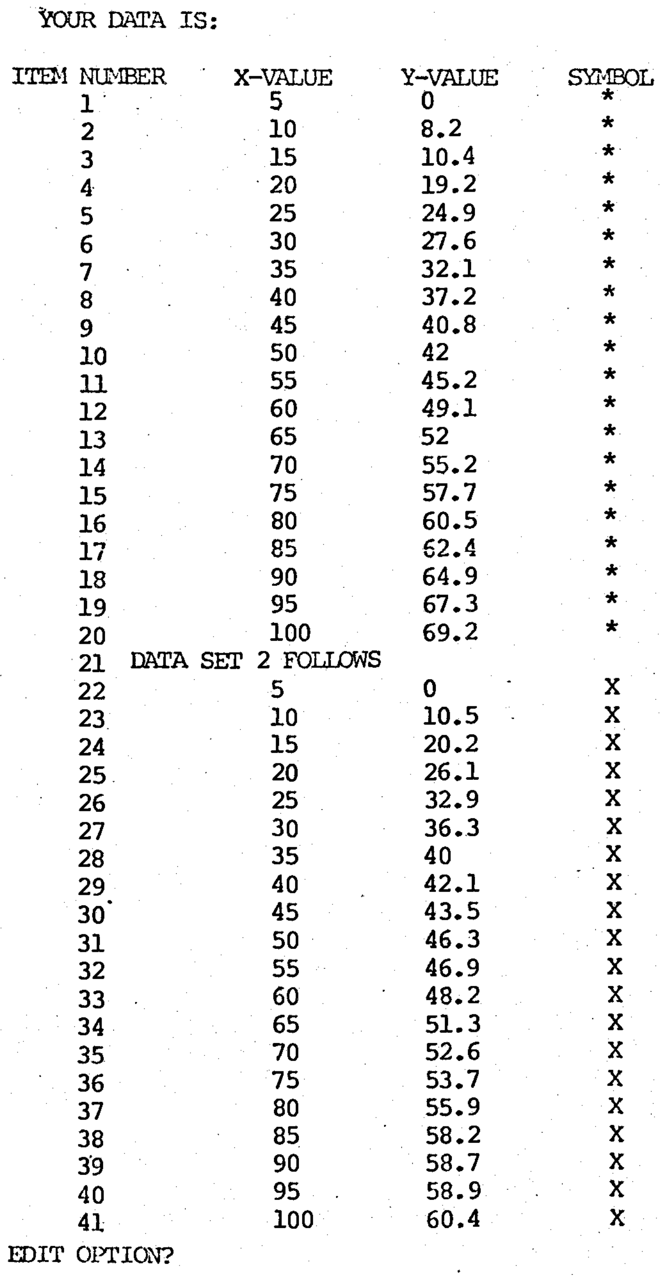
MENU:

```
EDIT OPTION?0
OPTIONS    REASONS
   1       DISPLAY DATA
   2       CHANGE 1 OR MORE ITEMS
   3       INSERT ITEMS
   4       SWITCH A PAIR OF ITEMS
   5       DELETE AN ITEM
   6       CHANGE PLOT SYMBOL FOR A SET
   7       DATA OK, PROCEED
EDIT OPTION?
```

ADVICE:

- Option 1 - It is almost essential that you display the data at least once if you plan to edit it. The editing depends on item numbers assigned to the pairs of points as shown in Figure 4.
- Options 2 through 6 - These lead you to interactive routines in which you will be asked which items to change, switch, delete, etc. After completing the data changes, these editing options return to the question EDIT OPTION? It is a good idea to respond with 1 to see if the edited data is correct.
- Option 6 - This option is illustrated in Figure 5 where plot symbols are changed to display the data points connected by line segments rather than as individual symbols.
- Option 7 - This returns you to the main sequence of questions. You may re-enter the editing sequence later as option 4 of the question WHAT NEXT?

The following is an example of the display generated by option 1. Subsequent editing depends on the item numbers. For example, the first pair of points in the second data set must be referred to as item 22.



```
 YOUR DATA IS:

ITEM NUMBER    X-VALUE      Y-VALUE      SYMBOL
       1          5           0             *
       2          10          8.2           *
       3          15          10.4          *
       4          20          19.2          *
       5          25          24.9          *
       6          30          27.6          *
       7          35          32.1          *
       8          40          37.2          *
       9          45          40.8          *
       10         50          42            *
       11         55          45.2          *
       12         60          49.1          *
       13         65          52            *
       14         70          55.2          *
       15         75          57.7          *
       16         80          60.5          *
       17         85          62.4          *
       18         90          64.9          *
       19         95          67.3          *
       20         100         69.2          *
       21  DATA SET 2 FOLLOWS
       22         5           0             X
       23         10          10.5          X
       24         15          20.2          X
       25         20          26.1          X
       26         25          32.9          X
       27         30          36.3          X
       28         35          40            X
       29         40          42.1          X
       30         45          43.5          X
       31         50          46.3          X
       32         55          46.9          X
       33         60          48.2          X
       34         65          51.3          X
       35         70          52.6          X
       36         75          53.7          X
       37         80          55.9          X
       38         85          58.2          X
       39         90          58.7          X
       40         95          58.9          X
       41         100         60.4          X
EDIT OPTION?
```

FIGURE 4. Example of Editing Option to Display Data

The example below illustrates a change from the plot symbol X to the special plot symbol -. The program will connect your data points when the plot symbol is a dash (-). In that case, the results will look funny if the data is not ordered properly, because data is not sorted by EZPLOT.

```
EDIT OPTION?6
SYMBOL CHANGE
FIRST ITEM NUMBER?22
LAST ITEM NUMBER?41
ENTER NEW SYMBOL?-
EDIT OPTION?7
```

Further interactions resulted in Figure 5.

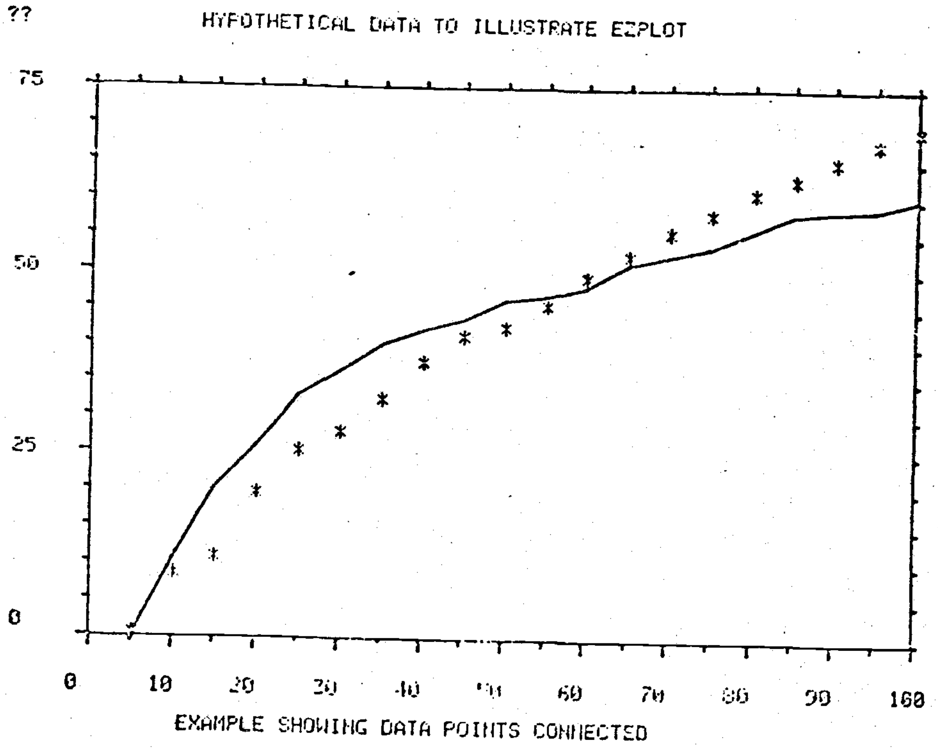


FIGURE 5. CRT Display after Editing to Change Plot Symbol of Second Set of Data.

QUESTION: TYPE OF GRAPH?

LINE NUMBER: 800

MENU:

```
TYPE OF GRAPH?0
1 = LINEAR
2 = LOG Y VS LINEAR X
3 = LINEAR Y VS LOG X
4 = LOG Y VS LOG X
TYPE OF GRAPH?
```

ADVICE:

- Option 1 - Figure 1 used this option.
- Log Scales (Options 2, 3, or 4) - The program uses log to the base 10. Any data items less than or equal to zero will be ignored. You will get a message to this effect prior to the drawing of the graph, but the graph itself will show no trace of such out-of-range data.
- Option 3 - Figures 9 and 10 used this option.
- General - This question is a powerful tool in the analysis of data. Use it freely to explore alternate ways of displaying graphs of the data. After display of one type of graph, you may return to this question with option 2 of the question, WHAT NEXT?

QUESTION: AUTO SCALE?

LINE NUMBER: 1270

MENU:

```
AUTO SCALE?0

1 = AUTOMATIC SCALE TO FIT DATA
2 = USER WILL SUPPLY SCALE FACTORS

AUTO SCALE?
```

ADVICE:

- Option 1 - The program will adjust the scale of the graph to fit your data by setting the endpoints of each axis to the minima and maxima of your data. The axes will be subdivided into ten equal parts. Usually the results are not pleasing to the eye although they are mathematically correct. For example, if the range of the data is from 1 to 10, the "tic" marks on the axis will be located at 1.0, 1.9, 2.8, 3.6, 4.5, etc. Figure 6 illustrates automatic scaling of a linear graph. Automatic scaling of log plots is discussed on page 19.
- Option 2 - User supplied scale factors give nicer looking results (Compare the example of automatic scaling in Figure 6 with the user supplied scale in Figure 7.), but requires more time by the user. In general, use auto scale in analysis of data and use your own scale factors when preparing graphs to communicate results to others.

  The following example illustrates the interactions between the program and the user in defining the scale factors for one axis of the graph (user responses underlined).

```
X-AXIS:

              MIN X     MAX X
DATA            5        100
AXIS            0          0
MINIMUM VALUE-?0
MAXIMUM VALUE-?100
X-AXIS GOES FROM 0 TO 100
WITHIN THIS RANGE OF 100 UNITS
HOW MANY UNITS DO YOU WANT BETWEEN TIC MARKS?5
THAT WILL MEAN 20 TIC MARKS ON THE AXIS
LABEL FACTOR?2
```

After setting the endpoints at zero and 100, the user specified that at every 5 units there would be a tic mark. The last question, LABEL FACTOR?, allows the user to choose which of the tic marks will be labeled. In this case the user asked that every other tic be labeled.

```
Y-AXIS:

              MIN Y     MAX Y
DATA            0        69.2
AXIS            0           0
MINIMUM VALUE?0
MAXIMUM VALUE?75
Y AXIS GOES FROM 0 TO 75
WITHIN THIS RANGE OF 75 UNITS
HOW MANY UNITS DO YOU WANT BETWEEN TIC MARKS?5
THAT WILL MEAN 15 TIC MARKS ON THE AXIS
LABEL FACTOR?5
```

The result of these selections is shown in Figure 7 and is to be compared with Figure 6.

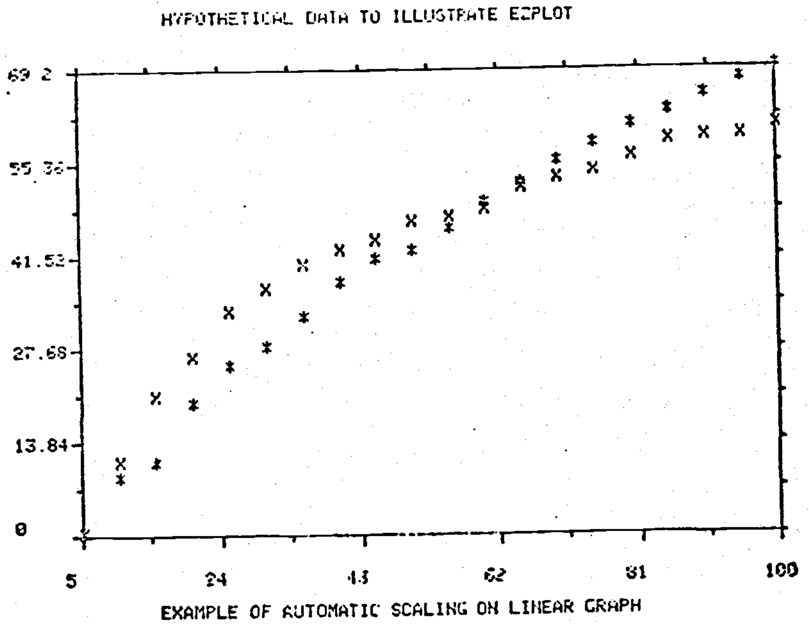


FIGURE 6. CRT Display after selection of Automatic Scaling for a Linear Graph.

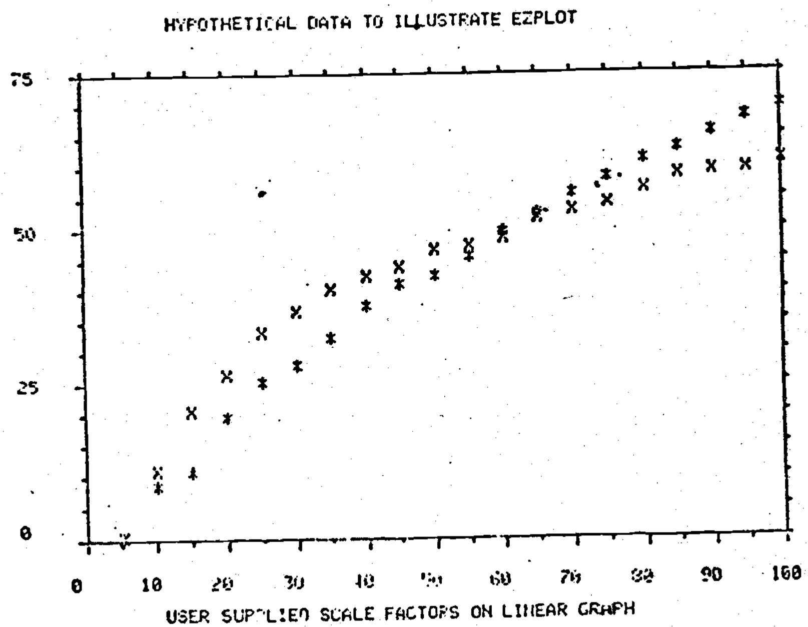


FIGURE 7. CRT Display after Input of User's Scale Factors for a Linear Graph.

- Option 1 with LOG SCALES - The logic for automatic scaling of log axes is the same as that for linear ones. As shown in Figure 8, this results in plots which are mathematically correct, although the axis divisions are unconventional

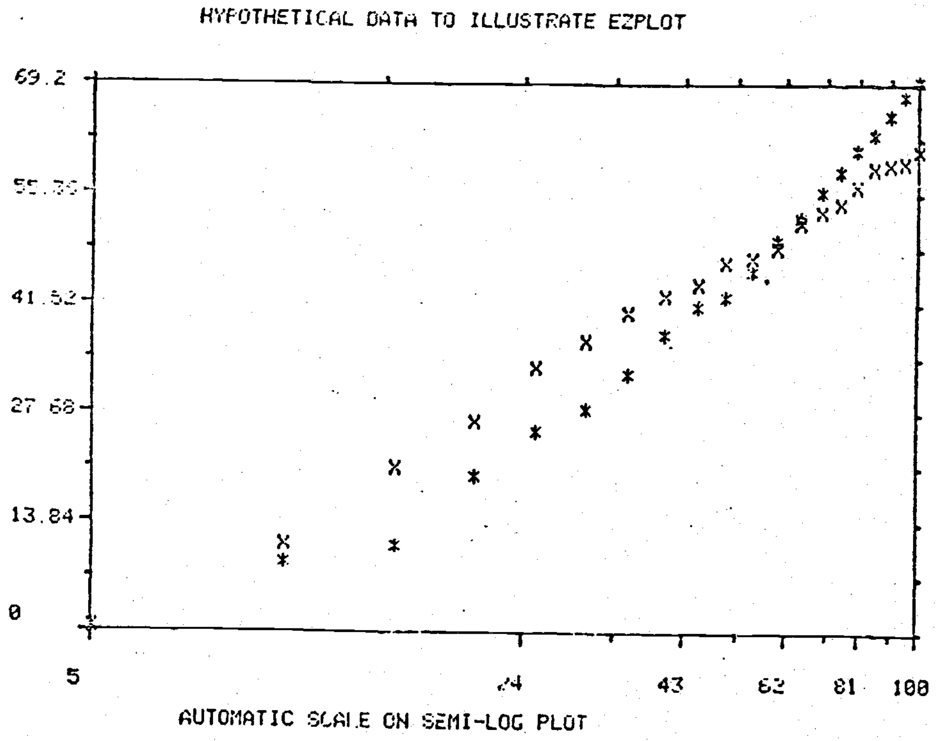


FIGURE 8. CRT Display after Selection of Automatic Scaling for a Semi-Log Plot.

•Option 2 with LOG SCALES - The procedure for scaling log axes is different than that for linear ones. You will be asked for the location of each tic mark along the axis. To place these properly, think of the axes in terms of decades. For example, if the endpoints are at 1 and 100 you have 2 decades. In that case, the tics would look appropriate at 2,4,5,8,10,20,40,60, and 80.

Remember that the axis cannot include zero or negative values on a log scale.

An example of user supplied scaling factors for a log axis follows.

```
X-AXIS:

              MIN X     MAX X
DATA            5        100
AXIS            0        100
MINIMUM VALUE-?5
MAXIMUM VALUE-?100
X-AXIS GOES FROM 5 TO 100
IN ADDITION TO THESE TWO ENDPOINTS
HOW MANY OTHER TIC MARKS ARE WANTED (MAX. 10)?6

X VALUE FOR TIC 1?7.5
X VALUE FOR TIC 2?10
X VALUE FOR TIC 3?20
X VALUE FOR TIC 4?40
X VALUE FOR TIC 5?60
X VALUE FOR TIC 6?80
```

The resulting graph is shown in Figure 9. Figure 10 is an example of a semi-log scale which is smaller than the data range. The user elected to restrict the X-axis to run from 10 to 100. The excluded data points (at X = 5) are shown on the left margin of the screen.

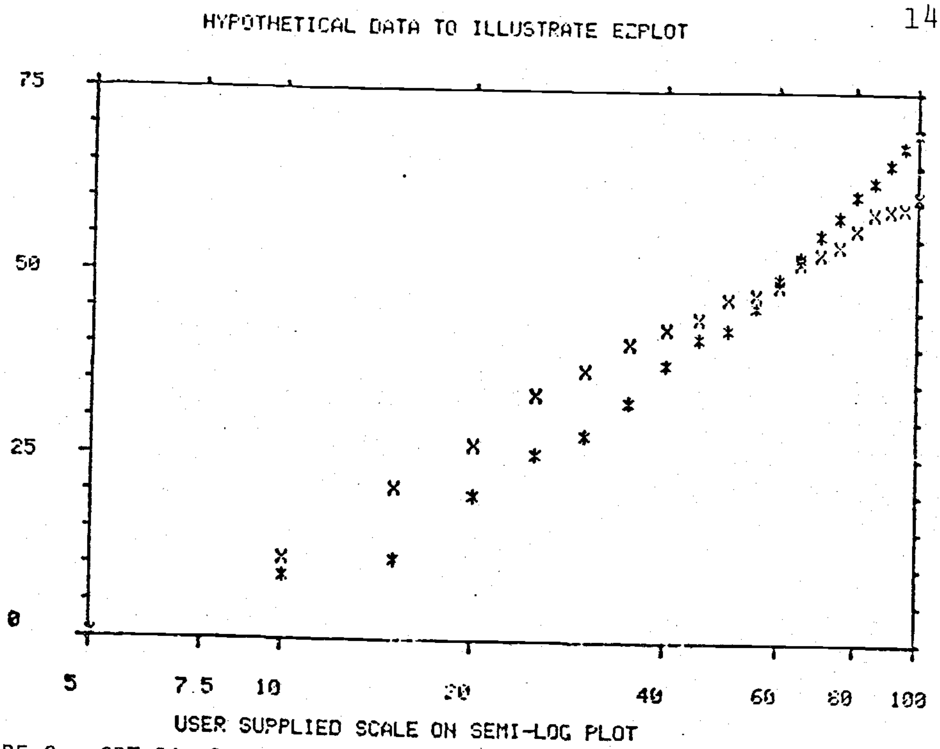


FIGURE 9. CRT Display after Input of User's Scale Factors for a Semi-Log Plot.

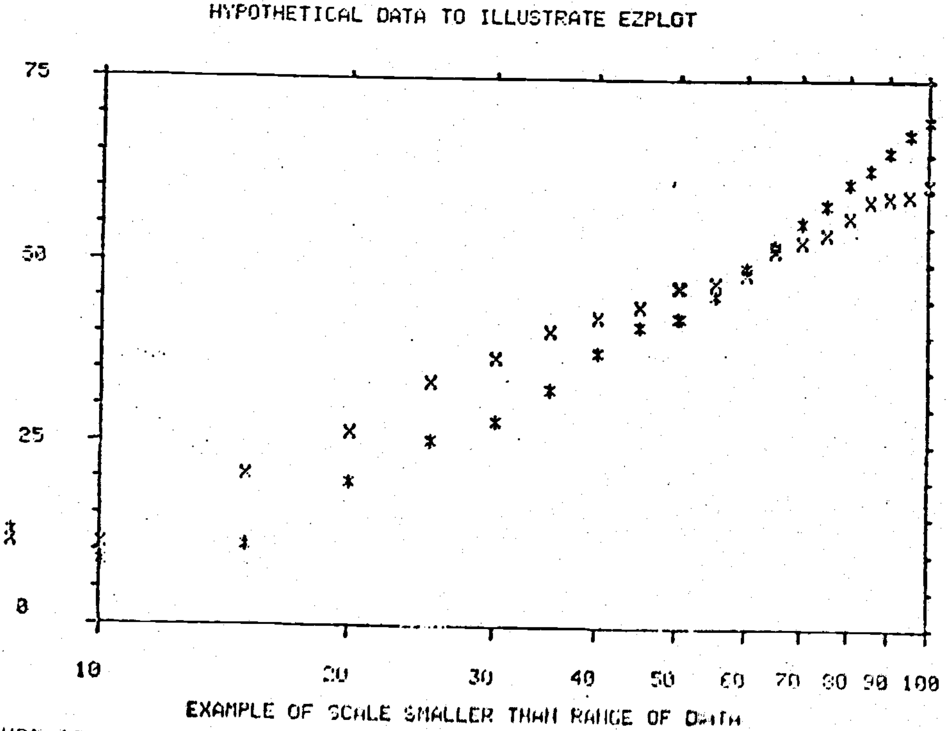


FIGURE 10. CRT Display after Input of User's Scale Factors which are Smaller than Range of Data on Semi-Log Plot.

QUESTION: GRID?

LINE NUMBER: 2621

MENU:

```
GRID?0

1 = AUTOMATIC GRID AT EACH TIC
2 = NO GRID AT ALL
3 = USER WILL SUPPLY GRID LOCATIONS

GRID?
```

ADVICE:

- Option 1 - Try to visualize how the graph will look with grid lines at each tic location. Remember that the program has no control over the intensity of the image on the screen. If there are more than a few tic marks, a graph with full grid usually obscures the data.

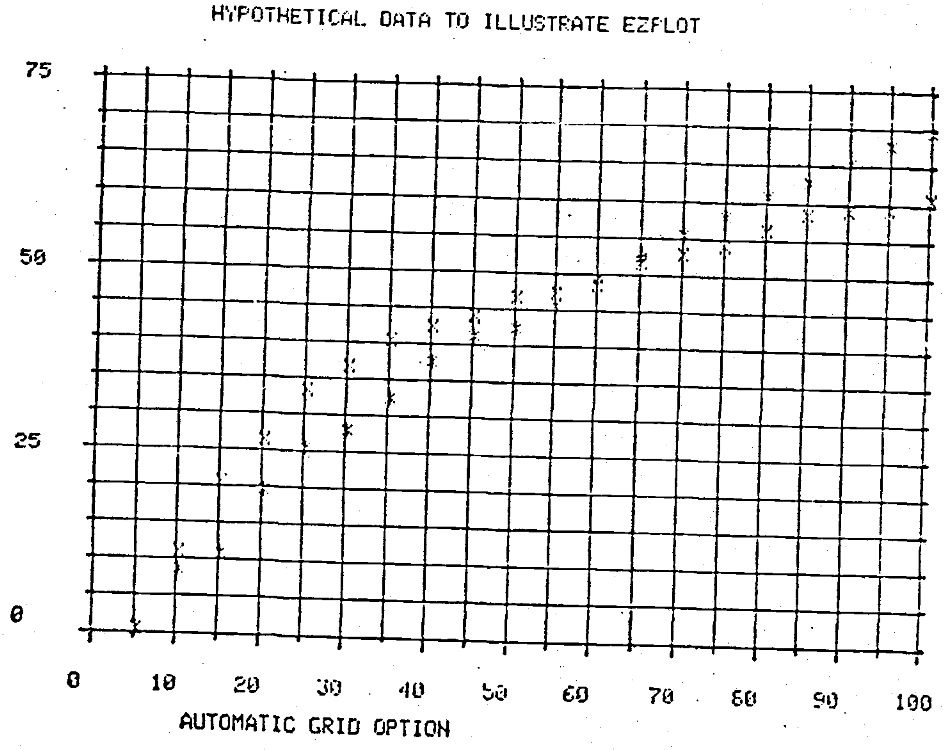


FIGURE 11. CRT Display after Selection of Automatic Grid Feature.

- Option 2 - Most of the time you will elect to skip the grid feature. To locate specific data points it is usually possible to use a straight edge with reasonable accuracy.

- Option 3 - As shown in Figure 12, user selected grids are often useful enhancements to the graph. You will supply each location, for example, in Figure 12 horizontal grid lines were requested at y = 25 and y = 50 and vertical lines at x = 20, x = 40, x = 60 and x = 80.

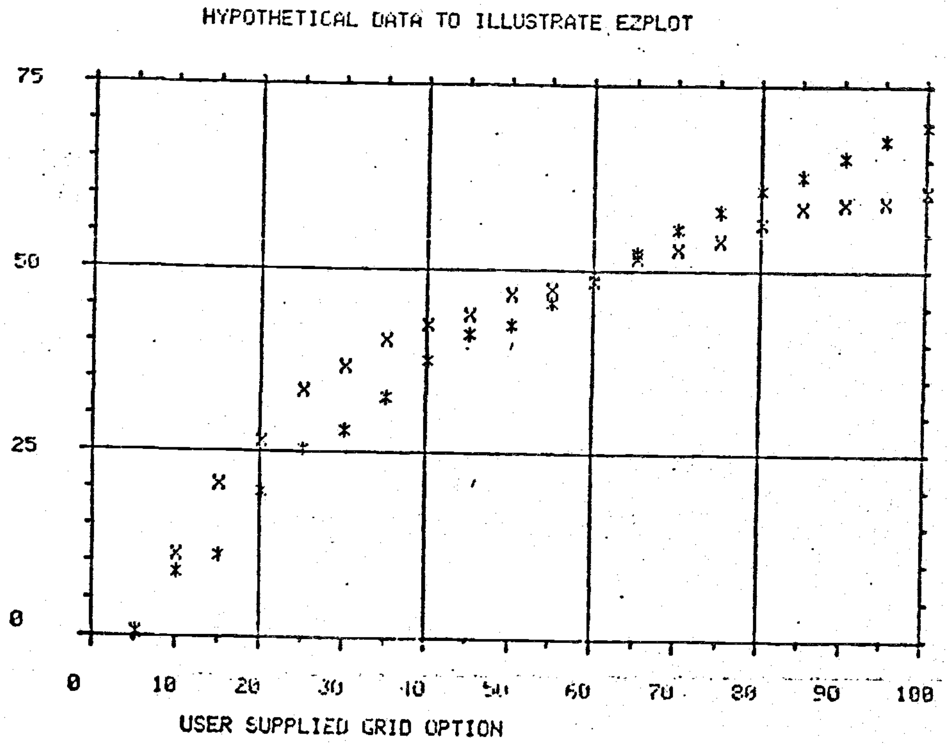


FIGURE 12. CRT Display after Input of User's Choice of Grid Line Locations.

- Alternate uses of grid - The grid option 3 allows you to do things not usually associated with a grid. For example, if your Y-axis runs from -10 to +10, you might want a horizontal line across the graph at Y = 0. This is easy under option 3; simply ask for one horizontal grid line Y = 0.

  Another use of the grid feature is to highlight interesting aspects of your data. For example, you may wish to call attention to the intersection of two curves by using a pair of grid lines which also intersect there as shown in Figure 13.

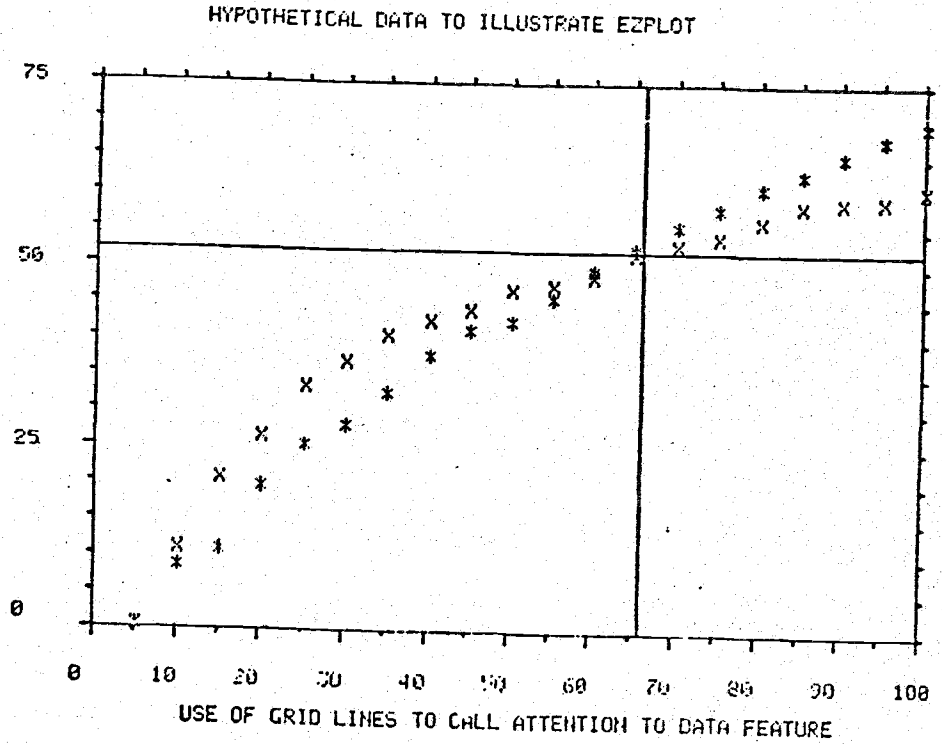


FIGURE 13. CRT Display Illustrating Use of Grid Lines to Call Attention to Data Feature.

QUESTION: TOP TITLE OK?

LINE NUMBER: 2630

MENU:

```
TOP TITLE OK?0
YOU MAY:
(1) HAVE:*** YOUR PREVIOUSLY ENTERED TITLE WILL PRINT HERE ***
    DISPLAYED AT TOP OF GRAPH
OR
(2) ENTER A NEW TITLE FOR THE TOP OF THE GRAPH
TOP TITLE OK?
```

ADVICE:

- General - The choice of wording of the top title is up to the user. Most people make the top line a general description of the data, using the bottom line to identify the variables.
- Option 2 - If you decide to re-enter the top title, the program will solicit your new title, then ask again TOP TITLE OK?. It is good practice to respond with zero so that the program will display the line you have just entered. This will help avoid misspellings or lines that were too long to fit on the graph.

QUESTION: BOTTOM LINE OK?

LINE NUMBER: 2990

MENU:

```
BOTTOM LINE OK?0
ARE YOU:
(1)  SATISFIED WITH:*** YOUR PREVIOUSLY ENTERED LINE WILL
                        PRINT HERE ***
OR DO YOU
(2)  WANT A DIFFERENT BOTTOM LINE
BOTTOM LINE OK?
```

ADVICE:

- General - Most investigators use the bottom line to identify the axes units. For example:

  UPTAKE (IN MG PER L) VS TIME (IN HOURS)

- Option 2 - As with the top title, it is a good practice to ask the program to display your new bottom line by responding with zero when it asks BOTTOM LINE OK? after you entered the new bottom line.

QUESTION: WHAT NEXT?

LINE NUMBER: 4180

MENU:

```
WHAT NEXT?0

1 = PLOT DATA AS IT IS
2 = CHANGE TYPE OF GRAPH
3 = CHANGE SCALE
4 = EDIT DATA
5 = ADD FUNCTIONS
6 = GET NEW DATA FILE
7 = QUIT

WHAT NEXT?
```

ADVICE:

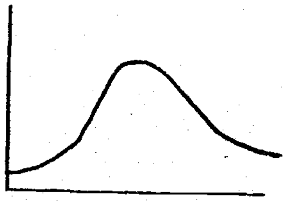
- Option 1 - The screen will erase, and your graph will be drawn. At the conclusion of the drawing, two question marks will appear at the upper left corner of the screen. The display will then remain on the screen until you hit the RETURN key. Then the screen will erase and you will be asked WHAT NEXT?
- Option 2 - This takes you back to the question TYPE OF GRAPH?
- Option 3 - This takes you back to the question AUTO SCALE?
- Option 4 - This will take you to the question EDIT OPTION? After editing the data you will be asked WHAT NEXT? so that you do not have to re-set the scale parameters.
- Option 5 - This will lead to an interactive routine in which you are invited to interrupt program execution to add as many as 5 functions. The program will return to WHAT NEXT?

  Two functions were added to the program to produce the graph shown in Figure 1. The following example illustrates how two such functions can be added. For help in expressing your functions in BASIC consult the CSC BASIC REFERENCE MANUAL or find a BASIC programmer who can help you.

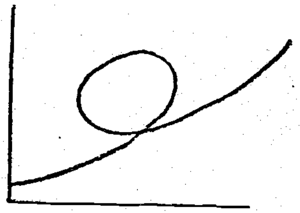
```
YOU MUST ENTER YOUR FUNCTIONS AS BASIC STATEMENTS
ENTER THE NUMBER OF FUNCTIONS YOU WISH TO DISPLAY
(LIMIT 5)?2
INSERT FUNCTIONS AT 4950, 4970, 4990, 5010, 5030
EXAMPLE:  4950 Y = SIN(X)
TO SEE THE RESULTS, TYPE "GO TO 9600"
HIT CONTROL-SHIFT-K NOW!!!

ATTENTION
READY

4950 Y = (SQRT (2 * X / 5))* 2

4970 Y = 4.56*EXP(1.223 * X) - 7.5E-4*X + 0.003429

GO TO 9600
```

When the user enters CONTROL-SHIFT-K the program EZPLOT is inoperative. Control is returned to the BASIC system. This allows the user to insert BASIC statements at the correct locations. Upon return to EZPLOT (via GO TO 9600) the functions may be drawn.

The logic for drawing functions is to step across the screen with increasing values of X. This means that functions such as the equation of a circle are not valid for EZPLOT.



Legal Function



Illegal Function

- Option 6 - This will go back to INPUT TYPE? All parameters will be cleared and you will be asked all questions over again.

IN CASE OF ERROR

Suppose that you make a mistake in entering your response. Even though you are not a computer programmer, a few simple rules will enable you to recover from error conditions without the loss of previously entered information. Three types of mistakes are common; each has an associated error recovery technique.

1) Error detected before RETURN key is depressed

Simultaneous pressing of the SHIFT and O (letter O) keys results in a backspace, allowing entry of the correct response. Underlined characters (-) will appear on the screen to indicate a logical backspace each time the SHIFT/O key is depressed.

Simultaneously pressing the CONTROL and X keys causes the entire line of information to be deleted, allowing re-entry of the correct information. The word DELETED will appear followed by a line feed to indicate that the computer is ready for the new line.

2) Error detected after RETURN key is depressed

You may proceed but will obtain a display that is not correct. At that point EZPLOT allows you to change parameters and obtain a modified graph. This method wastes time but is simple.

A more economical method is based on interrupting execution of the program and directing it to ask the question again. Each step of the program has a unique line number. Line numbers of the menu selecting questions are listed in the Summary section of this manual.

To interrupt the program, enter CONTROL-SHIFT-Z. The computer will respond with: READY To direct the program to repeat the question, enter:

GO TO line number; then hit RETURN.

CAUTION: Do not use this method of error correction unless you detect the error immediately after it is made. Strange things may happen if you attempt to skip back several questions.

3) Error detected by the computer

Errors such as entry of alphabetic information when the computer expects a number result in interruption of program execution. The computer will display a message of some sort, followed by

```
READY-
```

At this point you may enter GO TO line number to cause the question to be repeated.

Another procedure is to enter the word:

```
CONTINUE
```

This will cause the last line prior to the error to be repeated. Usually this will result in the display of a solitary question mark. Respond as if the previous question had been repeated.

Sometimes the computer itself makes mistakes. These usually appear as strange strings of characters in the middle of your graph. The problem is most often in telephone transmission of information from the computer to your remote terminal. Try to have the graph re-displayed. If it is still faulty, try the entire procedure over again, including hanging up and re-dialing the computer.

If things get too bad, it is possible to terminate a computer session any time the READY- condition exists. To terminate, respond with OFF.

SIGN-OFF PROCEDURE

To terminate a computer session, the following steps should be followed:

1. Type OFF when the Univac says READY-.
2. Erase the screen with the PAGE key.
3. Turn off power to: Terminal
   Hard Copy Unit
   Acoustic Coupler
4. Hang up the telephone.
5. Place bad copies in waste basket; take good ones with you.

SUMMARY OF QUESTIONS

In summary, the EZPLOT BASIC computer program can be operated by responding to a series of questions about your data. The following is a guide to the major questions asked by EZPLOT.

Line Number	Question	Options
230	INPUT TYPE?	1 File 2 Keyboard 3 File & Quick Linear 4 Functions Only
461	LIST AND?OR EDIT DATA?	1 Yes 2 No
7990	EDIT OPTION?	1 Display Data 2 Change Data 3 Insert 4 Switch 5 Delete 6 Symbol Change
800	TYPE OF GRAPH?	1 Linear 2 Log Y Linear X 3 Linear Y Log X 4 Log Y Log X
1270	AUTO SCALE?	1 Yes, Automatic Scale 2 User Supplied Scale
2621	GRID?	1 Automatic Grid 2 No Grid 3 User Supplied Grid
2630	TOP TITLE OK?	1 Yes 2 No
2990	BOTTOM LINE OK?	1 Yes 2 No
4180	WHAT NEXT?	1 Plot 2 Change Type 3 Change Scale 4 Edit 5 Add Functions 6 New Data 7 Quit