NASA Technical Memorandum 88293

User's Manual for EZPLOT Version 5.5 – A FORTRAN Program for Two-Dimensional Graphic Display of Data

Charles Garbinski, Paul C. Redin, and Gerald D. Budd

October 1988

NASA
National Aeronautics and
Space Administration

NASA Technical Memorandum 88293

User's Manual for EZPLOT Version 5.5 – A FORTRAN Program for Two-Dimensional Graphic Display of Data

Charles Garbinski, Paul C. Redin, and Gerald D. Budd
Ames Research Center, Dryden Flight Research Facility, Edwards, California

1988

National Aeronautics and
Space Administration
Ames Research Center
Dryden Flight Research Facility
Edwards, California 93523-5000

CONTENTS

ABSTRACT 1

INTRODUCTION 1

PROGRAM OVERVIEW 1
- Program Structure . . . 1
- Files and Parameters . . . 1
- Plot Features . . . 2

PROGRAM OPERATION 3
- Terminal Type . . . 3
- Standard Commands . . . 3
- Data Input Files . . . 4
- File and Parameter Names . . . 5
- Plot Commands . . . 6
- Screen Locations . . . 14
- Metafiles . . . 14
- Redirection and Batch Operation . . . 15
- Error and Warning Messages . . . 16
- Known Bugs . . . 20
- Tips for Efficient Usage . . . 20

PROGRAM TRANSMITTAL AND IMPLEMENTATION 21

APPENDIX A — EZPLOT SHELLFILE 23

APPENDIX B — GENERIC DATA INPUT FILES 26

APPENDIX C — PLOT FILE 28
- Interactive Mode . . . 28
- Batch Mode . . . 28

APPENDIX D — PROGRAM CATALOG 30
- Main Program . . . 30
- Subroutines . . . 30
- Functions . . . 34
- GetData Subprograms . . . 35
- EMBOS Subroutines . . . 35
- Intrinsics . . . 36

APPENDIX E — FILE ASSIGNMENTS 37

REFERENCES 38

FIGURES 38

ABSTRACT

EZPLOT is a computer applications program that converts data resident on a file into a plot displayed on the screen of a graphics terminal. This program generates either time history or *x-y* plots in response to commands entered interactively from a terminal keyboard. Plot parameters consist of a single independent parameter and from one to eight dependent parameters. Various line patterns, symbol shapes, axis scales, text labels, and data modification techniques are available. This user's manual describes EZPLOT as it is implemented on the Ames Research Center, Dryden Flight Research Facility ELXSI computer using DI-3000 graphics software tools.

INTRODUCTION

Flight test and other data can be analyzed more quickly and conveniently when plotted data are displayed on the screen of a graphics terminal rather than hand drawn on graph paper. This screen-display capability requires that a computer program be used to convert data read from a disk file into a plot on a graphics terminal. EZPLOT is an interactive computer applications program written at NASA Ames Research Center, Dryden Flight Research Facility (Ames-Dryden), to fulfill this requirement.

This user's manual describes the program's capabilities and how they may be used. This description includes features that may be unique to the Ames-Dryden implementation of EZPLOT on the ELXSI computer (ELXSI, San Jose, California) using EMBOS, the ELXSI message-based operating system (ref. 1). A user at another location or with another computer may have to modify the program to account for these system-dependent features.

This manual reflects EZPLOT Version 5.5. Subsequent versions may be reflected in updates to, or new releases of, this manual.

PROGRAM OVERVIEW

Program Structure

EZPLOT is a modular FORTRAN program that calls subroutines from the DI-3000 (Precision Visuals, Inc., Boulder, Colorado) graphics software library (ref. 2) and from GetData, a utility program for manipulating files of time history data (ref. 3). The main EZPLOT program and all applicable EZPLOT, DI-3000, and GetData subroutines and functions are compiled and bound (linked) to form an executable module. This module is then used interactively to generate time history and *x-y* data plots in response to commands and other information entered from a terminal keyboard.

EZPLOT may also be executed in a noninteractive mode called "batch." On the ELXSI computer, an EZPLOT batch job runs independently from any keyboard input. Commands and other inputs are read from a file. This mode of execution is described in the Redirection and Batch Operation section.

Files and Parameters

Data to be plotted are read from data input files during program execution. A file is a collection of data values residing on a computer storage device such as a disk or a tape. All files are identified to EZPLOT by a file name, which must follow the file-naming conventions of the host operating system. A parameter is a subset of the data values in a file and is identified to EZPLOT by an alphanumeric label called a parameter *name* or *ID*. Files and parameters are discussed further in the Data Input Files and File and Parameter Names sections.

The program plots one dependent parameter as a function of one independent parameter using data from as many as eight files. The program also plots as many as eight dependent parameters as functions of one independent parameter using data from one file. A maximum of 10,000 data values for each parameter may be plotted. This limit may be changed with a simple modification to the FORTRAN code.

Plot Features

This section briefly describes some of the general plot features. All plot features are discussed in detail in the Plot Commands section.

Figures 1 to 5 are representative of plots drawn by the EZPLOT program. The format of these plots may be easily modified to meet many analysis and reporting requirements. In some cases, copies of the screen display may be used as figures in technical reports.

Axes and scales.— A plot always consists of one independent parameter and from one to eight dependent parameters. Data are plotted on the screen relative to a set of orthogonal axes oriented with the independent parameter along the horizontal x axis and the dependent parameter or parameters along the vertical y axis. Each axis is divided into intervals by tick marks. By default, a plot is drawn with 10 intervals along the x axis and 8 intervals along the y axis, as illustrated in figure 1.

Scales may be linear or logarithmic, as illustrated in figure 2. Default scales are linear and are calculated automatically to fit all the data available for each plotted parameter. These default scales and the number of intervals may be overridden on any axis. If an axis is rescaled, data falling outside the new scale limits are not plotted. The number of intervals along a logarithmic scale is automatically adjusted to fit the minimum and maximum scale values.

A single common y axis and scale may be used for all dependent parameters, or a separate y axis and scale may be drawn for each dependent parameter. When separate y axes are used, there are two intervals for each parameter and the y axes scales are linear. Multiple y axes are illustrated in figure 3.

Plots without multiple y axes may display a background grid consisting of solid or dotted rectilinear lines aligned with the tick marks on the axes. A grid using dotted lines is illustrated in figure 1.

Lines and symbols.— The plotted data points may be connected with lines, marked with symbols, or both. If both lines and symbols are drawn, the lines are solid. If lines are drawn without symbols, the lines are dashed. Eight line patterns and eight symbol shapes are available. By default, their order of introduction follows that suggested in the NASA Publications Manual 1974 (ref. 4). This order may be overridden, if desired. Line patterns are illustrated in figure 1 and symbol shapes in figure 4.

Labels and legends.— By default, the x axis is labeled with the name of the independent parameter. If there is one dependent parameter, the y axis is labeled with the parameter name. If there are two or more dependent parameters, the y axis is not labeled. Instead, the parameter names are put in a legend. In addition to these labels, the tick marks along each axis are labeled with numbers representing the value of that location on the axis. Each plot may also have a title at the top, a figure description at the bottom, and up to nine optional lines of text placed anywhere on the plot. The text and the size of the characters in any of these labels may be changed. All letters are drawn in uppercase.

Data modification.— Before a plot is drawn, the data may be scaled, translated, thinned, or sorted. As a form of wild-point editing, plotted points may be deleted or replaced. Previously deleted or replaced points may be reinserted.

Curve fitting.— A single dependent parameter may be fitted with up to a ninth-order least-squares polynomial curve. A second-order curve fit is illustrated in figure 5. Some statistical characteristics of the data and the fit may be displayed.

PROGRAM OPERATION

Terminal Type

EZPLOT can be run from a Tektronix 4014, 4114, 4107, or 4109, a VT125, or any other terminal that can emulate a VT100 (Digital Equipment Corp., Maynard, Massachusetts) and a Tektronix 4014 (Tektronic, Inc., Beaverton, Oregon). The terminal must be able to display both the text and the graphics planes simultaneously. The terminal type must be specified so that the correct plotting escape codes and DI-3000 device driver can be used. On the ELXSI, this specification is made through the EZPLOT shellfile (see app. A and ref. 1). Product addresses for Selanar, GraphOn, and Modgraph terminals are as follows: Selanar Corp., Santa Clara, California; GraphOn, Campbell, California; and Modgraph, Inc., Waltham, Massachusetts.

An ELXSI plotting session is begun by typing the command *ezplot* which tells the operating system to execute the EZPLOT shellfile. The shellfile responds with an invitation to select a graphics terminal type from a menu similar to the following:

```
EZPLOT Version 5.5

Friday, December 21, 1987 08:41:27 AM PST

The following terminals are supported:

    (T)ektronix 4014/4114
    (G)raphon
    (S)elanar
    (M)odgraph
    (4)109 Tektronix
    (V)T125

Enter your terminal type, or "Q" to quit:  (T/G/S/M/4/V/Q)
```

The expected response to this prompt is one of the single characters listed in parentheses followed by a carriage return. A *Q* causes the session to end immediately. A character other than *Q* is used by the shellfile to select the executable module containing the relevant DI-3000 device driver for loading. Also, a single word identifying the terminal type is assigned to a parameter called a *shell variable*. This shell variable is later accessed by the program through a call to the system-dependent subroutine $VAR (ref. 1) where the identifying word is assigned to a FORTRAN variable. This FORTRAN variable is used by the program to select the plotting escape codes that match the terminial. The only other meaningful response to this prompt is the phrase *no commands*. See the Redirection and Batch Operation section for further explanation of this response.

All EZPLOT input is case insensitive on machines supporting uppercase and lowercase. The commands can be entered in any mix of uppercase and lowercase.

Standard Commands

After the terminal type is specified, the EZPLOT shellfile passes control to the FORTRAN code. From this point on, the following standard commands may be used in response to any prompt where their meaning is appropriate.

If a particular command is not appropriate in a given context, the program responds with an informative message followed by the same prompt. Each standard command is entered on a line by itself and is followed by a carriage return. Standard commands are described in the following tabulation:

Command	Meaning	Description
<return>	Select default	A carriage return with no other entry causes a default value to be selected; defaults, where permitted, are identified with *Default* = as part of the prompt
.	Terminate input	A period terminates a series of prompts for information such as file names, parameter names, or data values; if the prompt is the first in a series, a period causes the previous prompt to be displayed
?	Help	A question mark causes a list of choices or other helpful information to be displayed
QU	End the program	*QU* causes program execution to end immediately. Control is returned to the EZPLOT shellfile
DO cmd	Execute *cmd*	*DO cmd* causes the host operating system command *cmd* to be executed during a session from within EZPLOT

The *DO cmd* command uses system-dependent features that may have to be modified for other host operating systems (refer to the PROGRAM TRANSMITTAL AND IMPLEMENTATION section).

Data Input Files

The next prompt following that for terminal type is for selection of data input format. EZPLOT recognizes two formats for data input files. These formats are called *flight data* and *generic data*. Data input format is selected in response to a menu that looks like the following:

```
Choose from the following:

F = Plot FLIGHT data
G = Plot GENERIC data
I = INPUT Generic data from the keyboard
U = View Ezplot USERS GUIDE

Your choice? (F/G/I/U)
```

At Ames-Dryden, flight data exist as a series of binary records on files generated by flight test data reduction computer programs. These data are in a time history format that can be read by the GetData program (ref. 3). Each record contains values for several (sometimes hundreds of) parameters measured at essentially the same time. The number of records on a file is a function of the data interval start and stop times and sampling rate.

Generic data is in a flexible and generalized format consisting only of ASCII records and may be read from existing files or may be entered from the keyboard. The generic data format is described in appendix A.

Data for up to a maximum of 2000 parameters may be read from input files.

This menu also provides the option to view the on-line *EZPLOT USERS GUIDE*. The on-line user's guide contains information similar to that contained in this user's manual, but in abbreviated form.

File and Parameter Names

The method of file and parameter selection is the same for both flight and generic data. After data input format is selected, the program prompts for the names of the files containing the data and the names of the parameters to be plotted. Any default names are given in parentheses after each prompt. The name of the first data input file is entered in response to the following prompt:

```
Input File 1?
```

Data for one plot may be read from as many as eight different data input files. Their names are entered in response to a series of prompts similar to that for the first file. If fewer than eight names are entered, the series of prompts is terminated with a period (refer to the Standard Commands section). A verification prompt is provided in case an error was made while entering file names.

File names may be any valid ELXSI pathname and may include patterns (ref. 1). Wild card features of patterns may not be used to select more than one file during any one series of prompts. If desired, a list of file names may be entered by typing *read [listfile]* at the prompt for the first data input file. The string *[listfile]* is replaced with the name of a separate file containing the names of the data input files. These names must be placed in *[listfile]* prior to the start of an EZPLOT session.

Regardless of how they were entered, the names of the data input files may be saved on a separate file for convenience when the data from the same files are to be plotted again. This is done by typing *write [listfile]* at the point where the program asks for verification of the list of data input files. *[listfile]* is the name of the separate file receiving the names. This action implicitly answers *yes* to the verification prompt.

After the data input file names have been entered, the program prompts for the name of the independent parameter as follows:

```
Independent parameter (X)?
```

The independent parameter and at least one dependent parameter must be specified before a plot can be drawn. The prompt for the name of the first dependent parameter is as follows:

```
Dependent parameter 1 (Y1)?
```

If only one data input file name was entered, as many as eight different parameters from that file may be plotted. Their names are entered in response to a series of prompts similar to that for the first dependent parameter. If fewer than eight names are entered, the series of prompts is terminated with a period (refer to the Standard Commands section). A verification prompt is provided in case an error was made while entering parameter names.

If more than one data input file name was entered, only one dependent parameter may be plotted. In this case, both the independent and the dependent parameter must be present on all the files.

The names of the parameters available for plotting may be displayed by entering a question mark in response to the prompt for independent or dependent parameter names. The program "remembers" parameters previously entered and displays their names as defaults for easy reselection if desired.

All of the characters in the name of an available parameter do not need to be entered to select that parameter. The program first tries to match the entered name exactly with the name of one of the available parameters. Leading and embedded blanks are significant. If an exact match is not successful, the program then attempts to match only as many characters as were entered with the same number of characters in one of the available names. This match

is always anchored at the left-most character. For example, if the entered name is *abc* and two available names are *abcd* and *dabc*, there is no exact match. However, *abc* does match the first three characters of *abcd*; therefore, *abcd* is selected. Parameter *dabc* is not selected. If the list of available parameters contains two or more names that match the entered name, the first name that matches is selected.

Plot Commands

After the names of the data input file or files and the independent and dependent parameters have been entered, the program prompts for a plot command as follows:

```
COMMAND?
```

A plot command is a two-character string that directs the program to take some type of action. Possible actions are to draw the plot, to prompt for further entry, to toggle an option on or off, to modify the input data or the characteristics of the plot, to display helpful information, and to quit the program.

Many plot commands display brief messages that describe the progress of program execution or suggest some appropriate action. These messages are largely self-explanatory and are not included in this manual. Messages generated because of an error or other unusual condition are explained in the Error and Warning Messages section.

A list of all plot commands along with a brief description of the action the program takes in response to each command may be viewed interactively by entering a question mark instead of a command. Similar lists are given below. Basic commands are those commands necessary to produce plots under a reasonable amount of control without requiring a detailed knowledge of the program. Extended commands provide more complex and flexible control over the plot along with several ways to manipulate the input data prior to plotting. Text commands provide a way to place explanatory data anywhere on the plot. The second character of each text command is an integer line number in the range of 1 to 9.

Any plot command may be entered when the `COMMAND?` prompt is displayed.

Basic commands are described in the following tabulation:

Name	Meaning	Action
FI	Figure	Prompts for plot figure label
LE	Legend	Turns line and symbol legend on or off
LI	Lines	Turns point-connecting lines on or off
LL	Legend label	Prompts for legend labels
ME	Menu	Displays this list (menu) of commands
ML	Move legend	Moves location of legend
ND	New data	Prompts for new data format
NF	New files	Prompts for new data files
NP	New parameters	Prompts for new parameters
PL	Plot	Draws the plot
QU	Quit	Ends (quits) the program
SA	Save	Saves the plot on a metafile[a]
SY	Symbols	Turns point symbols on or off
TI	Title	Prompts for plot title
XL	*x* label	Prompts for *x*-axis label
XR	*x* range	Prompts for *x*-axis minimum and maximum values
YL	*y* label	Prompts for *y*-axis label
YR	*y* range	Prompts for *y*-axis minimum and maximum values

[a]Precision Visuals, Inc., Boulder, Colorado.

Extended commands are described in the following tabulation:

Name	Meaning	Action
CF	Curve fit	Prompts for curve-fit specifications
DE	Delete	Prompts for box enclosing points to be deleted
EP	Exchange parameters	Exchanges the *x* and *y* parameters
EX	Exit	Ends (exits) the program
GI	GIN location	Reads GIN locations from screen or from a file
GR	Grid	Turns background grid on or off
GT	Grid type	Sets grid type to lines or dots
LO	Log scale	Turns logarithmic scale on or off
MA	Move axes	Moves location of coordinate axes
MS	Move statistics	Moves location of curve-fit statistics
NI	Number of intervals	Prompts for number of intervals along axes
PR	Precision	Sets maximum number of digits for tick label
RE	Replace	Prompts for box enclosing points to be replaced
RS	Restore	Restores deleted or replaced points
RX	Reverse *x*	Reverses *x*-axis minimum-to-maximum range
RY	Reverse *y*	Reverses *y*-axis minimum-to-maximum range
SC	Scaling	Sets or turns off data scaling
SE	Select	Prompts for line patterns and symbol shapes
SI	Size	Prompts for size of tick marks and label text
SM	Smoothing	Turns line smoothing on or off
SO	Sorting	Turns sorting on or off
SS	Start–stop	Displays flight data start and stop times
ST	Statistics	Sets or turns off curve-fit statistics legend
SU	Save userinput	Saves or deletes file *EZPLOT.USERINPUT*
SW	Switch	Reads commands from *$stdin* or file *EZPLOT.COMMANDS*
TH	Thinning	Sets or turns off thinning percentage
TR	Translation	Sets or turns off data translation
UN	Units	Sets or turns off unit conversions
WH	Where is GIN?	Displays location of GIN cursor
YA	*y* axes	Selects single or multiple *y* axes

Text Commands are described in the following tabulation:

Name	Meaning	Action
Mn	Move line *n*	Moves location of text line *n*
Pn	Path line *n*	Sets path and justification of text line *n*
Sn	Size line *n*	Sets size of characters in text line *n*
Tn	Text line *n*	Prompts for or turns off text line *n*

All labels and text lines are displayed on the plot in uppercase characters even though all entry is case insensitive. To delete the contents of a label or a line of text, enter a null character. On many terminals, a null character may

be entered by pressing the escape (ESC) key followed by the control (CTRL) key, which is held down while the space bar (SPACE) is pressed. This keystroke sequence is referred to in the following pages as ESC CTRL SPACE. Pressing only the space bar followed by a carriage return usually causes the program to respond as if only a carriage return had been entered.

Detailed descriptions of individual commands may be viewed by typing a question mark followed by the two-character command name. The text of these descriptions is given in the following listing in alphabetical order by command name.

CF (curve fit) — Command *CF* toggles the curve fitting option on or off. When *CF* is on, data for a single dependent parameter may be fitted with up to a ninth-order, least-squares polynomial curve. This curve is identified and treated by the program as a second dependent parameter (see fig. 5). When *CF* is off (the initial default value), no curve-fitting parameters are calculated or displayed even if they were present on a previous plot. This command is illegal when two or more curves are displayed.

DE (delete) — Command *DE* provides a way to delete from the plot selected groups of data points. After a plot is drawn and the *DE* command is selected, a rectangular box is drawn around the area containing the points to be deleted (see fig. 6). This box is defined by marking the screen locations of the lower left-hand and upper right-hand corners of the box as described in the Screen Locations section. These locations are called graphics input (GIN) locations. This command may be used repeatedly, without redrawing the plot, to define up to 1000 groups of points to be deleted. When the plot is redrawn, the points within the box or boxes do not appear. If the plotted points are connected with a line, that section of the line falling within the box is drawn linearly between the two points along the curve just outside the box boundaries. Deleted points may be restored with the *RS* command. Command *DE* is illegal when multiple *y* axes are displayed or when an axis is reversed or scaled logarithmically. See commands *GI* and *WH* for more information.

EP (exchange parameters) — Command *EP* causes the dependent parameter or parameters to be plotted along the horizontal *x* axis and the independent parameter to be plotted along the vertical *y* axis. Axes scales and labels are adjusted accordingly. The position and orientation of the plot title, figure title, text lines, and legend labels are not affected. Any points previously deleted or replaced are restored. When command *EP* is reentered, normal parameter orientation is restored. This command is illegal when multiple *y* axes are displayed.

EX (exit) — Command *EX* terminates program execution. Any plot currently displayed remains visible. All messages and prompts are erased. All files opened by the program are closed, and control is returned to the EZPLOT shellfile. The session ends when control is passed from the shellfile back to the host operating system. An *EX* may be entered in response to any prompt.

FI (figure) — Command *FI* toggles the figure label on or off. When *FI* is on, the program prompts for the contents of the figure label. This label is a line of up to 80 characters of text that is centered at the bottom of the plot below the *x*-axis label (see figs. 1 to 6). As part of the prompt, the current contents, if any, are displayed as the default. Press the ESC CTRL SPACE keys to enter a null character. When *FI* is off (the initial default value), the figure label does not appear.

GI (GIN location) — Command *GI* toggles between reading GIN locations from the screen or from a file. For interactive plotting, GIN locations are usually read from the screen in a terminal-dependent manner (see the Screen Locations section). They may also be read from the keyboard or from *EZPLOT.COMMANDS*. When read from the keyboard or a file, the locations must always be X,Y pairs where X and Y may be integer or real numbers and must be separated by a space or a comma. For batch execution, *$stdin* must be redirected, and GIN locations are always read from either this redirected file or from *EZPLOT.COMMANDS*. The initial default choice is to read GIN locations from the screen. See commands *SW* and *WH* for more information.

GR (grid) — Command *GR* toggles the background grid on or off. When *GR* is on, a background grid of solid or dotted lines aligned with the axes tick marks is displayed (see fig. 1). The grid line pattern is selected with the

GT command. When *GR* is off (the initial default value), no background grid is displayed. This command is illegal when multiple *y* axes are displayed.

GT (grid type) — Command *GT* causes either solid or dotted lines to be used for the background grid. The solid-line grid (the initial default value) is drawn very rapidly. However, the weight of the grid lines is the same as the weight of the axes and curve lines. A lighter weight grid can be drawn using dotted lines (see fig. 1). However, this grid is drawn quite slowly on some terminals and should be avoided when rapid plot display is a consideration, especially when there are more than ten tick marks on an axis. For a logarithmically scaled axis, the grid line at each major tick mark is solid even if the grid line at each minor tick mark is dotted (see fig. 2). This command is illegal when multiple *y* axes are displayed.

LE (legend) — Command *LE* toggles the line and symbol legend on or off (see figs. 1 and 4). If only one curve is plotted, the line and symbol legend does not appear even if *LE* is on. This command is illegal when multiple *y* axes are displayed.

LI (lines) — Command *LI* toggles the plotted lines on or off. When *LI* is on (the initial default value), the program connects the plotted data points with a solid line or lines made up of a variety of dashed patterns (see fig. 1). If two or more dependent parameters are plotted, line visibility can be turned on or off for all curves as a group or separately for each individual curve. When *LI* is off, no lines appear. When both *LI* and *SY* are off, a marker symbol is drawn at each data point.

LL (legend label) — Command *LL* prompts for the contents of line and symbol legend labels. When two or more curves are plotted, a legend is displayed containing identifying information for the line patterns and symbol shapes used. Each curve is described on a separate line by a legend label containing up to 80 characters of text (see figs. 1, 4, and 5). As part of the prompt, the current text for each line is displayed as the default. Press the ESC CTRL SPACE keys to enter a null character. By default, when data from different files are plotted, the labels initially contain the names of files. If ELXSI pathnames are used, only those parts of the pathnames that are the names of the files are put in the legend. When two or more dependent parameters are plotted, the labels initially contain the names of the dependent parameters. No legend is displayed if only one curve is plotted or if command *LE* is off. Command *LL* is illegal when multiple *y* axes are displayed.

LO (log scale) — Command *LO* turns the logarithmic scale on or off. Either the *x* axis or the *y* axis, or both may be scaled linearly or logarithmically. When *LO* is on, the axis is scaled logarithmically (see fig. 2). Because it is not possible to plot data values less than or equal to zero on a logarithmic scale, the program plots such data on a linear scale regardless of the value of *LO*. If multiple *y* axes are drawn, only the *x* axis can be scaled by *LO* even if only one dependent parameter is plotted. The effects of this command may be partially overridden by the *XR* and *YR* commands. When *LO* is off (the initial default value), the number of intervals and the minimum and maximum values displayed are determined by linear scaling. Under certain conditions, the effects of *LO* are unpredictable (see the Known Bugs section).

MA (move axes) — Command *MA* provides a way to move the location and size of the plotting area on the screen. After a plot is drawn and the *MA* command is selected, the GIN locations of the lower left-hand and upper right-hand corners of the desired plotting area are marked as described in the Screen Locations section. The GIN location chosen for the intersection of the *x* and *y* axes is also marked. The tick mark numbers may be placed along the axes or along the borders of the plotting area. When the plot is redrawn, it appears in the area and with the configuration just defined. Any points previously deleted or replaced are restored. Figure titles, legends, and text lines are not moved with this command. Command *MA* is illegal when multiple *y* axes are displayed. See commands *GI* and *WH* for more information.

ME (menu) — Command *ME* causes a list (menu) of all the two-character plot commands to be displayed. This list includes a brief description of the action the program takes in response to each command.

ML (move legend) — Command *ML* provides a way to move the location of the line and symbol legend. After a plot is drawn and the *ML* command is selected, the GIN location chosen for the upper left-hand corner of the legend is marked as described in the Screen Locations section. When the plot is redrawn, the legend appears in the new location. This command is illegal when multiple *y* axes are displayed. See commands *GI* and *WH* for more information.

MS (move statistics) — Command *MS* provides a way to move the location of the curve-fit statistical legend. After a plot is drawn and the *MS* command is selected, the GIN location chosen for the upper left-hand corner of the statistical legend is marked as described in the Screen Locations section. When the plot is redrawn, the legend appears in the new location provided command *ST* is on. Command *MS* is illegal when multiple *y* axes are displayed. See commands *GI* and *WH* for more information.

Mn (move line *n*) — Command *Mn* provides a way to move the location of the *n*th text line (where $n = 1$ to 9). After a plot is drawn and the *Mn* command is selected, the GIN location chosen for the center or left edge of the text line is marked as described in the Screen Locations section. When the plot is redrawn, the text line appears in the new location provided command *Tn* is on. See commands *GI* and *WH* for more information.

ND (new data) — Command *ND* provides for the selection of a new data input format. The program presents the menu that prompts for a choice of input format of the data to be plotted. These data may come from a flight data file, a generic data file, or the keyboard. A *U* entered in response to this menu causes the text of an on-line EZPLOT user's guide to be displayed. After the new data input format is selected, the program continues with the normal sequence of prompts for file and parameter names. This command is illegal when the independent and dependent parameters are exchanged.

NF (new files) — Command *NF* provides for the selection of new data input files. The program prompts for the name or names of files from which data are to be read. The previous files selected are presented as defaults. After the file names are entered, the program continues with the prompt for the names of the parameter to be plotted. This command is illegal when the independent and dependent parameters are exchanged.

NI (number of intervals) — Command *NI* provides a way to specify the number of tick-marked intervals along the axes. Values from 1 to 99 are allowed. The initial default values are 10 intervals along the *x* axis and 8 intervals along the *y* axis (see fig. 1). When multiple *y* axes are used, there are two intervals for each dependent parameter even if only one dependent parameter is plotted. Command *NI* is illegal for multiple *y* axes and logarithmically scaled axes.

NP (new parameters) — Command *NP* provides for the selection of new parameters. The program prompts for the names of the parameters to be plotted. The previous names selected are presented as defaults. If a default parameter is selected containing points that were previously deleted or replaced, these points are restored. Command *NP* is illegal when the independent and dependent parameters are exchanged.

PL (plot) — Command *PL* causes a plot to be drawn on the screen. All text and graphics information currently displayed are erased prior to drawing the plot. The program does not automatically draw a plot after the dependent parameter or parameters have been chosen. This allows for selection of other than default values for various plot attributes before drawing the first in a series of plots.

PR (precision) — Command *PR* prompts for the number of digits to be used to label each tick mark. Values from 5 to 20 are allowed. The initial default value is 8. A label may interfere with an adjacent label if too many digits are used.

Pn (path line *n*) — Command *Pn* prompts for the character path and justification of text line *n*. Path, the directional orientation of the line, may be horizontal or vertical. Horizontal lines read from left to right. Vertical lines read from bottom to top. Justification, the location within the line used to position the line on the plot, may be the left (or bottom) edge or the center.

QU (quit) — Command *QU* terminates program execution. All plots, messages, and prompts currently displayed are erased. All files opened by the program are closed, and control is returned to the EZPLOT shellfile. The session ends when control is passed from the shellfile back to the host operating system. A *QU* may be entered in response to any prompt.

RE (replace) — Command *RE* provides a way to replace selected groups of data points. After a plot is drawn and the *RE* command is selected, a rectangular box is drawn around the area containing the points to be replaced (see fig. 6). This box is defined by marking the screen locations of the lower left-hand and upper right-hand corners of the box as described in the Screen Locations section. This command may be used repeatedly, without redrawing the plot, to define up to 1000 groups of points to be replaced. When the plot is redrawn, the points within the box or boxes are replaced with an equal number of points lying along a straight line connecting the two points along the curve just outside the box boundaries. Replaced points may be restored with the *RS* command. Command *RE* is illegal when multiple *y* axes are displayed or when an axis is reversed or scaled logarithmically. See commands *GI* and *WH* for more information.

RS (restore) — Command *RS* causes all points that have been deleted or replaced to be reinserted at their original location on the plot. This command is illegal when multiple *y* axes are displayed.

RX (reverse x) — Command *RX* causes the current direction for plotting minimum to maximum values along the *x* axis to be reversed. The data may be plotted with the minimum value on the left and the maximum value on the right, or conversely. By default, the minimum to maximum values are plotted from left to right. This command is illegal for a logarithmically scaled axis.

RY (reverse y) — Command *RY* causes the current direction for plotting minimum to maximum values along the *y* axis to be reversed. The data may be plotted with the minimum value on the bottom and the maximum value on the top, or conversely. By default, the minimum to maximum values are plotted from bottom to top. This command is illegal for a logarithmically scaled axis.

SA (save) — Command *SA* causes the current plot to be saved on a metafile. The action of this command is fully described in the Metafiles section.

SC (scaling) — Command *SC* toggles separate scaling (multiplying) factors on or off. When *SC* is on, the independent and dependent parameter data values are multiplied by factors entered from the keyboard. If two or more dependent parameters are plotted, scaling can be turned on or off for all curves as a group or separately for each individual curve. When *SC* is off (the initial default value), no scaling is done. When this command is used, any points previously deleted or replaced are restored. Scaling is turned off when command *ND*, *NF*, or *NP* is entered.

SE (select) — Command *SE* prompts for line pattern and symbol shape choice. By default, when only one curve is plotted, the line is solid and the symbol is a circle. When two or more curves are plotted, the second and subsequent lines are dash–patterned if no symbols are drawn (see fig. 1). If symbols are drawn, the lines are solid and the symbols are variously shaped (see figs. 2 and 4). The default order of patterns and shapes may be overridden. The number of the line pattern and the symbol shape correspond to the number of the curve and not to the parameter name. That is, line pattern 1 is used to draw the first curve, line pattern 2 is used to draw the second curve, and so on. Any line pattern or symbol shape may be selected repeatedly and used to draw more than one curve. Visibility of lines and symbols is determined by the value of the *LI* and *SY* commands. When both *LI* and *SY* are off, a marker symbol is drawn at each data point.

SI (size) — Command *SI* prompts for the horizontal and vertical size of the characters in the *x* axis, *y* axis, plot title, figure, legend, statistics, and tick-mark labels. Each size must be in the range 0.001 to 100.0. The default size of the plot title and figure label is 2.0, 2.0. The default size of the other labels is 1.5,1.5.

SM (smoothing) — Command *SM* toggles line smoothing on or off. When *SM* is on, a smoothed (curved) line is drawn between points using an algorithm called splines under tension (see ref. 2). If two or more dependent parameters are plotted, smoothing can be turned on or off for all curves as a group or separately for each individual

curve. When *SM* is off (the initial default value), a straight line is drawn between successive data points. This command is effective only if command *LI* is on and sometimes fails if there are more than about 50 data points (see the Known Bugs section).

SO (sorting) — Command *SO* toggles sorting on or off. When *SO* is on, all subsequent input data are sorted by increasing value of the independent parameter before any plots are drawn. If the data have been previously thinned, the first and last points may not represent the minimum and maximum values, respectively, in the unthinned data set. Sorting is most useful when lines are displayed. When *SO* is off (the initial default value), no sorting is done.

SS (start–stop) — Command *SS* causes the start and stop times for the data interval on a flight data input file to be displayed as `[hh:mm:ss.mil]`.

ST (statistics) — Command *ST* provides for selection of what statistical data to display on the plot. When the curve–fit command *CF* is enabled, the mean deviation, variance, and standard deviation are calculated and may be displayed on the text plane, the graphics plane, or both (see fig. 5). Also displayed is the form of the equation and the value of the coefficients. For example, `y = AX + BX2 + CX3 + ...`

SU (save userinput) — Command *SU* toggles the option to save file *EZPLOT.USERINPUT* on or off. All commands and other information entered during program execution are written on a temporary file. When *SU* is on, the contents of this temporary file are copied to file *EZPLOT.USERINPUT* which is saved in the current directory after the temporary file is deleted and program execution stops. When *SU* is off (the initial default value), neither file is saved. For jobs run in batch mode, the name *EZPLOT.USERINPUT* is prefixed automatically with the characters *j[jobID]*. where *[jobID]* is the job identification number.

SW (switch) — Command *SW* toggles between *$stdin* (usually the keyboard) and file *EZPLOT.COMMANDS* for command entry. At startup, after the terminal type has been entered, the program attempts to read and execute commands from file *EZPLOT.COMMANDS* in the current directory. If this file is empty or an *SW* command is encountered, the next command is read from *$stdin*. If an *SW* is encountered on *$stdin* (for example, entered from the keyboard), the program attempts to read the next command from *EZPLOT.COMMANDS*. This process continues until a *QU* or an *EX* command is read and executed from either source. *$stdin* may be redirected from the keyboard to a file.

SY (symbols) — Command *SY* toggles the plotted symbols on or off. When *SY* is on, symbols with a variety of shapes are drawn at the plotted points (see fig. 4). If two or more dependent parameters are plotted, symbol visibility can be turned on or off for all curves as a group or separately for each individual curve. When *SY* is off (the default value), no symbols are drawn. When both *LI* and *SY* are off, a marker symbol is drawn at each data point.

Sn (size line *n*) — Command *Sn* prompts for the horizontal and vertical sizes of the characters in text line *n*. Each size must be in the range 0.001 to 100.0. The default size is 2.0, 2.0.

TH (thinning) — Command *TH* toggles data point thinning on or off. When *TH* is on, the program prompts for the percentage of data points to be plotted out of the total number available. This percentage may vary from 1 to 100, where 100 percent means to plot all the points. Point deletion is evenly distributed along the curve. The first and last points are always retained. When *TH* is off (the initial default value), all available points are plotted.

TI (title) — Command *TI* toggles the plot title on or off. When *TI* is on, the program prompts for the contents of the plot title. This title is a line of up to 80 characters of text that is centered at the top of the plot above the plotting area (see fig. 1). As part of the prompt, the previous contents, if any, are displayed as the default. Press the ESC CTRL SPACE keys to enter a null character. When *TI* is off (the initial default value), the plot title does not appear.

TR (translation) — Command *TR* toggles separate translation (bias) factors on or off. When *TR* is on, the independent and dependent parameter data values are biased by factors entered from the keyboard. If two or more dependent parameters are plotted, translation can be turned on or off for all curves as a group or separately for each individual curve. When *TR* is off (the initial default value), no translation is done. When this command is used,

any points previously deleted or replaced are restored. Translation is turned off when command *ND*, *NF*, or *NP* is entered.

Tn (text line *n*) — Command *Tn* toggles the *n*th text line on or off. When *Tn* is on, the program prompts for the contents of the line. Up to nine lines containing up to 80 characters of text each may be displayed on a plot (see fig. 1). As part of the prompt, the current contents, if any, are displayed as the default. Press the ESC CTRL SPACE keys to enter a null character. When *Tn* is off (the initial default value), line *n* does not appear.

UN (units) — Command *UN* toggles unit conversion factors on or off. When *UN* is on, the independent and dependent parameter data values are scaled (multiplied) and translated (biased) as follows: adjusted data value = $A *$ (unadjusted data value) + B where A and B are factors entered from the keyboard. If two or more dependent parameters are plotted, unit conversion can be turned on or off for all curves as a group or separately for each individual curve. When *UN* is off (the initial default value), no unit conversion is done. This command makes use of the same factors and operates in the same way as commands *SC* and *TR* combined. When command *UN* is used, any points previously deleted or replaced are restored. Unit conversion is turned off when command *ND*, *NF*, or *NP* is entered.

WH (where is GIN?) — Command *WH* provides a way to display the present location of the GIN cursor. After a plot is drawn and command *WH* is selected, the GIN cursor is moved to the location to be displayed in a terminal-dependent manner (see the Screen Locations section). World coordinates and engineering units are both shown when the location is within the plot axes. When the location is outside the plot axes, only world coordinates are shown. When multiple *y* axes are displayed, the *y* location is shown in world coordinates only.

XL (*x* label) — Command *XL* toggles the *x*-axis label on or off. When *XL* is on (the initial default value), a line of up to 80 characters of text appears centered below the *x*-axis tick-mark numbers (see figs. 1 to 6). The name of the independent parameter is the initial default contents of the *x*-axis label. If the label is changed, the new contents remain the default until a subsequent change or until a new independent parameter is selected. The name of the new independent parameter then becomes the default. Press the ESC CTRL SPACE keys to enter a null character. When *XL* is off, this label does not appear.

XR (*x* range) — Command *XR* prompts for the minimum and maximum values to be displayed on the *x* axis. Current minimum and maximum values are displayed as defaults. The initial minimum and maximum values are calculated automatically to allow for all the input data to be plotted. The effect of changing these values is to increase or decrease the *x*-axis range. This is useful for "windowing" the data or for creating a more suitable interval between tick marks. Data that subsequently fall outside the plot borders are not displayed. Because the results of this command may be affected by command *LO*, windowing on a logarithmically scaled axis should be done after *LO* is selected. Command *XR* may then be used to change the number of intervals on the logarithmic axis, if desired. The *x*-axis scale is automatically recalculated as required any time a *ND*, *NF*, or *NP* command is selected.

YA (*y* axes) — Command *YA* provides for selection of single or multiple *y* axes. When multiple axes are chosen, there is one *y* axis for each of up to eight dependent parameters (see fig. 3). Each *y* axis is scaled independently from the others. If only one dependent parameter is plotted, there is only one *y* axis. There is always only one *x* axis. Commands *DE*, *EP*, *GR*, *GT*, *LE*, *LL*, *MA*, *ML*, *MS*, *RE*, and *RS* are illegal when multiple *y* axes are displayed.

YL (*y* label) — Command *YL* toggles the *y*-axis label on or off. When *YL* is on (the initial default value), the label is displayed. The form of this label varies according to the number of *y* axes displayed and the number of dependent parameters plotted. For a single *y* axis and one dependent parameter, this label is a line of up to 80 characters of text that appears centered to the left of the *y*-axis tick-mark numbers (see figs. 2 and 6). The text reads from bottom to top with the characters rotated 90° counterclockwise with respect to the *x*-axis label. For a single *y* axis and more than one dependent parameter, the label is optional and a line and symbol legend is drawn (see figs. 1, 4, and 5). For more than one *y* axis, there is one label for each axis plus an additional label for all *y* axes together (see fig. 3). Labels for individual *y* axes should not be longer than the relevant axis to avoid interference with adjacent labels.

For all cases, the initial default content of this label is the name or names of the dependent parameter or parameters if all the data come from one data input file, or the names of the files when there is more than one data input file. If the label is changed, the new contents remain the default until a subsequent change or until a new dependent parameter or a new data input file is selected. The name or names of the new dependent parameter or parameters or data input files then become the default. Press the ESC CTRL SPACE keys to enter a null character. When *YA* is off, this label does not appear.

YR (*y* range) — Command *YR* prompts for the minimum and maximum values to be displayed on the *y* axis. Current minimum and maximum values are displayed as defaults. The initial minimum and maximum values are calculated automatically to allow for all the input data to be plotted. The effect of changing these values is to increase or decrease the *y*-axis range. This is useful for "windowing" the data or for creating a more suitable interval between tick marks. Data that subsequently fall outside the plot borders are not displayed. Because the results of this command may be affected by command *LO*, windowing on a logarithmically scaled axis should be done after *LO* is selected. Command *YR* may then be used to change the number of intervals on the logarithmically scaled axis, if desired. The *y*-axis scale is automatically recalculated as required any time a *ND, NF,* or *NP* command is selected. When multiple *y* axes are displayed, the range of each axis can be specified separately.

Screen Locations

Plot Commands *DE, MA, ML, MS, Mn, RE* and *WH* make use of the DI-3000 virtual graphics input function *LOCATOR* (ref. 2) to provide for specification of screen locations. These locations are often specified in a terminal-dependent manner. Many graphics terminals can display a crosshair of intersecting lines that can be moved across the screen until their intersection indicates the position on the screen to be specified. For most terminals, the lines are moved by pressing the arrow keys on the keyboard. Pressing the shift key in conjunction with an arrow key may speed up the motion. Some terminals have more than one set of arrow keys. If one set produces no response, try the other set. For Tektronix 4107 and 4109 terminals, a joydisk on the keyboard is used instead of arrow keys. Other terminals may use a joystick, thumbwheels, a light pen, or a mouse to move the lines or some other marker.

When the lines intersect at the desired location, the space bar is pressed once or twice, depending on the terminal, to specify or "mark" this position. On some terminals, other keys or buttons may be used instead of the space bar. If the COMMAND? or some other prompt appears immediately after the space bar is pressed once, there is no need to press it again. If the lines disappear and a blinking cursor appears over the selected location but no prompt appears, the space bar must be pressed one more time. This action transmits the display coordinates of the location to the program. These coordinates are also briefly displayed on the screen.

Screen locations may also be read from a file as pairs of numbers representing X,Y display coordinates (see commands *GI* and *WH*). The effect within the program is the same as when the locations are entered directly from the screen. The previous coordinates of the screen location do not change if an error occurs when the numbers are read.

Metafiles

Plots may be saved during program execution on DI-3000 metafiles (ref. 5). Each plot on a metafile is called a picture and is stored as a series of records containing digitized plot information that can be interpreted by DI-3000 Metafile Translator utilities. After a session ends, plots saved on a metafile may be reviewed on the screen or plotted on an auxiliary plotter without reexecuting EZPLOT.

Plots are saved with the *SA* command and have the attributes in effect when the command is executed. To speed the process, plots are not drawn to the screen when saved. Therefore, it is often a good idea to review the plot prior to saving it by entering a *PL* command just before a *SA* command. For the first plot (picture), a name for the metafile

is entered followed by a name (title) for the picture. File names may contain up to 24 characters, while picture titles may contain up to 50 characters. For the second and following pictures, only the picture title is entered. Pictures are stored sequentially on a metafile. During any one EZPLOT session, only one metafile is generated. Therefore, if plots are to be stored on more than one metafile, the current session must be terminated with the plot command *QU* and a new session begun with the ELXSI command *ezplot*.

Metafile pictures may be drawn with a DI-3000 Metafile Translator linked with the appropriate device driver. The device drivers available on the ELXSI may be listed by entering the command *ListGraphicsDrivers* in response to the normal EMBOS prompt.

Here is an example. After the ELXSI command *metrans.t14* is entered, where *t14* is the ELXSI device driver abbreviation for graphics terminals that emulate the Tektronix 4014, the Metafile Translator prompt `M>` is displayed. The following command sequence is then entered in response to a prompt for each line. The shorter form of each command is shown in parentheses. *MF.XY* is the name of the metafile.

Set metafile 1 MF.XY	*(S MF 1 MF.XY)*
Directory 1	*(DIR 1)*
Draw picture 2	*(D P 2)*
Draw picture 3	*(D P 3)*
Draw picture 4	*(D P 4)*
Quit	*(Q)*

In this example, three separate plots are displayed as the *draw picture* command is executed for the second, third, and fourth picture on the file. The first picture is always blank and should not be drawn. The space bar must be pressed once or twice after each picture is drawn to redisplay the Metafile Translator prompt. Other keys or buttons may be used instead of the space bar depending on the terminal. The *directory* command displays the titles of the pictures on the file and may be omitted.

Metafiles saved in the current interactive session are corrupted if command *EX* is used to end execution. Command *QU* should be used in this situation. For batch execution, command *EX* is equivalent to *QU* and does not corrupt files saved with command *SA*.

An alternative way to save plots is described in appendix C.

Redirection and Batch Operation

EZPLOT is normally run interactively by entering commands and responses from the keyboard. An alternative way to run the program is to put these keyboard entries into a file, one entry per record, and redirect *$stdin* to this file. For example, the command line *ezplot.cmd > ezplot* directs the EZPLOT shellfile to take entries from file *ezplot.cmd* instead of the keyboard. This is a convenient way to store and use what may sometimes be a lengthy set of keyboard entries to draw a series of similar plots.

Here is a sample of what a typical file named *ezplot.cmd* might contain. The following comments in parentheses are given here for explanation and are not part of the records on the file.

g	select Graphon terminal type
g	(select generic data)
zgen	(name of file containing generic data)
.	(standard command, end file name entry)
y	(yes, names are correct)
x	(name of independent parameter)
y	(name of dependent parameter 1)
z	(name of dependent parameter 2)
w	(name of dependent parameter 3)
.	(standard command, end parameter name entry)
y	(yes, names are correct)
GR	(turn on grid)
PL	(draw plot)
EX	(end execution)

This procedure does not allow for any keyboard entry after the program has started execution. When all commands and responses have been processed, the normal ELXSI prompt reappears and the plot remains on the screen. To generate another plot, the entire sequence of entries is processed. Be sure that the file does not contain a *QU* command. If it does, the plot is erased when the ELXSI prompt reappears.

If a file named *EZPLOT.COMMANDS* is present in the current directory when this sample job is executed, the operation described is modified slightly. After the terminal type is read from *ezplot.cmd*, the next response is read from file *EZPLOT.COMMANDS*. If this response is the command *SW*, all commands and responses that remain on *ezplot.cmd* are read and executed. If this response is not the command *SW*, the program continues to read and execute commands and responses from *EZPLOT.COMMANDS* until an *SW* command or an end of file is reached. All commands and responses remaining on *ezplot.cmd* are then read and executed.

If the current directory contains an *EZPLOT.COMMANDS* file that should not be used and that cannot be deleted, enter *no commands* in response to the prompt for terminal type at program initiation. When the prompt is repeated, enter the appropriate terminal type to begin execution.

To help the user create files similar to *ezplot.cmd*, all commands and responses, including GIN locations entered through the keyboard during an interactive session, are written on a file named *EZPLOT.USERINPUT*. This file may be saved, if desired, and used as the basis of an *ezplot.cmd* file. See command *SU* for more information.

This procedure may also be executed in a noninteractive batch mode, which, on the ELXSI computer, runs independently from any user input. Therefore, *$stdin* must be redirected to a file that contains all the information normally entered from the keyboard to generate a plot, including GIN locations, where appropriate. The plot is not drawn on the screen and should be saved on a metafile to be plotted at a later time.

If no *EZPLOT.COMMANDS* file exist in a directory, any number of users may execute any combination of interactive and batch EZPLOT jobs in that directory at the same time. Be aware, however, that metafiles and file *EZPLOT.USERINPUT* previously written by one user may be overwritten by another user at some later time.

If an *EZPLOT.COMMANDS* file does exist in a directory, only one interactive job may be executed in that directory at any given time. If no interactive jobs are executing, any number of batch jobs may be executed at the same time with the understanding that they all use the same *EZPLOT.COMMANDS* file.

Error and Warning Messages

EZPLOT messages.—If an error or other abnormal condition is detected by EZPLOT, a message is displayed identifying the problem. The text of each error and warning message follows in alphabetical order.

`CANNOT FIT LESS THAN 2 POINTS` — This message, generated by subroutine CURFIT, indicates that an attempt was made to fit a line through only one data point.

`CANNOT OPEN FILE "xxxxx"` — This message, generated by subroutines FILES and GENRIC, indicates that a selected file could not be opened for access to input data.

`CANNOT WRITE IN THIS DIRECTORY` — This message, generated by subroutines FILES and SAVEPL, indicates that the user does not have permission to write a file in the current directory.

`CANNOT WRITE METAFILE WHEN USING PLOTFILE` — This message, generated by subroutine USER, indicates that an attempt was made to write plot data to a metafile when the shell variable *PlotFile* was active (see app. C).

`EZPLOT DETECTS ERROR AT OR NEAR RECORD nnnnn, FILE "xxxxx"` — This message, generated by subroutine GENRIC, indicates a problem with the format of generic data input file *xxxxx*. Remember that comments, if present, are record number 1.

`FILE "xxxxx" DOES NOT EXIST` — This message, generated by subroutine FILES, indicates that a file named *xxxxx* does not exist in the specified EMBOS directory, usually the current directory. Common causes are spelling errors and incomplete or incorrect pathnames.

`FIRST CHARACTER MAY NOT BE A NUMBER` — This message, generated by subroutine HELP, indicates that help information was requested about a string that began with a number. This is illegal.

`ILLEGAL COMMAND` — This message, generated by subroutine USER, indicates that a two-character string has been entered that does not match any of the legal commands.

`ILLEGAL COMMAND: LONGER THAN 2 CHARACTERS` — This message, generated by subroutine USER, indicates that a string of more than two characters was entered as a plot command.

`ILLEGAL GENERIC DATA VALUE` — This message, generated by subroutine GENRIC, indicates that an illegal value was encountered in the data on a generic file. A common cause is the presence of a nonnumeric character in the data values—except for *E* as part of an exponential number, which is permitted. Another possible cause is a number too large or too small for single-precision arithmetic. If the file is formatted, the line containing the error is displayed.

`ILLEGAL INPUT` — This message indicates that an illegal string has been entered from the keyboard. The specific problems that might be encountered are described here for each subroutine that generates this message.

NUMINT — An illegal value has been entered for the number of tick marks along an axis. Legal values are from 1 to 99.

PRECIS — An illegal value has been entered for the number of digits used to label each tick mark. Legal values are from 5 to 20.

SCALER, TRANSL, UNITS, XRANGE, and YRANGE — A nonnumeric character has been used in specifying a numerical value. Legal entries must be numbers.

SIZLBL and SIZTXT — An illegal value has been entered for the size of the characters in a label or a line of text. Legal values are from 0.001 to 100.

STATS — A noninteger or an integer other than from 1 to 5 has been entered when choosing which statistical information to display.

THIN — An illegal value has been entered for the thinning factor. Legal values are from 1 to 100.

`ILLEGAL INPUT (ENTER "?" FOR HELP)` — This message, generated by subroutine FILES, FLIGHT, GETDAT, GETGEN, and GINPUT indicates that an illegal entry, usually an inappropriate character or a misspelled name, has been made from the keyboard. A question mark causes helpful information to be displayed.

`ILLEGAL NUMBER OF PARAMETERS` — This message, generated by subroutine GENRIC, indicates that the value of the integer specifying the number of parameters on a generic file was less than one. If the file is formatted, the illegal value is displayed.

`ILLEGAL PARAMETER NAME` — This message, generated by subroutine GENRIC, indicates that the syntax of a parameter name on a generic file is incorrect. Look for missing or extra apostrophes in the list of names. Also, the number of parameter names may not match the value of the integer specifying how many names to read. If the file is formatted, the line containing the error is displayed.

`ILLEGAL WITH EXCHANGED PARAMETERS` — This message, generated by subroutine USER, indicates that a request for new data, new files or new parameters was entered when the independent and dependent parameters were exchanged. Commands *ND, NF*, and *NP* are illegal in this situation.

`ILLEGAL WITH LOGARITHMIC SCALE` — This message, generated by subroutine USER, indicates that a request was entered to reverse the direction of plotting along a logarithmically scaled axis or to delete or replace a group of data points when one or both axes are scaled logarithmically. Commands *DE, RE, RX*, and *RY* are illegal in these situations.

`ILLEGAL WITH MORE THAN ONE CURVE` — This message, generated by subroutine USER, indicates that a curve fit has been requested when more than one curve is displayed. The command *CF* is illegal in this situation.

`ILLEGAL WITH MULTIPLE Y AXES` — This message, generated by subroutine USER, indicates that a command has been entered that cannot be executed when multiple *y* axes are displayed. For this situation, the following commands are illegal: *DE, EP, GR, GT, LE, LL, MA, ML, MS, RE,* and *RS*.

`ILLEGAL WITH REVERSED AXIS` — This message, generated by subroutine USER, indicates that a request has been entered to impose a logarithmic scale on an axis along which the direction of plotting is reversed, or to delete or replace a group of data points when the direction of plotting is reversed on one or both axes. Commands *DE, RE*, and *LO* are illegal in these situations.

`MATRIX IS EFFECTIVELY SINGULAR. TRY AGAIN` — This message, generated by subroutine CURFIT, indicates that insufficient data were supplied to perform the desired curve fit. The curve fit option is turned off.

`MAXIMUM NUMBER OF DATA POINTS READ ON FILE "xxxxx"/REMAINDER WILL BE IGNORED` — This message, generated by subroutines FLIGHT and GENRIC, indicates that there were more than 10,000 data points on file *xxxxx*.

`MAXIMUM NUMBER OF PARAMETERS REACHED ON FILE "xxxxx"/ONLY THE FIRST "nnnn" PARAMETERS WILL BE AVAILABLE` — This message, generated by subroutines FLIGHT and GENRIC, indicates that the input file(s) contain data for more than 2000 parameters. This limit was reached while reading file *xxxxx*.

`NO MORE COMMANDS ON FILE "EZPLOT.COMMANDS"; USE KEYBOARD TO ENTER COMMANDS` — This message, generated by function READSW, indicates that an end of file has been reached on file *EZPLOT.COMMANDS*. Either the file is empty or all commands on it have been executed. All further commands must be entered from the keyboard.

`NUMBER OF DIGITS MUST BE BETWEEN 5 AND 20` — This message, generated by subroutine PRECIS, indicates that an illegal value has been entered for the number of digits to be written in the label for each axis tick mark. Legal values are from 5 to 20.

`NUMBER OF INTERVALS MUST BE BETWEEN 1 AND 99` — This message, generated by subroutine NUMINT, indicates that an illegal value has been entered for the number of tick marks along an axis. Legal values are from 1 through 99.

`NO PARAMETER NAMES READ; CHECK FILES` — This message, generated by subroutines FLIGHT and GENRIC, indicates that the parameter names on a data input file cannot be read. A common cause is the lack of an integer on a line by itself specifying the number of parameters on a generic file.

`"OPENR" FAILED TRYING TO OPEN FILE "xxxxx"` —This message, generated by subroutine FLIGHT, indicates that file *xxxxx* could not be opened for access to input data.

`ORDER MUST BE 1` — This message, generated by subroutine CURFIT, indicates an attempt was made to fit something other than a straight line through two data points.

`ORDER MUST BE BETWEEN 1 AND n` — This message, generated by subroutine CURFIT, indicates the current limits on the order of curve fit allowed. The maximum order of fit allowed may be less than nine if the number of data points is small.

`PARAMETER "yyyyy" NOT FOUND/ENTER "?" TO DISPLAY AVAILABLE PARAMETERS` — This message, generated by subroutine FLIGHT and GETGEN, indicates that the parameter named *yyyyy* is not available to be plotted. A question mark causes the names of available parameters to be displayed. If there are two or more data files, only the names of the parameters available on the first file are displayed.

`PARAMETER "yyyyy" NOT FOUND ON FILE "xxxxx"/WILL PLOT WHAT'S AVAILABLE` — This message, generated by subroutine GETGEN, indicates that when plotting from two or more files, the parameter name *yyyyy* was found on at least one file but was not found on file *xxxxx*. The data that was found is plotted. Logarithmical scaling is inhibited for parameter *yyyyy*. The contents of file *xxxxx* may be listed and checked with the standard command *DO cmd*.

`SIGNAL yyyyy NOT FOUND. USING 0.` —This message, generated by a GetData subroutine, indicates that parameter name *yyyyy* cannot be found on the data input file. Be sure the requested name is spelled correctly and exists on the data input file.

`SLOPE IS INFINITE; CANNOT FIT` — This message, generated by subroutine CURFIT, indicates an attempt was made to fit a straight line between two points with the same value of independent variable.

`THAT'S NOT LEGAL HERE` —This message, generated by various subroutines, indicates that a period has been entered at an inappropriate step during program execution.

`THERE IS NO DEFAULT` — This message, generated by subroutines FILES, FLIGHT, and SAVEPL, indicates that a default file or parameter name is not available at this step in program execution. A name must be entered from the keyboard.

`TOPIC NOT FOUND` — This message, generated by subroutine HELP, indicates that help information was requested about a string that cannot be located anywhere in the command menu text.

`"sssss" IS NOT A 2-LETTER COMMAND; WILL SEARCH FOR TOPIC` —This message, generated by subroutine HELP, indicates that help information was requested about a string that is not a two-letter EZPLOT command. Every line in the command menu that contains string *sssss* is displayed.

`YOU MUST ENTER AN INTEGER HERE` — This message, generated by subroutine GINPUT, indicates that an attempt was made to enter something other than an integer as the number of parameters to be input in data format "I".

`YOU MUST ENTER AT LEAST 2 AND NO MORE THAN n PARAMETERS` — This message, generated by subroutine GINPUT, indicates an attempt was made to enter less than two or more than *n*, the maximum number of curves that can be plotted, as the number of parameters to be input in data format "I". Currently, $n = 8$.

`YOU MUST ENTER DIGITS IN THE RANGE 1 THROUGH 8` — This message, generated by subroutine SELECT, indicates that a 9, a 0, or a noninteger has been entered when specifying line patterns or symbol styles. Legal values are integers from 1 to 8.

`WARNING: PICTURE TITLE TRUNCATED TO THE FIRST 50 CHARACTERS` — This message, generated by subroutine SAVEPL, indicates that the text entered as the title of a metafile picture contained more than 50 characters.

Operating system messages.—Errors and other abnormal conditions detected by the operating system may send messages to the screen. Errors detected by EMBOS while trying to execute a *DO cmd* cause a system-generated message to be displayed but usually do not cause a program session to end. Corrective action depends on the nature of the error and is often system dependent. Irrecoverable errors such as those caused by floating-point overflow often display a message beginning with the line `EXCEPTION #nnn` followed by several additional lines of information. These errors may cause the program session to end immediately. Corrective action usually involves selection of different parameters to be processed or modification of the input data.

Known Bugs

When there are more than about 1000 data points, the *SM* command sometimes fails during execution due to a floating-point overflow error. This error causes an EZPLOT session to end immediately. Therefore, until it has been tested for error-free execution, command *SM* should not be used when many plots are being generated and saved on a metafile.

When data from more than one file are plotted and one or both of the selected parameters cannot be found on all the files, the associated axes cannot be scaled logarithmically even if no values are less than, or equal to, zero.

No message is generated to indicate a mismatch between the requested data format type and the data actually on the file. For example, when flight data are requested and a file containing generic data is selected, the only indication of an error is that no parameters are available to be plotted.

Tips for Efficient Usage

Here are some guidelines for using EZPLOT easily and efficiently.

1. Plotting several thousand points for each of several parameters can be quite time consuming even at baud rates of 9600. Thinning may be used to significantly reduce the plotting time.

2. Because the dot grid is drawn relatively slowly, it should not be used when maximum screen plotting speed is desired. The line grid is drawn rapidly and usually does not require any significant waiting time.

3. Curve fitting more than 1000 points can be very slow. Polynomials of order greater than six are not recommended because of long execution times and uncertain results.

4. When many plots are being generated and saved on a metafile, the program session should be ended with the *QU* command every 10 to 15 plots. This helps safeguard the results of a lengthy program session in case an error occurs that ends the session prematurely (see Operating System Messages and Known Bugs sections on the previous page).

5. Before metafile or data file names are chosen, the *DO cmd* may be used to list the names of files that already exist. This helps avoid writing on files that should not be changed.

6. The *DO cmd* may be used to execute programs from within EZPLOT. This allows for new data to be generated and plotted during a session.

7. Several types of help are available in response to the COMMAND? prompt. A ? by itself displays a list and brief description of all two-letter plot commands. This list (menu) is also displayed in response to the *ME* command. A ? followed by a two-letter plot command displays a more detailed description of that individual command. A ? followed by a one-word topic displays all the lines in the list (menu) of plot commands containing that topic. A ? entered in response to any other prompt displays information relevant to that particular prompt.

PROGRAM TRANSMITTAL AND IMPLEMENTATION

The EZPLOT FORTRAN source code and the text of the online user's guide and help files will be transmitted in the following format unless explicit instructions request otherwise. The program will be sent on a nine-track, 1600 BPI unlabeled tape. The tape will contain ASCII-coded card images. Each card image will be a fixed-length, 80-character record. Records will be blocked in fixed-length, 800-character blocks. Each block will contain 10 records with no padding. Records will not span blocks. The program contains about 10,000 lines of code. The online users's guide and help files together contain about 1000 lines of text.

DI-3000 source code is not available from the government. GetData source code is available from the government (see ref. 3) but will not be included with the EZPLOT source code.

EZPLOT was developed and implemented in single precision on an ELXSI 6400 32-bit mainframe computer. Appendix D contains a brief description of the main program and all subprograms referenced explicitly in EZPLOT except DI-3000 subprograms. File assignments are described in appendix E. The FORTRAN code uses the FORTRAN 77 Full Language and is ANSI Standard (ref. 6) with the following exceptions: statements that use system-dependent features and the *do i=m,n* and *end do* statements.

On the ELXSI computer, EZPLOT executes the system-dependent command *cmdstr* through the following subroutine.

```
subroutine system(cmdstr)
character*(*) cmdstr
CALL $SETEXITONERROR(%VAL(.FALSE.))
CALL $SHELL(%STRING(cmdstr(1:ITRIM(cmdstr))))
CALL $SETEXITONERROR(%VAL(.TRUE.))
CALL $ERRORMSG
return
end
```

The call to *$SHELL (...)* causes EMBOS to execute a command. The calls to *$SETEXITONERROR (...)* prevent an EMBOS error from terminating EZPLOT execution. The call to *$ERRORMSG* displays any EMBOS-generated error messages. In addition to the call in this subroutine, calls to *$SHELL (...)* are used throughout the program whenever an escape code is sent to the terminal to toggle between text and graphics mode or to clear the screen.

The program displays the text of the online user's guide and help files through calls to subroutine SYSTEM. These calls contain the pathname of the requested help file. EZPLOT help files are not in the same format as EMBOS help files and may not be listed with the EMBOS *help* command (ref. 1).

The syntax of all system-dependent calls may have to be changed for other host computers.

Ames Research Center
Dryden Flight Research Facility
National Aeronautics and Space Administration
Edwards, California, February 4, 1987

APPENDIX A — EZPLOT SHELLFILE

This appendix includes the text of the EMBOS shellfile used to prompt for terminal type, select the executable module with the appropriate device driver, and initiate execution of the EZPLOT FORTRAN code.

The string *<ESC>* is used in this appendix to identify the presence of an escape control character within a string of characters transmitted to the terminal.

```
- Shellfile to set up for EZPLOT execution.

parm message +switch default=-
checkargs

- Declare shell variables.

declare ezplotSCR
declare ezjobID
declare eztmpID
declare ezusrID
declare PlotFile
declare TermType
declare UseCommands
declare slowTek

- Set usage information and shell variables
- used to identify working files.

set ezplotSCR [date +s]" "[time]" "[userID]
set ezjobID j[jobID].
set eztmpID /Dryden/EZPLOT/temp/
set ezusrID EZID.[userID]

- Display message if 'message = +'.

if [message]  then
    /embos/commands/list /Dryden/EZPLOT/msg
end if

if [var $BATCH]  then
    echo  'E Z P L O T  version 5.5 (BATCH)'
    timestamp
    set TermType 'BATCH'
    set PlotFile [find $ max=1]
    goto eXecution
else
    echo '<ESC>[2J'
    echo '<ESC>[1;1H<ESC>[1m<ESC>[4m'
    cat  'E Z P L O T'  [echo '<ESC> [Om'] '  Version 5.5'
```

```
    echo ''
    timestamp

label SetTermType

    echo ''
    echo 'The following terminals are supported:'
    echo ''
    echo '      (T)ektronix 4014/4114'
    echo '      (G)raphon'
    echo '      (S)elanar'
    echo '      (M)odgraph'
    echo '      (4)109 Tektronix'
    echo '      (V)T125'
    echo ''
    echo ' Enter your terminal type, or "Q" to quit: (T/G/S/M/4/V/Q)'
    set TermType [find \$ max=1]
   if [strcmp [TermType] '=' 'G'] then
       set TermType 'GRAPHON'
       goto eXecution
   end if
   if [strcmp [TermType] '='  'S']  then
       set TermType 'SELANAR'
       goto eXecution
   end if
   if [strcmp [TermType]  '='  'M'] then
       set TermType 'MODGRAPH'
       goto eXecution
   end if
   if [strcmp [TermType] '=' 'T'] then
       set TermType 'TEKTRONIX'
       set slowTek '+'
       goto eXecution
   end if
   if [strcmp [TermType] '=' '4'] then
       set TermType '4109'
       goto eXecution
   end if
   if [strcmp [TermType] '=' 'V'] then
       set TermType 'VT125'
       goto eXecution
   end if
   if [strcmp [TermType] '=' 'Q'], &
       [strcmp [TermType] '=' 'QU'] then
       if [not [var \$BATCH]] then
           echo '<ESC>[2J'
           echo '<ESC>[1;1H'
       end if
       exit
```

```
    end if
    if [strcmp [TermType]  '='  'B'] then
        set TermType  'BATCH'
        goto eXecution
    end if
    if [strcmp [TermType]  '='  'DEBUG'] then
        set TermType  'DEBUG'
        goto eXecution
    end if
    if [strcmp [TermType]  '='  'NO COMMANDS'], &
        [strcmp [TermType]  '='  'NOCOMMANDS']  then
        set UseCommands 'NO'
        echo ''
        echo "FILE 'EZPLOT.COMMANDS' WILL NOT BE USED"
        goto SetTermType
    end if
    echo ''
    echo 'ILLEGAL INPUT'
    goto SetTermType
end if

label eXecution

echo 'Setting up for execution...'
if [strcmp [TermType] '=' '4109'] then
    run /Dryden/EZPLOT/bound/ezplot.409
    exit
end if
if [strcmp [TermType] '=' 'VT125'] then
    run /Dryden/EZPLOT/bound/ezplot.125
    exit
end if
run /Dryden/EZPLOT/bound/ezplot.t14e}
```

APPENDIX B — GENERIC DATA INPUT FILES

A generic data input file contains the following records:

1. Comment record — This record must appear first and may contain any number of lines of arbitrary text, including blank lines. The one exception is that an integer may not appear by itself on a line if more comment lines follow (see the following paragraph describing the number of parameters record). Comment lines are optional and may be omitted completely.

2. Number of parameters record — This record contains a single integer on one line. The value of the integer is the number of parameters on the file, up to a maximum of 2000. If there are no comments, this integer appears on the first line in the file.

3. Name record — This record follows the line giving the number of parameters and contains the parameter names, one name for each parameter. Each name may be from 1 to 16 characters long, including leading and embedded blanks, which are significant. Trailing blanks and characters beyond the 16th are ignored. Also, each name must be enclosed within apostrophes and separated by one or more spaces, a comma, or an end–of–line from the other names.

A line may contain as many names as fit. A name may not be split between two lines. There may be as many lines of names as necessary.

If two or more parameters have the same name and that name is entered as a plot parameter, the first parameter with that name in the list of names is plotted. This rule applies even if the same name is entered in response to more than one parameter–name prompt. Therefore, it is highly recommended that in any one data input file, each parameter name should be different from any other name on that file.

4. Value record — This record contains the data values and must start on a new line following the name record. The data values must correspond one–to–one with the parameter names. Also, there must be the same number of values for each parameter. Values may be delimited by one or more spaces, a comma, or an end–of–line. The data values are grouped into sets of data, each set representing a single data point for each of the parameters. Data values may be spread over any number of lines, but each new set of data values must start on a new line.

The following example of a generic data input file declares five parameters and three data points or sets of data. The words in parentheses at the end of each line are comments added here for clarity. They are not part of the actual file.

Created on June 20, 1986 by EZPLOT	(Comment line)
	(Comment line)
5	(Number of parameters)
'TIME' 'BETA' 'MACH'	(Parameter labels
'CANARD' 'ALPHA'	enclosed in apostrophies)
3443.534 1.00 2.35 -2234.34 12.4	(First set of 5 values)
3467.098, 0.17, 2.66, -4498.23, 8.7,	(Second set of 5 values)
3456.656 0.05 2.44	(Third set of 5 values)
-3234.55 10.6	

There are three ways to create a generic data input file:

1. data may be entered directly onto a file using a text editor;

2. a program may be executed that generates a generic data file from data already existing in another format; or

3. the capability built in to EZPLOT for creating or saving generic data may be selected.

Options (1) and (2) provide for file creation prior to executing EZPLOT.

Option (3) provides for interactive file creation while EZPLOT is executing. Comments may be entered on a generic file only through options (1) and (2).

Option (3) is implemented by entering the letter "I" in response to the data input file format menu. The program then prompts for the number of parameters to be entered and their names. These names must follow the same conventions as given previously in the description of the name record except that they are not enclosed in apostrophes when typed on the keyboard. After the names are entered, the data values may be entered in either of the following ways:

1. enter all data values for one parameter, followed by all data values for the next parameter, and so on, until data values have been entered for all parameters; or

2. enter the first data value for each parameter, followed by the second data value for each parameter, and continue until all the data values have been entered.

The method of data value entry is selected by entering a *1* or a *2*. The program then prompts for individual data values.

When data for three or more parameters are being entered, an asterisk (*) may be used in place of a number to indicate that a value of 1.OE+99 is to be inserted. This large value is not plotted and serves only to fill a location when each parameter does not have the same number of legitimate values.

Data entry may be restarted at any time by entering an *s*. Data entry is terminated by entering a period. After all values have been entered, the data may be saved on a file, may be immediately plotted, or both. This save file may be formatted or unformatted.

APPENDIX C — PLOT FILE

An option that allows plots to be written either to the screen or to a plot file other than a metafile is described in this appendix. This choice is available in both interactive and batch mode. Information on this plot file may subsequently be routed to a laser printer or other suitable device for plotting.

Interactive Mode

To use this capability in the interactive mode, begin by setting the value of a shell variable named *PlotFile* to the name of the file that is to receive the plots. For example, if the plots are to be written to file *myplots*, enter the following command line before starting EZPLOT:

```
set PlotFile myplots +declare
```

Then enter *ezplot*, followed by the desired plotting commands. The plots do not appear on the screen because the coded plotting information normally written to the screen is, instead, written to file *myplots*. Any previous contents of *myplots* are overwritten. When finished, end the session with a *QU* command.

If desired, the plots may now be displayed on the screen by entering the command line

```
myplots> copy
```

The plot or plots do not remain on the screen. When the end of file is reached, the screen is cleared, and the terminal is usually left in its graphics mode rather than its text mode.

Alternately, if the terminal type was not 4109 or VT125, file *myplots* may be routed to the Talaris laser printer (Talaris Systems Inc., La Jolla, California) for plotting on single sheets of paper. This is most conveniently done with the *printTalaris* command by entering the command line

```
printTalaris myplots +Tek
```

If some other suitable laser printer or device is available, the plot file may be routed to it for plotting.

Metafiles cannot be written if *PlotFile* is used. Conversely, if shell variable *PlotFile* does not exist in any scope, or if its value is *.none.* or blank *(' ')*, the plots are displayed on the screen as usual and metafiles can be written (see the Metafiles section). Also, for this case, the program "forgets" *PlotFile* when the plotting session ends.

Batch Mode

To use this capability in batch mode, insert the name of the file to which plots are to be written in the first line of the redirected command file in place of the terminal type (see the Redirection and Batch Operation section). For example, the following command sequence on file *eztm.cmd* will write one plot on file *myplots*. The comments in parenthesis are given here for explanation and are not part of the records on the file.

myplots	(name of plot file)
g	(select generic data)
zgen	(name of file containing generic data)
.	(standard command, end file name entry)
y	(yes, name is correct)

x	(name of independent parameter)
y	(name of dependent parameter)
.	(standard command, end parameter name entry)
y	(yes, names are correct)
PL	(draw plot)
QU	(end execution)

The job is submitted in batch mode with a command line that looks like

```
batch 'eztm.cmd> ezplot' +notify
```

If all plot commands are on file *EZPLOT.COMMANDS*, the redirected file need only consist of one line containing the name of the plot file. If the first line of the redirected command file is blank, contains only one character (a terminal type designation), or contains the string *.none.*, the plots go nowhere unless they are saved on a metafile with the *SA* command. Metafiles cannot be written if the first line contains a file name. If the redirected command file is used for interactive plotting (not very commonly done, but possible, and occasionally useful), the first line must contain the terminal type.

As for all files created by EZPLOT while running in batch mode, the plot file name is prefixed by the job identifier *j[jobID]*. This permits multiple batch jobs to be running in the same directory at the same time.

All EZPLOT jobs run in batch mode use the DI3000 *t14e* driver. Therefore, the code on file *myplots* is Tektronics 4014 code, not 4109 or VT125 code.

Keep in mind that metafiles are still available, are flexible, and work for all terminal types. Plots on metafiles may be manipulated easily with metafile translators and may be output to a variety of devices.

APPENDIX D — PROGRAM CATALOG

The purpose and general operation of the main program and each of the subprograms in EZPLOT are briefly described in this appendix. Four GetData subprograms and four EMBOS subroutines are also briefly described. A list of FORTRAN intrinsics used in EZPLOT is given. Other intrinsics may be used in GetData and DI-3000. DI-3000 subprograms are not described here (see ref. 2).

Main Program

EZPLOT is the name of the main program that controls overall execution of the statements necessary to generate a plot. The program begins by setting initial default values including those graphics characteristics that depend on the type of terminal being used. Next, the data input format to be read and the choice of parameters to be plotted are determined. Then any requested input data processing is done followed by scaling calculations. Next, all necessary graphics elements such as segments, viewports, windows, and planes are specified; the terminal is set to graphics mode; and a plot is drawn. The program then either loops back for the next plot or terminates.

Subroutines

AUTOSC (RMIN,RMAX,NTICS,PLTMIN,PLTMAX) — Subroutine AUTOSC adjusts the minimum and maximum value on each axis so that the increment between tick marks is 1, 2, or 5 times 10 to an integer power. The calculated range is always greater than the range of the data to be plotted.

CAPS (STRING) — Subroutine CAPS converts all lowercase characters in STRING to uppercase.

CFSTAT(XIN, YIN, YFIT, NPTS, COEFFS, FITORD, MEAN, VAR, STDDEV, AVGERR, EZID, TMPID) — Subroutine CFSTAT calculates the mean value, variance, and standard deviation for a polynomial curve fit.

CURFIT (XIN, YIN, XFIT, YFIT, IFIT, NPTS, FITORD, COEFFS, MEAN, VAR,STDDEV,AVGERR,TERTYP, EZID, TMPID, QUEX) — Subroutine CURFIT performs a one-dimensional, least-squares polynomial curve fit.

DRAWFI (FITEXT,XFI,YFI,XSIZFI,YSIZFI) — Subroutine DRAWFI draws the figure label.

DRAWLE (LETEXT,XYL,VUPORT,NCURVS,XSIZLE,YSIZLE) — Subroutine DRAWLE draws the *y*-axis legend labels.

DRAWST (XST,YST,OPTLST,ORDER,MEAN,AVGERR,VAR,STDDEV, COEFFS,XSIZST,YSIZST) — Subroutine DRAWST draws the statistical data.

DRAWTI (TITEXT,XTI,YTI,XSIZTI,YSIZTI) — Subroutine DRAWTI draws the plot title.

DRAWTX (TLINE,TXLOC,TYLOC,TXSIZE,TYSIZE,TPATH,TJUST) — Subroutine DRAWTX draws a text line.

DRAWXL (XLTEXT,XXL,YXL,XSIZXL,YSIZXL) — Subroutine DRAWXL draws the *x*-axis labels.

DRAWYL (YLTEXT,XYL,YYL,XSIZYL,YSIZYL) — Subroutine DRAWYL draws the *y*-axis labels.

ERASER (ER,DE,ERASE,DEORRP,PLXMIN,PLXMAX,PLYMIN,PLYMAX,TERTYP,GIN,QUEX) — Subroutine ERASER marks the relevant area for deleting or replacing points.

FIGURE (FITEXT,FI,TERTYP,QUEX) — Subroutine FIGURE prompts for the figure label text.

FILES (TERTYP,NFILES,INFILE,NEWFIL,LMENU,QUEX) — Subroutine FILES prompts for data input file names. If there are two or more data input files, the number of curves is set equal to the number of files and the names of the files are assigned to the legend line text.

FLIGHT(TERTYP, NCURVS, NPTSIN, NFILES, NFILPR, XINDAT, YINDAT, NAVAIL, PAVAIL, NEWFIL, INDPAR,DEPPAR,LETEXT,XLTEXT,YLTEXT,LMENU,INFILE,ITIME,TIME,QUEX) — Subroutine FLIGHT retrieves data from flight data input files by calling GetData subprograms (see ref. 3).

GAUSS (A,C,COEFFD,FITORD,SNGULR) — Subroutine GAUSS solves *n* linearly independent equations in FITORD unknowns using Gaussian elimination with partial pivoting. This solution is part of the curve fitting option.

GENRIC(TERTYP, NCURVS, NPTSIN, NFILES, NFILPR, XINDAT, YINDAT, NAVAIL, PAVAIL, NEWFIL, INDPAR,DEPPAR,LETEXT,XLTEXT,YLTEXT,LMENU,INFILE,NGPARS,GENDAT,GENLBL,IPLOT, USERIN,QUEX) — Subroutine GENRIC retrieves data from generic data input files.

GETDAT (TERTYP,ND,DATTYP,GENDAT,GENLBL,NPTSIN,NAVAIL,IPLOT,EZID,TMPID,QUEX) — Subroutine GETDAT prompts for data input format and provides for inputting generic data from the keyboard.

GETGEN(GENDAT,GENLBL,NPTSIN,NCURVS,NFILES,NFILPR,LETEXT,INDPAR,DEPPAR,TERTYP, INFILE,XINDAT,YINDAT,PAVAIL,NAVAIL,XLTEXT, YLTEXT,NEWFIL,LMENU,QUEX) — Subroutine GETGEN prompts for parameter names, gets generic data from the data array, and sets default labels, including legend label text, where appropriate.

GINPUT(TERTYP,GENDAT,GENLBL,NPTSIN,NAVAIL,IPLOT,EZID,TMPID,QUEX) — Subroutine GINPUT prompts for keyboard entry of generic data, including parameter names and values, and provides for writing the entered data on a file.

GRID (PLXMIN, PLXMAX, NXINTV, XINT, PLYMIN, PLYMAX, NYINTV, YINT, GT, LOGXSC, LOGYSC, ABXMIN,ABYMIN) — Subroutine GRID draws a background grid of solid or dotted lines.

HELP (USERIN) — Subroutine HELP displays the requested help message.

HMS (TSEC,IHRS,IMIN,ISEC,IMIL) — Subroutine HMS converts time in total seconds to hours, minutes, seconds, and milliseconds.

JAXES(VXMIN, VXMAX, VYMIN, VYMAX, VXICPT, VYICPT, NXINTV, NYINTV, IXTIC, IYTIC, IXDRAW, IYDRAW) — Subroutine JAXES draws the coordinate axes, including tick marks, assuming that a segment is open.

JLABEL (XMIN,XMAX,IXPOS,YMIN,YMAX,IYPOS,IWIDTH,ITEXT,IAXLBL, IXLDIR,IYLDIR) — Subroutine JLABEL draws numbers along the coordinate axes, assuming that a segment is open and that a call has been made to subroutine JAXES.

JLOGAX(VXMIN,VXMAX,VYMIN,VYMAX,VXICPT,VYICPT, NXINTV, NYINTV, IXTIC, IYTIC, IXDRAW, IYDRAW) — Subroutine JLOGAX draws logarithmically spaced minor tick marks between the major tick marks on the designated axis, assuming that a segment is open.

JLOGLB(XMIN,XMAX,IXPOS,YMIN,YMAX,IYPOS,IWIDTH,ITEXT,IAXLBL,IXLDIR,IYLDIR,LOGXSC, LOGYSC) — Subroutine JLOGLB draws numbers along the coordinate axes, including logarithmic axes, assuming that a segment is open and that a call has been made to subroutine JLOGAX.

LEGEND (XLE,YLE, LETEXT, SY, LI, NCURVS, USERIN, ICOLOR, ISTYLE, ISYMBL, XSIZLE, YSIZLE) — Subroutine LEGEND draws legend symbols, lines, and text if those options are selected and more than one curve are plotted.

LEGLBL (LETEXT,DEPPAR,INFILE,NCURVS,NFILES,TERTYP, QUEX) — Subroutine LEGLBL prompts for legend label text.

LINES (LI,DEPPAR,INFILE,NCURVS,NFILES,TERTYP,QUEX) — Subroutine LINES prompts for line visibility choice.

MINMAX (NPTS, NCURVS, YMIN, YMAX, XDATA, YDATA, ABXMIN, ABXMAX, ABYMIN, ABYMAX) — Subroutine MINMAX finds the minimum and maximum data for the independent parameter and all dependent parameters.

MOVAXI(PLXMIN, PLXMAX, PLYMIN, PLYMAX, XINT, YINT, IXTIC, IYTIC, NXINTV, TERTYP,IAXLBL, GIN,QUEX) — Subroutine MOVAXI prompts for locations used to reposition and resize the coordinate axes.

MOVLEG (XLE,YLE,TERTYP,GIN,QUEX) — Subroutine MOVLEG prompts for the location used to reposition the upper left-hand corner of the legend.

MOVSTA (XST,YST,TERTYP,GIN,QUEX) — Subroutine MOVSTA prompts for the location used to reposition the upper left-hand corner of the statistical data.

MOVTXT (TEXTXL,TEXTYL,ITEXT,TERTYP,GIN,QUEX) — Subroutine MOVTXT prompts for the location of a text line.

NOSPAC (STRING) — Subroutine NOSPAC removes leading and embedded spaces from STRING.

NUMINT (NXINTV,NYINTV,AX,LOGXSC,LOGYSC,TERTYP,QUEX) — Subroutine NUMINT prompts for type of, and interval between, tick marks.

ORDER(N,MODE,ARRAY,INDEXS) — Subroutine ORDER sorts ARRAY (a scalar array) containing N real numbers into ascending or descending order according to the contents of MODE. The maximum and minimum values are isolated at opposite ends of the array while the sort converges toward the middle. The array INDEXS holds indexes that indicate the former position of each sorted number.

PATTXT (ITXPTH,ITXJST,ITEXT,TERTYP,QUEX) — Subroutine PATTXT prompts for the path (horizontal or vertical) and justification (left or center) of a line of text.

PLOT (ICURVE,NPTS,SY,LI,SM,USERIN,ICOLOR,ISTYLE,ISYMBL,XSCDAT,YSCDAT,PLXMIN,PLXMAX, PLYMIN,PLYMAX) — Subroutine PLOT draws a line or symbols, or both, for each dependent variable plotted on multiple *y* axes. The line or lines, if visible, may have either straight or smoothed (curved) segments.

PLOTS (NCURVS, NPTSPR, SY, LI, SM, USERIN, ICOLOR, ISTYLE, ISYMBL, XSCDAT, YSCDAT, PLXMIN, PLXMAX,PLYMIN,PLYMAX) — Subroutine PLOTS draws a line or symbols, or both, for each dependent variable plotted on a single *y* axis. The line or lines, if visible, may have either straight or smoothed (curved) segments.

PRECIS(LPRCSH,TERTYP,QUEX) — Subroutine PRECIS prompts for the desired decimal precision of the numbers identifying the tick mark positions along the coordinate axes.

PROCES (XINDAT, YINDAT, YPRDAT, NPTSIN, NPTSPR, SC, TR, TH, ITHIN, XSCALE, YSCALE, XTRANS, YTRANS,ER,SO,NCURVS,ERASE,DE,YMIN,YMAX,ABXMIN,ABXMAX,ABYMIN,ABYMAX,PLXMIN, PLXMAX,PLYMIN,PLYMAX) — Subroutine PROCES performs thinning, sorting, scaling, and translation and data point replacement.

REDUCE (STRING) — Subroutine REDUCE left adjusts STRING to eliminate leading spaces and truncates (removes) any trailing zeros.

SAVEPL(TERTYP,SA,METFIL,CALFIL,EZID,PICTIT,IOK,QUEX) — Subroutine SAVEPL saves the current plot as a metafile picture.

SCALER(XSCALE,YSCALE,INDPAR,DEPPAR,INFILE,NCURVS,NFILES, TERTYP, QUEX) — Subroutine SCALER prompts for a scaling (multiplying) factor.

SCALEX(NPTSPR, NCURVS, ABXMIN, ABXMAX, XPRDAT, XSCDAT, IXLDIR) — Subroutine SCALEX scales the independent parameter values.

SCALEY (NPTSPR,NCURVS,YMIN,YMAX,ABYMIN,ABYMAX,YPRDAT, YSCDAT,AX,IYLDIR) — Subroutine SCALEY scales the dependent parameter values.

SELECT(LI,ISTYLE,SY,ISYMBL,TERTYP,QUEX) — Subroutine SELECT prompts for selection of line patterns and symbol shapes.

SETFMT (VALUE,WIDTH,FMTSTR) — Subroutine SETFMT formats the display of statistical data.

SIZLBL(X1,X2,X3,X4,X5,X6,X7,Y1,Y2,Y3,Y4,Y5,Y6,Y7,TERTYP,QUEX) — Subroutine SIZLBL prompts for size of the characters in the labels.

SIZTXT (TEXTXS,TEXTYS,ITEXT,TERTYP,QUEX) — Subroutine SIZTXT prompts for size of the characters in the text lines.

SMOOTH(SM,DEPPAR,INFILE,NCURVS,NFILES,TERTYP,QUEX) — Subroutine SMOOTH prompts for line-smoothing choice.

STATS(ST,OPTLST,TERTYP,QUEX) — Subroutine STATS prompts for the choice of statistical parameters to be displayed. The order of the curve fit along with the coefficient values, mean value, variance, standard deviation, and average error may be displayed.

SYMBLS (SY,DEPPAR,INFILE,NCURVS,NFILES,TERTYP,QUEX) — Subroutine SYMBLS prompts for symbol visibility choice.

SYMBOL(LN, NP, XSCPTS, YSCPTS, PLXMIN, PLXMAX, PLYMIN, PLYMAX) — Subroutine SYMBOL draws a symbol at a data point location.

SYMSHP (N,PLXMIN,PLXMAX,PLYMIN,PLYMAX) — Subroutine SYMSHP draws various symbol shapes. Symbols are individually defined within a 10 × 10 matrix and then scaled to the window size with subroutines XSIZE and YSIZE.

SYSTEM(CMDSTR) — Subroutine SYSTEM executes the host operating system command contained in CMDSTR.

TERMIN(TERTYP,ICODE) — Subroutine TERMIN sends control sequences to the terminal to select text or graphics mode or to clear the screen.

TEXT (TEXTLN,ITEXT,TERTYP,QUEX) — Subroutine TEXT prompts for the contents of a text line.

THIN (ITHIN,DEPPAR,INFILE,NCURVS,NFILES,TERTYP,QUEX) — Subroutine THIN prompts for the percentage of data points to be plotted.

TITLE (TITEXT,TI,TERTYP,QUEX) — Subroutine TITLE prompts for the plot title text.

TRANSL (XTRANS,YTRANS,INDPAR,DEPPAR,INFILE,NCURVS,NFILES, TERTYP, QUEX) — Subroutine TRANSL prompts for a translation (bias) factor.

UNITS(XSCALE, YSCALE, XTRANS, YTRANS, INDPAR, DEPPAR, INFILE, NCURVS, NFILES, TERTYP, QUEX) — Subroutine UNITS prompts for unit conversion factors equivalent to scaling and translation combined.

UPDATE (NCURVS, NCURPR, NFILES, NFILPR, INPUTX, INPUTY,XLTEXT, YLTEXT, LETEXT, INDPAR, DEPPAR,INFILE,NEWFIL) — Subroutine UPDATE changes the axis and legend labels and calls for automatic scaling as necessary when transitioning from the previous plot to the current plot.

USER (USERIN, TERTYP, NCURVS, NFILES, INDPAR, DEPPAR, NYAXES, TIME, ERASE, EZID, GIN, AX, DE,EP,ER,FI,GR,GT,LE,LI,PF,SA,SC,SM,SO,ST,SU,SY,TH,TI,TR,TX,UN,XL,AXMIN,AXMAX,AYMIN, AYMAX,AXMINP,AXMAXP,AYMINP,AYMAXP,YMIN,YMAX,YMINPR,YMAXPR,PLXMIN,PLXMAX, PLYMIN, PLYMAX, VPXMIN, VPXMAX, VPYMIN, VPYMAX, XLTEXT, YLTEXT, TITEXT, FITEXT, LETEXT,PICTIT,METFIL,CALFIL,PLTFIL,XXL,YXL,YL,XYL,YYL,XTI,YTI,XFI,YFI,XLE,YLE,XST, YST, NEWXAB, NEWYAB, XSCALE, YSCALE, XTRANS, YTRANS, XINT, YINT, NXINTV, NYINTV, IXTIC, IYTIC, IAXLBL, IXLDIR, IYLDIR, INFILE, NEWFIL, LOGXSC, LOGYSC, TEXTLN, TEXTXL, TEXTYL, TEXTXS, TEXTYS, ITXPTH, ITXJST, XSIZXL, YSIZXL, XSIZYL, YSIZYL, XSIZTI, YSIZTI, XSIZFI, YSIZFI, XSIZLE, YSIZLE, XSIZST, YSIZST, XSIZNO, YSIZNO, NPTSIN, XINDAT, YINDAT, ICOLOR, ISTYLE, ISYMBL, LPRCSN, NPTSPR, XPRDAT, XSCDAT, YPRDAT, YSCDAT, ITHIN, IFIT, FITORD, NPTS, COEFFS, MEAN, VAR, STDDEV, AVGERR, OPTLST, QUEX) — Subroutine USER accepts any valid two-character plot command entered from the keyboard and responds appropriately. System commands are accepted as the argument of the *DO cmd* (see subroutine SYSTEM).

WAIT(NSECS) — Subroutine WAIT executes a do-nothing loop that causes a program delay of about NSECS seconds.

XLABEL (XLTEXT,XL,TERTYP,QUEX) — Subroutine XLABEL prompts for the *x*-axis label text.

XLOGSC (NPTSPR,NCURVS,ABXMIN,ABXMAX,XPRDAT,XSCDAT,IXLDIR) — Subroutine XLOGSC scales the independent parameter values to fit a logarithmic scale.

XRANGE (ABXMIN,ABXMAX,AXMINP,AXMAXP,TERTYP,QUEX) — Subroutine XRANGE prompts for minimum and maximum values for the *x* axis.

YLABEL (YLTEXT,YL,TERTYP,QUEX) — Subroutine YLABEL prompts for the *y*-axis label text.

YLOGSC (NPTSPR,NCURVS,YMIN,YMAX,ABYMIN,ABYMAX,YPRDAT, YSCDAT,AX,IYLDIR) — Subroutine YLOGSC scales the dependent parameter values to fit a logarithmic scale.

YRANGE (YMIN,YMAX,YMINPR,YMAXPR,ABYMIN,ABYMAX,AYMINP,AYMAXP,AX,NCURVS,NFILES, DEPPAR,INFILE,TERTYP,QUEX)—Subroutine YRANGE prompts for minimum and maximum values for the *y* axis.

Functions

BIG — Function BIG returns the real value +1.0E+18.

IRANGE (IVAL,IMIN,IMAX) — Function IRANGE returns IVAL unchanged if within the range of IMIN and IMAX. If outside that range, IVAL takes the value of IMIN or IMAX, depending on which bound was exceeded.

ITRIM (STRING) — Function ITRIM returns the length of STRING, not including trailing blanks.

LINDEX (TARGET,SEARCH) — Function LINDEX returns the position of the last occurrence of SEARCH in TARGET.

LINT (A) — Function LINT returns the nearest integer less than, or the integer equal to, the real number A.

NEAR (A,B,BIAS) — Function NEAR returns a one if MAX(A,B) is zero or if the fraction ABS(A-B)/MAX(A,B) is less than BIAS. Otherwise, a zero is returned.

PRESSR (QUEX)—Function PRESSR displays the prompt *Press <return> to continue...* on the screen and writes the string *<CR>* on unit 9. QUEX contains *QU, EX, EE,* or a blank.

RANGE (RVAL,RMIN,RMAX) — Function RANGE returns RVAL unchanged if within the range of RMIN and RMAX. If outside that range, RVAL takes the value of RMIN or RMAX, depending on which bound was exceeded.

READSW (IOBUFF,QUEX) — Function READSW returns the characters entered in response to a prompt as the contents of IOBUFF. QUEX contains *QU, EX, EE* or a blank.

SCALE (RVAL,RMIN,RMAX,SMIN,SMAX) — Function SCALE calculates a scaling factor based on the position of RVAL relative to RMIN and RMAX in terms of a percentage of the difference between SMIN and SMAX.

SMALL — Function SMALL returns the real value −1.0E+18.

GetData Subprograms

The following GetData subprograms are explicitly mentioned in the EZPLOT code. Other GetData subprograms included in the executable module are not described here.

CLOSER (UNIT) — Subroutine CLOSER closes a flight data file.

FREAD (UNIT,TIME,DATA) — Function FREAD reads the next record from a flight data file. UNIT, TIME, and DATA are names of variables in GetData.

OPENR (UNIT,FNAME,NCHS) — Function OPENR opens a flight data file and automatically determines the file format. UNIT, FNAME, and NCHS are names of variables in GetData.

SIGSR (UNIT,NAMES,NCHS) — Subroutine SIGSR specifies channels to read by signal name for FREAD. UNIT, NAMES, and NCHS are names of variables in GetData.

EMBOS Subroutines

The following EMBOS system-dependent subroutines are explicitly called from the EZPLOT code. The syntax of these calls may have to be changed for other host computers.

$ERRORMSG(...) — Display any system-generated error messages.

$SETEXITONERROR(...) — Set the value of the error exit flag.

$SHELL(...) — Execute a shell command line.

$VAR(...) — Assign the value of a shell variable to a FORTRAN variable.

Intrinsics

The following FORTRAN intrinsics are used in EZPLOT. Additional intrinsics may be used in the DI-3000 and GetData code.

ABS — Absolute value.

CHAR—Convert integer to character.

FLOAT — Convert integer to real.

ICHAR—Convert character to integer.

INDEX— Position of substring.

INT — Convert real to integer by truncating.

LEN — Length of character expression.

LOG10 — Common logarithm.

MAX — Maximum value.

MIN — Minimum value.

MOD — Remainder (modulo).

NINT — Convert real to integer by rounding.

SNGL — Convert double precision to single precision.

APPENDIX E — FILE ASSIGNMENTS

EZPLOT assigns the following FORTRAN unit numbers to files which are opened, read, written, and closed at various points in the program. While open, these files cannot be accessed by other EZPLOT jobs. To minimize this kind of interference, the names of some of these files are prefixed by a job identifier, symbolized below by the tag *j[jobID]*. The effect of this identifier is to isolate the file from other EZPLOT jobs.

Unit 1, /Dryden/EZPLOT/temp/j[jobID].EZPLOT.SCR — This file is used in subroutine CFSTAT to hold statistical data for display by subroutine USER when requested. It is also used in subroutine GINPUT to hold generic data entered through *$stdin* until this data is written to a user-named file by EMBOS redirection.

Unit 2, /Dryden/EZPLOT/usage — This file is accessed in the main program to record the start time and date of a plotting session. It is opened with "appendonly" permission to prevent inadvertent corruption by users. If this file happens to be opened by another user, the program will keep trying to access it at about 1-sec intervals until successful.

Unit 3, /Dryden/EZPLOT/temp/j[jobID].EZPLOT.COMMANDS — This file is accessed in function *READSW* and may contain a list of plotting commands similar to those entered through *$stdin*. This list of commands must initially be on a file named *EZPLOT.COMMANDS* (unit 4) in the user's current directory. See command *SW* for more information.

Unit 4 EZPLOT.COMMANDS — This file may contain a list of plotting commands used to draw a plot. If present in the user's current directory, this file is accessed in subroutine SYSOC and is copied to unit 3 during program startup prior to plotting.

Unit 7 (named by user)—This file is accessed in subroutine FILES and is used to store a list of the names of the files that contain data to be plotted.

Unit 8 (named by user) — This file is accessed in subroutine SAVEPL and contains data in DI-3000 metafile format (ref. 5). For jobs run in batch mode, the metafile name is prefixed automatically with the *j[jobID]*. tag.

Unit 9, /Dryden/EZPLOT/temp/j[jobID].EZID.[userID] — This file is used throughout the program to store a list of all commands and other responses entered through *$stdin* or *EZPLOT.COMMANDS* (unit 4). If desired, a copy of unit 9 may be saved in the user's current directory at program completion. If saved, the file will be named *EZPLOT.USERINPUT*. For jobs run in batch mode, this name is prefixed automatically with the *j[jobID]*. tag. See command *SU* for more information.

Units 11 to 18 (named by user) — These files are accessed in either subroutine FLIGHT or subroutine GENRIC and contain the data to be plotted. Flight data is read by GetData subroutines.

REFERENCES

1. *EMBOS User's Guide*, Volume 1, ELXSI, 2334 Lundy Place, San Jose, California 95131, 1983.
2. *DI-3000 User's Guide*, Precision Visuals, Inc., 6260 Lookout Road, Boulder, Colorado 80301, 1984.
3. Maine, Richard E., *Manual for GetData Version 3.1, a FORTRAN Utility Program for Time History Data*, NASA TM-88288, 1987.
4. *NASA Publications Manual*, NASA SP-7013, 1974.
5. *METAFILE System User's Guide*, Precision Visuals, Inc., 6260 Lookout Road, Boulder, Colorado 80301, 1984.
6. *American National Standard Programming Language FORTRAN, ANSI X3.9-1978*, American National Standards Institute, 1430 Broadway, New York, New York, 1978.

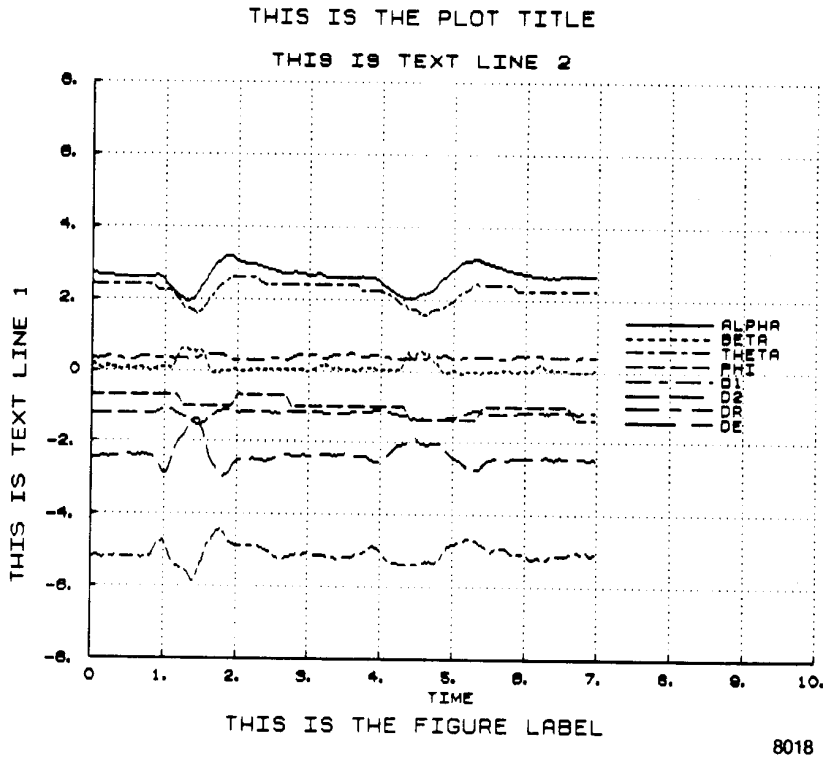

Figure 1. Representative plot showing default axis intervals, dotted grid lines, and various line patterns.

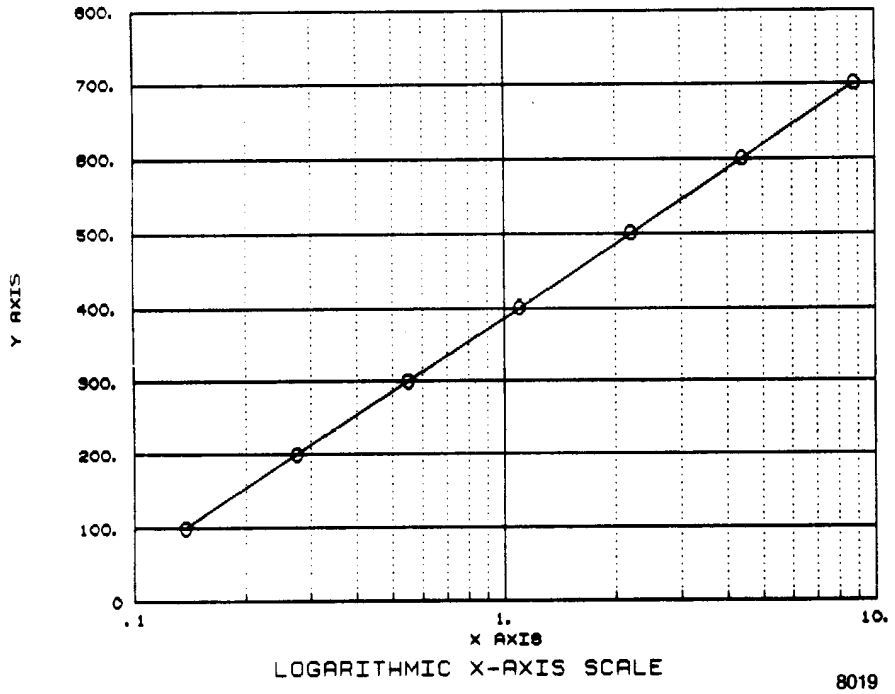

Figure 2. Representative plot showing linear and logarithmic scales.

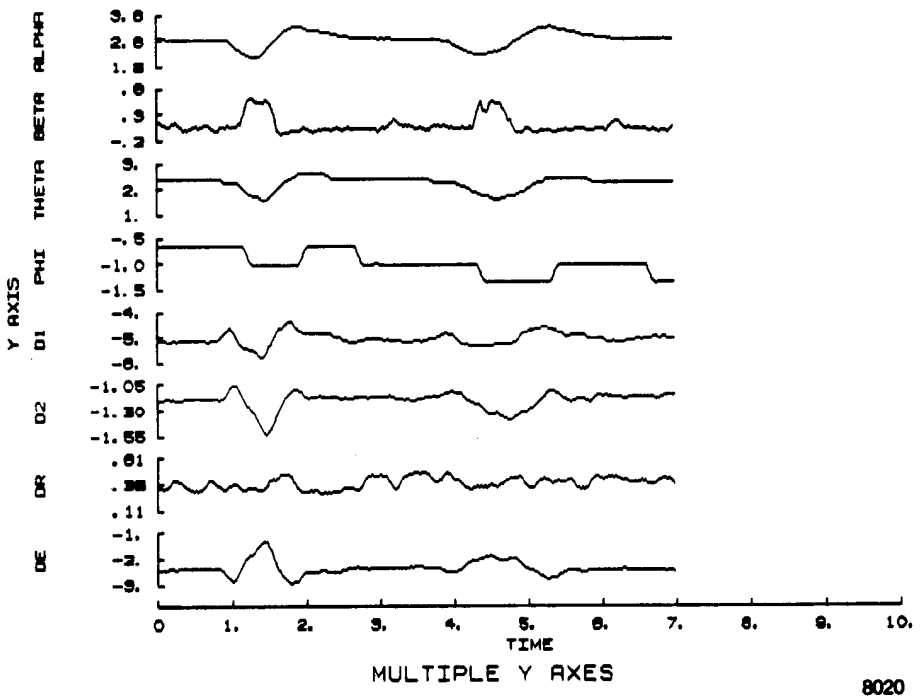

Figure 3. Representative plot showing multiple axes.

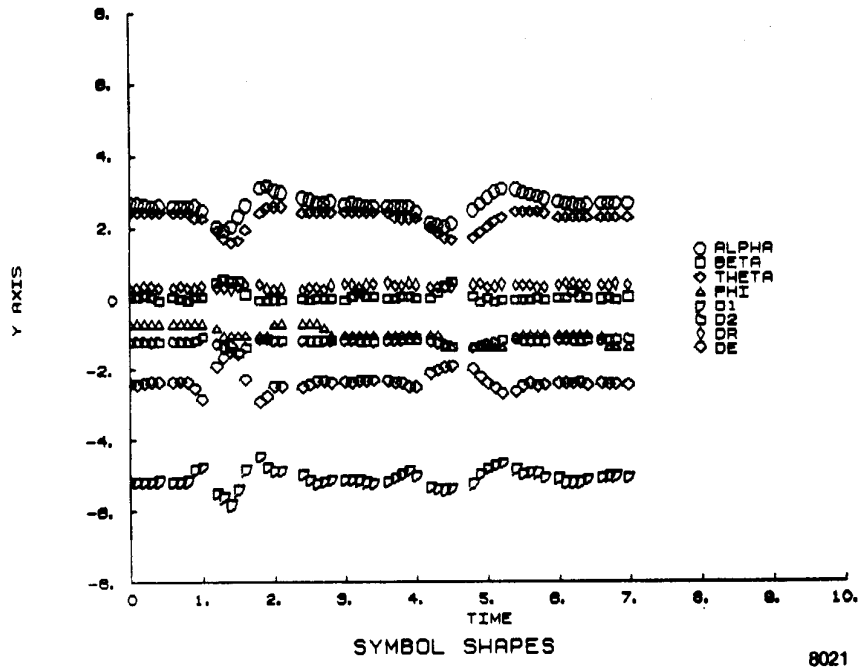

Figure 4. Representative plot showing various symbol shapes.

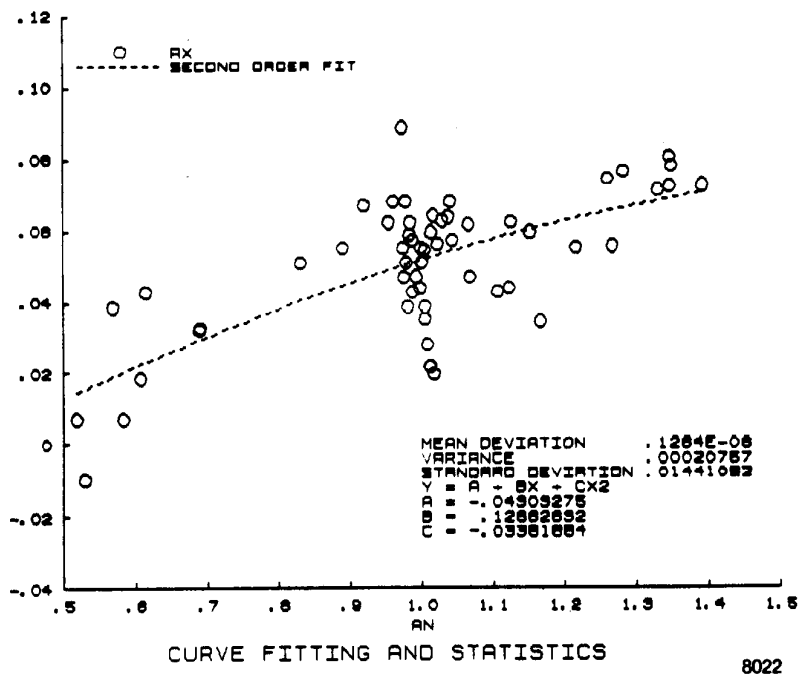

Figure 5. Representative plot showing curve fitting and statistical characteristics of the data.

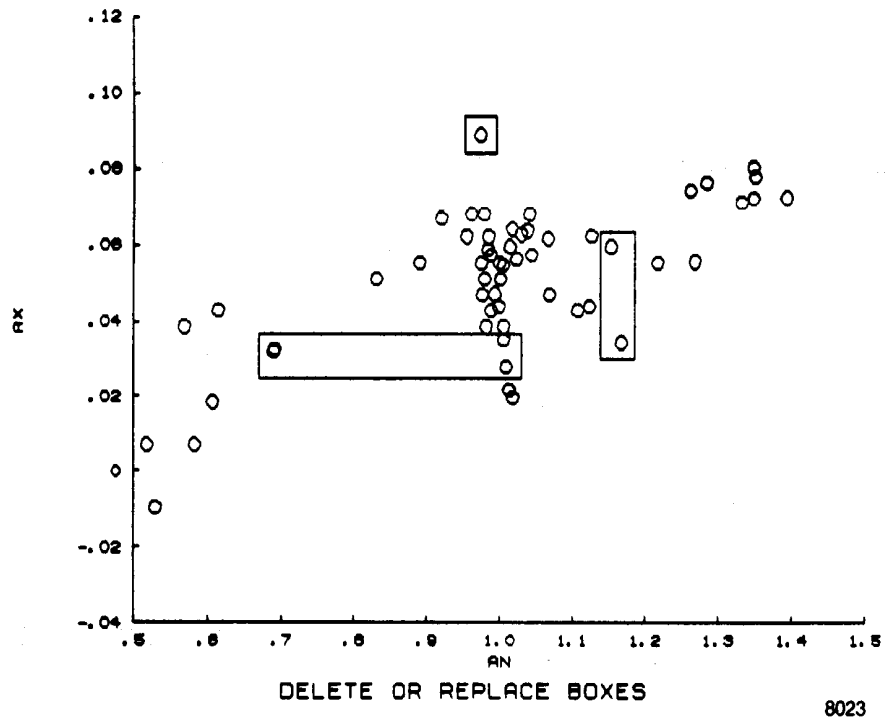

Figure 6. Representative plot showing boxes around points to be deleted or replaced.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-88293		

4. Title and Subtitle	5. Report Date
User's Manual for EZPLOT Version 5.5—A FORTRAN Program for Two-Dimensional Graphic Display of Data	October 1988
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Charles Garbinski, Paul C. Redin and Gerald D. Budd	H-1410
	10. Work Unit No.
9. Performing Organization Name and Address	RTOP 533-02-51
NASA Ames Research Center Dryden Flight Research Facility P.O. Box 273, Edwards, CA 93523-5000	11. Contract or Grant No.
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

EZPLOT is a computer applications program that converts data resident on a file into a plot displayed on the screen of a graphics terminal. This program generates either time history or x–y plots in response to commands entered interactively from a terminal keyboard. Plot parameters consist of a single independent parameter and from one to eight dependent parameters. Various line patterns, symbol shapes, axis scales, text labels, and data modification techniques are available. This user's manual describes EZPLOT as it is implemented on the Ames Research Center, Dryden Flight Research Facility ELXSI computer using DI-3000 graphics software tools.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Computer program Data analysis Graphics	Unclassified — Unlimited Subject category 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	45	A03

NASA FORM 1626 OCT 86

For sale by the National Technical Information Service, Springfield, VA 22161-2171.