The SCD Graphics Utilities

Editor: Gregory R. McArthur
The SCD Graphics Utilities

Table of Contents

CHAPTER 1: OVERVIEW OF THE SCD GRAPHICS UTILITIES

Philosophy ............................................................... 1-1
Capabilities ........................................................................ 1-1
Picture Quality vs. Cost ................................................... 1-1
Flexibility ........................................................................... 1-2
Portability ........................................................................... 1-3

CHAPTER 2: HOW TO USE THIS MANUAL

General ............................................................................. 2-1
The Graphics Utilities ...................................................... 2-2

CHAPTER 3: THE SCD GRAPHICS UTILITIES

Alphabetical listing by Utility Name

AUTOGRAPH ........................................................................ AUTOGRAPH-1
CONRAN ............................................................................. CONRAN-1
CONRAQ ............................................................................. CONRAQ-1
CONRAS ............................................................................. CONRAS-1
CONREC ............................................................................. CONREC-1
CONRECQCK ..................................................................... CONRECQCK-1
CONRECSMTH .................................................................. CONRECSMTH-1
CONRECSUPR .................................................................. CONRECSUPR-1
DASHCHAR ......................................................................... DASHCHAR-1
DASHLINE ......................................................................... DASHLINE-1
DASHSMTH ......................................................................... DASHSMTH-1
DASHSUPR ......................................................................... DASHSUPR-1
EZMAP .............................................................................. EZMAP-1
HAFTON ............................................................................. HAFTON-1
ISOSRF .............................................................................. ISOSRF-1
ISOSRFHR ......................................................................... ISOSRFHR-1
APPENDICES

Appendix A: The Unabridged AUTOGRAPH
Appendix B: Transformations
Appendix C: Processing Parts of Arrays
CHAPTER 1: OVERVIEW OF THE SCD GRAPHICS UTILITIES

PHILOSOPHY

At NCAR, the approach has been to have a generally static system plot package and supplement this with FORTRAN utility routines as needs arise. The Scientific Computing Division program library contains a comprehensive selection of utility routines for generating scientific computer graphics. Generally, these represent the most up-to-date plotting methods consistent with NCAR's graphics philosophy and available hardware.

Because of the nature of scientific computer graphics, algorithms for certain capabilities must be present, and these algorithms must have certain attributes that are often conflicting. For example, some users want the quickest algorithm possible regardless of picture quality while others desire directly publishable plots regardless of the cost. Some users want very simple argument lists, while others need more versatility. Portability of utility routines is also important, and excellent documentation is vital.

CAPABILITIES

The capabilities required for the atmospheric scientist include plotting characters including Greek and Roman alphabets, plotting geographic maps in various projections; automatic generation of plots of curves or families of curves from user supplied arrays; plotting two-dimensional fields via contour maps, half-tone pictures of surfaces with hidden lines removed; displaying two-dimensional velocity fields by drawing velocity vectors or streamlines; and displaying of three-dimensional arrays by drawing contour surfaces or processing two-dimensional slices through the fields. Utility routines for all of these processes have been implemented or created at NCAR.

PICTURE QUALITY VS. COST

There is generally a trade-off between picture quality and algorithm complexity (and resulting code length and execution time.) The simplest, smallest, quickest algorithms generally produce the least attractive results; while the most complicated, largest, slowest algorithms generally produce the best looking results. Since no single algorithm can be at both extremes, a family of algorithms is often needed to solve a particular problem. Users can then choose the level of picture quality consistent with their needs by carefully examining the attributes for each package in the table of contents and studying the sample pictures in each write-up.

The construction of dashed lines can illustrate such a hierarchy of routines. The dd80 hardware is capable of producing dashed lines. This hardware, however, fails to
produce pleasing results when drawing very short vector segments or segments of varying length. To remedy the problems, a package was written to produce dashed line patterns by software. This package had entry points analogous to the pattern setting and line drawing entries in the system plot package, allowing the user who wished to pay for the slight amount of extra space and time to get better looking results with minimal change to the user program. Subsequently, a version of this package was written that allowed characters to be part of the dashed line pattern and be plotted at the appropriate positions on the line. Finally, a version was written that automatically fitted a curve through the given points using splines under tension. By selecting one of these four packages, a user may produce whatever level of picture quality required.

Applications of these dashed line generators is not restricted to user programs; they can also be used by utility routines. Since the user entry points have the same names in all the software dashed line packages, by calling these routines in higher level packages, the range of picture quality, routine size and speed of the higher level algorithm is obtained by merely exchanging dashed line packages. For example, various levels of sophistication in the automatic graphing routines are obtained by using different software dashed line packages with the same driver routines. In the same way, various levels of contouring routines are obtained by using the various dashed line packages with a common driver. Similarly, there are levels of quality available in the character generators, ranging from the dd80 and Dicroed hardware characteristics to very high quality software characters. To prevent name conflicts when two high level packages use the same lower level packages, as in CONREC and AUTOPRINT, both using DASHCHAR, the routines FWRY and DASHCHAR have been added to the binary file and are automatically loaded (only once) when higher level packages reference them. Users wishing to make modifications to these routines may still access them from ULIB. The documentation is written as if the lower level code were present on the file with the higher level code, since loading of the lower level code is transparent to the user.

FLEXIBILITY

The problem of choosing argument lists which possess both simplicity and flexibility is solved by having one of each type. For example, half-tone pictures can be generated by calling an inflexible routine with a short argument list, CALL EZHFTN (Z, M, N), or a flexible routine with a longer argument list, CALL HAFTON (Z, L, M, N, FLO, HT, NLEV, NOPT, NPERM, ISPV, SPVAL). The routine with the short argument list merely calls the one with the
long argument list with reasonable values supplied for the additional arguments. The use of clearly identified internal parameters for any value the user might want to alter further increases flexibility and readability.

PORTABILITY

Because the individual scientist is mobile, and because NCAR is becoming the central computer node on the remote job entry network of atmospheric science departments, the portability of the graphics utility routines is extremely important. The portability problem is attacked in two ways: First, a standard FORTRAN is used as much as possible, and, second, only a subset of the system plot package entry points is used in the FORTRAN utility routines. Thus, an implementor at a remote site is faced with a total of less than a dozen referenced system plot package entry points for transporting all of NCAR's utility routines that have been brought up to this standard. Alternatively, the system plot package can be transported; contact the SCD Software Librarian for details.
CHAPTER 2: HOW TO USE THIS MANUAL

GENERAL
This manual documents the Graphics Utilities used by the Scientific Computing Division (SCD) at NCAR. The utilities are listed alphabetically by utility name.

In order to provide users with a rapid-scan reading capability, the documentation for each utility is divided into topical categories, arranged in the following order:

Utility Name
The name of the utility (generally an acronym).

Dimension of Arguments
Any variables in the calling sequence that are used in dimension statements are included here with an indication of minimum requirements for variable dimensions.

Latest Revision
The month and year in which the utility was most recently modified.

Purpose
A brief description of the particular capabilities of the named graphic utility.

Usage
A brief explanation of how to use the access cards to obtain the utility. In some cases, underlying assumptions are indicated which are designed to give the user some idea of the utility default options.

Arguments
All arguments to the utility are defined and specify the required values and functions.

Entry Points
All entry points included in the file are listed. These include the names of programs, functions, subroutines, and entry points.

Common Blocks
A list of the common blocks and their lengths.

I/O
An indication of the kind of input/output performed by this utility.

Precision
Either single or double precision is indicated.

Required Library Routines
Other library routines which require access cards of their own are listed here.

Language
The name of the source language(s) used for the utility.

History
Information about the original routine is included here as well as dates when the utility was standardized.
Algorithm

A brief description of the algorithm used is given.

Space Required

The number of octal words of central memory required with the decimal equivalent stated. The figure is obtained from one of NCAR's local computers.

Accuracy

A brief description of the limits on accuracy appropriate for understanding the results of the utility.

Timing

The length of time the utility takes to perform a typical task on one of NCAR's local computers.

Portability

An indication of the portability of the utility as well as portability restrictions.

Plotting Routines Used

A list of routines used by the utility to perform plotting.

Required Resident Routines

A list of basic external functions included in the NCAR system. These routines are not part of the utility.

Internal Parameters

A list of the parameters used internally by the utility. The list includes the name of the parameter, its default value, and a description of the parameter function.

Sample Plots

Illustrations of the types of plots obtained via the utility.

THE GRAPHICS UTILITIES

The following utilities are documented in this manual:

- AUTOGRAPH
- CONRAN
- CONRAS
- CONRAQ
- CONREC
- CONRECQCK
- DASHCHAR
- DASHLINE
- DASHSMTH
- DASHSUPR
- EZMAP
- HAFTON
- ISOSRF
- ISOSRFHR
- PWRITX
- PWRITY
- PWRZI
- PWRZS
- PWRZT
- SCROLL
- SRFACE
- STRMLN
- SUPMAP
- THREED
- VELVCT
- WINDOW

A brief description of each utility may be found in the document INTRODUCTION TO THE SCD GRAPHICS SYSTEM.

Each utility is arranged alphabetically by name. Users will find the name of the utility on the bottom of the page along with the page number.
PURPOSE
To draw graphs, each with a labeled background and each displaying one or more curves.

USAGE
AUTOGRAPH has been designed for two general classes of users: (1) those who require only simple X-Y coordinate graphics capabilities, and (2) those who wish to produce more sophisticated, customized graphical output.

With this dichotomy in mind, the documentation for AUTOGRAPH is divided into two separate sections. This document is an abridged version of the entire AUTOGRAPH package; designed for those users who wish only to use the more elemental features of the package to produce simple graphs. The second section, found in Appendix A, contains the unabridged documentation for AUTOGRAPH. It documents the full range of features which are available to users who desire more complex graphical manipulation of their data.

AUTOGRAPH Routines

Each of the following routines draws a complete graph with one call. The name of the routine appears in the left-hand margin. The argument(s) to the routine are enclosed in parentheses immediately to the right of the routine name. Each routine is implemented by a set of calls to the lower-level AUTOGRAPH routines AGSTUP, AGCURV, and AGBACK (see below).

EZY (YDRA,NPTS,GLAB)
Draws a graph of the curve defined by the data points ((I,YDRA(I)),I=1,NPTS), with a graph label specified by GLAB.

EZXY (XDRA,YDRA,NPTS,GLAB)
Draws a graph of the curve defined by the data points ((XDRA(I),YDRA(I)),I=1,NPTS), with a graph label specified by GLAB.

EZMY (YDRA,IDXY,MANY,NPTS,GLAB)
Draws a graph of the family of curves defined by data points (((I,YDRA(I,J)),I=1,NPTS),J=1,MANY), with a graph label specified by GLAB. The order of the subscripts of YDRA may be reversed. (See the routine DISPLA, argument LROW).

EZMXY (XDRA,YDRA,IDXY,MANY,NPTS,GLAB)
Draws a graph of the family of curves defined by the data points (((XDRA(I),YDRA(I,J)),I=1,NPTS),J=1,MANY), with a graph label specified by GLAB. XDRA may be doubly-
subscripted and the order of the subscripts of XDRA and YDRA may be reversed. (See the routine DISPLA, argument LROW).

**IDiot**

(XDRA, YDRA, NPTS, LTYP, LUSH, LABX, LABY, LABG, LFRA)

This is an implementation of the routine from which AUTOGRAPH grew. It is not recommended for use.

**AUTOGRAPH Control Parameter Routines**

The following routines provide user access to the AUTOGRAPH control parameters (in the labeled common block AGCONP).

**ANOTAT**

(XLAB, YLAB, LBAC, LSET, NDSH, DSHL)

May be used to change the X- and Y-axis (non-numeric) labels, the background type, the way in which graphs are positioned and scaled, and the type of dash patterns to be used in drawing curves.

**DISPLA**

(LFRA, LROW, LTYP)

May be used to specify when, if ever, the EZ... routines do a frame advance, how input arrays for EZMY and EZMXY are dimensioned, and the linear/log nature of graphs.

**AGSETP**

(TPGN, FURA, LURA)

A general purpose parameter setting routine, used to set the group of parameters specified by TPGN, using values obtained from the array (FURA(I),I=1,LURA).

**AGSETF**

(TPGN, FUSR)

Used to set the single parameter specified by TPGN, giving it the floating-point value FUSR.

**AGSETI**

(TPGN, IUSR)

Used to set the single parameter specified by TPGN, giving it the integer equivalent of the value of the single parameter specified by TPGN.

**AGGETP**

(TPGN, FURA, LURA)

A general purpose parameter getting routine, used to get the group of parameters specified by TPGN, putting the result in the array (FURA(I),I=1,LURA).

**AGGETF**

(TPGN, FUSR)

Used to get, in FUSR, the floating-point value of the single parameter specified by TPGN.

**AGGETI**

(TPGN, IUSR)

Used to get, in IUSR, the integer equivalent of the value of the single parameter specified by TPGN.
The following are lower-level routines which may be used to draw graphs of many different kinds. The EZ... routines call these. They are intended to be called by user programs as well.

**AGSTUP**

\[ \text{AGSTUP} \quad (\text{XDRA, NIX, IIVX, NEVX, IIEX, YDRA, IVIY, IIVY, NEVY, IIEY}) \]

This routine must be called prior to the first call to either of the two routines AGBACK and AGCURV, to force the set-up of secondary parameters controlling the behavior of those routines. After any parameter setting call, AGSTUP must be called again before calling either AGBACK or AGCURV again. AGSTUP calls the routine "SET", in the Plot Package, so that user X/Y coordinates in subsequent calls will map properly into the plotter space.

**AGBACK**

Draws the background defined by the current state of the AUTOGRAPH control parameters.

**AGCURV**

\[ \text{AGCURV} \quad (\text{XVEC, IIEX, YVEC, IIEY, NEVY, KDSH}) \]

Draws the curve defined by the arguments, positioning it as specified by the current state of the AUTOGRAPH control parameters.

**ARGUMENTS TO THE ROUTINES**

This section contains descriptions of the arguments to AUTOGRAPH routines.

**Routines EZ...**

In calls to the routines EZY, EZXY, EZMY, and EZMXY:

**XDRA**

is an array of X coordinates dimensioned as implied by the current value of the AUTOGRAPH control parameter "ROW." (see the description of the argument LROW, below). The value of the AUTOGRAPH parameter "NULL/1." (1.E36, by default), when used as an X-coordinate, implies a missing data point. The curve segments on either side of such a point are not drawn.

**YDRA**

is an array of Y coordinates dimensioned as implied by the current value of the AUTOGRAPH control parameter "ROW." (see the description of the argument LROW, below). The value of the AUTOGRAPH parameter "NULL/1." (1.E36, by default), when used as a Y-coordinate, implies a missing data point. The curve segments on either side of such a point are not drawn.

**IDXY**

is the first dimension of the arrays XDRA (if it has two dimensions) and YDRA.

**MANY**

is the number of curves to be drawn by the call to EZ... Normally, the second dimension of XDRA (if it has two dimensions) and YDRA.
NPTS is the number of points defining each curve to be drawn by the routine EZ... Normally, the first (or only) dimension of XDRA and YDRA.

GLAB is a character string (or an array containing a character string) defining a label to be placed at the top of the graph. The string may not be more than 40 characters long. If it is fewer than 40 characters long, its last character must be a dollar sign. The dollar sign is NOT a part of the label. It is stripped off. A zero may be used in place of the argument GLAB, indicating that the previous label, whatever it was, should continue to be used (the initial graph label consists of blanks). Note that the use of a zeroed GLAB is non-standard. The routines AGSETP, AGSETF, and AGSETI may be used to define labels in a more nearly standard manner.

Routine ANOTAT In calls to the routine ANOTAT:

XLAB and YLAB resemble GLAB (see above) and define labels for the X and Y axes. The default X-axis label is the single character X, the default Y-axis label is the single character Y. Note that one may use a zeroed XLAB (YLAB) to indicate that the X-axis (Y-axis) label is not to be changed from what it was previously.

LBAC if non-zero, specifies a new value for the AUTOGRAPH control parameter "BACKGROUND.", as follows:

1 - A PERIMETER BACKGROUND
2 - A GRID BACKGROUND
3 - AN AXIS BACKGROUND
4 - NO BACKGROUND

The default value of "BACKGROUND." is 1.

LSET if non-zero, specifies a new value for the AUTOGRAPH control parameter "SET.". This parameter may be negated to suspend the drawing of curves by the EZ... routines so that a call to one of them will produce only a background. The absolute value of "SET." affects the way in which AUTOGRAPH determines the position and shape of the graph and the scaling of the axes, as follows:
1- Restores the default values of the AUTOGRAPH parameters in question. AUTOGRAPH will set up an appropriate call to the Plot Package routine "SET", overriding any prior call to that routine.

2- Tells AUTOGRAPH to use arguments 1-4 and 9 of the last "SET" call. Arguments 1-4 specify where the graph should fall on the plotter frame. Argument 9 specifies whether the graph is linear/linear, linear/log, etc.

3- Tells AUTOGRAPH to use arguments 5-8 and 9 of the last "SET" call. Arguments 5-8 specify the scaling of the axes. Argument 9 specifies whether the graph is linear/linear, linear/log, etc.

4- A combination of 2 and 3 above. Arguments 1-4 of the last "SET" call specify the position, arguments 5-8 the scaling, and argument 9 the linear/log nature of the graph.

NOTE: The Plot-Package routine "SET" is described in the NCAR System Plot Package documentation. It is not a part of AUTOGRAPH.

If the routine DISPLA is called with its argument LTYP non-zero, the linear/log nature of the graph will be that specified by LTYP, not that specified by the last SET call, irrespective of the value of the control parameter "SET."

The default value of "SET." is 1.

NDSH if non-zero, specifies a new value of the AUTOGRAPH control parameter "DASH/SELECTOR." (and, therefore a new set of dashed-line patterns), as described below. Note: the default value of the dashed-line parameters is such that all curves will be drawn using solid lines. If that is what you desire, use a zero for NDSH.

If the value of "DASH/SELECTOR." is negative, curves produced by subsequent calls to EZMY or EZMXY will be drawn using the set of alphabetic dashed-line patterns defined internally to AUTOGRAPH. The first curve produced by a given call will be labeled "A", the second "B", ..., the twenty-sixth "Z", the twenty-seventh "A" again, and so on. Curves drawn by calls to EZY and EZXY are unaffected.

If the value of "DASH/SELECTOR." is positive, it must be less than or equal to 26. The next argument, DSHL, is an array containing NDSH dashed-line patterns. All curves produced by subsequent calls to EZY, EZXY, EZMY, and EZMXY will be drawn using the dashed-line patterns in AUTOGRAPH - 5
The first curve produced by a given call will have the pattern specified by DSHL(1), the second that specified by DSHL(2), the third that specified by DSHL(3), . . . the NDSH+1st that specified by DSHL(1), . . . etc. Each element DSHL(I) is either a sixteen-bit integer (sixteen bits, right-justified, with zero fill to the left), in which a 1-bit represents a solid line portion and a 0-bit a gap or a character string (as many as will fit in a floating-point word) in which a dollar sign stands for a solid-line segment, a quote stands for a gap, and other characters stand for themselves. (See the documentation for the "DASHCHAR" routine).

DSHL (if NDSH is greater than zero) is an array of dashed-line patterns, as described above. Note that integer values 0 and 1, which used to give a solid line, no longer work in that manner. If you want a solid line, use the integer value 2**16-1 (65535, DECIMAL).

**Routine DISPLA**

In calls to the routine DISPLA:

- **LFRA** if non-zero, specifies a new value for the AUTOGRAPH control parameter "FRAME.". Possible values are as follows:
  
  1 = The EZ... routines do a frame advance after drawing.

  2 = No frame advance is done by the EZ... routines.

  3 = The EZ... routines do a frame advance before drawing.

  The default value of "FRAME." is 1.

- **LROW** if non-zero, specifies a new value for the AUTOGRAPH control parameter "ROW.". This parameter tells AUTOGRAPH how the argument arrays XDRA and YDRA, in calls to the routines EZMY and EZMXY, are subscripted, as follows:

  1. If "ROW." is positive, this implies that the first subscript of YDRA is a point number and the second subscript is a curve number.
     If "ROW." is negative, the order is reversed.

  2. If the absolute value of "ROW." is 1, this implies that XDRA is singly-subscripted, by point number only.
     If the absolute value of "ROW." is 2 or greater, this implies that XDRA is doubly-subscripted, just like YDRA.

  The default value of "ROW." is 1, specifying that XDRA is
singly-subscripted and that YDRA is doubly-subscripted by point number and curve number, in that order.

LTYP if non-zero, specifies new values for the AUTOGRAPH control parameters "X/LOGARITHMIC." and "Y/LOGARITHMIC.", which determine whether the X and Y axes are linear or logarithmic. Possible values are as follows:

1 - X AXIS LINEAR - Y AXIS LINEAR
2 - X AXIS LINEAR - Y AXIS LOGARITHMIC
3 - X AXIS LOGARITHMIC - Y AXIS LINEAR
4 - X AXIS LOGARITHMIC - Y AXIS LOGARITHMIC

The default values of these parameters make both axes linear.

If the parameters "X/LOGARITHMIC." and "Y/LOGARITHMIC." are reset by the routine DISPLA, they are given values which makes them immune to being reset when "SET." = 2, 3, or 4 (see the discussion of the argument LSET, above).

In calls to the routines AGSETP, AGSETF, AGSETI, AGGETP, AGGETF, and AGGETI:

TPGN is a character string identifying a group of AUTOGRAPH control parameters. It is of the form "K1/K2/K3/.../KN." Each KI is a keyword. The keyword K1 specifies a group of control parameters, K2 a subgroup of that group, K3 a subgroup of that subgroup, etc.

FURA is an array, from which control parameter values are to be taken (the routine AGSETP) or into which they are to be stored (the routine AGGETP). Note that the array is real: all of the AUTOGRAPH parameters are stored internally as reals.

LURA is the length of the user array FURA.

FUSR is a variable from which a single control parameter value is to be taken (the routine AGSETF) or into which it is to be stored (the routine AGGETF). Note that the variable is real.

IUSR is a variable from which a single control parameter value is to be taken (the routine AGSETI) or into which it is to be stored (the routine AGGETI). Note that since the control parameters are stored internally as reals, each of the routines AGSETI and AGGETI does a conversion from integer to real or vice-versa.
Routine AGSTUP

In calls to the routine AGSTUP:

**XDRA** is an array of X coordinates of user data. Usually, but not necessarily, the same data which will later be used in calls to the routine AGCURV.

**NVIX** is the number of vectors of data in XDRA. If XDRA is doubly-dimensioned, NVIX would normally have the value of its second dimension, if XDRA is singly-dimensioned, NVIX should be a 1.

**IIVX** is the index increment between vectors in XDRA. If XDRA is doubly-dimensioned, IIVX would normally have the value of its first dimension. If XDRA is singly-dimensioned, IIVX should have a dummy value.

**NEVX** is the number of elements in each data vector in XDRA. If XDRA is doubly-dimensioned, NEVX would normally have the value of its first dimension. If XDRA is singly-dimensioned, it has the value of that single dimension.

**IIEX** is the index increment between elements of a data vector in XDRA, normally a 1.

**NOTE:** YDRA, NVIY, IIVY, NEVY, and IIEY are analogous to XDRA, NVIX, IIVX, NEVX, and IIEX, but define Y-coordinate data.

Routine AGCURV

In calls to the routine AGCURV:

**XVEC** is a vector of X-coordinate data.

**IIEX** is the index increment between elements in XVEC. AGCURV will use XVEC(1), XVEC(1+IIEX), XVEC(1+2*IIEX), etc.

**YVEC** is a vector of Y-coordinate data.

**IIEY** is the index increment between elements in YVEC. AGCURV will use YVEC(1), YVEC(1+IIEY), YVEC(1+2*IIEY), etc.

**NEXY** is the number of points defining the curve to be drawn.

**KDSH** is a dashed-line selector. Possible values are:
= 0 If KDSH is zero, AUTOGRAPH will assume that the user has called the routine DASHD (see the DASHCHAR package) to define the dashed-line pattern to be used.

< 0 If KDSH is less than zero and has an absolute value M, AUTOGRAPH will use the Mth (MODULO 26) alphabetic dashed line pattern. Each of these patterns defines a solid line interrupted every so often by a letter of the alphabet.

> 0 If KDSH is greater than zero and has the value M, AUTOGRAPH will use the Mth (MODULO N) dashed-line pattern in the group of N dashed-line patterns defined by the AUTOGRAPH control parameters in the group named "DASH.". The default values of these parameters specify a single solid-line pattern.

ENTRY POINTS

Except for seven routines which are included in the package for historical reasons (EZY, EZXY, EZMY, ESMXY, IDIOT, ANOTAT, and DISPLA), the AUTOGRAPH routines have six-character names beginning with the prefix "AG". An alphabetized list follows:

AGAXIS AGBACK AGCPYL AGCTKO AGCURV AGDASH
AGDFLT AGEXAX AGEXUS AGFTOL AGGETF AGGETI
AGGETP AGGINIT AGKURV AGLBS AGMAXI AGMINI
AGNUMB AGQURV AGSCAN AGSETF AGSETI AGSETP
AGSRCH AGSTUP AGXTOX

NOTE The routine AGDFLT is a block-data routine containing the default values of AUTOGRAPH control parameters.

SPECIAL CONDITIONS

Under certain conditions, AUTOGRAPH may print an error message (via the routine ULIBER) and stop. Each error message includes the name of the routine which issued it. A description of the condition which caused the error may be found in the complete AUTOGRAPH documentation in Appendix A.

For error messages issued by the routine AGNUMB, see the write-up of the routine AGSTUP.

If you get an error in the routine ALOG10, this probably means that you are using a log axis and some of the coordinate data along that axis is zero or negative.

COMMON BLOCKS

The AUTOGRAPH common blocks are AGCONP, AGMCHP, and AGIBSP. AGCONP contains the AUTOGRAPH control parameters and is 458 words in length. AGMCHP contains machine-dependent constants and is eight words long. AGIBSP is used for communication between AGSCAN and AGSRCH and is three words long.

AUTOGRAPH - 9
**I/O**

No I/O is done directly by AUTOGRAPH. It calls routines in the System Plot Package to produce graphic output and it may call the routine ULIBER to write error messages to the printer.

**REQUIRED ULIB Routines**

AUTOGRAPH uses the software dashed-line package DASHCHAR, for which the access card is:

```
*FORTRAN,S=ULIB,SN=DASHCHAR
```

Of course, either of the packages DASHSMTH or DASHSUPR may be used instead to get smoother curves.

**LANGUAGE**

FORTTRAN

**HISTORY**

AUTOGRAPH represents the culmination of many years of software development designed to provide users with a wide range of graphing capabilities. The original routine, IDIOT, was written by Dave Robertson in the early 1960's. It was intended to provide users with a simple X-Y graph drawing capability. In time, as users began to adapt IDIOT to perform more sophisticated tasks, it became apparent that additional software capabilities were required. In 1974, Dan Anderson undertook a complete revision of IDIOT and produced the first AUTOGRAPH package. It allowed the user to put more than one curve on a graph, to use more sophisticated backgrounds, to specify coordinate data in a variety of ways, and to more easily control the scaling and positioning of graphs. As users became familiar with this version of AUTOGRAPH, suggestions for its improvement and enhancement necessitated yet another revision of the program. As a consequence, in 1977, Dave Kennison entirely re-wrote AUTOGRAPH. This latest version incorporates many desireable new features while maintaining the ease of use for generating simple graphs—the principal virtue of the original package. At the same time AUTOGRAPH still provides users with as much control as possible in formatting and producing graphic output. In keeping with the overall philosophy of the Graphics Project at NCAR, AUTOGRAPH is highly portable.

**SPACE REQUIRED**

AUTOGRAPH is big— the price one pays for its capabilities. On the 7600 it occupies about 23000 (OCTAL) or 9500 (DECIMAL) locations.

**PORTABILITY**

AUTOGRAPH may be ported with few modifications to most systems having a FORTRAN compiler which meets the ANSI-66 standard. Possible problem areas are as follows:

1. The block-data routine AGDFLT contains definitions for eight device-dependent constants, all of which are declared in the labeled common block AGMCHIP. These may or may not have to be changed, depending on
the graphics device being used.

2. The Hollerith constant "1H$" (one-aitch-dollar sign) appears in two different data statements (one in AGDFLT and one in AGCURV). The first of these defines the default "LINE/END." character and the second defines the code for a solid-line segment in character string dashed-line patterns. Some compilers will not allow the use of a dollar sign.

3. The labeled common blocks AGCONP and AGMCHP may have to be declared in a portion of the user program which is always core-resident so that variables in them will maintain their values between one call to AUTOGRAPH and the next. This problem may arise when AUTOGRAPH is placed in an overlay or when some sort of memory-paging scheme is used.

4. When an AGSETP call is executed to define a mapping function, a dashed-line pattern array, or the text of a label line, AUTOGRAPH saves only the memory address of the entity defined (obtained by use of the function "LOC"). During following calls, out-of-array indexing is used to access the function or array. On a machine with dynamic storage allocation, this scheme would not work. It is difficult to say what changes would have to be made in the package. The variables in which addresses are saved are all in the labeled common block AGCONP and are as follows:

QUIML(I), FOR I FROM 1 TO 4
QADP
FLLN(4,J), FOR J FROM 1 TO 16

A few AUTOGRAPH routines are called with arguments whose type does not match that declared in the routine itself. This should work unless the compiler passes arguments of different types in different ways.

5. Hollerith label names are treated by AUTOGRAPH as if they were floating-point. If this causes underflow, overflow or some other heinous problem, all user programs should use floating-point numbers for the label names and the names of the pre-defined labels "L", "R", "B", and "T" should be changed to floating-point numbers (in the block data routine AGDFLT).

REQUIRED RESIDENT Routines

AUTOGRAPH uses the following resident routines:

1. DASHCHAR routines:

DASHD FRSTD LASTD LINED VECTED
2. The System Plot Package routines:

   FLUSH FRAME GETSET GETSI LINE
   OPTN PWRIT SET

3. The System Plot Package support routines:

   GETCHR LOC SETCHR ULIBER

4. The FORTRAN library routines:

   ALOG10 ATAN2 COS SIN SQRT
SAMPLE PLOT

DEMONSTRATING EZY ENTRY OF AUTOGRAPH

AUTOGRAPH - 13
DEMONSTRATING EZMXY ENTRY OF AUTOGRAPH

$y = x^3 + \cos(x)$

$A$, $B$, $C$, $D$, $E$
CONRAN

SUBROUTINE CONRAN(XD, YD, ZD, NDP, WK, IWK, SCRARR)
STANDARD AND SMOOTH VERSIONS OF CONRAN

DIMENSION OF ARGUMENTS

XD(NDP), YD(NDP), ZD(NDP), WK(13*NDP)
IWK((27+NCP)*NDP)
NCP = 4 by default
SCRARR(Resolution**2) where resolution is described in
the SSZ option below. Resolution is 40 by default.

LATEST REVISION

August 1981

OVERVIEW

CONRAN performs contouring of irregularly distributed data. It is used for the standard and smooth members of the CONRAN family. This version will plot contours and smooth them using splines under tension (if the DASHSMTH package is loaded), plot a perimeter or grid, title, produce a message on map variables, plot the input data on the map and label the contour lines.

PURPOSE

CONRAN plots contour lines using random, sparse or irregular data sets. The data is triangulated and then contoured. Contouring is performed using interpolation of the triangulated data.

USAGE

CALL CONRAN(XD, ZD, NDP, WK, IWK, SCRARR)

An option setting routine can also be invoked, see write-up below. FRAME must be called by the user.

ARGUMENTS

On Input

XD
Array of dimension NDP containing the X-coordinates of the data points.

YD
Array of dimension NDP containing the Y-coordinates of the data points.

ZD
Array of dimension NDP containing the data values at the points.

NDP
Number of data points (must be 4 or greater)

WK
Real work array of dimension at least 13*NDP

IWK
Integer work array of dimension at least
IWK((27+NCP)*NDP). NCP = 4 by default.

SCRARR
Real work array of dimension at least (RESOLUTION**2) where resolution is described in the SSZ option below. Resolution is 40 by default.

On Output
All arguments remain unchanged except the scratch arrays IWK, WK, and SCRARR which have been written into. If making multiple runs on the same triangulation IWK and WK must be saved and returned to the next invocation of CONRAN.

ENTRY POINTS
CONRAN, CONDET, CONINT, CONCAL, CONLOC, CONTNG, CONDRW, CONSET, CONCLS, CONSTP, CONSWV, CONBDN, CONTLK, CONPDV, CONOP1, CONOP2, CONOP3, CONOP4, CONXCH, CONREQ, CONCOM, CONCLB, CONCLD, CONPMN, CONGEN, CONLOD, CONECD, CONOUT, CONOT2

COMMON BLOCKS
CONRA1, CONRA2, CONRA3, CONRA4, CONRA5, CONRA6, CONRA7, CONRA8, CONRA9, CONRA10, CONRA11, CONRA12, INTFR (from the DASH package)

I/O
Plots the contour map and, via PORT library routines, outputs messages to the message output unit. At NCAR this unit is the printer. The option values are all listed on the standard PORT output unit. At NCAR this unit is the printer.

PRECISION
Single

REQUIRED LIBRARY Routines
Standard version—DASHCHAR which, at NCAR, is loaded by default.
Smooth version—DASHSMTH which must be requested at NCAR.
Both versions require the PORT library error routines.

LANGUAGE
1966 ANSI FORTRAN

ALGORITHM
The sparse data is triangulated and a virtual grid is laid over the triangulated area. Each virtual grid point receives an interpolated value. The grid is scanned once for each contour level and all contours at that level are plotted. The triangulation and interpolation scheme is based on Lawson's C1 surface interpolation algorithm, which has been refined by Hirosha Akuma. Parts of Akima's algorithm are used in this package. See reference heading for publications.

PORTABILITY
CONRAN will run on NCAR's CRAY-1 and 7600. It is designed to run on most other machines by including some required local routines which are described in the Implementor's Guide.
CALL CONRAN (XD, YD, ZD, NDP, WK, IWK, SCRARR)

FRAME must be called by the user.

Listed below are options which can enhance your plot.

Calling the CONOP routine prior to the CONRAN call can activate or deactivate the options. Each option command must be invoked with separate CONOP calls. There are 4 CONOP routines: CONOP1, CONOP2, CONOP3, CONOP4. The numerical value of the name determines the number of parameters in the calling sequence. If you want to plot a perimeter use CONOP1. To change the data output format use CONOP4. Only the first two characters on each side of the equal sign are scanned. Therefore only 2 characters for each option are required on input to CONOP (i.e. 7HSCA=PRI or 5HSC=PR would both be accepted for setting the scaling option to the prior mode). Remember that the number of input points must be at least 4. This is equal to the default number of data points to be used for estimation of partial derivatives at each data point. NCP contains the number of these points to be used. The estimated partial derivatives are used for the construction of the interpolating polynomial's coefficients. NOTE: option settings remain in effect between calls to any CONRAN entry point.

OPTIONS

CHL. This flag determines how the high and low contour values are set. These contour values may be set by the program or by the user. If CHL=OFF, the program examines the user's input data and uses both the high and low values. If CHL=ON, the user must specify the desired high (hi) and low (flo) value should he wish to use different contour values from his input data. The default is CHL=OFF.

If Program Set: CALL CONOP1(7HCHL=OFF)
If User Set: CALL CONOP3(6HCHL=ON, hi, flo)

Example: CALL CONOP3(6HCHL=ON, 5020, 2000.) where the hi value desired is 5020 and the flo value desired is 2000. These are floating point numbers.

Note: The values supplied for contour increment and contour high and low values assumes the unscaled data...
values. See the SDC flag, below.

**CIL**

This flag determines how the contour increment (cinc) is set. The increment is either calculated by the program (CIL=OFF) using the range of high and low values from the user's input data, or set by the user (CIL=ON).

If Program Set: CALL CONOP1(7HCIL=OFF)

If User Set: CALL CONOP2(6HCIL=ON,cinc)

Example: CALL CONOP2(6HCIL=ON,15.) where 15. represents the contour increment desired by the user. This is a floating point number.

*Note:* The default option is CIL=OFF. By default, the program will examine the user's input data and determine the contour interval (cinc) at some appropriate range between the level of high and low values supplied, usually generating between 15 and 20 contour levels.

**CON**

This flag determines how the contour levels are set. If CON=ON, the user must specify the array and number of contour levels (ncl). A maximum of 30 contour levels are permitted for both array and ncl. If CON=OFF, default values are used. In this case, the program will calculate the values for the array and ncl using input data.

If Program Set: CALL CONOP1(7HCON=OFF)

If User Set: CALL CONOP3(6HCON=ON,array,ncl)

Example:
DATA RLIST(1),RLIST(2),...,RLIST(5)/1.,2.,3.,0.,12./
CALL CONOP3(6HCON=ON,RLIST,5) where 'RLIST' is the array of contour levels, and 5 is the number of user specified contour levels (ncl). The array must contain floating point numbers and the count (ncl) must be an integer.

*Note:* The array (array) contains the contour levels (floating point only) and ncl is the number of levels. The maximum number of contour levels allowed is 30.

**WARNING ON CONTOUR OPTIONS:**
It is illegal to use the CON option when either CIL or CHL are activated. In this error condition the option call that detected the error will NOT be executed. When assigning the contour array, it must be ordered from smallest to largest.
**DAS**

This flag determines which contours are represented by dashed lines. The user sets the dashed line pattern. The user may specify that dashed lines be used for contours whose value is less than, equal to, or greater than the dash pattern break point (see bp in the DBP option below), which is zero by default. If DAS=OFF (the default value), all solid lines are used.

All Solid Lines: CALL CONOP1(7HDAS=OFF)

If Greater: CALL CONOP2(7HDAS=GTR,ipat)

If Equal: CALL CONOP2(7HDAS=EQU,ipat)

If Less: CALL CONOP2(7HDAS=LSS,ipat)

If All Same: CALL CONOP2(7HDAS=ALL,ipat)

**Note:** ipat must be a ten character Hollerith string with a dollar sign ($) for DASH and a single quote ('') for blank. This is not the repetition count for line labels. That is controlled by an internal parameter, NREP, found in common block CONR11.

Example: CALL CONOP2(7HDAS=GTR,10H$$$$$$$$$$)

**DBP**

This flag determines how the dash pattern break point (bp) is set. If DBP=ON, bp must be set by the user by specifying bp. If DBP=OFF the program will set bp at the default value which is zero.

If Program Set: CALL CONOP1(7HDBP=OFF)

If User Set: CALL CONOP2(6HDBP=ON,bp)

Example: CALL CONOP2(6HDBP=ON,5.) where 5. is the user specified break point.

**Note:** bp is a floating point number where the break for GTR and LSS contour DASH patterns are defined. bp is assumed to be given relative to the untransformed contours.

**DEF**

Reset flags to default values. Activating this option sets all flags to the default value. DEF has no 'ON' of 'OFF' states.

To Activate: CALL CONOP1(3HDEF)

**EXT**

Flag to set extrapolation.

To turn ON: CALL CONOP1(6HEXT=ON)
Note: Normally all CONRAN versions will only plot the boundaries of the convex hull defined by the users data to fill the rectangular area of the frame. Use the EXT switch.

FMT Flag for the format of the plotted input data values. If FMT=OFF, the default values for \( f_t \), \( l \), and \( if \) are used. The default values are:

\[
f_t = (G10.3)\\
l = 7 \text{ characters including the parentheses}\\
if = 10 \text{ characters printed in the output field by the format}
\]

If FMT=ON, the user must specify values for \( f_t \), \( l \), and \( if \). All user specified values MUST be given in the correct format.

If Program Set: CALL CONOP1(7HFM=OFF)

If User Set: CALL CONOP4(6HFM=ON, \( f_t \), \( l \), \( if \))

Example: CALL CONOP4(6HFM=ON,(G30.2),7,30)

Note: \( f_t \) is a Hollerith String containing the format. The format must be enclosed by parentheses. Any format allowed at your installation (up to 10 characters) will be accepted. \( l \) is the number of characters in \( f_t \). \( if \) is the length of the field created by the format. The numerical parameters are type integer.

WARNING: CONRAN WILL NOT TEST FOR A VALID FORMAT AND THE FORMAT IS ONLY ALLOWED TO BE 10 CHARACTERS LONG.

GRI Flag for the Grid.

To turn ON: CALL CONOP1(6HGRI=ON)

To turn OFF: CALL CONOP1(7HGRI=OFF)

Note: If the flag is ON, the X and Y tick interval will be given. This is the interval in user coordinates that each tick mark represents.

INT Flag to determine the intensities of the contour lines and other parts of the plot. If INT=OFF, \( ival \) is the default value. If INT=ON, the user must specify \( ival \).

If Program Set: CALL CONOP1(7HINT=OFF)

Major Lines: CALL CONOP2(7HINT=MAJ, \( ival \))
Minor Lines: CALL CONOP2(7HINT=MIN, ival)
Title and Message: CALL CONOP2(7HINT=LAB, ival)
Data Values: CALL CONOP2(7HINT=DAT, ival)

Note: This relates to the plotted data values and the plotted maximums and minimums.

All the Same: CALL CONOP2(7HINT=ALL, ival)
Example: CALL CONOP2(7HINT=ALL, 110)

Note: ival is the intensity desired. For an explanation of the option value settings see the OPTN routine in the System Plot Package documentation. Briefly, ival values range from 0 to 255 or the character strings 2HLO and 2HHI. The default is 2HHI except for INT=MIN which is set to 2HLO. Numerical values are type integer.

LAB
This flag can be set to either label the contours (LAB=ON) or not (LAB=OFF). The default value is LAB=ON.

To Turn On: CALL CONOP1(6HLAB=ON)
To Turn Off: CALL CONOP1(7HLAB=OFF)

LOT
Flag to list options on the printer. The default value is set to OFF, and no options will be displayed.

To turn ON: CALL CONOP1(6HLOT=ON)
To turn OFF: CALL CONOP1(7HLOT=OFF)

Note: If users want to print the option values, they should turn this option ON. The option values will be sent to the standard output unit as defined by the port routine IMACH.

LSZ
This flag determines the label size. If LSZ=OFF, the default isizelsz value will be used. If LSZ=ON, the user should specify isizelsz. The default value is 9 PWRIT units.

If Program Set: CALL CONOP1(7HLSZ=OFF)
If User Set: CALL CONOP2(6HLSZ=ON, isizelsz)
Example: CALL CONOP2(6HLSZ=ON, 4) where 4 is the user desired integer PWRIT units.

Note: isizelsz is the requested character size in integer PWRIT units.
MES  Flag to plot a message.
To turn ON:  CALL CONOP1(6HMES=ON)
To turn OFF: CALL CONOP1(7HMES=OFF)
Note: If MES=ON a message is printed below the plot giving contour intervals and execution time in seconds. If PER or GRI are ON, the message also contains the X and Y tick interval.

MIN  Flag for setting the number of minor lines between each major line. Minor lines are low intensity and unlabeled. Major lines are high intensity and labeled. The default (set when option is turned off) is two minor lines between each major line.
To turn ON:  CALL CONOP2(6HMIN=ON,micnt).
To turn OFF: CALL CONOP1(7HMIN=OFF)
micnt is a type integer data value.

NCP  Flag for data points used for partial derivative estimation. If NCP=OFF, num is set to 4, which is the default value. If NCP=ON, the user must specify num greater than or equal to 2.
If Program Set:  CALL CONOP1(7HNCP=OFF)
If User Set:  CALL CONOP2(6HNCP=ON,num)
Example:  CALL CONOP2(6HNCP=ON,3)
Note: num = number of data points used for estimation. Changing this value affects the contours produced and the size of input array ITWK. num is type integer.

PDV  Flag to plot the input data values. The default value is PDV=OFF.
To turn ON:  CALL CONOP1(6HPDV=ON)
To turn OFF: CALL CONOP1(7HPDV=OFF)
Note: If PDV=ON, the input data values are plotted relative to their location on the contour map. If you only wish to see the locations and not the values, set PDV=ON and change FMT to produce an asterisk (*) such as (I1).

PER  Flag to set the perimeter. The default value is PER=ON, which causes a perimeter to be drawn around the contour plot.
To turn ON: CALL CONOP1(6HPER=ON)
To turn OFF: CALL CONOP1(7HLPER=OFF)

**Note:** If MES is ON, the X and Y tick interval will be given. This is the interval in user coordinates that each tick mark represents.

**PM**
Flag to plot relative minimums and maximums.

To Turn Off: CALL CONOP1(7HPMM=OFF)
To Turn On: CALL CONOP1(6HPMM=ON)

**REP**
Flag indicating the use of the same data, but a new execution. The default value is set to OFF.

To turn ON: CALL CONOP1(6HREP=ON)
To turn OFF: CALL CONOP1(7HREP=OFF)

**Note:** If REP=ON, the same data and triangulation are to be used but it is assumed the user has changed contour values or resolution for this run. Scratch arrays WK and IWK must remain unchanged.

**SC**
Flag for scaling of the plot on a frame.

User Scaling: CALL CONOP1(7HSCA=OFF)
Program Scaling: CALL CONOP1(6HSCA=ON)
Prior Set Call: CALL CONOP1(7HSCA=PRI)

**Note:** With SCA=OFF, plotting instructions will be issued using the user's input coordinates, unless they are transformed via FX and FY transformations. Users will find an extended discussion in the Interfacing to Other Graphics Routines section below. The SCA option assumes the user has previously used a SET call and that all input data falls into the range of that SET call. With SCA=ON, the entry point will perform the SET call so that the user's plot will fit into the center 90 percent of the frame. When SCA=PRI, the program maps the user's plot instructions into the portion of the frame defined by the last SET call. SCA=PRI should be used when interfacing to SUPMAP.

**SDC**
Flag to determine how to scale the data on the contours. If SDC=OFF, the floating point value is given by scale. If SDC=ON, the user may specify scale (see Note below.)
If Program Set: CALL CONOP1(7HSDC=OFF)
If User Set: CALL CONOP2(6HSDC=ON, scale)
Example: CALL CONOP2(6HSDC=ON, 100.)

**Note:** The data plotted on contour lines and the data plotted for relative minimums and maximums will be scaled by the floating point value given by `scale`. Typical scale values are 10., 100., 1000., etc. The original data values are multiplied by `scale`. It must be a floating point number and is displayed in the message (see MES). The default value for `scale` is 1.

**SML** Flag to determine the size of minimum and maximum labels. If SML=OFF, the value of `isizesml` is 15, which is the default. If SML=ON, the user must specify `isizesml`.

If Program Set: CALL CONOP1(7HSML=OFF)
If User Set: CALL CONOP1(6HSML=ON, `isizesml`)
Example: CALL CONOP1(6HSML=ON, 12)

**Note:** `isizesml` is an integer number which is the size of labels in PWRIT units.

**SPD** Flag for the size of the plotted input data values. If SPD=OFF, the value of `isizespd` is 8, which is the default. If SPD=ON, the user must specify `isizespd`.

If Program Set: CALL CONOP1(7HSPD=OFF)
If User Set: CALL CONOP2(6HSPD=ON, `isizespd`)
Example: CALL CONOP2(6HSPD=ON, 6)

**Note:** `isizespd` is an integer number giving the size of the data values in PWRIT units.

**SSZ** Flag to determine the resolution (number of steps in each direction). If SSZ=ON, the user sets `istep`, or, if SSZ=OFF, the program will automatically set `istep` at the default value of 40.

If Program Set: CALL CONOP1(7HSSZ=OFF)
If User Set: CALL CONOP2(6HSSZ=ON, `istep`)
Example: CALL CONOP2(6HSSZ=ON, 25) This `istep` value will produce a coarse contour. See note below.
**Note:** istep = type integer. t is the density of the virtual grid. In most cases, the default value of 40 produces pleasing contours. For coarser but quicker contours, lower the value. For smoother but longer time, raise the value. NOTE: For step sizes greater than 200 in CONRAQ, the array OV in common CONRA1, and ITLOC (not in the quick version) in common CONRA9 must be expanded to about 10 more than the size of SSZ. See CONRA1 and CONRA9 comments below for more information.

**STL** Flag to determine the size of the title. isizestl may be set by the user (STL=ON), or the program will set it at the default size of 16 PWRIT units (STL=OFF).

If Program Set: CALL CONOP1(STL=OFF)
If User Set: CALL CONOP2(STL=ON,isizestl)
Example: CALL CONOP2(STL=ON,13)

**Note:** When 30 or 40 characters are used for the title, the default size of 16 PWRIT units works well. When titles are larger, a smaller title size will be required. Title size is an integer value.

**TEN** Flag to determine the tension factor applied when smoothing contour lines. The user may set tens or allow the program to set the value. If user set, tens must have a value greater than zero and less than or equal to 30. The default value is 2.5.

If Program Set: CALL CONOP1(HTEN=OFF)
If User Set: CALL CONOP2(HTEN=ON,tens)
Example: CALL CONOP2(HTEN=ON,14.)

**Note:** tens is not available in the standard version of CONRAN. Tension is a floating point number.

Smoothing of contour lines is accomplished with splines under tension. To adjust the amount of smoothing applied, adjust the tension factor. The tension must be greater than ZERO. The default is 2.5. Setting tens very large (i.e. 30), effectively shuts off smoothing.

**TFR** Flag to advance frame before triangulation. The default value is TFR=ON.

If Program Set: CALL CONOP1(TFR=ON)
To turn OFF: CALL CONOP1(TFR=OFF)
**Note:** Triangles are plotted after the contouring is completed. If the user wished to see the triangles over the contours, turn this switch OFF.

**TLE** Flag to place a title at the top of the plot. If TLE=ON, the user must specify ichars and inum. The default value is OFF. *ichars* is the Hollerith character string for the title. *inum* is the number of characters in *ichars*.

To turn ON: \( \text{CALL CONOP3}(6\text{HTLE}=\text{ON},\text{ichars},\text{inum}) \)

Example: \( \text{CALL CONOP3}(6\text{HTLE}=\text{ON},13\text{HVVECTOR REVIEW},13) \)

To turn OFF: \( \text{CALL CONOP1}(7\text{HTLE}=\text{OFF}) \)

**Note:** The title array is 32 words long, so the maximum title is 32 times the number of characters per word on your machine. Longer titles will require increasing the size of array ISTRNG found in CONRA7. *inum* is an integer value.

**TOP** Flag to plot triangle only.

To Turn Off: \( \text{CALL CONOP1}(7\text{HTOP}=\text{OFF}) \)

To turn ON: \( \text{CALL CONOP1}(6\text{HTOP}=\text{ON}) \)

**Note:** The user may wish to overlay the triangles on some other plot. This option ON will allow that. This option when activated (TOP=ON), will set TRI=ON, and TFR=OFF. If the user wants TFR=ON, it should be set after TOP is set. If the user sets TOP=OFF it will set TRI=OFF and TFR=ON. If the user wants TRI and TFR different, then set them after the TOP call.

**TRI** Flag to plot the triangulation.

To turn ON: \( \text{CALL CONOP1}(6\text{HTRI}=\text{ON}) \)

To turn OFF: \( \text{CALL CONOP1}(7\text{HTRI}=\text{OFF}) \)

**Note:** Plotting the triangles will indicate to the user where good and bad points of interpolation are occurring in the contour map. Equilateral triangles are optimal for interpolation. Quality degrades as triangles approach a long and narrow shape. The convex hull of the triangulation is also a poor point of interpolation.
Optional Default Values

Below are listed the program set default values for the various options given above. Unless the user specifies otherwise, these values will be used in execution of the various options.

CHL=OFF  LAB=ON  SDC=OFF
CIL=OFF  LOT=OFF  SML=OFF
CON=OFF  LSZ=OFF  SPD=OFF
DAS=OFF  MES=ON  SSZ=OFF
DBP=OFF  MIN=OFF  STL=OFF
EXT=OFF  NCP=OFF  TEN=OFF
FMT=OFF  PDV=OFF  TFR=ON
GRI=OFF  PER=ON  TLE=OFF
LAB=ON  PMM=OFF  TOP=OFF
LOT=OFF  REP=OFF

Default Values for User Specified Parameters

The OPTION DEFAULT VALUES given above, if used, will set default values for the following parameters:

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DEFAULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>array</td>
<td>Up to 30 contour levels allowed. Values are computed by the program, based on input</td>
</tr>
<tr>
<td>bp</td>
<td>0. Note: there is no dash break point in CONRAQ.</td>
</tr>
<tr>
<td>cinc</td>
<td>Computed by program based on the range of hi and lo values of the input data.</td>
</tr>
<tr>
<td>flo</td>
<td>Computed by program based on the lowest unscaled input data.</td>
</tr>
<tr>
<td>ft</td>
<td>(G10.3) Parentheses must be included.</td>
</tr>
<tr>
<td>hi</td>
<td>Computed by program based on the highest unscaled input data.</td>
</tr>
<tr>
<td>minct</td>
<td>two minor lines</td>
</tr>
<tr>
<td>ichars</td>
<td>No title</td>
</tr>
<tr>
<td>if</td>
<td>10 characters</td>
</tr>
<tr>
<td>inum</td>
<td>No title</td>
</tr>
<tr>
<td>ipat</td>
<td>10H$$$$$$$$$$ This is a 10 character Hol-lerith string.</td>
</tr>
<tr>
<td>isizelsz</td>
<td>9 PWRIT units</td>
</tr>
</tbody>
</table>

CONRAN - 13
isizesml  15 PWRIT units
isizepsd  8 PWRIT units
isizestl  16 PWRIT units
istep  40
ival  2HHI for all except minor contour lines which are 2HLO.

1  7 characters including both parentheses
ncl  Computed by program based on input data. Up to 30 contour levels are permitted.
num  4 data points
scale  1 (no scaling performed)
tens  2.5 Note: Tension Option does not exist in standard CONRAN.

OPTIONS WHICH EFFECT THE CONTOURS

To create different styled contours use options NCP and SSZ. NCP will modify the interpolating functions and therefore cause changes in the plots produced. Increasing NCP causes more of the surrounding data to influence the point of interpolation. Some data sets cause difficulty when trying to produce meaningful contours (this is a useful feature when triangles are long and narrow). By modifying NCP and thereby changing the interpolation functions, a user can fine-tune a plot. Increasing SSZ will smooth out the contour lines and pick up more detail (new contours will appear as SSZ increases and old ones will sometimes break into more distinct units).

INTERFACING INTO OTHER GRAPHICS ROUTINES

The scaling option will be used in some way normally not in the SCA=ON mode. The statement functions, FX and FY can be modified to perform other mappings. Users are referred to Appendix B of the Graphics Utilities document for a complete description of FX and FY transformation. The routines having FX and FY transformations are:

CONDRW, CONPDV, CONTLK, CONPMM, CONGEN

In most cases mapping can be performed before calling a CONRAN entry point, saving the user from modifying the file. If reasonable results cannot be obtained then the statement functions will have to be replaced.
NOTE IF,NCP,GR 25,arrays DSQ0,and IPCO in CONDET must be adjusted accordingly. Also NCPSZ in CONBDN (25 by default), will have to equal NCP. The default value of NCP which is 4, produces pleasing pictures in most cases. However, fine tuning of the interpolation can be obtained by increasing the size of NCP, with a corresponding linear increase in work space. CONRAN calls SETI to guarantee 1-1024 integer range for PWRIT.

AUTHOR John Humbrecht NCAR, Boulder, Colorado

REFERENCES
Akima, Hirosha
A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points.
ACM Transactions on Mathematical Software
VOL 4, NO. 2, June 1978, Pages 148-159

Lawson, C.L.
Software for C1 Surface Interpolation
JPL Publication 77-30
August 15, 1977

CONRAN ERROR MESSAGES

<table>
<thead>
<tr>
<th>ERROR</th>
<th>ROUTINE</th>
<th>MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONRAN</td>
<td>INPUT PARAMETER NDP LT NCP</td>
</tr>
<tr>
<td>2</td>
<td>CONRAN</td>
<td>NCP GT MAX SIZE OR LT 2</td>
</tr>
<tr>
<td>3</td>
<td>CONTING</td>
<td>ALL COLINEAR DATA POINTS</td>
</tr>
<tr>
<td>4</td>
<td>CONTING</td>
<td>IDENTICAL INPUT DATA POINTS FOUND</td>
</tr>
<tr>
<td>5</td>
<td>CONSET</td>
<td>UNDEFINED OPTION</td>
</tr>
<tr>
<td>6</td>
<td>CONCLS</td>
<td>CONSTANT INPUT FIELD</td>
</tr>
<tr>
<td>7</td>
<td>CONSET</td>
<td>INCORRECT CONOP CALL USED</td>
</tr>
<tr>
<td>8</td>
<td>CONSET</td>
<td>ILLEGAL USE OF CON OPTION WITH CIL OR CHL OPTIONS</td>
</tr>
<tr>
<td>9</td>
<td>CONSET</td>
<td>NUMBER OF CONTOUR LEVELS EXCEEDS 30</td>
</tr>
<tr>
<td>10</td>
<td>CONDRW</td>
<td>CONTOUR STORAGE EXHAUSTED</td>
</tr>
</tbody>
</table>

The errors are all defined as recoverable in the port error processor. See the port error processing writeup for detailed usage. However no user action is required to execute CONRAN.
CONRAQ

SUBROUTINE CONRAQ(XD, YD, ZD, NDP, WK, IWK)

QUICK VERSION OF CONRAN

DIMENSION OF

ARGUMENTS

XD(NDP), YD(NDP), ZD(NDP), WK(13*NDP)

IWK((27+NCP)*NDP)

NCP = 4 by default

LATEST REVISION

August 1981

OVERVIEW

CONRAQ performs contouring of irregularly distributed data. It is the fastest member of the CONRAN family. This version will plot contours from your input data. It has the ability to draw a perimeter or grid, title the plot, produce a message on map variables, and plot the input data on the map. This version is well suited for testing of your data. Besides being the fastest version it also is the smallest.

PURPOSE

CONRAQ plots contour lines using random, sparse or irregular data sets. The data is triangulated and then contoured. Contouring is performed using interpolation of the triangulated data.

USAGE

CALL CONRAQ(XD, ZD, NDP, WK, IWK)

An option setting routine can also be invoked, see write-up below. FRAME must be called by the user.

ARGUMENTS

On Input

XD

Array of dimension NDP containing the X-coordinates of the data points.

YD

Array of dimension NDP containing the Y-coordinates of the data points.

ZD

Array of dimension NDP containing the data values at the points.

NDP

Number of data points (must be 4 or greater)

WK

Real work array of dimension at least 13*NDP

IWK

Integer work array of dimension at least IWK((27+NCP)*NDP). NCP = 4 by default.
On Output
All arguments remain unchanged except the scratch arrays IWK and WK which have been written into. If making multiple runs on the same triangulation IWK and WK must be saved and returned to the next invocation of CONRAN.

ENTRY POINTS
CONRAQ, CONDET, CONINT, CONCAL, CONLOC, CONTNG, CONDRW, CONSET, CONCLS, CONSTP, CONSWV, CONBDN, CONTLK, CONPDV, CONOP1, CONOP2, CONOP3, CONOP4, CONXCH, CONOUT, CONOT2

COMMON BLOCKS
CONRA1, CONRA2, CONRA3, CONRA4, CONRA5, CONRA6, CONRA7, CONRA8, CONRA9, CONR10, CONR11, CONR12, INTPR FROM THE DASH PACKAGE

I/O
Plots the contour map and, via PORT library routines, outputs messages to the message output unit. At NCAR this unit is the printer. The option values are all listed on standard PORT output unit, at NCAR this unit is the printer.

PRECISION
Single

REQUIRED LIBRARY ROUTINES
PORT library error routines.

LANGUAGE
1966 ANSI FORTRAN

ALGORITHM
The sparse data is triangulated and a virtual grid is laid over the triangulated area. Each virtual grid point receives an interpolated value. The grid is scanned once for each contour level and all contours at that level are plotted. The triangulation and interpolation scheme is based on Lawson's C1 surface interpolation algorithm, which has been refined by Hiroshi Akima. Parts of Akima's algorithm are used in this package. See reference heading for publications.

PORTABILITY
CONRAQ will run on NCAR's CRAY-1 and 7600. It is designed to run on most other machines by including some required local routines which are described in the implementor's writeup. This writeup is available from NCAR.

REQUIRED PLOTTING ROUTINES
FL2INT, GETSET, PERIM, PLOTIT, PWRIT, SET, OPTN, SETI, GRID

REQUIRED RESIDENT ROUTINES
GETCHR, SETCHR, ENCODE, ISHIFT, IAND, IOR, Port Error Handling Routines.

OPERATION
CALL CONRAQ (XD, YD, ZD, NDP, WK, IWK)
FRAME must be called by the user.
Listed below are options which can enhance your plot.

Calling the CONOP routine prior to the CONRAQ call can activate or deactivate the options. Each option command must be invoked with separate CONOP calls. There are 4 CONOP routines: CONOP1, CONOP2, CONOP3, CONOP4. The numerical value of the name determines the number of parameters in the calling sequence. If you want to plot a perimeter use CONOP1. To change the data output format use CONOP4. Only the first two characters on each side of the equal sign are scanned. Therefore only 2 characters for each option are required on input to CONOP (i.e. 7HSCA=PRI or 5HSC=PR would both be accepted for setting the scaling option to the prior mode). Remember that the number of input points must be at least 4. This is equal to the default number of data points to be used for estimation of partial derivatives at each data point. NCP contains the number of these points to be used. The estimated partial derivatives are used for the construction of the interpolating polynomial's coefficients. Listed are all the options available to CONRAQ. NOTE: option settings remain in effect between calls to any CONRAN entry point.

OPTIONS

CHL. This flag determines how the high and low contour values are set. These contour values may be set by the program or by the user. If CHL=OFF, the program examines the user's input data and uses both the high and low values. If CHL=ON, the user must specify the desired high (hi) and low (flo) value should he wish to use different contour values from his input data. The default is CHL=OFF.

If Program Set: CALL CONOP1(7HCHL=OFF)

If User Set: CALL CONOP3(6HCHL=ON,hi,flo)

Example: CALL CONOP3(6HCHL=ON,5020.,2000.) where the hi value desired is 5020 and the flo value desired is 2000. These are floating point numbers.

Note: The values supplied for contour increment and contour high and low values assumes the unscaled data values. See the SDC flag, below.

CIL. This flag determines how the contour increment (cinc) is set. The increment is either calculated by the program (CIL=OFF) using the range of high and low values from the user's input data, or set by the user (CIL=ON).

If Program Set: CALL CONOP1(7HCIL=OFF)
If User Set: CALL CONOP2(6HCIL=ON,cinc)

Example: CALL CONOP2(6HCIL=ON, 15.) where 15. represents the contour increment desired by the user. This is a floating point number.

**Note:** The default option is CIL=OFF. By default, the program will examine the user's input data and determine the contour interval (cinc) at some appropriate range between the level of high and low values supplied, usually generating between 15 and 20 contour levels.

**CON** This flag determines how the contour levels are set. If CON=ON, the user must specify the array and number of contour levels (ncl). A maximum of 30 contour levels are permitted for both array and ncl. If CON=OFF, default values are used. In this case, the program will calculate the values for the array and ncl using input data.

If Program Set: CALL CONOP1(7HCON=OFF)

If User Set: CALL CONOP3(6HCON=ON, array, ncl)

Example:

DATA RLIST(1), RLIST(2), ..., RLIST(5)/1., 2., 3., 10., 12./

CALL CONOP3(6HCON=ON, RLIST, 5) where 'RLIST' is the number of contour levels in the array, and 5 is the number of user specified contour levels (ncl). The array must contain floating point numbers and the count ncl must be integer.

**Note:** The array (array) contains the contour levels (floating point only) and ncl is the number of levels. The maximum number of contour levels allowed is 30.

**WARNING ON CONTOUR OPTIONS:**
It is illegal to use the CON option when either CIL or CHL are activated. In this error condition the option call that detected the error will NOT be executed. When assigning the contour array, it must be ordered from smallest to largest.

**DEF** Reset flags to default values. Activating this option sets all flags to the default value. DEF has no 'ON' of 'OFF' states.

To Activate: CALL CONOP1(3HDEF)
**EXT** Flag to set extrapolation.

To turn ON: CALL CONOP1(6HEXT=ON)

To turn OFF: CALL CONOP1(7HEXT=OFF)

*Note:* Normally all CONRAN versions will only plot the boundaries of the convex hull defined by the users data to fill the rectangular area of the frame. Use the EXT switch.

**FMT** Flag for the format of the plotted input data values. If FMT=OFF, the default values for \( ft, l, \) and \( if \) are used. The default values are:

\[

t = (G10.3) \\
1 = 7 \text{ characters including the parentheses} \\
If = 10 \text{ characters printed in the output field by the format}
\]

If FMT=ON, the user must specify values for \( ft, l, \) and \( if \). All user specified values **MUST** be given in the correct format.

If Program Set: CALL CONOP1(7HFMT=OFF)

If User Set: CALL CONOP4(6HFMT=ON,\( ft, l, if \))

Example: CALL CONOP4(6HFMT=ON,(G30.2),7,30)

*Note:* \( ft \) is a Hollerith String containing the format. The format must be enclosed by parenthesis. Any format allowed at your installation (up to 10 characters) will be accepted. \( l \) is the number of characters in \( ft \). \( if \) is the length of the field created by the format. The numerical parameters are type integer.

**WARNING:** CONRAQ WILL NOT TEST FOR A VALID FORMAT AND THE FORMAT IS ONLY ALLOWED TO BE 10 CHARACTERS LONG.

**GRI** Flag for the Grid.

To turn ON: CALL CONOP1(6HGRI=ON)

To turn OFF: CALL CONOP1(7HGRI=OFF)

*Note:* If the flag is ON, the X and Y tick interval will be given. This is the interval in user coordinates that each tick mark represents.

**INT** Flag to determine the intensities of the contour lines and other parts of the plot. If INT=OFF, \( ival \) is the default value. If INT=ON, the user must specify \( ival \).
If Program Set: CALL CONOP1(7HINT=OFF)

Major Lines: CALL CONOP2(7HINT=MAJ,ival)

**Note:** CONRAQ (the QUICK version) has no major or minor lines. All contour line intensities will be controlled by INT=MAJ.

Title and Message: CALL CONOP2(7HINT=LAB,ival)

Data Values: CALL CONOP2(7HINT=DAT,ival)

**Note:** This relates to the plotted data values and the plotted maximums and minimums.

All the Same: CALL CONOP2(7HINT=ALL,ival)

Example: CALL CONOP2(7HINT=ALL,110)

**Note:** ival is the intensity desired. For an explanation of the option value settings see the OPTN routine in the NCAR System Plot Package documentation. Briefly, ival values range from 0 to 255 or the character strings 2HLO and 2HHI. The default is 2HHI except for INT=MIN which is set to 2HLO. Numerical values are type integer.

**LOT** Flag to list options on the printer. The default value is set to OFF, and no options will be displayed.

To turn ON: CALL CONOP1(6HLOT=ON)

To turn OFF: CALL CONOP1(7HLOT=OFF)

**Note:** If users want to print the option values, they should turn this option ON. The option values will be sent to the standard output unit as defined by the PORT routine I1MACH.

**MES** Flag to plot a message.

To turn ON: CALL CONOP1(6HMES=ON)

To turn OFF: CALL CONOP1(7HMES=OFF)

**Note:** If MES=ON a message is printed below the plot giving contour intervals and execution time in seconds. If PER or GRI are ON, the message also contains the X and Y tick interval.

**NCP** Flag for data points used for partial derivative estimation. If NCP=OFF, num is set to 4, which is the default value. If NCP=ON, the user must specify num greater than or equal to 2.
If Program Set: CALL CONOP1(7HNCP=OFF)

If User Set: CALL CONOP2(6HNCP=ON,num)

Example: CALL CONOP2(6HNCP=ON,3)

Note: num = number of data points used for estimation. Changing this value effects the contours produced and the size of input array IWK. num is type integer.

PDV Flag to plot the input data values. The default value is PDV=OFF.

To turn ON: CALL CONOP1(6HPDV=ON)

To turn OFF: CALL CONOP1(7HPDV=OFF)

Note: If PDV=ON, the input data values are plotted relative to their location on the contour map. If you only wish to see the locations and not the values, set PDV=ON and change FMT to produce an asterisk (*) such as (I1).

PER Flag to set the perimeter. The default value is PER=ON, which causes a perimeter to be drawn around the contour plot.

To turn ON: CALL CONOP1(6HPER=ON)

To turn OFF: CALL CONOP1(7HLPER=OFF)

Note: If MES is ON, the X and Y tick interval will be given. This is the interval in user coordinates that each tick mark represents.

REP Flag indicating the use of the same data, but a new execution. The default value is set to OFF.

To turn ON: CALL CONOP1(6HREP=ON)

To turn OFF: CALL CONOP1(7HREP=OFF)

Note: If REP=ON, the same data and triangulation are to be used but it is assumed the user has changed contour values or resolution for this run. Scratch arrays WK and IWK must remain unchanged.

SCA Flag for scaling of the plot on a frame.

User Scaling: CALL CONOP1(7HSCA=OFF)

Program Scaling: CALL CONOP1(6HSCA=ON)
Prior Set Call: CALL CONOP1(7HSCA=PRI)

**Note:** With SCA=OFF, plotting instructions will be issued using the user’s input coordinates, unless they are transformed via FX and FY transformations. Users will find an extended discussion in the Interfacing to Other Graphics Routines section below. The SCA option assumes the user has previously used a SET call and that all input data falls into the range of that SET call. With SCA=ON, the entry point will perform the SET call so that the user’s plot will fit into the center 90 percent of the frame. When SCA=PRI, the program maps the user’s plot instructions into the portion of the frame defined by the last SET call. SCA=PRI should be used when interfacing to SUPMAP.

**SPD** Flag for the size of the plotted input data values. If SPD=OFF, the value of isizespd is 8, which is the default. If SPD=ON, the user must specify isizespd.

If Program Set: CALL CONOP1(7HSPD=OFF)

If User Set: CALL CONOP2(6HSPD=ON,isizespd)

Example: CALL CONOP2(6HSPD=ON,6)

**Note:** isizespd is an integer number giving the size of the data values in PWRIT units.

**SSZ** Flag to determine the resolution (number of steps in each direction). If SSZ=ON, the user sets istep, or, if SSZ=OFF, the program will automatically set istep at the default value of 40.

If Program Set: CALL CONOP1(7HSSZ=OFF)

If User Set: CALL CONOP2(6HSSZ=ON,istep)

Example: CALL CONOP2(6HSSZ=ON,25) This istep value will produce a coarse contour. See note below.

**Note:** istep =type integer=rt. It is the density of the virtual grid. In most cases, the default value of 40 produces pleasing contours. For coarser but quicker contours, lower the value. For smoother but longer time, raise the value. NOTE: For step sizes greater than 200 in CONRAQ, the array OV in common CONRA1, and ITLOC (not in the quick version) in common CONRA9 must be expanded to about 10 more than the size of SSZ. See CONRA1 and CONRA9 comments below for more information.
**STL** Flag to determine the size of the title. `isizestl` may be set by the user (STL=ON), or the program will set it at the default size of 16 PWRIT units (STL=OFF).

If Program Set: CALL CONOP1(STL=OFF)

If User Set: CALL CONOP2(STL=ON,isizestl)

Example: CALL CONOP2(STL=ON,13)

**Note:** When 30 or 40 characters are used for the title, the default size of 16 PWRIT units works well. When titles are larger, a smaller title size will be required. Title size is an integer value.

**TFR** Flag to advance frame before triangulation. The default value is TFR=ON.

If Program Set: CALL CONOP1(6HTFR=ON)

To turn OFF: CALL CONOP1(7HTFR=OFF)

**Note:** Triangles are plotted after the contouring is completed. If the user wished to see the triangles over the contours, turn this switch OFF.

**TLE** Flag to place a title at the top of the plot. If TLE=ON, the user must specify `ichars` and `inum`. The default value is OFF. `ichars` is the Hollerith character string for the title. `inum` is the number of characters in `ichars`.

To turn ON: CALL CONOP3(6HTLE=ON,ichars,inum)

Example: CALL CONOP3(6HTLE=ON,13HVECTOR REVIEW,13)

To turn OFF: CALL CONOP1(7HTLE=OFF)

**Note:** The title array is 32 words long, so the maximum title is 32 times the number of characters per word on your machine. Longer titles will require increasing the size of array ISTRNG found in CONRA7. `inum` is an integer value.

**TOP** Flag to plot triangle only.

To Turn Off: CALL CONOP1(7HTOP=OFF)

To turn ON: CALL CONOP1(6HTOP=ON)

**Note:** The user may wish to overlay the triangles on some other plot. This option ON will allow that. This option when activated (TOP=ON), will set TRI=ON, and TFR=OFF. If the user wants TFR=ON, it should be set after TOP is
set. If the user sets TOP=OFF it will set TRI=OFF and TFR=ON. If the user wants TRI and TFR different, then set them after the TOP call.

**TRI** Flag to plot the triangulation.

To turn ON: CALL CONOP1(6HTRI=ON)

To turn OFF: CALL CONOP1(7HTRI=OFF)

**Note:** Plotting the triangles will indicate to the user where good and bad points of interpolation are occurring in the contour map. Equilateral triangles are optimal for interpolation. Quality degrades as triangles approach a long and narrow shape. The convex hull of the triangulation is also a poor point of interpolation.

**OPTIMAL DEFAULT VALUES**

- CHL=OFF
- LOT=OFF
- SPD=OFF
- CIL=OFF
- MES=ON
- SSZ=OFF
- CON=OFF
- NCP=OFF
- STL=OFF
- EXT=OFF
- PDV=OFF
- TFR=ON
- FMT=OFF
- PER=ON
- TLE=OFF
- GRI=OFF
- REP=OFF
- TOP=OFF

**DEFAULT VALUES FOR USER SPECIFIED PARAMETERS**

The OPTION DEFAULT VALUES given above, if used, will set default values for the following parameters:

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DEFAULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>array</td>
<td>Up to 30 contour levels allowed. Values are computed by the program, based on input</td>
</tr>
<tr>
<td>cinc</td>
<td>Computed by program based on the range of hi and lo values of the input data.</td>
</tr>
<tr>
<td>flo</td>
<td>Computed by program based on the lowest unscaled input data.</td>
</tr>
<tr>
<td>ft</td>
<td>(G10.3) Parentheses must be included.</td>
</tr>
<tr>
<td>hi</td>
<td>Computed by program based on the highest unscaled input data.</td>
</tr>
<tr>
<td>ichars</td>
<td>No title</td>
</tr>
<tr>
<td>if</td>
<td>10 characters</td>
</tr>
<tr>
<td>inum</td>
<td>No title</td>
</tr>
<tr>
<td>isizelsz</td>
<td>9 PWRIT units</td>
</tr>
<tr>
<td>isizesml</td>
<td>15 PWRIT units</td>
</tr>
</tbody>
</table>
isizespd  8 PWRIT units
isizestl  16 PWRIT units
istep     40
ival      2HHI for all options.
l         7 characters including both parentheses
ncl       Computed by program based on input data. Up to 30 contour levels are permitted.
num       4 data points
scale     1 (no scaling performed)

OPTIONS WHICH EFFECT THE CONTOURS

To create different styled contours use options NCP and SSZ. NCP will modify the interpolating functions and therefore cause changes in the plots produced. Increasing NCP causes more of the surrounding data to influence the point of interpolation. Some data sets cause difficulty when trying to produce meaningful contours (this is a useful feature when triangles are long and narrow). By modifying NCP and thereby changing the interpolation functions, a user can fine-tune a plot. Increasing SSZ will smooth out the contour lines and pick up more detail (new contours will appear as SSZ increases and old ones will sometimes break into more distinct units).

INTERFACING INTO OTHER GRAPHICS ROUTINES

The scaling option will be used in some way normally not in the SCA=ON mode. The statement functions, FX and FY can be modified to perform other mappings. See the Graphics Utilities documentation for examples of interfaces to other graphics routines when using CONREC. The routines having FX and FY transformations are:

CONTOR, CONPDV, CONTLK

In most cases mapping can be performed before calling CONRAQ, saving the user from modifying the file. If reasonable results cannot be obtained then the statement functions will have to be replaced.

NOTE

If, NCP, GT 25 , arrays DSQO and IPCO in CONDET must be adjusted accordingly. Also NCPSZ in CONBDN (25 by default), will have to equal NCP. The default value of NCP which is 4, produces pleasing pictures in most cases. However, fine tuning of the interpolation can be obtained by increasing the size of NCP, with a corresponding linear increase in work space. CONRAQ calls SETI to
garantee 1-1024 integer range for PWRIT.

AUTHOR
John Humbrecht NCAR, Boulder, Colorado

REFERENCES
Akima, Hirosha
A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points.
ACM Transactions on Mathematical Software
VOL 4, NO. 2, June 1978, Pages 148-159

Lawson, C.L.
Software for C1 Surface Interpolation
JPL Publication 77-30
August 15, 1977

CONRAQ ERROR MESSAGES

<table>
<thead>
<tr>
<th>ERROR</th>
<th>ROUTINE</th>
<th>MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONRAQ</td>
<td>INPUT PARAMETER NDP LT NCP</td>
</tr>
<tr>
<td>2</td>
<td>CONRAQ</td>
<td>NCP GT MAX SIZE OR LT 2</td>
</tr>
<tr>
<td>3</td>
<td>CONTNG</td>
<td>ALL COLINEAR DATA POINTS</td>
</tr>
<tr>
<td>4</td>
<td>CONTNG</td>
<td>IDENTICAL INPUT DATA POINTS FOUND</td>
</tr>
<tr>
<td>5</td>
<td>CONSET</td>
<td>UNDEFINED OPTION</td>
</tr>
<tr>
<td>6</td>
<td>CONCLS</td>
<td>CONSTANT INPUT FIELD</td>
</tr>
<tr>
<td>7</td>
<td>CONSET</td>
<td>INCORRECT CONOP CALL USED</td>
</tr>
<tr>
<td>8</td>
<td>CONSET</td>
<td>ILLEGAL USE OF CON OPTION WITH CIL OR CHL OPTIONS</td>
</tr>
<tr>
<td>9</td>
<td>CONSET</td>
<td>NUMBER OF CONTOUR LEVELS EXCEEDS 30</td>
</tr>
</tbody>
</table>

The errors are all defined as recoverable in the port error processor. See the port error processing writeup for detailed usage. However no user action is required to execute CONRAQ.
CONRAS

SUBROUTINE CONRAS(XD, YD, ZD, NDP, WK, IWK, SCRARR)
SUPER VERSION OF CONRAN

DIMENSION OF ARGUMENTS

XD(NDP), YD(NDP), ZD(NDP), WK(13*NDP)
IWK((27+NCP)*NDP)
NCP = 4 by default
SCRARR(Resolution**2) where resolution is described in
the SSZ option below. Resolution is 40 by default.

LATEST REVISION
August 1981

OVERVIEW
CONRAS performs contouring of irregularly distributed
data. It is the super member of the CONRAN family. This
version will plot contours and smooth them using splines
under tension; plot a perimeter or grid; plot a title;
produce a message on map variables; plot the input data
on the map; label the contour lines and eliminate crowd-
ing of contour lines. This version is the biggest and
slowest member of the family. It is intended to produce
publication quality maps.

PURPOSE
CONRAS plots contour lines using random, sparse or irreg-
ular data sets. The data is triangulated and then con-
toured. Contouring is performed using interpolation of
the triangulated data.

USAGE
CALL CONRAS(XD, ZD, NDP, WK, IWK, SCRARR)
An option setting routine can also be invoked, see write-
up below. FRAME must be called by the user.

ARGUMENTS

On Input

XD
Array of dimension NDP containing the X-coordinates
of the data points.

YD
Array of dimension NDP containing the Y-coordinates
of the data points.

ZD
Array of dimension NDP containing the data values at
the points.

NDP
Number of data points (must be 4 or greater)

WK
Real work array of dimension at least 13*NDP
IWK

Integer work array of dimension at least IWK((27+NCP)*NDP). NCP = 4 by default.

SCRARR

Real work array of dimension at least (RESOLUTION**2) where resolution is described in the SSZ option below. Resolution is 40 by default.

On Output

All arguments remain unchanged except the scratch arrays IWK, WK, and SCRARR which have been written into. If making multiple runs on the same triangulation IWK and WK must be saved and returned to the next invocation of CONRAS.

ENTRY POINTS

CONRAS, CONDET, CONINT, CONCAL, CONLOC, CONTNG, CONDRW, CONSET, CONCLS, CONSTP, CONSWV, CONBDN, CONTLK, CONPDV, CONOP1, CONOP2, CONOP3, CONOP4, CONXCH, CONREO, CONCOM, CONCLE, CONCLD, CONPMS, CONGEN, CONLOD, CONECD, CONOUT, CONOT2

COMMON BLOCKS

CONRA1, CONRA2, CONRA3, CONRA4, CONRA5, CONRA6, CONRA7, CONRA8, CONRA9, CONR10, CONR11, CONR12, INTPR FROM THE DASH PACKAGE

I/O

Plots the contour map and, via PORT library routines, outputs messages to the message output unit. At NCAR this unit is the printer. The option values are all listed on standard PORT output unit, at NCAR this unit is the printer.

PRECISION

Single

REQUIRED LIBRARY ROUTINES

DASHSUPR, which must be requested.

All four versions require the PORT library error routines.

ALGORITHM

The sparse data is triangulated and a virtual grid is laid over the triangulated area. Each virtual grid point receives an interpolated value. The grid is scanned once for each contour level and all contours at that level are plotted. The triangulation and interpolation scheme is based on Lawson's C1 surface interpolation algorithm, which has been refined by Hirosha Akima. Parts of Akima's algorithm are used in this package. See reference heading for publications.

PORTABILITY

CONRAS will run on NCAR's CRAY-1 and 7600. It is designed to run on most other machines by including some required local routines which are described in the implementor's writeup. This writeup is available from NCAR.
CALL CONRAS (XD,YD,ZD,NDP,WK,IWK,SCRARR)

FRAME must be called by the user.

Listed below are options which can enhance your plot.

Calling CONOP prior to the CONRAS call can activate or deactivate the options. Each option command must be invoked with separate CONOP calls. There are 4 CONOP routines: CONOP1, CONOP2, CONOP3, CONOP4. The numerical value of the name determines the number of parameters in the calling sequence. If you want to plot a perimeter use CONOP1. To change the data output format use CONOP4. Only the first two characters on each side of the equal sign are scanned. Therefore only 2 characters for each option are required on input to CONOP (i.e. 7HSCA=PRI or 5HSC=PR would both be accepted for setting the scaling option to the PRI mode). Remember that the number of input points must be at least 4. This is equal to the default number of data points to be used for estimation of partial derivatives at each data point. NCP contains the number of these points to be used. The estimated partial derivatives are used for the construction of the interpolating polynomial's coefficients. NOTE: option settings remain in effect between calls to any CONRAN entry point.

OPTIONS

CHL. This flag determines how the high and low contour values are set. These contour values may be set by the program or by the user. If CHL=OFF, the program examines the user's input data and uses both the high and low values. If CHL=ON, the user must specify the desired high (hi) and low (flo) value should he wish to use different contour values from his input data. The default is CHL=OFF.

If Program Set: CALL CONOP1(7HCHL=OFF)

If User Set: CALL CONOP3(6HCHL=ON,hi,flo)

Example: CALL CONOP3(6HCHL=ON,5020.,2000.) where the hi value desired is 5020 and the flo value desired is 2000. These are floating point numbers.

Note: The values supplied for contour increment and contour high and low values assumes the unscaled data values. See the SDC flag, below.
CIL This flag determines how the contour increment (cinc) is set. The increment is either calculated by the program (CIL=OFF) using the range of high and low values from the user's input data, or set by the user (CIL=ON).

If Program Set: CALL CONOP1(7HCIL=OFF)

If User Set: CALL CONOP2(6HCIL=ON,cinc)

Example: CALL CONOP2(6HCIL=ON,15.) where 15. represents the contour increment desired by the user. This is a floating point number.

Note: The default option is CIL=OFF. By default, the program will examine the user's input data and determine the contour interval (cinc) at some appropriate range between the level of high and low values supplied, usually generating between 15 and 20 contour levels.

CON This flag determines how the contour levels are set. If CON=ON, the user must specify the array and number of contour levels (ncl). A maximum of 30 contour levels are permitted for both array and ncl. If CON=OFF, default values are used. In this case, the program will calculate the values for the array and ncl using input data.

If Program Set: CALL CONOP1(7HCON=OFF)

If User Set: CALL CONOP3(6HCON=ON,array,ncl)

Example:

DATA RLIST(1),RLIST(2),...,RLIST(5)/1.,2.,3.,10.,12./

CALL CONOP3(6HCON=ON,RLIST,5) where 'RLIST' is the number of contour levels in the array, and 5 is the number of user specified contour levels (ncl). The array must contain floating point numbers and the count ncl must be an integer.

Note: The array (array) contains the contour levels (floating point only) and ncl is the number of levels. The maximum number of contour levels allowed is 30.

WARNING ON CONTOUR OPTIONS:
It is illegal to use the CON option when either CIL or CHL are activated. In this error condition the option call that detected the error will NOT be executed. When assigning the contour array, it must be ordered from smallest to largest.
**DAS** This flag determines which contours are represented by dashed lines. The user sets the dashed line pattern. The user may specify that dashed lines be used for contours whose value is less than, equal to, or greater than the dash pattern breakpoint (see bp in the DBP option below), which is zero by default. If DAS=OFF (the default value), all solid lines are used.

All Solid Lines: CALL CONOP1(7HDAS=OFF)

If Greater: CALL CONOP2(7HDAS=GTR,ipat)

If Equal: CALL CONOP2(7HDAS=EQU,ipat)

If Less: CALL CONOP2(7HDAS=LSS,ipat)

If All Same: CALL CONOP2(7HDAS=ALL,ipat)

**Note:** ipat must be a ten character Hollerith string with a dollar sign ($) for DASH and a single quote (') for blank. This is not the repetition count for line labels. That is controlled by an internal parameter, NREP, found in common block CONR11.

Example: CALL CONOP2(7HDAS=GTR,10H$$$$$$$$$$)

**DBP** This flag determines how the dash pattern break point (bp) is set. If DBP=ON, bp must be set by the user by specifying bp. If DBP=OFF, the program will set bp at the default value which is zero.

If Program Set: CALL CONOP1(7HDBP=OFF)

If User Set: CALL CONOP2(6HDBP=ON,bp)

Example: CALL CONOP2(6HDBP=ON,5.) where 5. is the user specified break point.

**Note:** bp is a floating point number where the break for GTR and LSS contour DASH patterns are defined. bp is assumed to be given relative to the untransformed contours.

**DEF** Reset flags to default values. Activating this option sets all flags to the default value. DEF has no 'ON' of 'OFF' states.

To Activate: CALL CONOP1(3HDEF)

**EXT** Flag to set extrapolation.

To turn ON: CALL CONOP1(6HEXT=ON)
CALL CONOP1(7HEXT=OFF)

Note: Normally all CONRAN versions will only plot the boundaries of the convex hull defined by the users data to fill the rectangular area of the frame. Use the EXT switch.

**FMT**
Flag for the format of the plotted input data values. If FMT=OFF, the default values for ft, l, and if are used. The default values are:

\[
\begin{align*}
ft &= (G10.3) \\
l &= 7 \text{ characters including the parentheses} \\
if &= 10 \text{ characters printed in the output field by the format}
\end{align*}
\]

If FMT=ON, the user must specify values for ft, l, and if. All user specified values **MUST** be given in the correct format.

If Program Set: CALL CONOP1(7HFMT=OFF)
If User Set: CALL CONOP4(6HFMT=ON, ft, l, if)
Example: CALL CONOP4(6HFMT=ON, (G30.2), 7, 30)

Note: ft is a Hollerith String containing the format. The format must be enclosed by parenthesis. Any format allowed at your installation (up to 10 characters) will be accepted. l is the number of characters in ft. if is the length of the field created by the format. The numerical parameters are type integer.

**WARNING:** CONRAS WILL NOT TEST FOR A VALID FORMAT AND THE FORMAT IS ONLY ALLOWED TO BE 10 CHARACTERS LONG.

**GRI**
Flag for the Grid.

To turn ON: CALL CONOP1(6HGRI=ON)
To turn OFF: CALL CONOP1(7HGRI=OFF)

Note: If the flag is ON, the X and Y tick interval will be given. This is the interval in user coordinates that each tick mark represents.

**INT**
Flag to determine the intensities of the contour lines and other parts of the plot. If INT=OFF, ival is the default value. If INT=ON, the user must specify ival.

If Program Set: CALL CONOP1(7HINT=OFF)
Major Lines: CALL CONOP2(7HINT=MAJ, ival)
Minor Lines: CALL CONOP2(7HINT=MIN, ival)
Title and Message: CALL CONOP2(7HINT=LAB, ival)
Data Values: CALL CONOP2(7HINT=DAT, ival)

**Note:** This relates to the plotted data values and the plotted maximums and minimums.

All the Same: CALL CONOP2(7HINT=ALL, ival)

Example: CALL CONOP2(7HINT=ALL, 110)

**Note:** $ival$ is the intensity desired. For an explanation of the option value settings see the OPTN routine in the System Plot Package documentation. Briefly, $ival$ values range from 0 to 255 or the character strings 2HLO and 2HHI. The default is 2HHI except for INT=MIN which is set to 2HLO. Numerical values are type integer.

**LAB**
This flag can be set to either label the contours (LAB=ON) or not (LAB=OFF). The default value is LAB=ON.

To turn on: CALL CONOP1(6HLAB=ON)
To turn off: CALL CONOP1(7HLAB=OFF)

**LOT**
Flag to list options on the printer. The default value is set to OFF, and no options will be displayed.

To turn on: CALL CONOP1(6HLOT=ON)
To turn off: CALL CONOP1(7HLOT=OFF)

**Note:** If users want to print the option values, they should turn this option ON. The option values will be sent to the standard output unit as defined by the port routine I1MACH.

**LSZ**
This flag determines the label size. If LSZ=OFF, the default $isizelsz$ value will be used. If LSZ=ON, the user should specify $isizelsz$. The default value is 9 PWRIT units.

If Program Set: CALL CONOP1(7HLSZ=OFF)
If User Set: CALL CONOP2(6HLSZ=ON, $isizelsz$)

Example: CALL CONOP2(6HLSZ=ON, 4) where 4 is the user desired integer PWRIT units.

**Note:** $isizelsz$ is the requested character size in integer PWRIT units.
MES Flag to plot a message.

To turn ON: CALL CONOP1(6HMES=ON)
To turn OFF: CALL CONOP1(7HMES=OFF)

Note: If MES=ON a message is printed below the plot giving contour intervals and execution time in seconds. If PER or GRI are ON, the message also contains the X and Y tick interval.

MIN Flag for setting the number of minor lines between each major line. Minor lines are low intensity and unlabeled. Major lines are high intensity and labeled. The default (set when option is turned off) is two minor lines between each major line.

To turn ON: CALL CONOP2(6HMIN=ON,micnt).
To turn OFF: CALL CONOP1(7HMIN=OFF)
micnt is a type integer data value

NCP Flag for data points used for partial derivative estimation. If NCP=OFF, num is set to 4, which is the default value. If NCP=ON, the user must specify num greater than or equal to 2.

If Program Set: CALL CONOP1(7HNCP=OFF)
If User Set: CALL CONOP2(6HNCP=ON,num)
Example: CALL CONOP2(6HNCP=ON,3)

Note: num = number of data points used for estimation. Changing this value effects the contours produced and the size of input array IWK. num is type integer.

PDV Flag to plot the input data values. The default value is PDV=OFF.

To turn ON: CALL CONOP1(6HPDV=ON)
To turn OFF: CALL CONOP1(7HPDV=OFF)

Note: If PDV=ON, the input data values are plotted relative to their location on the contour map. If you only wish to see the locations and not the values, set PDV=ON and change FMT to produce an asterisk (*) such as (I1).

PER Flag to set the perimeter. The default value is PER=ON, which causes a perimeter to be drawn around the contour plot.
To turn ON: CALL CONOP1(6HPER=ON)
To turn OFF: CALL CONOP1(7HPER=OFF)

**Note:** If MES is ON, the X and Y tick interval will be given. This is the interval in user coordinates that each tick mark represents.

**PMN**  
Flag to plot relative minimums and maximums.

To Turn Off: CALL CONOP1(7HPMM=OFF)
To Turn On: CALL CONOP1(6HPMM=ON)

**REP**  
Flag indicating the use of the same data, but a new execution. The default value is set to OFF.

To turn ON: CALL CONOP1(6HREP=ON)
To turn OFF: CALL CONOP1(7HREP=OFF)

**Note:** If REP=ON, the same data and triangulation are to be used but it is assumed the user has changed contour values or resolution for this run. Scratch arrays WK and IWK must remain unchanged.

**SCA**  
Flag for scaling of the plot on a frame.

User Scaling: CALL CONOP1(7HSCA=OFF)
Program Scaling: CALL CONOP1(6HSCA=ON)
Prior Set Call: CALL CONOP1(7HSCA=PRI)

**Note:** With SCA=OFF, plotting instructions will be issued using the user's input coordinates, unless they are transformed via FX and FY transformations. Users will find an extended discussion in the Interfacing to Other Graphics Routines section below. The SCA option assumes the user has previously used a SET call and that all input data falls into the range of that SET call. With SCA=ON, the entry point will perform the SET call so that the user's plot will fit into the center 90 percent of the frame. When SCA=PRI, the program maps the user's plot instructions into the portion of the frame defined by the last SET call. SCA=PRI should be used when interfacing to SUPMAP.

**SDC**  
Flag to determine how to scale the data on the contours.

If SDC=OFF, the floating point value is given by scale.
If SDC=ON, the user may specify scale (see Note below.)
CALL CONOP1(7HSDC=OFF)

If User Set: CALL CONOP2(6HSDC=ON, scale)

Example: CALL CONOP2(6HSCD=ON, 100.)

**Note:** The data plotted on contour lines and the data plotted for relative minimums and maximums will be scaled by the floating point value given by `scale`. Typical `scale` values are 10., 100., 1000., etc. The original data values are multiplied by `scale`. It must be a floating point number and is displayed in the message (see MES). The default value for `scale` is 1.

**SML** Flag to determine the size of minimum and maximum labels. If SML=OFF, the value of `isizesml` is 15, which is the default. If SML=ON, the user must specify `isizesml`.

If Program Set: CALL CONOP1(7HSML=OFF)

If User Set: CALL CONOP1(6HSML=ON, isizesml)

Example: CALL CONOP1(6HSML=ON, 12)

**Note:** `isizesml` is an integer number which is the size of labels in PWRIT units.

**SPD** Flag for the size of the plotted input data values. If SPD=OFF, the value of `isizespd` is 8, which is the default. If SPD=ON, the user must specify `isizespd`.

If Program Set: CALL CONOP1(7HSPD=OFF)

If User Set: CALL CONOP2(6HSPD=ON, isizespd)

Example: CALL CONOP2(6HSPD=ON, 6)

**Note:** `isizespd` is an integer number giving the size of the data values in PWRIT units.

**SSZ** Flag to determine the resolution (number of steps in each direction). If SSZ=ON, the user sets `istep`, or, if SSZ=OFF, the program will automatically set `istep` at the default value of 40.

If Program Set: CALL CONOP1(7HSSZ=OFF)

If User Set: CALL CONOP2(6HSSZ=ON, istep)

Example: CALL CONOP2(6HSSZ=ON, 25) This `istep` value will produce a coarse contour. See note below.
**Mlote:** istep = type integer. It is the density of the virtual grid. In most cases, the default value of 40 produces pleasing contours. For coarser but quicker contours, lower the value. For smoother but longer time, raise the value. **NOTE:** For step sizes greater than 200 in CONRAQ, the array OV in common CONRA1, and ITLOC (not in the quick version) in common CONRA9 must be expanded to about 10 more than the size of SSZ. See CONRA1 and CONRA9 comments below for more information.

**STL.** Flag to determine the size of the title. isizestl may be set by the user (STL=ON), or the program will set it at the default size of 16 PWRIT units (STL=OFF).

If Program Set: CALL CONOP1(STL=OFF)
If User Set: CALL CONOP2(STL=ON,isizestl)
Example: CALL CONOP2(STL=ON,13)

**Note:** When 30 or 40 characters are used for the title, the default size of 16 PWRIT units works well. When titles are larger, a smaller title size will be required. Title size is an integer variable.

**TEN** Flag to determine the tension factor applied when smoothing contour lines. The user may set tens or allow the program to set the value. If user set, tens must have a value greater than zero and less than or equal to 30. The default value is 2.5.

If Program Set: CALL CONOP1(HTEN=OFF)
If User Set: CALL CONOP2(HTEN=ON,tens)
Example: CALL CONOP2(HTEN=ON,14.)

Smoothing of contour lines is accomplished with splines under tension. To adjust the amount of smoothing applied, adjust the tension factor. The tension must be greater than ZERO. The default is 2.5. Setting tens very large (i.e. 30), effectively shuts off smoothing.

**TFR** Flag to advance frame before triangulation. The default value is TFR=ON.

If Program Set: CALL CONOP1(HTFR=ON)
To turn OFF: CALL CONOP1(HTFR=OFF)

**Note:** Triangles are plotted after the contouring is completed. If the user wished to see the triangles over the contours, turn this switch OFF.
TLE  Flag to place a title at the top of the plot. If TLE=ON, the user must specify ichars and inum. The default value is OFF. ichars is the Hollerith character string for the title. inum is the number of characters in ichars.

To turn ON: CALL CONOP3(6HTLE=ON,ichars,inum)

Example: CALL CONOP3(6HTLE=ON,13HVECTOR REVIEW,13)

To turn OFF: CALL CONOP1(7HTLE=OFF)

Note: The title array is 32 words long, so the maximum title is 32 times the number of characters per word on your machine. Longer titles will require increasing the size of array ISTRNG found in CONRA7.

TOP  Flag to plot triangle only.

To turn OFF: CALL CONOP1(7HTOP=OFF)

To turn ON: CALL CONOP1(6HTOP=ON)

Note: The user may wish to overlay the triangles on some other plot. This option ON will allow that. This option when activated (TOP=ON), will set TRI=ON, and TFR=OFF. If the user wants TFR=ON, it should be set after TOP is set. If the user sets TOP=OFF it will set TRI=OFF and TFR=ON. If the user wants TRI and TFR different, then set them after the TOP call.

TRI  Flag to plot the triangulation.

To turn ON: CALL CONOP1(6HTRI=ON)

To turn OFF: CALL CONOP1(7HTRI=OFF)

Note: Plotting the triangles will indicate to the user where good and bad points of interpolation are occurring in the contour map. Equilateral triangles are optimal for interpolation. Quality degrades as triangles approach a long and narrow shape. The convex hull of the triangulation is also a poor point of interpolation.

OPTIONAL DEFAULT VALUES

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHL</td>
<td>OFF</td>
</tr>
<tr>
<td>LAB</td>
<td>ON</td>
</tr>
<tr>
<td>SDC</td>
<td>OFF</td>
</tr>
<tr>
<td>CIL</td>
<td>OFF</td>
</tr>
<tr>
<td>LOT</td>
<td>OFF</td>
</tr>
<tr>
<td>SML</td>
<td>OFF</td>
</tr>
<tr>
<td>CON</td>
<td>OFF</td>
</tr>
<tr>
<td>LSZ</td>
<td>OFF</td>
</tr>
<tr>
<td>SPD</td>
<td>OFF</td>
</tr>
<tr>
<td>DAS</td>
<td>OFF</td>
</tr>
<tr>
<td>MES</td>
<td>ON</td>
</tr>
<tr>
<td>SSZ</td>
<td>OFF</td>
</tr>
<tr>
<td>DBP</td>
<td>OFF</td>
</tr>
<tr>
<td>MIN</td>
<td>OFF</td>
</tr>
<tr>
<td>STL</td>
<td>OFF</td>
</tr>
<tr>
<td>EXT</td>
<td>OFF</td>
</tr>
<tr>
<td>NCP</td>
<td>OFF</td>
</tr>
<tr>
<td>TEN</td>
<td>OFF</td>
</tr>
<tr>
<td>FMT</td>
<td>OFF</td>
</tr>
<tr>
<td>PDV</td>
<td>OFF</td>
</tr>
<tr>
<td>TFR</td>
<td>ON</td>
</tr>
<tr>
<td>GRI</td>
<td>OFF</td>
</tr>
<tr>
<td>PER</td>
<td>ON</td>
</tr>
<tr>
<td>TLE</td>
<td>OFF</td>
</tr>
<tr>
<td>PMM</td>
<td>OFF</td>
</tr>
<tr>
<td>TOP</td>
<td>OFF</td>
</tr>
<tr>
<td>REP</td>
<td>OFF</td>
</tr>
</tbody>
</table>
The OPTION DEFAULT VALUES given above, if used, will set default values for the following parameters:

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DEFAULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>array</td>
<td>Up to 30 contour levels allowed. Values are computed by the program, based on input</td>
</tr>
<tr>
<td>bp</td>
<td>0.</td>
</tr>
<tr>
<td>cinc</td>
<td>Computed by program based on the range of hi and lo values of the input data.</td>
</tr>
<tr>
<td>flo</td>
<td>Computed by program based on the lowest unscaled input data.</td>
</tr>
<tr>
<td>ft</td>
<td>(G10.3) Parentheses must be included.</td>
</tr>
<tr>
<td>hi</td>
<td>Computed by program based on the highest unscaled input data.</td>
</tr>
<tr>
<td>minct</td>
<td>two minor lines</td>
</tr>
<tr>
<td>ichars</td>
<td>No title</td>
</tr>
<tr>
<td>if</td>
<td>10 characters</td>
</tr>
<tr>
<td>inum</td>
<td>No title</td>
</tr>
<tr>
<td>ipat</td>
<td>10H$$$$$$$$$$ This is a 10 character Hollerith string.</td>
</tr>
<tr>
<td>isizelsz</td>
<td>9 PWRIT units</td>
</tr>
<tr>
<td>sizesml</td>
<td>15 PWRIT units</td>
</tr>
<tr>
<td>sizespd</td>
<td>8 PWRIT units</td>
</tr>
<tr>
<td>sizesstl</td>
<td>16 PWRIT units</td>
</tr>
<tr>
<td>istep</td>
<td>40</td>
</tr>
<tr>
<td>ival</td>
<td>2HHI for all except minor contour lines which are 2HLO.</td>
</tr>
<tr>
<td>l</td>
<td>7 characters including both parentheses</td>
</tr>
<tr>
<td>ncl</td>
<td>Computed by program based on input data. Up to 30 contour levels are permitted.</td>
</tr>
<tr>
<td>num</td>
<td>4 data points</td>
</tr>
</tbody>
</table>
To create different styled contours use options NCP and SSZ. NCP will modify the interpolating functions and therefore cause changes in the plots produced. Increasing NCP causes more of the surrounding data to influence the point of interpolation. Some data sets cause difficulty when trying to produce meaningful contours (this is a useful feature when triangles are long and narrow). By modifying NCP and thereby changing the interpolation functions, a user can fine-tune a plot. Increasing SSZ will smooth out the contour lines and pick up more detail (new contours will appear as SSZ increases and old ones will sometimes break into more distinct units).

The scaling option will be used in some way normally not in the SCA=ON mode. The statement functions, FX and FY can be modified to perform other mappings. See the Graphics Manual for examples of interfaces to other graphics routines when using CONREC. The routines having FX and FY transformations are:

CONDRW, CONPDV, CONTLK, CONPMS, CONGEN

In most cases mapping can be performed before calling a CONRAS entry point, saving the user from modifying the file. If reasonable results cannot be obtained then the statement functions will have to be replaced.

If NCP GT 25 arrays DSQO and IPCO in CONDET must be adjusted accordingly. Also NCP$SZ$ in CONBDN (25 by default), will have to equal NCP. The default value of NCP which is 4, produces pleasing pictures in most cases. However, fine-tuning of the interpolation can be obtained by increasing the size of NCP, with a corresponding linear increase in work space. CONRAS calls SETI to guarantee 1-1024 integer range for PWRIT.

John Humbrecht NCAR, Boulder, Colorado

REFERENCES
Akima, Hirosha
A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points.
ACM Transactions on Mathematical Software
VOL 4, NO. 2, June 1978, Pages 148-159

Lawson, C.L.
Software for C1 Surface Interpolation
JPL Publication 77-30
August 15, 1977

<table>
<thead>
<tr>
<th>ERROR</th>
<th>ROUTINE</th>
<th>MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONRAS</td>
<td>INPUT PARAMETER NDP LT NCP</td>
</tr>
<tr>
<td>2</td>
<td>CONRAS</td>
<td>NCP GT MAX SIZE OR LT 2</td>
</tr>
<tr>
<td>3</td>
<td>CONTNG</td>
<td>ALL COLINEAR DATA POINTS</td>
</tr>
<tr>
<td>4</td>
<td>CONTNG</td>
<td>IDENTICAL INPUT DATA POINTS FOUND</td>
</tr>
<tr>
<td>5</td>
<td>CONSET</td>
<td>UNDEFINED OPTION</td>
</tr>
<tr>
<td>6</td>
<td>CONCLS</td>
<td>CONSTANT INPUT FIELD</td>
</tr>
<tr>
<td>7</td>
<td>CONSET</td>
<td>INCORRECT CONOP CALL USED</td>
</tr>
<tr>
<td>8</td>
<td>CONSET</td>
<td>ILLEGAL USE OF CON OPTION WITH CIL OR CHL OPTIONS</td>
</tr>
<tr>
<td>9</td>
<td>CONSET</td>
<td>NUMBER OF CONTOUR LEVELS EXCEEDS 30</td>
</tr>
<tr>
<td>10</td>
<td>CONDRW</td>
<td>CONTOUR STORAGE EXAuATED THIS ERROR IS TRAPPED AND NULLIFIED BY CONRAN. IT SERVES TO SIGNAL THE USER THAT A CONTOUR LEVEL MAY NOT BE COMPLETE.</td>
</tr>
</tbody>
</table>

The errors are all defined as recoverable in the port error processor. See the port error processing writeup for detailed usage. However no user action is required to execute CONRAS.
SAMPLE PLOT

DEMONSTRATION PLOT FOR CONRAS

CONTOUR FROM -4.0000 TO 26.0000
X INTERVAL: 1.0000 Y INTERVAL: 1.0000

CONTOR INTERVAL OF 2.0000

CONRAS - 16
SUBROUTINE CONREC (Z,L,M,N,FLO,HI,FINC,NSET,NHI,NDOT)

DIMENSION OF ARGUMENTS
Z(L,N)

LATEST REVISION
April 1979

PURPOSE
CONREC draws a contour map from data stored in a rectangular array. Contour lines are labelled.

USAGE
If the following assumptions are met, use
CALL EZCNTR (Z,M,N)

Assumptions:
All of the array is to be contoured,
Contour levels are picked internally
Contouring routine picks scale factors,
Highs and Lows are marked,
Negative lines are drawn with a dashed line pattern,
EZCNTR calls FRAME after drawing the contour map.

If these assumptions are not met, use
CALL CONREC (Z,L,M,N,FLO,HI,FINC,NSET,NHI,NDOT)

ARGUMENTS

On Input for EZCNTR

Z
M by N array to be contoured

M
First dimension of Z

N
Second dimension of Z

On Output for EZCNTR

All arguments are unchanged.
On Input for CONREC

Z
The origin of the array to be contoured. Z is L by N.

L
The first dimension of Z in the calling program.

M
The number of data values to be contoured in the x-direction (the first subscript direction). When plotting an entire array, L = M. See Appendix 1 of the Graphics Utilities Used at NCAR for an explanation of using this argument list to process any part of an array.

N
The number of data values to be contoured in the y-direction (the second subscript direction).

FLO
The value of the lowest contour level. If FLO = HI = 0., a value rounded up from the minimum Z is generated by CONREC.

HI
The value of the highest contour level. If HI = FLO = 0., a value rounded down from the maximum Z is generated by CONREC.

FINC
> 0 Increment between contour levels
= 0 A value which produces between 10 and 30 contour levels at nice values is generated by CONREC.
< 0 The number of levels generated is near ABS(FINC).

NSET
Flag to control scaling:
= 0 CONREC calls SET to properly scale the plotting instructions to the standard configuration. PERIM is called and puts tick marks corresponding to the data points.
> 0 CONREC assumes that SET has been called by the user in such a way as to properly scale the plotting instructions generated by CONREC. PERIM is not called.
< 0 CONREC calls SET in such a way as to place the contour plot within the limits of the user's last SET call. PERIM is not called.
NHI
Flag to control extra information on the contour plot:

= 0 Highs and lows are marked with an H or L as appropriate, and the value of the high or low is plotted under the symbol.

> 0 The values in each Z are plotted at each Z point, with the center indicating the data point location.

< 0 Neither of the above are done.

NDOT
A 10-bit constant designating the desired dashed line pattern for the contour lines in the plot. If |NDOT| is set to 0, 1 or 1777B, solid lines are drawn.

> 0 Pattern is used for all lines.

< 0 Pattern is used for negative-valued contour lines.

See DASHD write-up in DASHCHAR, in this section.

On Output for CONREC
All arguments are unchanged.

NOTE See the listed reference for directions for the most common modifications.

Comparison of Contouring  Contouring Write-up
Non-uniform contouring levels Comments in CLGEN
Transformations of contour lines Appendix 2 to Graphics
lines Internal parameters Used at NCAR
Internal parameters such as Internal parameters (below)
label sizes, Unknown data points, drawing placement, etc.

ENTRY POINTS  CONREC, CLGEN, REORD, ENCD, STLNE, DRLINE, MINMAX, PNTVAL, CALCN, EZCNTR

COMMON BLOCKS  CONREC1 2
                CONREC2 1014B
                DASHD1 147B
                INTPR 10

I/O Plots contour map

PRECISION Single
REQUIRED ULIB ROUTINES: None

LANGUAGE: FORTRAN

HISTORY: Replaces old contouring package called CALCNT at NCAR

ALGORITHM: Each line is followed to completion, points along a line are found on the boundaries of the (rectangular) cells, points along a line are connected by straight lines using the software dashed line package, which labels the lines.

SPACE REQUIRED: About 7000 octal not including the System Plot Package.

TIMING: Varies widely with size and smoothness of Z. Average smoothness 30 by 30 array takes less than 30 seconds on the Control Data 7600.

An Implementor's Write-up is available.

PORTABILITY: For a portable version, contact the Graphics Project.

PLOTTING ROUTINES USED: SET, GETSET, PERIM, FRSTPT, VECTOR, MXY, CURVE, OPTION (to set intensity), DASHLN and DASHD, CURVED, FRSTD, VECTD, LINED, PSYMD, PWRY, ENCD

REQUIRED RESIDENT ROUTINES: ALGO10, SORT, ATAN2, SIN, COS
### INTERNAL PARAMETERS

<table>
<thead>
<tr>
<th>NAME</th>
<th>DEFAULT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIZEL</td>
<td>1.5</td>
<td>Size of line labels</td>
</tr>
<tr>
<td>SIZEM</td>
<td>2.5</td>
<td>Size of labels for minimum and maximums</td>
</tr>
<tr>
<td>SIZEP</td>
<td>1.0</td>
<td>Size of labels for data point values</td>
</tr>
<tr>
<td>NLA</td>
<td>16</td>
<td>Approximate number of contour levels when internally generated</td>
</tr>
<tr>
<td>NLM</td>
<td>40</td>
<td>Maximum number of contour levels. If this is to be increased, the dimensions of CL and TWORK must be increased by the same amount in CONREC</td>
</tr>
<tr>
<td>XLT</td>
<td>.05</td>
<td>Left-hand edge of the plot (0.0=left edge of frame, 1.0=right of frame)</td>
</tr>
<tr>
<td>YBT</td>
<td>.05</td>
<td>Bottom edge of the plot (0.0=bottom of frame, 1.0=top of frame)</td>
</tr>
<tr>
<td>SIDE</td>
<td>0.9</td>
<td>Length of longer edge of plot (see also EXT)</td>
</tr>
<tr>
<td>NREP</td>
<td>6</td>
<td>Number of repetitions of the dash pattern between line labels. This cannot be larger than 9 at the present time because of a dimension limit in DASHD</td>
</tr>
<tr>
<td>NCRT</td>
<td>4</td>
<td>Number of plotter units per element (bit) in the dash pattern</td>
</tr>
<tr>
<td>ILAB</td>
<td>1</td>
<td>Flag to control the drawing of line labels</td>
</tr>
</tbody>
</table>

ILAB non-zero means label the lines
ILAB=0 means do not label the lines

**CONREC - 5**
### INTERNAL PARAMETERS (Con't)

<table>
<thead>
<tr>
<th>NAME</th>
<th>DEFAULT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULBLL</td>
<td>3</td>
<td>Number of unlabeled lines between labeled lines. For example, when NULBLL = 3, every fourth level is labeled.</td>
</tr>
<tr>
<td>IOFFD</td>
<td>0</td>
<td>Flag to control normalization of label numbers:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IOFFD = 0 means include decimal point when possible (do not normalize unless required).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IOFFD non-zero means normalize all label numbers and output a scale factor in the message below the graph.</td>
</tr>
<tr>
<td>EXT</td>
<td>.25</td>
<td>Lengths of the sides of the plot are proportional to $M$ and $N$ (when CONREC calls SET) except in extreme cases, namely, when the $\text{MIN}(M,N)/\text{MAX}(M,N)$ is less than EXT. Then CONREC produces a square plot.</td>
</tr>
<tr>
<td>IOFFP</td>
<td>0</td>
<td>Flag to control special value feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IOFFP = 0 means special value feature not in use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IOFFP non-zero means special value feature is in use. SPVAL is set to the special value. Contour lines will then be omitted from any cell with any corner equal to the special value.</td>
</tr>
<tr>
<td>SPVAL</td>
<td>0</td>
<td>Contains the special value when IOFFP is non-zero.</td>
</tr>
<tr>
<td>IOFFM</td>
<td>0</td>
<td>Flag to control message below plot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IOFFM = 0 means message is plotted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IOFFM non-zero means the message is not plotted.</td>
</tr>
<tr>
<td>ISOLID</td>
<td>1777B</td>
<td>Dash pattern for non-negative lines</td>
</tr>
</tbody>
</table>
SAMPLE PLOT

DEMONSTRATION PLOT FOR CONREC ENTRY OF CONREC

CONTOUR FROM -4.5000 TO 4.5000 CONTOUR INTERVAL OF 0.30000 PT(3.3) = -1.9066

CONREC - 7
SIMPLE PLOT

DEMONSTRATION PLOT FOR EZCNTR ENTRY OF CONREC

CONTOUR FROM -4.5000 TO 4.5000 CONTOUR INTERVAL OF 0.50000

PT(3,3) = -1.9066

CONREC - 8
A QUICKENED VERSION OF THE CONTOUR PACKAGE CONREC

DIMENSION OF ARGUMENTS

Z(L,N)

LATEST REVISION

January 1978

PURPOSE

Draws a contour map from data stored in a rectangular array. No line labels.

USAGE

See CONREC write-up

ARGUMENTS

See CONREC write-up

See DASHLN comments

Note: NDOT is a 10-bit constant designating the desired dashed line pattern for all the contour lines in the plot. If NDOT is set to 0, 1, or 1777B, solid lines are drawn.

ENTRY POINTS

CONREC, CLGEN, ENCD, MINMAX, QUICK, CALCNT, EZCNTR

COMMON BLOCKS

CONREC 2 words

I/O

Plots contour map

PRECISION

Single

REQUIRED ULIB ROUTINES

None

LANGUAGE

FORTRAN

HISTORY

A faster version of CONREC without line labelling capabilities

ALGORITHM

The grid space is divided into (M-1)*(N-1) cells. Each cell is processed in turn, drawing all contour lines found in a particular cell, until the entire rectangular space is contoured. (This could result in a relatively long plot time on mechanical plotters but does not affect plot time on CRT-based plotters.)

SPACE REQUIRED

About 3000 octal not including system plot package.
TIMING  Varies widely with size and smoothness of Z Takes 1/2 as long as CONREC (Standard). The sample figure took .06 seconds on the 7600.

PORTABILITY  For a portable version, contact the Graphics group

PLOTTING ROUTINES USED  SET, GETSET, PERIM, FRSTPT, MXMM, LINE, DASHLN, PWRT, FRAME

REQUIRED RESIDENT ROUTINES  ALOG10

INTERNAL PARAMETERS

<table>
<thead>
<tr>
<th>NAME</th>
<th>DEFAULT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISIZEM</td>
<td>2</td>
<td>Size of labels for minimums and maximums</td>
</tr>
<tr>
<td>ISIZEP</td>
<td>1</td>
<td>Size of labels for data point values</td>
</tr>
</tbody>
</table>

For explanations of MLA, SLT, YBT, SIDE, IDFFD, EXT, IDFFP, SPVAL, IDFFM, ISOLID, see CONREC write-up.

A sample picture follows:
SAMPLE PLOT

DEMONSTRATION PLOT FOR CONRECQCK ENTRY OF CONRECQCK

CONTOUR FROM -4.5000 TO 4.5000 CONTOUR INTERVAL OF 0.30000

PT(3,3) = -1.9066
DEMONSTRATION PLOT FOR EZCNR TR ENTRY OF CONRECQCK

CONTOUR FROM -4.5000 TO 4.5000 CONTOUR INTERVAL OF 0.50000 PT(3,3) = -1.9066

CONRECQCK - 4
A SMOOTHING VERSION OF THE CONTOUR PACKAGE, CONREC

DIMENSION OF ARGUMENTS

Z(L,N)

LATEST REVISION

April 1979

PURPOSE

CONRECSMTH draws a contour map from data stored in a rectangular array. Contour lines are labeled and smoothed.

USAGE

If the following assumptions are met, use

CALL EZCNTR (Z,M,N)

Assumptions:

• All of the array is to be contoured.
• Contour levels are picked internally.
• Contouring routine picks scale factors.
• Highs and Lows are marked.
• Negative lines are drawn with a dashed line pattern.
• EZCNTR calls FRAME after drawing the contour map.

If these assumptions are not met, use

CALL CONREC (Z,L,M,N,FLO,HI,FINC,NSET,NHI,NDOT)

ARGUMENTS

See CONREC write-up.

NOTE

• CONRECSMTH is CONREC with the software dashed line package with character capability (DASHCHAR) replaced by the software dashed line package with character capability and smoothing (DASHSMTH).

• Mods to CONREC can also be used on this file without change.

• With the following exceptions, everything is the same as the CONREC write-up.
ENTRY POINTS

CONREC, CLGEN, REORD, ENCD, STLINE, DRLINE, DASHD, CRVED, PSYMD, PWRY, CURVED, CALCNT, FRSTD, KURV1, KURV2S, MINMAX, PNTVAL, VECT, LINED, VECTD, LASTD, FRST

TIMING

About 1.6 times as long as CONREC

Space Required

117338
SAMPLE PLOT

DEMONSTRATION PLOT FOR CONREC ENTRY OF CONRECSMTH

CONTOUR FROM -4.5000 TO 4.5000 CONTOUR INTERVAL OF 0.30000 PT(3.3)= -1.9066

CONRECSMTH - 4
THE FANCIEST VERSION OF THE CONTOURING PACKAGE, CONREC

DIMENSION OF ARGUMENTS
Z(L,N)

LATEST REVISION
August 1975

PURPOSE
CONRECSUPR draws a contour map from data stored in a rectangular array. Contour lines are labeled and smoothed and crowded lines are eliminated.

USAGE
Same as CONREC, that is,

CALL EZCNTR (Z,M,N)

or

CALL CONREC (Z,L,M,N,FLO,HI,FINC,NSET,NHI,NDOT)

ARGUMENTS
See CONREC.

NOTE
• CONRECSUPR is CONREC with the software dashed line package with character capability (DASHCHAR) replaced by the software dashed line package with character capability, smoothing, and capability of removing crowded lines (DASHSUPR).

• With the following exceptions, everything is the same as the CONREC write-up.

COMMON BLOCKS
CONRE1 3
CONRE2 10148
DASHD1 1478
DASHD2 1
MARK1 441718
MARK2 1

REQUIRED ULIB ROUTINES
PWRX

ENTRY POINTS
CONREC, CLGEN, REORD, ENCD, STLINE, DRLINE, DASHD, CRVED, PSYMD, PWRY, CURVED, CALCNT, FRSTD, KURV1, KURV2S, MIN-MAX, PNTVAL, VECT, LINED, VECTD, LASTD, FRST, LAST, VCTOR, HIDDEN, MARKL, MARKP, PWRM

TIMING
About 4 times as long as CONRECSMTH
PORTABILITY Poor. Machine word-length dependent arrays are used to store crowded line information.

SPACE REQUIRED About 57000Q not including system plot package.

INTERNAL PARAMETERS CONRECSUPR has all the same internal parameters that CONREC and CONRECSMTH have, and in addition, it has the following parameter.

<table>
<thead>
<tr>
<th>NAME</th>
<th>DEFAULT</th>
<th>FUNCTION</th>
</tr>
</thead>
</table>
| IHILO | 3       | Flag to control the labelling of highs and lows if NHI = 0. When IHILO:
|      |         | = 0 Neither highs nor lows are labelled
|      |         | = 1 Highs are labelled and lows are not
|      |         | = 2 Lows are labelled and highs are not
|      |         | = 3 Both highs and lows are labelled |
SAMPLE PLOT

DEMONSTRATION PLOT FOR CONREC ENTRY OF CONRECSUPR

CONTOUR FROM -4.5000 TO 4.5000 CONTOUR INTERVAL OF .30000 PT(3.31= -1 9066

CONRECSUPR - 3
DEMONSTRATION PLOT FOR EZCNTR ENTRY OF CONRECSUPR
CONTOUR FROM -4.5000 TO 4.5000 CONTOUR INTERVAL OF .5000

CONRESUPR - 4
SOFTWARE DASHED LINE PACKAGE WITH CHARACTER CAPABILITY

LATEST REVISION December 1979

PURPOSE DASHCHAR is a software dashed line package. Some hardware dashed line generators fail to produce pleasing results when drawing very short vector segments or vector segments of varying lengths. DASHCHAR does not have this problem. In addition, this package can put characters at intervals in the line to label the line.

USAGE First,

CALL RESET

RESET performs no function. It is used in DASHCHAR and DASHSMTH for compatibility with CONREC and AUTOGRAPH which both use DASHSUPR.

CALL DASHD (IPAT, NC, JCRT, JSIZE)

then, call any of the following:

CALL CURVED (X, Y, N)
CALL FRSTD (X, Y)
CALL VECTD (X, Y)
CALL LINED (XA, YA, XB, YB)

ARGUMENTS Except for DASHD, all arguments match those in the corresponding routine in the System Plot Package. Please refer to the System Plot Package documentation for an explanation of the arguments.

<table>
<thead>
<tr>
<th>DASHCHAR PACKAGE</th>
<th>SYSTEM PLOT PACKAGE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURVED (X, Y, N)</td>
<td>CURVE (X, Y, N)</td>
<td>12</td>
</tr>
<tr>
<td>FRSTD (X, Y)</td>
<td>FRSTPT (X, Y)</td>
<td>10</td>
</tr>
<tr>
<td>VECTD (X, Y)</td>
<td>VECTOR (X, Y)</td>
<td>10</td>
</tr>
<tr>
<td>LINED (XA, YA, XB, YB)</td>
<td>LINE (XA, YA, XB, YB)</td>
<td>11</td>
</tr>
</tbody>
</table>

Wherever the System Plot Package allows either fixed or floating point arguments, the software dashed line package does also.
On Input for DASHD

IPAT
A dash line pattern that may be defined in one of two ways:

- If IPAT is an integer in the range 0-177777B, the last 3 arguments in the call are neither needed nor used. These 16 bits are treated as a code for solid line segment or gap with a one bit as 3 plotter address units solid, and a zero bit as 3 plotter address units of gap. Thus, the pattern 147070B=1100111000111000 means 6 ON, 6 OFF, 9 ON, 27 OFF, etc.

- If IPAT is not of the form above, it is assumed to be a Hollerith String NC characters long. NC may be any number, although about 60 seems to be a practical limit. A solid line segment is indicated by a $. A gap is indicated by a quote (a quote (') = 8,4 punch on the keypunch, (") when printed). Blanks are ignored. Any other Hollerith characters become part of the pattern with a hole of appropriate length being left in the line for the character string. No single Hollerith String to be printed as such may be longer than 15 characters.

- When IPAT is to represent a bit pattern, the other parameters to DASHD have to be set to zero. On machines with a word length greater than 16 bits, these parameters can be left out completely if the compiler allows that. Users who are concerned about the portability of their programs should always supply dummy parameters to DASHD (zeros).

NC
The number of characters in IPAT

JCRF
The length in plotter address units per $ or quote

JSIZE
Is the size of the PWRIT character.

- If between 0 and 3, it is 1., 1.5., 2., and 3. times the 3 plotter address units.

- If greater than 3, it is the character width in user plotter address units.
On Output
All arguments are unchanged for all routines.

- When using FRSTD and VECTD, LASTD may be called (no arguments needed). If the dashed line package was leaving a space for characters but the line ended before there was enough room for the characters, the space left will be filled in if LASTD is called.

- When switching from the regular plotting routines to a dashed line package, the first call should not be to VECTD. When switching from a dashed line package to the regular plotting routines, the first call should not be to VECTOR.

ENTRY POINTS
DASHD, CURVED, FRSTD, VECTD, LINED, RESET, LASTD, CFVLD, DRAWPV, DASHBD

COMMON BLOCKS
DASHD1 3158
DSAVE1 138
DSAVE2 7
DSAVE3 2
DDFLAG 1
CFFLAG 1
DCFLAG 1

I/O
Plots solid or dashed lines, possibly with characters at intervals in the line.

PRECISION
Single

REQUIRED ULIB
None

LANGUAGE
FORTRAN

HISTORY
Written in 1969, standardized November 1972. All the Dashed-Line packages were merged into one master file by Virginia Kallal ending in June, 1976. The routine was made portable in September 1977 by Robert Ubelmesser for use on all computer systems which support plotters with up to 15 bit resolution. The routines of the 1977 System Plot Package were used.

ALGORITHM
Position in dash pattern is remembered as points are processed. Distance traversed in plotter address space is used to determine whether to draw segments, parts of segments, characters, or nothing.

SPACE REQUIRED
About 32108
ACCURACY  
+.5 plotter units per call. There is no cumulative error.

TIMING  
For solid or blank lines, there is almost no overhead. For dashed lines, about 4 times as long as the System Plot Package (but the line drawing software in the System Plot Package is so fast that the increase will not be noticeable in most programs). For lines with characters, not much longer.

PORTABILITY  
Do not port the ULIB files, as this results in much extra work. Please contact the specialist.

PLOTTING ROUTINES USED  
PLOTIT, FL2INT, PWRIT, GETSI, ISHIFT, IOR, IAND, ULIBER, PERROR, GETCHR, SETCHR

REQUIRED RESIDENT ROUTINES  
SQRT, ATAN2, SIN, COS
### INTERNAL PARAMETERS

<table>
<thead>
<tr>
<th>NAME</th>
<th>DEFAULT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPAU</td>
<td>3</td>
<td>Number of plotter addresses per element (bit) in the dash pattern when ≤ 1777B. Thus, the pattern is repeated every ICRT*10 plotter units.</td>
</tr>
<tr>
<td>FPART</td>
<td>1.</td>
<td>Multiplying factor for first solid line segment. This can be used to off-set labels. For example, if FPART = .5, the first solid line segment is only one-half as long as it is the rest of the time. This moves all labels on this line towards the beginning, which reduces the probability of the label being written on top of a label of a nearby line drawn with FPART = 1.</td>
</tr>
<tr>
<td>IGP</td>
<td>9</td>
<td>Flag to control the space left for characters = 9 gap is left = 0 no gap is left</td>
</tr>
<tr>
<td>ICLOSE</td>
<td>6</td>
<td>An internal or external call to set the pen (pen-UP) to a specific position is only executed if this position is more than ICLOSE metacode address units away from the current pen position (distance=difference in X-coordinates + difference in Y-coordinates).</td>
</tr>
</tbody>
</table>
C X,Q,R,S,T, AND U ARE PREVIOUSLY DEFINED ONE DIMENSIONAL C ARRAYS
C SET AND PERIML HAVE BEEN CALLED
CALL DASHD(20H$\ldots$A, 20, 10, 2.)
CALL CURVED(X,Q,N)
CALL DASHD(20H$\ldots$B, 20, 10, 2.)
CALL CURVED(X,R,N)
CALL DASHD(20H$\ldots$C, 20, 10, 2.)
CALL CURVED(X,S,N)
CALL DASHD(20H$\ldots$D, 20, 10, 2.)
CALL CURVED(X,T,N)
CALL DASHD(20H$\ldots$E, 20, 10, 2.)
CALL CURVED(X,U,N)
CALL FRAME
SAMPLE PLOT

DEMONSTRATION PLOT FOR DASHCHAR

IPAT=DADADADADADADAD, K=1

IPAT=DDDDDDADADDDDDD, K=2

IPAT=DDDDADDDADDDDDA, K=3

IPAT=DDDDAAAAADDDDDD, K=4

IPAT=DDDADDDADDDDDD, K=5

IN IPAT STRINGS, A AND D SHOULD BE INTERPRETED AS APOSTROPHE AND DOLLAR SIGN
SOFTWARE DASHLINE PACKAGE

LATEST REVISION December 1979

PURPOSE DASHLINE is a software dash line package. Some hardware dash line generators fail to produce pleasing results when drawing very short vector segments or vector segments of varying length. This package does not have this problem.

USAGE First,

CALL DASHD (IPAT)

then, call any of the following:

CALL CURVED (X,Y,N)
CALL FRSTD (X,Y)
CALL VECTD (X,Y)
CALL LINED (XA,YA,XB,YB)

ARGUMENTS

On Input All arguments match those in the corresponding routine in the System Plot Package. See NCAR 7600 System Library Routines Manual, Chapter 3 under PLOT for explanation of the arguments.

<table>
<thead>
<tr>
<th>DASHLINE PACKAGE</th>
<th>SYSTEM PLOT PACKAGE</th>
<th>NCAR LRM</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASHD (IPAT)</td>
<td>DASHLN (IPAT)</td>
<td>5.26</td>
</tr>
<tr>
<td>CURVED (X,Y,N)</td>
<td>CURVE (X,Y,N)</td>
<td>3.191</td>
</tr>
<tr>
<td>FRSTD (X,Y)</td>
<td>FRSTPT (X,Y)</td>
<td>3.187</td>
</tr>
<tr>
<td>VECTD (X,Y)</td>
<td>VECTOR (X,Y)</td>
<td>3.187</td>
</tr>
<tr>
<td>LINED (XA,YA,XB,YB)</td>
<td>LINE (XA,YA,XB,YB)</td>
<td>3.188</td>
</tr>
</tbody>
</table>

Whenever the System Plot Package allows either fixed or floating point arguments, the software dash line package does also.

On Output All arguments are unchanged.

NOTE When switching from the regular plotting routines to a dashed line package, the first call should not be to VECTD. When switching from a dashed line package to the regular plotting routines the first call should not be to VECTOR.
<table>
<thead>
<tr>
<th>ENTRY POINTS</th>
<th>DASHD,CURVED,FRSTD,VECTD,LINED, LASTD, CFVLD, DRAWPV, DASHBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMON BLOCKS</td>
<td>DSHD 22B</td>
</tr>
<tr>
<td></td>
<td>DSHDA 1</td>
</tr>
<tr>
<td></td>
<td>DSHDB 11B</td>
</tr>
<tr>
<td></td>
<td>DSHDC 2</td>
</tr>
<tr>
<td>I/O</td>
<td>Plots solid or dashed lines.</td>
</tr>
<tr>
<td>PRECISION</td>
<td>Single</td>
</tr>
<tr>
<td>REQUIRED ULIB ROUTINES</td>
<td>None</td>
</tr>
<tr>
<td>LANGUAGE</td>
<td>FORTRAN</td>
</tr>
<tr>
<td>HISTORY</td>
<td>Written in 1969, standardized November, 1972. All the Dashed-Line packages were merged into one master file by Virginia Kallal ending in June, 1976. The routine was made portable in September, 1977 by Robert Ubelmesser for use on all computer systems which support plotters with up to 15 bit resolution. The routines of the 1977 System Plot Package were used.</td>
</tr>
<tr>
<td>ALGORITHM</td>
<td>Position in dash pattern is remembered as points are processed. Distance traversed in plotter space is used to determine whether to draw segments, parts of segments, or nothing.</td>
</tr>
<tr>
<td>SPACE REQUIRED</td>
<td>About $1000 \frac{3}{2}$</td>
</tr>
<tr>
<td>ACCURACY</td>
<td>+.5 plotter address units per call. There is no cumulative error.</td>
</tr>
<tr>
<td>TIMING</td>
<td>For solid or blank lines, there is almost no overhead. For dashed lines, about 4 times as long as the System Plot Package (but the line drawing software in the system plot package is so fast that the increase will not be noticeable in most programs).</td>
</tr>
<tr>
<td>PORTABILITY</td>
<td>Do not port the ULIB files, as this results in much extra work. Please contact the specialist.</td>
</tr>
<tr>
<td>PLOTTING ROUTINES USED</td>
<td>PLOTIT, FL2INT, GETSI, ISHIFT, IOR, IAND, ULIBER, PERROR</td>
</tr>
<tr>
<td>REQUIRED RESIDENT ROUTINES</td>
<td>SQRT</td>
</tr>
</tbody>
</table>

DASHLINE - 2
### INTERNAL PARAMETERs

<table>
<thead>
<tr>
<th>NAME</th>
<th>DEFAULT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPAU</td>
<td>3</td>
<td>Number of plotter units per element (bit) in the dash pattern. Thus, the pattern is repeated every ICRT*10 plotter units.</td>
</tr>
<tr>
<td>ICLOSE</td>
<td>6</td>
<td>An internal or external call to set the pen (pen=UP) to a specific position is only executed if this position is more than ICLOSE meta-code address units away from the current pen position. (Distance=difference in X-coordinates + difference in Y-coordinates).</td>
</tr>
</tbody>
</table>
SAMPLE PLOT

DEMONSTRATION PLOT FOR DASHLINE

IPAT=0001110001111111

IPAT=1111000011110000
/
/
- - - - - ... N N
7 N N /
7 - -'-

IPAT=1111111100000000
->
I 7 /
K.
7-7 7

IPAT=1111111111111000
_-/7 K-_ _-'I

DASHLINE - 4
DASHSMTH
SOFTWARE DASHED LINE PACKAGE WITH CHARACTER CAPABILITY AND SMOOTHING

LATEST REVISION
December 1979

PURPOSE
DASHSMTH is a software dashed line package with smoothing capabilities. DASHSMTH is DASHCHAR with smoothing features added.

USAGE
First,

CALL RESET
RESET performs no function. It is used in DASHCHAR and DASHSMTH for compatibility with CONREC and AUTOGRAPH which both use DASHSUPR. CALL DASHD (IPAT,NC,ICRT,ISIZE)

Then, call the following:

CALL CURVED (X,Y,N)
or,

CALL FRSTD (X,Y)
CALL VECTD (X,Y)
CALL LASTD

FRSTD processes the first point of a line. VECTD is called for the remaining points of a line. LASTD is called only after the last point of a line has been processed in VECTD.

The following may also be called, but no smoothing will result:

CALL LINED (XA,YA,XB,YB)

ARGUMENTS
Except for DASHD, all arguments match those in the corresponding routines in the System Plot Package. Please refer to this document for an explanation of the arguments.

<table>
<thead>
<tr>
<th>DASHCHAR PACKAGE</th>
<th>SYSTEM PLOT PACKAGE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURVED (X,Y,N)</td>
<td>CURVE (X,Y,N)</td>
<td>12</td>
</tr>
<tr>
<td>FRSTD (X,Y)</td>
<td>FRSTPT (X,Y)</td>
<td>10</td>
</tr>
<tr>
<td>VECTD (X,Y)</td>
<td>VECTOR (X,Y)</td>
<td>10</td>
</tr>
<tr>
<td>LINED (XA,YA,XB,YB)</td>
<td>LINE (XA,YA,XB,YB)</td>
<td>11</td>
</tr>
</tbody>
</table>
On Input for DASHD

IPAT

A dash line pattern that may be defined in one of two ways:

- If IPAT is an integer in the range of 0-177777B, these 16 bits are treated as a code for solid or gap with a one bit as 3 plotter address units solid, and a zero bit as 3 plotter address units of gap. For example, the pattern 147000B = 1100111000000000 means 6 On, 6 OFF, 9 ON, 27 OFF, etc.

- If IPAT is not of the form above, it is assumed to be a Hollerith String NC characters long. NC may be any number although about 60 seems to be a practical limit. A solid is indicated by a dollar sign ($). A gap is indicated by a single quote ('). (Quote = 8,4 punch on the keypunch, resulting in an ($) when printed). Blanks are ignored. Any other Hollerith characters become part of the pattern with a hole of appropriate length being left in the line for the character string. No single Hollerith String to be printed as such may be longer than 15 characters.

- When IPAT is to represent a bit pattern the other parameters to DASHD have to be set to zero. On machines with a word length greater than 16 bits, these parameters can be left out completely if the compiler allows that. Users who are concerned about the portability of their programs should always supply dummy parameters to DASHD (zeros).

NC

The number of characters in IPAT

JCRT

The length in user plotter address units per $ or quote.

JSIZE

JSIZE is the size of the PWRIT characters.

- If between 0 and 3, it is 1., 1.5., 2., and 3. times the 8 plotter address unit width.

- If greater than 3, it is the character width in user plotter address units.
On Output All arguments are unchanged for all routines.

Note: When using FRSTD and VECTD, LASTD must be called (no arguments needed). LASTD sets up the calling routines KURV1S and KURV2S.

When switching from the regular plotting routines to a dashed line package the first call should not be to VECTD. When switching from a dashed line package to the regular plotting routines, the first call should not be to VECTOR.

ENTRY POINTS DASHD, CURVED, FRSTD, VECTD, LINED, RESET, LASTD, KURV1S, KURV2S, CFVLD, FDVDLD, DRAWPV, DASHBD.

COMMON BLOCKS DASHD, 3158
DSAVE1 13  DSAVE2 7  DSAVE3 2  DFLAGS 2268
FDFLAG 1  DDFLAG 1  CFFLAG 1  CDFLAG 1

I/O Plots solid or dashed lines, possibly with characters at intervals in the line. The lines may also be smoothed.

PRECISION Single

REQUIRED ULIB None

LANGUAGE FORTRAN

HISTORY Written in October 1973; standardized October 1973. All the Dashed-Line packages were merged into one master file by Virginia Kallal ending in June, 1976. The routine was made portable in September, 1977 by Robert Ubelmesser for use on all computer systems which support plotters with up to 15 bit resolution. The routines of the 1977 System Plot Package were used.

ALGORITHM Points for each line segment are processed and passed to the routines KURV1S and KURV2S, which compute splines under tension passing through these points. New points are generated between the given points, resulting in smooth lines.

SPACE REQUIRED About 46008

ACCURACY ±.5 plotter units per call. There is no cumulative error.

TIMING About three times as long as DASHCHAR
PORTABILITY
Do not port the ULIB files, as this results in much extra work. Please contact the specialist.

PLOTTING ROUTINES USED
PLOTIT, FL2INT, PWRIT, GETSI, ISHIFT, IOR, IAND, ULIBER, PERROR, GETCHR, SETCHR

REQUIRED RESIDENT ROUTINES
SQRT, ATAN2, SIN, COS, EXP

INTERNAL PARAMETERS

<table>
<thead>
<tr>
<th>NAME</th>
<th>DEFAULT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPAU</td>
<td>3</td>
<td>Number of plotter address units per element (bit) in the dash pattern when less than or equal to 177777B. Thus, the pattern is repeated very IPAU*16 plotter address units.</td>
</tr>
<tr>
<td>FPART</td>
<td>1.0</td>
<td>Multiplying factor for first solid line segment. This can be used to off-set labels. For example, if FPART=.5, the first solid line segment is only half as long as it is the rest of the time. This moves all labels on this line towards the beginning, which reduces the probability of the label being written on top of a label of a nearby line drawn with FPART=1.</td>
</tr>
<tr>
<td>IGP</td>
<td>9</td>
<td>Flag to control whether a gap is left for characters when plotting. = 9 gap is left = 0 no gap is left</td>
</tr>
<tr>
<td>IOFFS</td>
<td>0</td>
<td>Flag to turn on smoothing code =0 smoothing =1 no smoothing</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>TENSN</td>
<td>2.5</td>
<td>Tension factor. Must be greater than 0. A large tension factor (30.) would essentially turn off smoothing.</td>
</tr>
<tr>
<td>NP</td>
<td>150</td>
<td>Twice the maximum number of interpolated points on a horizontal line with length equal to that of the grid.</td>
</tr>
<tr>
<td>SMALL</td>
<td>128</td>
<td>Minimum distance in plotter units between points. When points on a line are being processed and two or more consecutive points are less than 4 plotter units apart, only one of these consecutive points is saved. This procedure is to prevent cusps.</td>
</tr>
<tr>
<td>L1</td>
<td>70</td>
<td>The maximum number of points saved at one time. If there are more than 70 points on a given line, 70 points are processed, then the next 70, until the entire line is processed. Smoothness between segments is maintained automatically. If L1 is increased, the dimensions of XSAVE, YSAVE, XP, YP, and TEMP in FRSTD must be increased to the new value of L1.</td>
</tr>
<tr>
<td>ICLOSE</td>
<td>6</td>
<td>An internal or external call to set the pen (pen-UP) to a specific position is only executed if this position is more than ICLOSE metacode address units away from the current pen position (distance=difference in X-coordinates + difference in Y-coordinates).</td>
</tr>
</tbody>
</table>
SAMPLE PLOT  Compare this plot with that for DASHCHAR.

DEMONSTRATION PLOT FOR DASHSMTH
DEMONSTRATION PLOT FOR DASHSMTH

IPAT=DADADADADADADAD, K=1

IPAT=DDDDDDADADDDDDD, K=2

IPAT=DDDDADDDADDDDA, K=3

IPAT=DDDDAADDDAADDDDA, K=4

IPAT=DDDAODDDAADDDADD, K=5

IN IPAT STRINGS, A AND D SHOULD BE INTERPRETED AS APOSTROPHE AND DOLLAR SIGN
SOFTWARE DASHED LINE PACKAGE WITH CHARACTER CAPABILITY, SMOOTHING, AND CAPABILITY OF REMOVING CROWDED LINES

LATEST REVISION December 1979

PURPOSE DASHSUPR is a software dashed line package with smoothing capabilities and the capability of removing crowded lines.

USAGE First use,

CALL RESET

RESET initializes the model picture array and should be called before each new frame to be produced.

Then use,

CALL DASHD (IPAT, NC, JCRT, JSIZE)

then, call the following:

CALL CURVED (X, Y, N)

or,

CALL FRSTD (X, Y)
CALL VECTD (X, Y)
CALL LASTD

FRSTD processes the first point of a line. VECTD is called for the remaining points of a line. LASTD is called only after the last point of a line has been processed in VECTD.

The following may also be called, but no smoothing will result:

CALL LINED (XA, YA, XB, YB)

or

CALL PWRM (X, Y, IDPC, NC, SIZ, IOR, ICNT)

PWRM can be called to draw characters and mark the regions where the characters have been drawn in the model picture.

ARGUMENTS Except for DASHD, all arguments match those in the corresponding routine in the System Plot Package. Please refer to this document for an explanation of the arguments.
Wherever the System Plot Package allows either fixed or floating point arguments, the software dashed line package does also.

**On Input for DASHD**

**IPAT** A dash line pattern that may be defined in one of two ways:

- If IPAT is an integer in the range 0-177777B, the last 3 arguments in the call are neither needed nor used. These 16 bits are treated as a code for solid line segment or gap with a one bit as 3 plotter address units solid, and a zero bit as 3 plotter address units of gap. Thus, the pattern $147070B=1100111000111000$ means 6 ON, 6 OFF, 9 ON, 27 OFF, etc.

- If IPAT is not of the form above, it is assumed to be a Hollerith String NC characters long. NC may be any number, although about 60 seems to be a practical limit. A solid line segment is indicated by a $. A gap is indicated by a quote (a quote (')) = 8,4 punch on the keypunch, (")) when printed. Blanks are ignored. Any other Hollerith characters become part of the pattern with a hole of appropriate length being left in the line for the character string. No single Hollerith String to be printed as such may be longer than 15 characters.

- When IPAT is to represent a bit pattern, the other parameters to DASHD have to be set to zero. On machines with a word length greater than 16 bits, these parameters can be left out completely if the compiler allows that. Users who are concerned about the portability of their programs should always supply dummy parameters to DASHD (zeros).
The number of characters in IPAT

The length in plotter address units per $ or quote

Is the size of the PWRIT character.

* If between 0 and 3, it is 1., 1.5, 2., and 3. times the 8 plotter address units.

* If greater than 3, it is the character width in user plotter address units.

All arguments are unchanged for all routines.

When using FRSTD and VECTD, LASTD must be called (no arguments needed). LASTD sets up the calls to the smoothing routines KURV1S and KURV2S.

When switching from the regular plotting routines to a dashed line package the first call should not be to VECTD. When switching from a dashed line package to the regular plotting routines the first call should not be to VECTOR.

DASHD, CURVED, FRSTD, VECTD, LINED, LASTD, KURV1S, KURV2S, CPVLD, FDVLD, DASHBD, CUTUP, RESET, REMOVE, DRAWPV, MARKL, PWRM

DASHD1 3158
DSAVE1 138
DSAVE2 7
DSAVE3 2
DSAVE4 4
DSAVE5 2268

MODEL 440008 for 60 bit words and 1024 by 1024 model

I/O Plots solid or dashed lines, possibly with characters at intervals in the line. The lines may also be smoothed.
PRECISION Single
REQUIRED ULIB None
LANGUAGE FORTRAN
HISTORY Modification of DASHSMTH by Tom Reid of Texas A & M, based on CALCNTSUPR. All the Dashed-Line packages were merged into one master file by Virginia Kallal ending in June, 1976. The routine was made portable in September 1977 by Robert Ubelmesser for use on all computer systems which support plotters with up to 15 bit resolution. The routines of the 1977 System Plot Package were used.

ALGORITHM Points for each line segment are processed and passed to the routines KURV1S and KURV2S, which compute splines under tension passing through these points. New points are generated between the given points, resulting in smooth lines. As each line is drawn a test is done to see if that part of the plotting plane has been drawn on. If it has, then that line, or part of that line, is not drawn. As the lines are drawn, they are also marked into the model picture.

Note: Since this algorithm eliminates the lines in the order that they are drawn, the user should draw all lines of major importance first and lines of minor importance last.

SPACE REQUIRED About 520008 not including system plot package
ACCURACY +.5 plotter units per call. There is no cumulative error.
TIMING About four times as long as DASHSMTH
PORTABILITY Do not port the ULIB files, as this results in much extra work. Please contact the specialist.
PLOTTING ROUTINES USED PLOTIT, FL2INT, PWRT, GETSI, ISHIFT, IOR, IAND, ULIBER, PERROR, GETCHR, SETCHR
REQUIRED RESIDENT ROUTINES SQRT, ATAN2, SIN, COS, EXP
<table>
<thead>
<tr>
<th>NAME</th>
<th>DEFAULT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPAU</td>
<td>3</td>
<td>Number of plotter addresses per element (bit) in the dash pattern when ≤ 1777B. Thus, the pattern is repeated every ICRT*10 plotter units.</td>
</tr>
<tr>
<td>FPART</td>
<td>1.</td>
<td>Multiplying factor for first solid line segment. This can be used to off-set labels. For example, if FPART = .5, the first solid line segment is only one-half as long as it is the rest of the time. This moves all labels on this line towards the beginning, which reduces the probability of the label being written on top of a label of a nearby line drawn with FPART - 1.</td>
</tr>
<tr>
<td>IGP</td>
<td>9</td>
<td>Flag to control the space left for characters = 9 gap is left = 0 no gap is left</td>
</tr>
<tr>
<td>ICLOSE</td>
<td>6</td>
<td>An internal or external call to set the pen (pen-UP) to a specific position is only executed if this position is more than ICLOSE metacode address units away from the current pen position (distance=difference in X-coordinates + difference in Y-coordinates).</td>
</tr>
<tr>
<td>Variable</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| IOFFS    | 0     | Flag to turn on smoothing code  
= 0  Smoothing and marking (DASHSUPR)  
= -1 Smoothing but no marking (DASHSMTH)  
= 1  No smoothing or marking (DASHCHAR) |
| TENSN    | 2.5   | Tension factor. Must be greater than 0.  
A large tension factor (30.) would essentially turn off smoothing. |
<p>| NP       | 150   | Twice the maximum number of interpolated points on a horizontal line with length equal to that of the grid. More points per unit length are interpolated for short lines than for long lines. |
| SMALL    | 4.    | Minimum distance in plotter units between points. When points on a line are being processed and two or more consecutive points are less than 4 plotter units apart, only one of these consecutive points is saved. This procedure is to prevent cusps. |</p>
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>70</td>
<td>The maximum number of points saved at one time. If there are more than 70 points on a given line, 70 points are processed, then the next 70, until the entire line is processed. Smoothness between segments is maintained automatically. If ( L1 ) is increased, the dimensions of ( XSAVE, YSAVE, XP, YP, ) and ( TEMP ) in ( FRSTD ) must be increased to the new value of ( L1 ).</td>
</tr>
<tr>
<td>ADDLR</td>
<td>2</td>
<td>Number of plotter addresses added to each character string to the left and to the right as free space.</td>
</tr>
<tr>
<td>ADDTB</td>
<td>2</td>
<td>The number of plotter addresses added to each character string on the top and on the bottom as free space.</td>
</tr>
<tr>
<td>MLINE</td>
<td>384</td>
<td>The maximum length in each coordinate of a single line to be processed. Lines longer than ( MLINE ) metacode address units in a coordinate are cut up into smaller segments ad each of these segments is processed separately. This is done to prevent anomalies in the removal of long lines, because only starting points and end points of lines are checked in that process.</td>
</tr>
</tbody>
</table>
SAMPLE PLOT

DEMONSTRATION PLOT FOR DASHSUPR

IPAT=DADADADADADADAD, K=1

IPAT=DDDDDDADADDDDDD, K=2

IPAT=DDDDAADDADDDDDA, K=3

IPAT=DDDDDAAAAAADDDDD, K=4

IPAT=DDDAADDADDDADD, K=5

IN IPAT STRINGS, A AND D SHOULD BE INTERPRETED AS APOSTROPHE AND DOLLAR SIG.
SUBROUTINE EZMAP (JLTS,FLI,FL2,PL3,F14)

DIMENSION OF PL1(2),PL2(2),PL3(2),PL4(2)

ARGUMENTS

LATEST REVISION
July, 1978

PURPOSE
To plot continental, political, and/or U.S. state outlines according to one of nine projections. The origin and orientation of the projections are selected by the user. Points on the earth defined by latitude and longitude are transformed to points in the U,V plane, the plane of projection. The U and V axes are respectively parallel to the X and Y axes of the plotter. A rectangular frame parallel to the U and V axes is chosen and only material within the frame is plotted.

USAGE
EZMAP resides on CRAYLIB and runs on the CRAY-1 mainframe computer. This EZMAP should not be confused with the EZMAP entry of the SUPMAP utility on ULIB. The routine described here is on $NCARLIB; no special measures need be taken to ship it to the CRAY-1. EZMAP contains a SUPMAP entry whose usage is the same as SUPMAP on ULIB except that the absolute value of JGRID (see below) is used for JGRID. For a detailed discussion and examples of the various map projections, consult the documentation for the SUPMAP utility on ULIB.

This entry provides the user with a flexible, yet easy to use means of generating a map. For more direct control, see the routines listed below.

JLTS
JLTS can take on the following character values. It specifies one of four ways in which the limits of the map are defined by the parameters PL1, PL2, PL3, and PL4.

JLTS = 2HMA (MAXIMUM)
The maximum useful area produced by the projection is plotted.

JLTS = 2HCO (CORNERS)
In this case (PL1,PL2) and (PL3,PL4) are the coordinates of two points - (Latitude, Longitude) in degrees - which are to be at opposite corners of the map. (PL1,PL2) should be on the left edge, (PL3,PL4) on the right.

JLTS = 2HPO (POINTS)
PL1, PL2, PL3, and PL4 are two-element arrays giving the latitudes and longitudes of four
points which are to be on the edges of the rectangular map. PL1(1) and PL1(2) are respectively the latitude and longitude of a point on the left edge. Similarly PL2 lies on the right edge, PL3 lies on the lower edge and PL4 lies on the upper edge. Note that the calling program should include: DIMENSION PL1(2),PL2(2),PL3(2),PL4(2)

JLTS = 2HAN (ANGLES)
Here PL1 = AUMIN, PL2 = AUMAX, PL3 = AVMIN,
PL4 = AVMAX, where

AUMIN = Angular distance from origin to left edge of the map.
AUMAX = Angular distance from origin to right edge of the map.
AVMIN = Angular distance from origin to lower edge.
AVMAX = Angular distance from origin to upper edge.

The pole of the projection is chosen to be the north pole if all of PL1-PL4 are positive, the south pole otherwise. In the latter case, the absolute values of PL1 - PL4 are used.

JLTS = NONE OF THE ABOVE
The cylindrical default projection with pole at (0.0,0.0) is drawn.

The center of the projection, i.e. (POLAT, POLONG), is chosen automatically. For the azimuthal projections, the point (POLAT,POLONG) will be chosen to lie in the center of the desired area, except for JLTS = 2HAN. See below. For cylindrical projections POLAT will be on the equator, while POLONG will be in the center of the projection. The conical projection, Lambert conformal with two standard parallels, is only defined for JLTS = 2HCO or 2HPO in the obvious manner.

IF JLTS = 2HMA, THEN PL1, PL2, PL3, and PL4 may be set to JPROJ, POLAT, POLONG, and ROT respectively. (See below.) If PL1 is undecipherable, then default is used unless it is 2HLC otherwise the cylindrical default, 2HCD is used with POLE (0.,0.) and ROT=0.0
IF JLTS = 2HAN and default is not an azimuthal projection, then the azimuthal default, 2HAD, is used.

CONTROL ROUTINES
For more control the following routines should be used:

MAPROJ Define the projection to be used.
MAPSET Set the area you want to see.
MAPOPT Change the cosmetics.
MAPDRW Draw the map.

ON INPUT
The following are the input parameter descriptions for the routines which follow:

Transformation Specification

JPROJ JPROJ defines the projection type according to the following code:

The CONIC Projection:
LC LAMBERT CONFORMAL CONIC with two standard parallels

The AZIMUTHAL Projections:
ST Stereographic
OR Orthographic
LE Lambert equal area
GN Gnomonic
AE Azimuthal equidistant

The CYLINDRICAL Projections:
CE Cylindrical equidistant
ME Mercator
MO Mollweide type

POLAT, POLONG, ROT If JPROJ not equal to 2HLC:

- POLAT and POLONG define in degrees the latitude and longitude of the point on the globe which is to transform to the origin of the U,V plane.

-90 < POLAT < 90
-180 < POLONG < 180

Degrees of latitude north of the equator and degrees of longitude east of the Greenwich Meridian are positive. If the origin is at the north pole, "north" is considered to be in the direction of (POLONG+180). If the origin is at the south pole, "north" is in the direction of POLONG.

- ROT is the angle between the V axis and north at the origin. It is measured in degrees and is taken to be positive if the angular movement from north to
the V axis is counter-clockwise. For the cylindrical projections, the axis of the projection is parallel to the V axis. If JPROJ = 2HLC (Lambert Conformal Conic with two standard parallels)

POLONG = central meridian of projection in degrees.
POLAT, ROT are the two standard parallels in degrees.

Map Area Specification

JLTS can take on the following character values. It specifies one of five ways in which the limits of the map are defined by the parameters PL1, PL2, PL3, and PL4.

**JLTS = 2HMA (MAXIMUM)**
The maximum useful area produced by the projection is plotted. PL1, PL2, PL3, PL4 are not used and may be set to zero.

**JLTS = 2HCO (CORNERS)**
Are the coordinates of two points (latitude, longitude in degrees) which are to be at opposite corners of the map. (PL1, PL2) should be on the left edge, (PL3, PL4) on the right.

**JLTS = 2HPO (POINTS)**
PL1, PL2, PL3, and PL4 are two-element arrays giving the latitudes and longitudes of four points which are to be on the edges of the rectangular map. PL1(1) and PL1(2) are respectively the latitude and longitude of a point on the left edge. Similarly PL2 lies on the right edge, PL3 lies on the lower edge and PL4 lies on the upper edge. Note that the calling program should include:

```
DIMENSION PL1(2), PL2(2), PL3(2), PL4(2)
```

**JLTS = 2HAN (ANGLES)**
Here PL1 = AUMIN, PL2 = AUMAX, PL3 = AVMIN, PL4 = AVMAX, where

- **AUMIN** = Angular distance from origin to left edge of the map.
- **AUMAX** = Angular distance from origin to right edge of the map.
- **AVMIN** = Angular distance from origin to lower edge.
- **AVMAX** = Angular distance from origin to upper edge.

AUMIN, AUMAX, AVMIN, AVMAX must be positive and the
origin must be within the rectangular limits of the map. This option is useful for polar projections. It is not appropriate for the Lambert Conformal with two standard parallels. An error message is issued if an attempt is made to use JLTS=2HAN when JPROJ = 2HLC. (See below.)

**JLTS = 2HLI (LIMITS)**
The minimum and maximum values of U and V are specified by PL1 through PL4.

- \( PL1 = \text{UMIN} \)
- \( PL2 = \text{UMAX} \)
- \( PL3 = \text{VMIN} \)
- \( PL4 = \text{VMAX} \)

Knowledge of the transformation equations is necessary for this option to be used.

**Other Parameters**

**GRID**
GRID gives the interval, in degrees, at which lines of latitude and longitude are to be plotted. A value in the range 1. through 10. will usually be appropriate but higher or lower values are acceptable. If GRID < 0.0, no grids will be drawn.

**OUTLINES**
Specifies which data sets to plot:

- \( 2\text{HC} \) Continental
- \( 2\text{HU} \) U.S.
- \( 2\text{HP} \) Political

The digitized data bank has been divided into three sections:

I. Continental Outlines
   A digitization of the continental outlines of the world at a resolution of about five minutes of arc with an accuracy of six seconds. It is the most commonly used set.

II. U.S. Outlines
   A high resolution digitization of the North United States of America (as opposed to Mexico). Its resolution is near one minute of arc with an accuracy of one second.

III. International Political Boundaries
   This is the union of the previous two data sets together with the political boundaries for the countries of the world.
DOT = .FALSE. for continuous outlines.
    = .TRUE. for dotted outlines.

LABEL A logical value to specify whether to label the meridians and poles.

PERIMETER A logical value specifying whether to draw the perimeter around the map.

DEFAULT The default projection used by EZMAP. Initially it is 2HCE.

RESOLUTION The resolution of the target graphics device.

DASHPAT The DASH pattern to be used when drawing the grids.

CYLINDRICAL DEFAULT (CD) The default cylindrical projection used by EZMAP when called with JLTS either undecipherable or = 2HMA and PL1 is undecipherable. It is initially 2HCE.

AZIMUTHAL DEFAULT (AD) The projection used by EZMAP when called with JLTS = 2HAN and default is not azimuthal. Initially it is 2HST.

ON OUTPUT All arguments are unchanged.

IER Error flag located in common MAPCMB with the following meanings.

= 0 OPERATION SUCCESSFULLY COMPLETED.
= 17 INAPPROPRIATE PROJECTION FOR DEFAULT
= 18 UNDECIPHERABLE PROJECTION FOR DEFAULT
= 33 ATTEMPT TO USE NON-EXISTENT PROJECTION.
= 34 MAP LIMITS INAPPROPRIATE.
= 35 ANGULAR LIMITS TOO GREAT.
= 36 MAP HAS ZERO AREA.
= 43 UNDECIPHERABLE OPTION REQUEST
= 73 UNDECIPHERABLE OUTLINE
= 86 UNDECIPHERABLE MAP AREA SPECIFIER (JLTS)

NOTE Only the first two characters of any character string input parameter are examined.

ENTRY POINTS EZMAP, MAPDRW, MAPGRD, MAPINT, MAPIO,
MAPIT, MAPLSL, MAPLOT, MAPOPT, MAPROJ,
MAPRS, MAPSET, MAPTO, MAPTRN, MAPTRP,
MAPVP, MAPVEC, MAPFST, SUPMAP, SUPCON,
IDICTL
MAPROJ specifies the transformation from (LAT, LONG) to (U, V).

CALL MAPROJ (JPROJ, POLAT, POLONG, ROT)

MAPSET specifies the area of the projection to be plotted. That is, the portion of the U-V plane you wish to look at. Windowing in short.

CALL MAPSET (JLTS, PL1, PL2, PL3, PL4)

MAPOPT used to set the various options within this package.

CALL MAPOPT (IWHICH, OPVAL)

IWHICH the option to be set; a character string.

OPVAL the value to which the option is set.
<table>
<thead>
<tr>
<th>IWHICH</th>
<th>OPVAL</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTLINE</td>
<td>CHARACTER</td>
<td>Which data set to use. Default is continental.</td>
</tr>
<tr>
<td>DOT</td>
<td>LOGICAL</td>
<td>If true, draw dotted continental outlines. Default is false.</td>
</tr>
<tr>
<td>GRID</td>
<td>REAL</td>
<td>The desired GRID spacing. Default is 10.</td>
</tr>
<tr>
<td>LABEL</td>
<td>LOGICAL</td>
<td>If true, label the meridians and poles. Default is true.</td>
</tr>
<tr>
<td>PERIM</td>
<td>LOGICAL</td>
<td>If true, draw the perimeter. Default is true.</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>CHARACTER</td>
<td>Redefine the default projection to be used by EZMAP. Originally = 2HLC</td>
</tr>
<tr>
<td>RE</td>
<td>INTEGER</td>
<td>Resolution of the target graphics device. Default = 1024.</td>
</tr>
<tr>
<td>DASHPAT</td>
<td>INTEGER</td>
<td>Dashed line pattern for the grids.</td>
</tr>
<tr>
<td>CD</td>
<td>CHARACTER</td>
<td>Cylindrical default</td>
</tr>
<tr>
<td>AD</td>
<td>CHARACTER</td>
<td>Azimuthal default</td>
</tr>
</tbody>
</table>

MAPDRW Does the rest, i.e. it simply calls MAPINT, MAPLOT, MAPGRD, and MAPLBL.

MAPLOT Draws (or dots) the desired outline data sets.

CALL MAPLOT

MAPGRD This routine draws the dashed grid lines.

CALL MAPGRD

MAPLBL Labels the international date line, (ID), the equator, (EQ), Greenwich Meridian, (GM), and the poles, (NP and SP), as well as drawing the border around the map.
CALL MAPLBL

MAPTRN Once the transformation has been initialized (by calling MAPINT or otherwise) the subroutine MAPTRN may be called to transform a point, (latitude, longitude) to the corresponding point, (U, V) on the plane. Contours may thus be readily drawn against the map background. (See MAPIT, below).

CALL MAPTRN (RLAT, RLon, U, V)

ON INPUT: RLAT, RLon are the latitude and longitude of a point to be transformed to the U, V plane. \(-90. \leq RLAT \leq 90.\), \(-180. \leq RLon \leq 180.\).

ON OUTPUT: RLAT, RLon are unchanged. U, V are the transformed coordinates of the point (RLAT, RLon).

MAPRS Resets the mapping between the plane of the map projection and that of the plotter. (It recalls SET.)

CALL MAPRS

MAPINT Actually initializes the transformation.

CALL MAPINT

MAPIT

MAPFST

MAPVEC To facilitate drawing lines on the map plotting routines PLOTIT, FIRSTPT, and VECTOR. They are subject to the same restrictions as MAPTRN above.

CALL MAPIT (RLAT, RLon, IFST)
CALL MAPFST (RLAT, RLon)
CALL MAPVEC (RLAT, RLon)

MAPTO Performs the inverse of MAPOPT. That is, it returns the current value of WHICH in addition to returning those options set in MAPOPT. MAPTO will also return values for the following:

EZMAP - 9
<table>
<thead>
<tr>
<th>IWHICH</th>
<th>OPVAL</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>INTEGER</td>
<td>Current error status.</td>
</tr>
<tr>
<td>PR</td>
<td>CHARACTER</td>
<td>Projection</td>
</tr>
<tr>
<td>PT</td>
<td>REAL</td>
<td>Pole Latitude</td>
</tr>
<tr>
<td>PN</td>
<td>REAL</td>
<td>Pole Longitude</td>
</tr>
<tr>
<td>RO</td>
<td>REAL</td>
<td>Rotation</td>
</tr>
<tr>
<td>AR</td>
<td>CHARACTER</td>
<td>Map area specification (JLTS)</td>
</tr>
<tr>
<td>P1</td>
<td>REAL</td>
<td>PL1(1)</td>
</tr>
<tr>
<td>P2</td>
<td>REAL</td>
<td>PL2(1)</td>
</tr>
<tr>
<td>P3</td>
<td>REAL</td>
<td>PL3(1)</td>
</tr>
<tr>
<td>P4</td>
<td>REAL</td>
<td>PL4(1)</td>
</tr>
<tr>
<td>P5</td>
<td>REAL</td>
<td>PL1(2)</td>
</tr>
<tr>
<td>P6</td>
<td>REAL</td>
<td>PL2(2)</td>
</tr>
<tr>
<td>P7</td>
<td>REAL</td>
<td>PL3(2)</td>
</tr>
<tr>
<td>P8</td>
<td>REAL</td>
<td>PL4(2)</td>
</tr>
</tbody>
</table>

CALL MAPTPPO (IWHICH, OPVAL)

MAPVP Called by MAPQV to draw (DOT) lines on the plotter.

MAPTRP Performs interpolation to the edges of the frame.

MAPIO Performs all input of the data sets for MAPLOT.

SUPMAP Provides a single entry for more complex map specification. Its existence is to provide compatibility with the previous program.

CALL SUPMAP (JPROJ, POLAT, POLONG, ROT, PL1, PL2, PL3, PL4, JLTS, JGRID, IUSOUT, IDOT, IER)

SUPCON Also to provide compatibility with the previous program.

COMMON BLOCKS
NAME     LENGTH (Base 10)
MAPCM1   6
MAPCM2   7
MAPCM3   207
MAPCM4   20
MAPCM5   3
MAPCM6   6
MAPCM7   4
MAPCM8   2
MAPCM9   5
MAPCMA   3
MAPCMB   1
MAPCMC   9
MAPCMD   19

I/O Map plotted. Outline data is read from tape.

PRECISION Single

REQUIRED ULIB ROUTINES ULIBER (on system).

LANGUAGE FORTRAN

Revised July, 1974
Revised August, 1976
Revised July, 1978

ALGORITHM The latitudes and longitudes of successive outline points are transformed to coordinates in the plane of projection and joined by a vector.

Lee, Tso-Hwa, STUDENTS SUMMARY REPORTS, WORK-STUDY PROGRAM IN SCIENTIFIC COMPUTING. NCAR 1968.
Parker, R. L., 2UCSD SUPERMAP: WORLD PLOTTING PACKAGE.

ACCURACY The definition of the map produced is limited by two factors:
The resolution of the outline data.

The resolution of the graphics device.

**TIMING**  Less than one half second per map on the CRAY-1 depending upon projection, origin, and orientation. The cylindrical equidistant projection with POLAT = ROT = 0.0 is particularly fast, hence its use as the default.

**REQUIRED RESIDENT RUTINES**  ATAN, TAN, SIN, COS, ALOG, SQRT, ATAN2, ACOS
SUBROUTINE HAFTON (Z,L,M,N,FLO,HI,NLEV,NOPT,NPRM,ISPV,SPVAL)

DIMENSION OF ARGUMENTS
Z(L,M)

LATEST REVISION January 1978

PURPOSE HAFTON draws a half-tone picture from data stored in a rectangular array with the intensity in the picture proportional to the data value.

USAGE If the following assumptions are met, use

CALL EZHFTN (Z,M,N)

Assumptions:

All of the array is to be drawn,

Lowest value in Z will be at lowest intensity on reader/printer output,

Highest value in Z will be at highest intensity,

Values in between will appear linearly spaced,

Maximum possible number of intensities are used,

The picture will have a perimeter drawn,

FRAME will be called after the picture is drawn,

Z is filled with numbers that should be used (no unknown values).

If these assumptions are not met, use

CALL HAFTON (Z,L,M,N,FLO,HI,NLEV,NOPT,NPRM,ISPV,SPVAL)

ARGUMENTS

On Input for EZHFTN

Z M by N array to be used to generate a half-tone plot

M First dimension of Z
N Second dimension of Z

On Output for EZHFTN
All arguments are unchanged.

On Input for HAFTON

Z The origin of the array to be plotted

L The first dimension of Z in the calling program

M The number of data values to be plotted in the x-direction (the first subscript direction). When plotting all of an array, L=M. See Appendix C of the Graphics Utilities Used at NCAR for an explanation of using this argument list to process any part of an array.

N The number of data values to be plotted in the y-direction (the second subscript direction).

FLO The value of Z that corresponds to the lowest intensity. (When NOPT.LT.0, FLO corresponds to the highest intensity.) If FLO=HI=0, MIN(Z) is used for FLO.

HI The value of Z that corresponds to the highest intensity. (When NOPT.LT.0, HI corresponds to the lowest intensity.) If HI=FLO=0, MAX(Z) is used for HI.

NLEV The number of intensity levels desired. 16 maximum. If NLEV=0 or 1, 16 levels are used.

NOPT Flag to control the mapping of Z onto the intensities. The sign of NOPT controls the directness or inverseness of the mapping.

* NOPT positive yields direct mapping. That is, the largest Zs produce the densest dots. On mechanical plotters, big Zs will produce a dark area on the paper. With the film development methods used at NCAR, big Zs will produce many (white) dots on the film, also resulting in a dark area on reader printer paper.

* NOPT negative yields inverse mapping. That is, the smallest Zs produce the densest dots resulting in dark areas on the paper.

The absolute value of NOPT determines the mapping of Z onto the intensities. For IABS(NOPT)

= 0 The mapping is linear. That is, for each intensity there is an equal range in Z value.
= 1 The mapping is linear. That is, for each intensity there is an equal range in Z value.

= 2 The mapping is exponential. That is, for higher Zs there is a larger difference in intensity for relatively close Zs. Details in the bigger Zs are displayed at the expense of the smaller Zs.

= 3 The mapping is logarithmic, so details of smaller Zs are shown at the expense of larger Zs.

= 4 Sinusoidal mapping, so middle-size Zs are shown at the expense of extreme Zs.

= 5 Arcsine mapping, so extreme sized Zs are shown at the expense of middle-sized Zs.

**NPRM** Flag to control the drawing of a perimeter around the half-tone picture.

- NPRM = 0: Perimeter drawn with ticks pointing at data locations. (Side lengths proportional to number of data values.)
- NPRM positive: No perimeter, picture fills frame.
- NPRM negative: Picture within confines of user's last SET call.

**ISPV** Flag to tell if the special value feature is being used. The special value feature is used to mark areas where the data is not known or holes are wanted in the picture.

- ISPV = 0: Special value feature not in use. SPVAL is ignored.
- ISPV non-zero: Special value feature in use. SPVAL contains the special value. Where Z contains the special value, no half-tone is drawn. If ISPV = 0 Special value feature not in use. SPVAL is ignored.
- = 1 Nothing is drawn in special value area.
- = 2 Contiguous special value areas are surrounded by a polygonal line.
- = 3 Special value areas are filled with Xs.
- = 4 Special value areas are filled with the highest intensity.
The value used in Z to mark unknown areas. This argument is ignored if ISPV = 0.

On Output for HAFTON

All arguments are unchanged.

NOTE

This routine produces a huge number of plotter instructions per picture, averaging over 100,000 point calls per dd80 frame when M=N. The dd80 must be assigned on-line (*ASSIGN, DD80=ONLINE) when more than about 100 frames are produced. Consult CF Notes "39, page 7 for putting the dd80 on line.

ENTRY POINTS

EZHFTN, HAFTON, ZLSET, GRAY, BOUND

COMMON BLOCKS

HAFT01 3
HAFT02 7
HAFT03 142

I/O

Plots half-tone picture

PRECISION

Single

REQUIRED ULIB ROUTINES

None

LANGUAGE

FORTRAN

HISTORY

Rewrite of PHOMAP originally written by M. Perry of High Altitude Observatory, NCAR.

ALGORITHM

Bi-linear interpolation on plotter (resolution-limited) grid of normalized representation of data.

SPACE REQUIRED

24428

TIMING

About 1 second on the 7600 for this dd80 version, more or less regardless of the size of Z.

PORTABILITY

An implementation's write-up is available.

PLOTTING ROUTINES USED

FRAME, SET, GETSET, PERIM, PWRT, POINT, and LINE.

REQUIRED RESIDENT ROUTINES

None
## INTERNAL PARAMETERS

<table>
<thead>
<tr>
<th>NAME</th>
<th>DEFAULT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>XLT</td>
<td>0.1</td>
<td>Left-hand edge of the plot when NSET=0 (0.0 = left edge of frame, 1.0 = right edge of frame.)</td>
</tr>
<tr>
<td>YBT</td>
<td>0.1</td>
<td>Bottom edge of the plot when NSET=0 (0.0 = bottom of frame, 1.0 = top of frame.)</td>
</tr>
<tr>
<td>SIDE</td>
<td>0.8</td>
<td>Length of longer edge of plot (see also EXT)</td>
</tr>
<tr>
<td>EXT</td>
<td>.25</td>
<td>Lengths of the sides of the plot are proportional to M and N (when NSET=0) except in extreme cases, namely, when MIN(M,N)/MAX(M,N) is less than EXT. Then a square plot is produced. When a rectangular plot is produced, the plot is centered on the frame (as long as SIDE + 2<em>XLT = SIDE + 2</em>YBT = 1.0, as with the defaults.)</td>
</tr>
<tr>
<td>ALPHA</td>
<td>1.6</td>
<td>A parameter to control the extremeness of the mapping function specified by NOPT. (For IABS(NOPT) = 0 or 1, the mapping function is linear and independent of ALPHA.) For the non-linear mapping functions, when ALPHA is changed to a number closer to 1.0, the mapping function becomes more linear. When ALPHA is changed to a larger number, the mapping function becomes more extreme.</td>
</tr>
<tr>
<td>MXLEV</td>
<td>16</td>
<td>Maximum number of levels. Limited by plotter.</td>
</tr>
<tr>
<td>NCRTG</td>
<td>8</td>
<td>Number of plotter units per gray-scale cell. Limited by plotter.</td>
</tr>
<tr>
<td>NCRTF</td>
<td>1024</td>
<td>Number of plotter units per frame.</td>
</tr>
<tr>
<td>IL</td>
<td>(below)</td>
<td>An array defining which of the available intensities are used when less than the maximum number of intensities are requested.</td>
</tr>
<tr>
<td>NLEV</td>
<td>Intensities Used</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5, 11</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4, 8, 12</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3, 6, 10, 13</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2, 5, 8, 11, 14</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1, 4, 7, 9, 12, 15</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1, 4, 6, 8, 10, 12, 15</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1, 3, 5, 7, 9, 11, 13, 15</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1, 3, 4, 6, 8, 10, 12, 13, 15</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1, 3, 4, 6, 7, 9, 10, 12, 13, 15</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1, 2, 3, 5, 6, 8, 10, 11, 13, 14, 15</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 15</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 13, 14, 15</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15</td>
<td></td>
</tr>
</tbody>
</table>
SAMPLE PLOT

DEMONSTRATION PLOT FOR ENTRY EZHFTN OF HAFTON
DEMONSTRATION PLOT FOR ENTRY HAFTON OF HAFTON
SUBROUTINE ISOSRF (T,LU,MU,LV,MW,EYE,MUVWP2,SLAB,TISO,IFLAG)

DIMENSION OF ARGUMENTS
T(LU,LV,MW),EYE(3),SLAB(MUVWP2,MUVWP2)

LATEST REVISION
June 1980

PURPOSE
ISOSRF draws an approximation of an iso-valued surface from a three-dimensional array with hidden lines removed.

USAGE
If the following assumptions are met, use:

CALL EZISOS (T,MU,MV,MW,EYE,SLAB,TISO)

Assumptions:

All of the T array is to be used,
IFLAG is chosen internally,
FRAME is called by EZISOS.

If the assumptions are not met, use:

CALL ISOSRF (T,LU,MU,LV,MV,MW,EYE,MUVWP2,SLAB,TISO,IFLAG)

ARGUMENTS

On Input
T
Three dimensional array of data that is used to determine the iso-valued surface.

LU
First dimension of T in the calling program.

MU
The number of data values of T to be processed in the U direction (the first subscript direction). When processing the entire array, LU= MU (and LV=MV). See Appendix C of the Graphics Utilities Used at NCAR for an explanation of using this argument list to process any part of an array.

LV
Second dimension of T in the calling program.

MV
The number of data values of T to be processed in the V direction (the second subscript direction).
MW
The number of data values of T to be processed in the W direction (the third subscript direction).

EYE
The position of the eye in three-space. T is considered to be in a box with opposite corners \((1,1,1)\) and \((MU,MV,MW)\). The eye is at \((EYE(1), EYE(2), EYE(3))\), which must be outside the box that T is in. While gaining experience with the routine, a good choice for EYE might be \((5.0*MU,3.5*MV,2.0*MW)\).

MUVWP2
The maximum of \((MU,MV,MW) + 2\); that is, \(MUVWP2 = \text{MAX}(MU,MV,MW) + 2\)

SLAB
A work space used for internal storage. SLAB must be at least \(MUVWP2*MUVWP2\) words long.

TISO
The iso-value used to define the surface. The surface drawn will separate volumes of T that have value greater than or equal to TISO from volumes of T that have value less than TISO.

IFLAG
A flag which serves two purposes:

- First, the absolute value of IFLAG determines which types of lines are drawn to approximate the surface. Three types of lines are considered: lines of constant U, lines of constant V, and lines of constant W. The following table lists the types of lines drawn:

| \(|ABS(IFLAG)|\) | LINES OF CONSTANT |
|-----------------|------------------|
| 1               | No No Yes        |
| 2               | No Yes No        |
| 3               | No Yes Yes       |
| 4               | Yes No No        |
| 5               | Yes No Yes       |
| 6               | Yes Yes No       |
| 0,7 or more     | Yes Yes Yes      |

See the example picture for further clarification.

- Second, the sign of IFLAG determines what is inside and what is outside, hence, which lines are visible and what is done at the boundary of T.
For IFLAG:

positive; T values greater than TISO are assumed to be inside the solid formed by the drawn surface.

negative; T values less than TISO are assumed to be inside the solid formed by the drawn surface.

If the algorithm draws a cube, reverse the sign of IFLAG.

On Output T, LU, MU, LV, MV, MW, EYE, MUVWP2, TISO, and IFLAG are unchanged. SLAB has been written in.

NOTE

• This routine is for lower resolution arrays than ISOSRFHR. 40 by 40 by 40 is a practical maximum.

• Simple transformations can be achieved by adjusting scaling statement functions in ISOSRF, SET3D, and TR32.

• The hidden-line algorithm is not exact, so visibility errors can occur. The sample picture is typical.

• Three-dimensional perspective character labeling of ISOSRF plots is possible by using the ULIB routine PWRZI. For a description of the usage, see the PWRZI write-up.

ENTRY POINTS ISOSRF, EZISOS, SET3D, TRN32I, ZEROSC, STCNTR, DRCNTR, TR32, FRSTS, KURV1S, KURV2S, FRSTC, FILLIN, DRAWI, ISOSRB, MMASK
**COMMON BLOCKS**

<table>
<thead>
<tr>
<th>NAME</th>
<th>LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISOSR1</td>
<td>4</td>
</tr>
<tr>
<td>ISOSR2</td>
<td>4003 (OCTAL)</td>
</tr>
<tr>
<td>ISOSR3</td>
<td>7</td>
</tr>
<tr>
<td>ISOSR4</td>
<td>2</td>
</tr>
<tr>
<td>ISOSR5</td>
<td>22 (OCTAL)</td>
</tr>
<tr>
<td>ISOSR6</td>
<td>1015 (OCTAL)</td>
</tr>
<tr>
<td>ISOSR7</td>
<td>2</td>
</tr>
<tr>
<td>ISOSR8</td>
<td>44 (OCTAL)</td>
</tr>
<tr>
<td>ISOSR9</td>
<td>2</td>
</tr>
<tr>
<td>TEMPR</td>
<td>1</td>
</tr>
<tr>
<td>PWRZIT</td>
<td>12 (OCTAL)</td>
</tr>
</tbody>
</table>

**I/O**
Plots surface

**PRECISION**
Single

**REQUIRED ULIB ROUTINES**
If three-dimensional character writing is required, use PWRZI.

**LANGUAGE**
FORTRAN

**HISTORY**
Developed for users of ISOSRFHR with smaller arrays

**ALGORITHM**
Cuts through the three-dimensional array are contoured with a smoothing contourer which also marks a model of the plotting plane. Interiors of boundaries are filled in and the result is OR.ed into another model of the plotting plane which is used to test subsequent contour lines for visibility.

**SPACE REQUIRED**
About 110008 not including the system plot package

**TIMING**
Varies widely with size of T and the volume of the space enclosed by the surface drawn. The sample picture took about 1 second of 7600 time.

**PORTABILITY**
Portable

**PLOTTING ROUTINES USED**
PLOTIT, FRAME

**REQUIRED RESIDENT ROUTINES**
SQRT, COS, SIN, COS, ATAN2, EXP, SBYTES, GBYTES
<table>
<thead>
<tr>
<th>NAME</th>
<th>DEFAULT</th>
<th>FUNCTION</th>
</tr>
</thead>
</table>
| IREF | 1       | Flag to control drawing of axes  
  * IREF=non-zero means draw axes.  
  * IREF=0 means don't.       |
SAMPLE PLOT

DEMONSTRATION PLOT FOR ENTRY EZISOS OF ISOSRF

ISOSRF - 6
SAMPLE PLOT

DEMONSTRATION PLOT FOR ENTRY ISOSRF OF ISOSRF
ISO-SURFACE PACKAGE FOR HIGH RESOLUTION 3-DIMENSIONAL ARRAYS

DIMENSION OF ARGUMENTS

EYE(3), ST1(NV, NW, 2), IS2(LX, NY), S(4), IOBJS(MV, NW)

LATEST REVISION
January 1978

PURPOSE
This package of three routines produces a perspective picture of an arbitrary object or group of objects with the hidden parts not drawn. The objects are assumed to be stored in the format described below; a format which was chosen to facilitate the display of functions of three variables (Figure 1) or output from high resolution three-dimensional computer simulations (Figure 2).

USAGE
The object is defined by setting words in a three-dimensional array to one where the object is and zero where it is not. That is, the position in the array corresponds to a position in three-space, and the value of the array tells whether any object is present at that position or not. Because a large array is needed to define objects with good resolution, only a part of the array is passed to the package with each call.

There are three subroutines in the package. INIT3D is called at the beginning of a picture. This call can be skipped sometimes if certain criteria are met and certain precautions are taken. See the Timing section for details. DANDR (Draw AND Remember) is called successively to process different parts of the three-dimensional array. For example, in Figure 3, the nearer plane would be processed in the first call to DANDR, while the farther plane would be processed in a subsequent call. An example follows:

CALL INIT3D(EYE, NU, NV, NW, ST1, LX, NY, IS2, IU, S)
DO 1 IBKWD5=1, NU
   I=NU+1-IBKWD5
   C FORM OR READ SLAB I OF THE 3 DIMENSIONAL ARRAY
   C ONLY 1 OR 0 IN THE SLAB, CALLED IOBJS
   ...  
   1 CALL DANDR(NV, NW, ST1, LX, NX, NY, IS2, IU, S, IOBJS, MV)
ARGUMENTS

On Input

EYE An array 3 long containing the U, V, and W coordinates of the eye position. Objects are considered to be in a box with 2 extreme corners at (1,1,1) and (NU,NV,NW). The eye position must be positive coordinates away from the coordinate planes U=0,V=0, and W=0. While gaining experience with the package, use EYE(1)=5*NU, EYE(2)=4*NV, EYE(3)=3*NW.

NU U direction length of the box containing the objects

NV V direction length of the box containing the objects

NW W direction length of the box containing the objects

ST1 A scratch array at least NV*NW*2 words long.

LX The number of words needed to hold NX bits. Also, the first dimension of IS2. See NX and IS2. On the 7600, LX=(NX-1)/60+1.

NX Number of cells in the x-direction of a model of the image plane. A silhouette of the parts of the picture processed so far is stored in this model. Lines to be drawn are tested for visibility by examining the silhouette. Lines in the silhouette are hidden. Lines out of the silhouette are visible. The solution is approximate because the silhouette is not formed exactly. See IS2.

NY Number of cells in the y-direction of the image plane. Also the second dimension of IS2.

IS2 An array to hold the image plane model. It is dimensioned LX by NY. The model is NX by NY and packed densely. If hidden lines are drawn, decrease NX and NY (and LX if possible). If visible lines are left out of the picture, increase NX and NY (and LX if need be). As a guide, some examples showing successful choices are listed:

<table>
<thead>
<tr>
<th>GIVEN NU NV NW</th>
<th>RESULTING NX NY FROM TESTING</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 100 60</td>
<td>200 200</td>
</tr>
<tr>
<td>60 60 60</td>
<td>110 110</td>
</tr>
<tr>
<td>40 40 40</td>
<td>75 75</td>
</tr>
</tbody>
</table>

IU Unit number of scratch file for the package. ST1 will be written NU times on this file.
S  A real array 4 long which contains the plotter coordinates of the area where the picture is to be drawn. That is, all plotting coordinates generated will be bounded as follows: X coordinates will be between S(1) and S(2), Y coordinates will be between S(3) and S(4). To prevent distortion, have S(2)-S(1)=S(4)-S(3).

\[ 10.0 \leq S(I) \leq 1010.0 \quad I=1,2,3,4. \]

IOBJS  An NV by NW array (with actual first dimension MV in the calling program) describing the object. If this is call number 1 to DANDR, the part of the picture at U=NU+1-I is to be processed. IOBJS defines the objects to be drawn in the following manner: IOBJS(J,K)=1 if any object contains the point (NU+1-I,J,K) and IOBJS(J,K)=0 otherwise.

MV  Actual first dimension of IOBJS in the calling program. When plotting all of IOBJS, NV=MV. See Appendix ( ) of the graphics chapter for an explanation of using this argument list to process any part of an array.

On Output  EYE,NU,NV,NW,LXNX,NY,IU,S,IOBJS, and MV are unchanged. ST1 and IS2 have been written in.

NOTE  This routine is for large arrays, 40 x 40 x 30 is a practical minimum.

ENTRY POINTS  INIT3D, SETORG, PERSPC, and DANDR

COMMON BLOCKS  None

I/O  Plots visible surfaces, uses scratch file or tape

PRECISION  Single

REQUIRED ULIB  None

ROUTINES  FORTRAN

HISTORY  Originally developed at NCAR starting in late 1970

ALGORITHM  The basic method is to contour cuts through the array starting with a cut nearest the observer. The algorithm leaves out the hidden parts of the contours by suppressing lines enclosed within lines produced while processing preceding cuts. The technique is described in detail in the reference cited below.

The algorithm is not exact. However, reasonable pictures are produced.

This routine is very time consuming. If many pictures are produced with the same size arrays and eye position, much time can be saved by rewinding unit IU, filling IS2 with zeros, and skipping the call to INIT3D for other than the first picture.

Two machine-dependent constants are initialized in DANDR. SETORG has and ENTRY statement for PERSPC. In DANDR, .AND. and .OR. are used for masking operations.

LINE

SQRT, ACOS, SIN

SPACE REQUIRED 14748

ACCURACY

PORTABILITY

PLOTTING ROUTINES USED

REQUIRED RESIDENT ROUTINES
Figure 1. Four contour surfaces of the wave function of a 3-P electron in a one electron atom. 50 x 50 x 50 object cube, 100 x 100 screen model.
Figure 2. Output from a 3-dimensional cloud model. 100 x 100 x 60 object cube, 200 x 200 screen model.
Figure 3. Nearer plane would be processed in first call to DANDR, while the farther plane would be processed in a subsequent call.
SUBROUTINE PWRITX (X, Y, IDPC, NCHAR, JSIZE, JOR, JCTR)

PWRITX (X, Y, IDPC, NCHAR, JSIZE, JOR, JCTR)

DIMENSION OF ARGUMENTS
IDPC((NCHAR-1)/NCHPW+1)
(NCHPW = Number of Characters Per Word)

LATEST REVISION
September, 1980

PURPOSE
PWRITX is a character plotting routine. It produces high quality characters for annotating graphs, making movie titles, etc. It replaces the former character plotting routine PWRX.

User's Note: This subroutine replaces PWRX on ULIB.

USAGE
CALL PWRITX (X, Y, IDPC, NCHAR, JSIZE, JOR, JCTR)

ARGUMENTS

Common Block PUSER
By default PWRITX uses the complex character set (set used by former PWRX on ULIB). To make PWRITX use the duplex character set (set used by former PWRX on XLIB) the user has to define in his main program a common block PUSER containing 1 integer variable. If this variable is initialized to 1 before the first call to PWRITX, then the duplex character set is used by PWRITX.

Example

```
COMMON /PUSER/ MODE

MODE = 1

CALL PWRITX(....................)
```

NOTE: The character set cannot be changed after the first call to PWRITX.

INPUT

X, Y Positioning coordinates for the characters to be drawn. These can be integers in the range 1 through 1024 or floating point numbers, in which case they are scaled according to the most recent SET call. Also see JCTR.
IDPC Characters to be drawn and FUNCTION CODES. (See below).

NCHAR The number of characters in IDPC, including characters to be drawn and function codes.

JSIZE Size of the Character.

- If between 0 and 3, it is 1, 1.5, 2, or 3 times digitized character width. (see FUNCTION CODES below for these sizes.)

- If greater than 3 it is the desired plotter address units for principal character height, i.e. principal characters will be JSIZE plotter address units high, and indexical and cartographic characters will be scaled proportionally, such that:

  INDEXICAL = 13/21 PLA UNITS
  CARTOGRAPHIC = 9/21 PLA UNITS HIGH.

IOR Character string orientation in degrees counter-clockwise from the horizontal.

JCTR Centering option.

- = 0 (X,Y) is the center of the entire string.
- = -1 (X,Y) is the center of the left edge of the first character.
- = 1 (X,Y) is the center of the right edge of the first character.

On Output All arguments are unchanged.

FUNCTION CODES Function codes may be included in the character string IDPC to change font, case, etc. within a string of text. All function instructions must be enclosed in apostrophes. (The apostrophe = 3,4 punch on the keypunch, " when printed.) No punctuation is needed between functions except for a comma between adjacent numbers; however, commas may be used between functions to improve legibility. The following are the only legal function codes. Any other characters in a function string will be ignored except that an error message will be printed and, if more than 10 errors occur within a string, control will be returned to the main program. At the first call to PWRITX, SIZE, TYPE, and CASE are: Principal, Roman, and Upper.

PLA = Plotter address (for resolution 10)

PA = User plotter address (resolution as defined by user)
A. FONT DEFINITIONS

R ROMAN Type characters
G GREEK Type characters

B. SIZE DEFINITIONS

P Principal size, digitized to be 21 PLA units high. The total character including white space is 32 PLA units high. A carriage return or a Y increment will space down 32 PLA units. A blank or an X increment will space across 16 PLA units. NOTE: CHARACTERS VARY IN WIDTH.

I Indexical size, digitized to be 13 PLA units high and 20 PLA units high including white space. A carriage return or a Y increment is 20 PLA units. Blanks or X increments are 12 PLA units.

K Cartographic size, digitized to be 9 PLA units high and 14 PLA units high including white space. Carriage return or Y increments are 14 PLA units. Blanks or X increments are 8 PLA units.

C. CASE DEFINITIONS

U or UN. Upper Case
If U is followed by a number N (not separated by a comma) then N characters will be drawn in upper case, subsequent characters will be in lower case. (The UN option is particularly useful for capitalizing sentences.)

L or LN. Lower Case
If L is followed by a number N then N characters will be drawn in lower case and subsequent characters will be in upper case.

D. LEVEL DEFINITIONS

S or SN. Superscript Level

B or BN. Subscript Level

N or NN. Normal Level

When super- or sub-scripting, the character size will change depending on the previous character drawn. Principal base characters will be sub- or super-scripted with indexical characters, with a 10 PLA unit shift (scaled to
JSIZE) up or down. Indexical and cartographic base characters will be sub- or super-scripted with cartographic characters with a 7 PLA unit shift.

The case of the indexing characters will remain the same as that of the base character unless otherwise specified, except that a lower case indexical base will be super- or sub-scripted with upper case cartographic, as the cartographic type has no lower case alphabetic or numeric characters available.

If $S,B$ or $N$ is followed by a number $N$, ($N$ not equal to 0), then $N$ characters will be drawn as specified above, after which character size, case and position will be reset to the base character. If $N$ is negative its absolute value will be used instead. Do not overlap level definitions given for a specified number of characters. The $N$ option returns character case and size to the base, but maintains the current character position.

**Example:** "$U1"T"S1"E"ST will be written as:

```
E
TST
```

Whereas "$U1"T"S"E"N"ST will be written as:

```
E
T ST
```

E. COORDINATE DEFINITIONS (Descriptions assume normal UPA unit space.)

$H,HN,H,Q$. Increment in the X direction.

If this option appears without a number $N$, $N$ will be taken to be 1. $HN$ will shift the present X position $N$ UPA units. If $N$ is positive, the shift is to the right, if $N$ is negative the shift is to the left. If $HN$ is followed by $A Q$, the X position will be incremented by $N$ character widths (i.e., $N$ blanks) either right or left.
V,VN,VNQ. Increment in the Y direction.
If this option appears without a number
N, N will be taken to be 1. VN will shift the present Y position N UPA
units. If N is positive, the shift is up, if N is negative, the shift is down.
If VN is followed by A Q, the Y position will be incremented by N lines up or
down.

X,XN. SET X.
If X appears without a number N, this will act as a do-nothing statement.
Otherwise, the character position in the X direction will be set to the UPA
cordinate N, so that the next character drawn will be centered on N and
subsequent characters will be drawn from this position.

Y,YN. SET Y.
This works the same as SET X.

C
A carriage return will be done before the next character is plotted.

F. DIRECTION DEFINITIONS

D,DN. Write Down, Rather than Across the Frame.
If D appears without an N or if N=0, all characters will be written down, until
an "A" function is encountered. If D is followed by a number N, N characters
will be written down and subsequent characters will be written across the frame.
If N is negative, the absolute value of N is used instead.

A
Write across. Escape from D option.

G. DIRECT CHARACTER ACCESS

NNN. Numeric Character.
Character number NNN will be drawn.
NNN is base 8.

NOTE:  All characters in a given call are drawn in the same intensity. If JSIZE is less than or equal to 2,
characters are in low intensity, otherwise they are in high intensity. Return to the main program is
always in high intensity.

- On other than the first entry to PWRX, FONT, CASE, etc. are in the state last assumed in the previous call.

- FONT, SIZE, and CASE are reset to default values if they were only set for a specified number of characters in the previous call.

- The previous case is always reset to upper.

- The direction is always reset to down.

- Numbers for direct character access must not be signed. All other numbers can be signed.

PORTABILITY
Contact the specialist for a portable version.

PLOTTING ROUTINES
USED
PLOTIT, FL2INT, GEISI.

REQUIRED RESIDENT ROUTINES
SQRT, SIN, COS.

ENTRY POINTS
PWRITX, GTNUM, GTNUMB, XTCH, PWRXBD

COMMON BLOCKS
PWRCO 2496 (for 60-bit words only)
PWRCl 150
PSAV1 9
PSAV2 5
USER 1

I/O Plots characters.

PRECISION Single

REQUIRED ULIB ROUTINES
None

LANGUAGE FORTRAN 66

HISTORY Implemented by Dori Bundy to make Hershey's character set more usable. Made portable in 1978 by Robert Ubelmesser for use on all computer systems which support plotters with up to 15-bit resolution.
The following examples show all the characters available to PWRITX. Characters may be accessed in two ways. The set may be declared in a function code, then the desired character in that set is determined by KP, a keypunch character. For example, the six-character string 'KGL.'A would produce an @.

A character may be specified numerically. The numeric representation of a character is found by adding the appropriate number 'NUM' to the base number for the set in which the desired character is located. For example, the six-character string '1301' would produce an @.
### SAMPLE PLOT/CHARACTER ACCESS TABLE

<table>
<thead>
<tr>
<th>SET=</th>
<th>PRU</th>
<th>PRL</th>
<th>IRU</th>
<th>IRL</th>
<th>KRU</th>
<th>KRL</th>
<th>PGU</th>
<th>PGL</th>
<th>IGU</th>
<th>IGL</th>
<th>KGU</th>
<th>KGL</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASE=</td>
<td>0</td>
<td>100</td>
<td>200</td>
<td>0300</td>
<td>0400</td>
<td>0500</td>
<td>0600</td>
<td>0700</td>
<td>1000</td>
<td>1100</td>
<td>1200</td>
<td>1300</td>
</tr>
<tr>
<td>KP NUM</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>A</td>
<td>A</td>
<td>α</td>
<td>α</td>
<td>A</td>
<td>@</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>B</td>
<td>B</td>
<td>β</td>
<td>β</td>
<td>B</td>
<td>§</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>C</td>
<td>C</td>
<td>γ</td>
<td>γ</td>
<td>γ</td>
<td>γ</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>D</td>
<td>D</td>
<td>Δ</td>
<td>Δ</td>
<td>Δ</td>
<td>Δ</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>E</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>E</td>
<td>ε</td>
<td>ε</td>
<td>E</td>
<td>ε</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>F</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>F</td>
<td>F</td>
<td>θ</td>
<td>θ</td>
<td>ψ</td>
<td>ψ</td>
</tr>
<tr>
<td></td>
<td>G</td>
<td>G</td>
<td>g</td>
<td>g</td>
<td>g</td>
<td>g</td>
<td>G</td>
<td>G</td>
<td>Θ</td>
<td>Θ</td>
<td>Θ</td>
<td>Θ</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>h</td>
<td>h</td>
<td>h</td>
<td>h</td>
<td>h</td>
<td>h</td>
<td>h</td>
<td>θ</td>
<td>θ</td>
<td>θ</td>
<td>θ</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>i</td>
<td>i</td>
<td>i</td>
<td>i</td>
<td>i</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>J</td>
<td>j</td>
<td>j</td>
<td>j</td>
<td>j</td>
<td>j</td>
<td>J</td>
<td>J</td>
<td>Λ</td>
<td>Λ</td>
<td>Λ</td>
<td>Λ</td>
</tr>
<tr>
<td></td>
<td>K</td>
<td>k</td>
<td>k</td>
<td>k</td>
<td>k</td>
<td>k</td>
<td>K</td>
<td>K</td>
<td>M</td>
<td>μ</td>
<td>M</td>
<td>μ</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>l</td>
<td>l</td>
<td>l</td>
<td>l</td>
<td>l</td>
<td>L</td>
<td>L</td>
<td>\</td>
<td>μ</td>
<td>M</td>
<td>μ</td>
</tr>
</tbody>
</table>

PWRITX-8
### SAMPLE PLOT/CHARACTER ACCESS TABLE

<table>
<thead>
<tr>
<th>SET=</th>
<th>PRU</th>
<th>PRL</th>
<th>IRU</th>
<th>IRL</th>
<th>KRU</th>
<th>KRL</th>
<th>PGU</th>
<th>PGL</th>
<th>IGU</th>
<th>IGL</th>
<th>KGU</th>
<th>KGL</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASE=</td>
<td>0</td>
<td>100</td>
<td>200</td>
<td>300</td>
<td>400</td>
<td>500</td>
<td>600</td>
<td>700</td>
<td>1000</td>
<td>1100</td>
<td>1200</td>
<td>1300</td>
</tr>
<tr>
<td>KP NUM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M 15</td>
<td>M m</td>
<td>M m</td>
<td>M m</td>
<td>N</td>
<td>N v</td>
<td>N v</td>
<td>N v</td>
<td>N v</td>
<td>N v</td>
<td>N v</td>
<td>N v</td>
<td>N v</td>
</tr>
<tr>
<td>N 16</td>
<td>N n</td>
<td>N n</td>
<td>N n</td>
<td>/</td>
<td>Ε</td>
<td>Ε</td>
<td>Ε</td>
<td>Ε</td>
<td>Ε</td>
<td>Ε</td>
<td>Ε</td>
<td>Ε</td>
</tr>
<tr>
<td>O 17</td>
<td>O o</td>
<td>O o</td>
<td>O o</td>
<td>O /</td>
<td>O o</td>
<td>O o</td>
<td>O o</td>
<td>O o</td>
<td>O o</td>
<td>O o</td>
<td>O o</td>
<td>O o</td>
</tr>
<tr>
<td>P 20</td>
<td>P p</td>
<td>P p</td>
<td>P p</td>
<td></td>
<td>Π</td>
<td>Π</td>
<td>Π</td>
<td>Π</td>
<td>Π</td>
<td>Π</td>
<td>Π</td>
<td>Π</td>
</tr>
<tr>
<td>Q 21</td>
<td>Q q</td>
<td>Q q</td>
<td>Q q</td>
<td>\</td>
<td>Ρ</td>
<td>Ρ</td>
<td>Ρ</td>
<td>Ρ</td>
<td>Ρ</td>
<td>Ρ</td>
<td>Ρ</td>
<td>Ρ</td>
</tr>
<tr>
<td>R 22</td>
<td>R r</td>
<td>R r</td>
<td>R r</td>
<td>\</td>
<td>Σ</td>
<td>Σ</td>
<td>Σ</td>
<td>Σ</td>
<td>Σ</td>
<td>Σ</td>
<td>Σ</td>
<td>Σ</td>
</tr>
<tr>
<td>S 23</td>
<td>S s</td>
<td>S s</td>
<td>S s</td>
<td>-</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T 24</td>
<td>T t</td>
<td>T t</td>
<td>T t</td>
<td></td>
<td>Τ</td>
<td>Τ</td>
<td>Τ</td>
<td>Τ</td>
<td>Τ</td>
<td>Τ</td>
<td>Τ</td>
<td>Τ</td>
</tr>
<tr>
<td>U 25</td>
<td>U u</td>
<td>U u</td>
<td>U u</td>
<td></td>
<td>Φ</td>
<td>Φ</td>
<td>Φ</td>
<td>Φ</td>
<td>Φ</td>
<td>Φ</td>
<td>Φ</td>
<td>Φ</td>
</tr>
<tr>
<td>V 26</td>
<td>V v</td>
<td>V v</td>
<td>V v</td>
<td>\</td>
<td>Χ</td>
<td>Χ</td>
<td>Χ</td>
<td>Χ</td>
<td>Χ</td>
<td>Χ</td>
<td>Χ</td>
<td>Χ</td>
</tr>
<tr>
<td>W 27</td>
<td>W w</td>
<td>W w</td>
<td>W w</td>
<td></td>
<td>Ψ</td>
<td>Ψ</td>
<td>Ψ</td>
<td>Ψ</td>
<td>Ψ</td>
<td>Ψ</td>
<td>Ψ</td>
<td>Ψ</td>
</tr>
<tr>
<td>X 30</td>
<td>X x</td>
<td>X x</td>
<td>X x</td>
<td></td>
<td>Ω</td>
<td>Ω</td>
<td>Ω</td>
<td>Ω</td>
<td>Ω</td>
<td>Ω</td>
<td>Ω</td>
<td>Ω</td>
</tr>
</tbody>
</table>

PWRITX-9
SAMPLE PLOT/CHARACTER ACCESS TABLE

<table>
<thead>
<tr>
<th>KP NUM</th>
<th>PRU</th>
<th>PRL</th>
<th>IRU</th>
<th>IRL</th>
<th>KRU</th>
<th>KRL</th>
<th>PGU</th>
<th>PGL</th>
<th>IGU</th>
<th>IGL</th>
<th>KGU</th>
<th>KGL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
<td>200</td>
<td>0300</td>
<td>0400</td>
<td>0500</td>
<td>0600</td>
<td>0700</td>
<td>1000</td>
<td>1100</td>
<td>1200</td>
<td>1300</td>
<td></td>
</tr>
</tbody>
</table>

Y 31 Y Y Y y y Y y Y Y Y Y
Z 32 Z Z Z z z z z z z z z
0 33 0 : 0 : 0 : 0 → 0 → 0 0
1 34 1 ; 1 ; 1 ; 1 ↑ 1 ↑ 1 1
2 35 2 ! 2 ! 2 ! 2 ← 2 ← 2 2
3 36 3 ? 3 ? 3 ? 3 ↓ 3 ↓ 3 3
4 37 4 × 4 × 4 × 4 4 4 4
5 40 5 . 5 . 5 . 5 5 5 5 5
6 41 6 ÷ 6 ÷ 6 6 6 6 6 6
7 42 7 ≡ 7 ≡ 7 ≡ 7 7 7 7 7 7
8 43 8 < 8 < 8 \ 8 8 8 8 8
9 44 9 > 9 > 9 \ 9 9 9 9 9
### Sample Plot/Character Access Table

<table>
<thead>
<tr>
<th>SET</th>
<th>PRU</th>
<th>PRL</th>
<th>IRU</th>
<th>IRL</th>
<th>KRU</th>
<th>KRL</th>
<th>PGU</th>
<th>PGL</th>
<th>IGU</th>
<th>IGL</th>
<th>KGU</th>
<th>KGL</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASE</td>
<td>0</td>
<td>100</td>
<td>200</td>
<td>0300</td>
<td>0400</td>
<td>0500</td>
<td>0600</td>
<td>0700</td>
<td>1000</td>
<td>1100</td>
<td>1200</td>
<td>1300</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KP NUM</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>≤</td>
<td>+</td>
<td>≤</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>−</td>
<td>≥</td>
<td>−</td>
<td>≥</td>
<td>−</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td></td>
<td>*</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/</td>
<td>/</td>
<td></td>
<td>/</td>
<td></td>
<td>/</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(</td>
<td>(</td>
<td></td>
<td>(</td>
<td></td>
<td>(</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>)</td>
<td>)</td>
<td></td>
<td>)</td>
<td></td>
<td>)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>$</td>
<td></td>
<td>$</td>
<td></td>
<td>$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>=</td>
<td>=</td>
<td></td>
<td>=</td>
<td></td>
<td>=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>%</td>
<td></td>
<td></td>
<td></td>
<td>%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SUBROUTINE PWRITY (X, Y, ID, N, ISIZE, ITHETA, ICNT)

DIMENSION OF ARGUMENTS
ID((N-1)/NCPW+1)
(NCPW = Number of Characters Per Word)

LATEST REVISION
June 1979

PURPOSE
PWRY is a character plotting routine. It has some features not found in PWRT, but is not as fancy as PWRX.

User's Note: PWRITY replaces PWRY on ULIB.

USAGE
CALL PWRITY(X, Y, ID, N, ISIZE, ITHETA, ICNT)

ARGUMENTS

On Input

X, Y Positioning coordinates for the characters to be drawn. These can be integers in the range 1 through 1024 or floating point numbers, in which case they are scaled according to the most recent SET cal. Also see ICNT.

ID Characters to be drawn.

N The number of characters in ID.

ISIZE Size of the character.

* If between 0 and 3, the factor is chosen as 1, 1.5, 2, or 3 times an 8 plotter address character width

* If greater than 3 it is the character width in plotter address units.

ITHETA Angle counter clockwise from X axis at which the characters are plotted.

* ANGLE = ITHETA degrees.
ICNT Centering Option.

\[\begin{align*}
-1 \ (X,Y) & \text{ is the center of the left edge of the first character.} \\
=0 \ (X,Y) & \text{ is the center of the entire string.} \\
=1 \ (X,Y) & \text{ is the center of the right edge of the last character.}
\end{align*}\]

On Output

All arguments are unchanged.

ENTRY POINTS

PWRY, PWRYSO, PWRYGT, PWRTY, PWRYBD

COMMON BLOCKS

PWRCOM

I/O

Plots characters.

PRECISION

Single

REQUIRED ULIB Routines

None

LANGUAGE

FORTRAN

HISTORY

Implemented for use in DASHCHAR. Made portable in January 1977 by Robert Ubelmesser for use on computer systems which support plotters with up to 15-bits resolution.

ALGORITHM

Digitizations of the characters are stored internally and adjusted according to \(X, Y, SIZE\) and ICNT, then plotted.

SPACE REQUIRED

724 (OCTAL) on CDC 7600 at NCAR

TIMING

Slower than PWRT, faster than PWRX.

PORTABILITY

This is the most portable version of PWRY. A smaller version exists for CDC 60 bit CPU's.

PLOTTING ROUTINES USED

FL2INT, PLOTIT, GETSI

(See System Plot Package documentation).

REQUIRED RESIDENT ROUTINES

COS, SIN, GETCHR
SAMPLE PLOT

DEMONSTRATION PLOT FOR ENTRY PWRITY OF PWRY
SIZE TEST
SIZE TEST
SIZE TEST
SIZE TEST

CENTR TEST
CENTR TEST
CENTR TEST

PWRITY - 3
SUBROUTINE PWRZI (X,Y,Z,ID,N,ISIZE,LINE,ITOP,ICNT)

DIMENSION OF ARGUMENTS
ID((N-1)/NC+1) (NC = number of characters)

LATEST REVISION
March 1980

PURPOSE
PWRZI is a character plotting routine for plotting characters in three-space when using ISOSRF. For a large class of possible positions, the hidden character problem is solved.

USAGE
CALL PWRZI (X,Y,Z,ID,N,ISIZE,LINE,ITOP,ICNT)

Use CALL PWRZI after calling ISOSRF and before calling FRAME.

ARGUMENTS
On Input

X,Y,Z
Positioning coordinates for the characters to be drawn. These are floating point numbers in the same three-space as used in ISOSRF.

ID
Characters to be drawn.

N
The number of characters in ID.

ISIZE
Size of the character(s)
- If between 0 and 3 the factor is 1., 1.5, 2., or 3. times a standard width equal to 1/128th of the screen width.
- If greater than 3 it is the character width in CRT units, these CRT units being adjusted to the current grid resolution.

LINE
The direction in which the characters are to be written.

| 1 = +X | -1 = -X |
| 2 = +Y | -2 = -Y |
| 3 = +Z | -3 = -Z |
ITOP
The direction from the center of the first character to
the top of the first character. Note that LINE cannot
equal ITOP even in absolute value.

ICNT
Centering option:

-1 (X,Y,Z) Is the center of the left edge of the
first character.

0 (X,Y,Z) Is the center of the entire string.

1 (X,Y,Z) Is the center of the right edge of the
last character.

On Output
All arguments are unchanged.

NOTE
The hidden character problem is solved correctly for
characters near (but not inside) the three-space object.

ENTRY POINTS
PWRZI, INITZI, PWRZOI, PWRZGI

COMMON BLOCKS
PWRZ1I

I/O
Plots character(s)

PRECISION
Single

REQUIRED ULIB
ISOSRF

ROUTINES

LANGUAGE
FORTRAN

HISTORY
Implemented for use with ISOSRF

ALGORITHM
Digitizations of the characters are stored internally and
adjusted according to X, Y, Z, ISIZE, LINE, ITOP; they
are then plotted, using the hidden-line algorithm of
ISOSRF, if necessary.

SPACE REQUIRED
34418 = 182510

TIMING
Slightly slower than PWRY

PORTABILITY
Portable

PLOTTING ROUTINES
(USED)

PLOTIT
The following is an example of PWRZI used in conjunction with ISOSRF.

DEMONSTRATION PLOT FOR PWRZI
SUBROUTINE PWRZS (X,Y,Z,ID,N,ISIZE,LIME,ITOP,ICNT)

DIMENSION OF ARGUMENTS
ID((N-1)/NC+1) (NC = number of characters per word)

LATEST REVISION
March 1980

PURPOSE
PWRZS is a character plotting routine for plotting characters in a three-space when using SRFACE. For a large class of possible positions, the hidden character problem is solved.

USAGE
CALL PWRZS (X,Y,Z,ID,N,ISIZE,LIME,ITOP,ICNT)

Use CALL PWRZS after calling SRFACE and before calling FRAME.

NOTE: SRFACE will have to be called using IFR=0 in the common block to suppress the FRAME call. See IFR in SRFACE internal parameters.

ARGUMENTS

On Input
X,Y,Z
Positioning coordinates for the characters to be drawn. These are floating point numbers in the same three-space as used in SRFACE.

ID
Characters to be drawn.

N
The number of characters in ID.

ISIZE
Size of the character(s)
- If between 0 and 3 the factor is 1., 1.5, 2., or 3. times a standard width equal to 1/128th of the screen width.
- If greater than 3 it is the character width in CRT units, these CRT units being adjusted to the current grid resolution.

LINE
The direction in which the characters are to be written.
The direction from the center of the first character to the top of the first character. Note that LINE cannot equal ITOP even in absolute value.

**ICNT**
Centering option

-1 (X,Y,Z) Is the center of the left edge of the first character.

0 (X,Y,Z) Is the center of the entire string.

1 (X,Y,Z) Is the center of the right edge of the last character.

**NOTE**
The hidden character problem is solved correctly for characters near (but not inside) the three-space object.

**ENTRY POINTS**
PWRZS, INITZS, PWRZOS, PWRZGS

**COMMON BLOCKS**
PWRZ1S

**I/O**
Plots character(s)

**PRECISION**
Single

**REQUIRED ULIB Routines**
SRFACE

**LANGUAGE**
FORTRAN

**HISTORY**
Implemented for use with SRFACE

**ALGORITHM**
Digitizations of the characters are stored internally and adjusted according to X, Y, Z, ISIZE, LINE, ITOP; they are then plotted, using the hidden-line algorithm of SRFACE, if necessary.

**SPACE REQUIRED**
34418 = 182510
TIMING Slightly slower than PWRY

PORTABILITY Portable

PLOTTING ROUTINES USED FRSTPT, VECTOR

REQUIRED RESIDENT ROUTINES COS, SIN

SAMPLE PLOT The following is an example of PWRZS used in conjunction with SRFACE.

DEMONSTRATION PLOT FOR PWRZS
SUBROUTINE PWRZT (X,Y,Z,ID,N,ISIZE,LINE,ITOP,ICNT)

DIMENSION OF ARGUMENTS
ID((N-1)/NC+1) (NC = number of characters)

LATEST REVISION
March 1980

PURPOSE
PWRZT is a character plotting routine for plotting characters in three-space when using THREED. For a large class of possible positions, the hidden character problem is solved.

USAGE
CALL PWRZT (X,Y,Z,ID,N,ISIZE,LINE,ITOP,ICNT)

Use CALL PWRZT after calling THREED and before calling FRAME.

ARGUMENTS

On Input

X,Y,Z
Positioning coordinates for the characters to be drawn. These are floating point numbers in the same three-space as used in THREED.

ID
Characters to be drawn.

N
The number of characters in ID.

ISIZE
Size of the character(s)

- If between 0 and 3 the factor is 1., 1.5, 2., or 3. times a standard width equal to 1/128th of the screen width.
- If greater than 3 it is the character width in CRT units, these CRT units being adjusted to the current grid resolution.

LINE
The direction in which the characters are to be written.
1 = +X  -1 = -X
2 = +Y  -2 = -Y
3 = +Z  -3 = -Z

ITOP
The direction from the center of the first character to the top of the first character. Note that LINE cannot equal ITOP even in absolute value.

ICNT
Centering option

-1 (X,Y,Z) Is the center of the left edge of the first character.
0 (X,Y,Z) Is the center of the entire string.
1 (X,Y,Z) Is the center of the right edge of the last character.

On Output All arguments are unchanged.

NOTE The hidden character problem is solved correctly for characters near (but not inside) the three-space object.

ENTRY POINTS PWRZT, INITZT, PWRZOT, PWRZGT

COMMON BLOCKS PWRZ1T

I/O Plots character(s)

PRECISION Single

REQUIRED ULIB Routines

LANGUAGE FORTRAN

HISTORY Implemented for use with THREED

ALGORITHM Digitizations of the characters are stored internally and adjusted according to X, Y, Z, ISIZE, LINE, ITOP; they are then plotted, using the hidden-line algorithm of THREED, if necessary.

SPACE REQUIRED 34418 = 182510
TIMING  Slightly slower than FWRY
PORTABILITY  Portable
PLOTTING Routines USED  FRSTPT, VECTOR
REQUIRED RESIDENT Routines  COS, SIN
SAMPLE PLOT  The following is an example of PWRZT used in conjunction with THREED.
SUBROUTINE SCROLL (IBUF,LBUF,CARDS,NCARDS,NYST,NYFIN,TST,TMV,TFIN,MV)

DIMENSION OF ARGUMENTS
IBUF(LBUF), CARDS(NWPC,NCARDS), where NWPC is the number of words per 80-column card image.

LATEST REVISION
September, 1980

PURPOSE
To produce scrolled movie titles with a minimum of effort. Scrolling is possible.

USAGE
If the following assumptions are met, use

CALL MTITLE (IBUF,LBUF,MV)

Assumptions:

- Each group of lines of text is contained on one frame (i.e., no scrolling).
- Vertical spacing of titles is done automatically.
- Titles are centered horizontally.
- There is a maximum of 60 characters per line of text including PWRITX modifiers.
- In production mode, blank frames are generated before and after each group of title frames.

If these assumptions are not met, use

CALL SCROLL (IBUF,LBUF,CARDS,NCARDS,NYST,NYFIN,TST,TMV,TFIN,MV)

ARGUMENTS
For MTITLE the argument list and data input are different from SCROLL and are described separately below.

On Input For MTITLE

IBUF
Integer array of length LBUF used as buffer for plotting instructions. Typically 5000 words are required.

LBUF
Dimension of IBUF

MV
Switch to indicate whether this is a practice run or the movie is being made.
0 Movie is being made.

# 0 Practice run
Practice runs output an outlined frame of titles with a legend indicating how many seconds the frame will be shown. The number of blank frames that will be output before and after the title sequence are also indicated. When the movie is being made this practice output is suppressed.

Data Cards
Each group of data cards results in a frame of titles which is repeated to give enough time for reading. There can be any number of groups. MTITLE keeps processing groups until an NCARD=0 is read. A group consists of the following:

- A header card from which NCARD, TIME, and SIZE are read under FORMAT (I5,2F5.1).

NCARD
Number of text cards that follow. IF NCARD=0, MTITLE will return to the calling routine.

TIME
Time in seconds this frame should be displayed.

SIZE
Relative size of characters. This multiplies the PWRITX character height.

- Text cards, each containing one line of the movie title. PWRITX modifiers may be used (see PWRITX documentation). Characters should not appear beyond column 50.

On Output For MTITLE Workspace buffer (IBUF) has been used. Other arguments are unchanged.

On Input For SCROLL IBUF Scratch area dimensioned LBUF used as a buffer for plotting instructions. Typically 50000 words are required.

LBUF Dimension of IBUF

CARDS
An integer array, dimensioned NWPC by NCARDS, filled prior to calling SCROLL (either by internal manipulations or by reading cards). Each card image starts at a word boundary and contains the following:
Columns 1-5
The MX coordinate of this line of text on the scroll, or an indication that this line of text is a continuation of the previous line. MX is the coordinate of the middle of the line if ICNTR is 1, and the coordinate of the left edge of the first character if ICNTR is 0. See columns 11-15 for ICNTR.

The scroll of text is 1024 CRT units wide and any number of CRT units high. MX=-9999 is the continuation card indicator. Any number of continuation cards may be used. Trailing blanks are omitted from each card, including those followed by a continuation card.

Columns 6-10
The MY coordinate of this line of text on the scroll (MY may be outside the 1-1024 CRT unit range). In the case of a continuation card, columns 6-20 are ignored.

Columns 11-15
ICNTR - the centering option:

= 0 Start the text at MX

= 1 Center the text about MX

= 2 End the text at MX

Columns 16-20
SIZE, the relative size of characters. This multiplies the PWRITX character height. Recommended range 1. - 2.5 (PWRITX modifiers can also be used to change sizes).

Columns 21-80
Text for this line, or for continuation of a line when MX=-9999. Note: Every line of text, except continuation lines, must start with a PWRITX modifier.

NCARDS
Second dimension of CARDS, i.e., the number of card images in CARDS.

NYST
The scroll coordinate that will be at the center of the screen when the text is first displayed (see diagram for clarification).

NYFIN
As NYST but for final position

SCROLL - 3
TST
Time in seconds that the scroll will be stationary at NYST. One second is recommended.

TMV
Time to move the scroll from NYST to NYFIN. This should be the time required to read the text aloud at slow to normal speed.

TFIN
Time that the scroll will be stationary at NYFIN. One second is recommended.

MV
Switch to indicate whether this is a practice run or the movie is being made.

= 0 Movie being made
≠ 0 Practice run

Practice runs output outlined representative frames from the scroll with a legend indicating the number of seconds the frame will be shown at the start or finish, or the number of seconds into the total moving time that a particular frame represents. When the movie is being made this practice output is suppressed.

On Output For SCROLL
Workspace buffer (IBUF) has been used. Card buffer (CARDS) has been shuffled.

NOTE
See the internal parameter list below for directions for the most common modifications. SCROLL contains a general purpose windowing package.

ENTRY POINTS
DOTEDG, GAP, GENCHR, GETFLT, GETINT, MOVCHR, MTITLE, SCPLTW, PWRITV, SCPWTW, SCROLL, WNDOUT. SCRLBD IS THE BLOCK-DATA ROUTINE FOR SCROLL. (see also the file PWRITX, which is required.)

COMMON BLOCKS
SCRLDT, PWRSV1, PWRC0, PWRC1, PWRC2, PSAV1, PSAV2, PUSER, PINT6 (all but the first of these occur in SCPWTW, which is a slightly modified version of PWRITX.)

I/O
MTITLE
Reads data from cards and generates graphical output

SCROLL
Is passed all its input through the argument list and generates graphic output.
PRECISION Single

REQUIRED ULIB Routines

PWRITX

LANGUAGE FORTRAN

HISTORY Improves and refines old routines AUTOTITLE and SCROLL at NCAR.

Re-written by Dave Kennison in April, 1980. Changed to use the new PWRITX instead of the old PWRX and made portable.

ALGORITHM SCROLL in effect moves the body of text up through the screen window outputting the frames required to generate a movie sequence of duration specified by the user.

At each frame SCROLL skips plotting lines of text that are completely outside of the screen window, plots lines of text entirely within the window with PWRITX, and lines of text that are partially in the window with SCPWTW, which is similar to PWRITX except that it allows the portions of characters that are only partially within the window to be drawn without distortion. NOTE: SCPWTW uses a generalized windowing capability provided by subroutine SCPLTW, which may be used separately in other plotting applications where windowing is desirable.

SPACE REQUIRED About 14000, not including the user's plot buffer or the System Plot Package.

TIMING Varies widely

PORTABILITY Portable

SYSTEM PLOT PACKAGE ROUTINES USED ENCODE, FLASH1, FLASH2, FLASH3, FL2INT, FRAME, GETCHR, GETSI, ISHIFT, IMACH, PERROR, PLOTIT, PWRIT, Q8QST4, SETCHR, SETI, ULIBER

REQUIRED RESIDENT ROUTINES SIN, COS, SQRT, LOG

SCROLL - 5
## INTERNAL PARAMETERS

In MTITLE

<table>
<thead>
<tr>
<th>NAME</th>
<th>DEFAULT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICU</td>
<td>5</td>
<td>Input-card Unit.</td>
</tr>
<tr>
<td>ICO</td>
<td>1</td>
<td>Centering Option. Reset to 0 to get left edges lined up at X-coordinate 64, to 2 to get right edges lined up at X-coordinate 960.</td>
</tr>
<tr>
<td>PCHSZ</td>
<td>32.</td>
<td>Height of PWRITX character size for a given font (currently PRU)</td>
</tr>
<tr>
<td>GAPSZ</td>
<td>40.</td>
<td>Value of interline spacing for the above font</td>
</tr>
<tr>
<td>T1</td>
<td>1.</td>
<td>Number of seconds worth of blank frames generated before any title frames are produced (at 24 frames/second)</td>
</tr>
<tr>
<td>T2</td>
<td>.5</td>
<td>Number of seconds worth of blank frames generated between sets of title frames and after the last set of title frames</td>
</tr>
</tbody>
</table>
In SCROLL

<table>
<thead>
<tr>
<th>NAME</th>
<th>DEFAULT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NXST</td>
<td>512</td>
<td>Analogous to NYST and NYFIN in argument list. Allows for limited scrolling in the X or horizontal direction. NXST and NXFIN are confined to be within X limits of the window, with the additional constraint that text MUST leave the window through the top rather than the sides.</td>
</tr>
<tr>
<td>NXFIN</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>ICRTJP</td>
<td>300</td>
<td>MY scroll coordinate spacing between practice output frames.</td>
</tr>
<tr>
<td>LIML(4)</td>
<td>1,32767, 1,32767</td>
<td>Contains the four values that define the window, stated on a hypothetical grid with 2**15 addressable positions in both directions. Set in the Block Data Routine SCRLBD.</td>
</tr>
</tbody>
</table>
Example
This code illustrates how MTITLE is used.

```
DIMENSION PBUF(5000)
MOVIE= 1
C FOR PRACTICE RUN
.
.
CALL MTITLE (PBUF,3000,MOVIE)
.
.
```

Data Cards

```
3 3 1.5
'PRU'A
'PRU'F'L'RAME
'PRU'O'L'F 'PRU'T'L'ITLES

6 5 1.25
'PRU'P'L'RODUCED BY
'PRU'T'L'HE 'U1'NATIONAL 'U1'CENTER FOR
'PRU1'ATMOSPHERIC 'U1'RESEARCH 'U1',
'PRU1'BOULDER 'U1', 'U1'COLORADO 'U1'80302
'PRU1'L'NDER SPONSORSHIP OF THE
'PRU1'N'L'ATIONAL 'PRU1'S'L'CIENCE 'PRU1'F'L'OUNDAITION
.
.
```
A Frame Of Titles
Produced by
The National Center for
Atmospheric Research,
Boulder, Colorado 80302
Under sponsorship of the
National Science Foundation
view in screen initially

scroll coordinate
NYST at center of
screen for TST secs.

scroll moves from
NYST to NYFIN in
TMV secs.

scroll coordinate
NYFIN at center of
screen for TFIN secs.

final view in screen
SUBROUTINE SRFACE (X,Y,Z,M,MX,NX,MY,S,STEREO)

DIMENSION OF ARGUMENTS
X(NX),Y(NY),Z(MX,NY),M(2,NX,NY),S(6)

LATEST REVISION
May 1980

PURPOSE
SRFACE draws a perspective picture of a function of two variables with hidden lines removed. The function is approximated by a two-dimensional array of heights.

USAGE
If the following assumptions are met, use

CALL EZSRFC (Z,M,N,ANGH,ANGV,WORK)

Assumptions:
The entire array is to be drawn,
The data is equally spaced (in the X-Y plane), No STEREO pairs,
Scaling is chosen internally.

If these assumptions are not met, use

CALL SRFACE (X,Y,Z,M,MX,NX,MY,S,STEREO)

ARGUMENTS

On Input for EZSRFC

Z
The M by N array to be drawn

M
The first dimension of Z

N
The second dimension of Z

ANGH
Angle in degrees in the X-Y plane to the line of sight (counter-clock wise from the plus-X axis)

ANGV
Angle in degrees from the X-Y plane to the line of sight (positive angles are above the middle Z, negative below).

WORK
A scratch storage dimensioned at least
2 x M x N + M + N
On Output for Z, M, N, ANGH, ANGV are unchanged. WORK has been written EZSRFC in.

On Input for X SRFACE A linear array NX long containing the X coordinates of the points in the surface approximation. See NOTE, below.

Y The linear array NY long containing the Y coordinates of the points in the surface approximation. See NOTE, below.

Z An array MX by NY containing the surface to be drawn in NX by NY cells. \( Z(I,J) = F(X(I), Y(J)) \). See NOTE, below.

M Scratch array at least 2 x NX x NY words long

MX First dimension of Z

NX Number of points in the X direction in Z. When plotting an entire array, MX=NX. See Appendix C of the Graphics Utilities Used at NCAR for an explanation of using this argument list to process any part of an array.

NY Number of points in the Y direction in Z

S S defines the line of sight. The viewer's eye is at \((S(1), S(2), S(3))\) and the point looked at is at \((S(4), S(5), S(6))\). The eye should be outside the block with opposite corners \((X(1), Y(1), ZMIN)\) and \((X(NX), Y(NY), ZMAX)\) and the point looked at should be inside it. For a nice perspective effect, the distance between the eye and the point looked at should be 5 to 10 times the size of the block. See NOTE, below.

STEREO Flag to indicate if STEREO pairs are to be drawn. 0.0 means no STEREO pair (one picture). Non-zero means put out two pictures. The value of STEREO is the relative angle between the eyes. A value of 1.0 produces standard separation. Negative STEREO reverses the left and right figures.
On Output for SRFACE

X, Y, Z, MX, NX, NY, S, STEREO are unchanged. M has been written in.

NOTE

- The range of Z compared with the range of X and Y determines the shape of the picture. They are assumed to be in the same units and not wildly different in magnitude. S is assumed to be in the same units as X, Y, and Z.

- Picture size can be made relative to distance. See comments in SETR.

- TRN32S can be used to translate from 3 space to 2 space. See comments in TRN32S.

- Data with extreme discontinuities may cause visibility errors. If this problem occurs, use a distant eye position away from the +Z axis.

ENTRY POINTS

SRFACE, EZSRFC, SETR, DRAWS, TRN32S, CLSET, CTCELL, SRFABD

SPECIAL CONDITIONS

Three-dimensional perspective character labelling of SRFACE plots is possible by using the ULIB routine PWRZS. For a description of the usage, see the PWRZS write-up.

COMMON BLOCKS

SRFBLK (4072 length)
SRFIP1 (178 length)
PWRZ1S (128 length)

I/O

Plots

PRECISION

Single

REQUIRED ULIB Routines

None

LANGUAGE

FORTRAN

HISTORY


ALGORITHM

The data is processed from the near side of the surface toward the far side. Visibility information is stored (see Reference). Highest so far is visible from above.

REFERENCE

SPACE REQUIRED 11607, including SRFBLK, not including the System Plot Package

ACCURACY If the ends of a line segment are visible, the middle is assumed visible.

TIMING Proportional to NX*NY. 20 by 20 takes 0.5 seconds on the CDC N7600.

PORTABILITY Portable

PLOTTING ROUTINES USED FRAME, FRSTPT, VECTOR

REQUIRED STANDARD Routines SIN, COS, SQRT, ATAN2, ALOG10

INTERNAL PARAMETERS These parameters are set in the block data routine SRFABD.

<table>
<thead>
<tr>
<th>NAME</th>
<th>DEFAULT</th>
<th>FUNCTION</th>
</tr>
</thead>
</table>
| IFR  | 1       | -1 Call FRAME first  
0 Do not call FRAME  
+1 Call FRAME when done |
| ISTP | 0       | STEREO type if STEREO non-zero  
-1 Alternating frames, slightly offset (for movies. IROTS=0)  
0 Blank frame between (for STEREO slide. IROTS=1)  
+1 Both on same frame. (Left picture to left side. IROTS=0) |
| IROTS| 0       | 0 +Z in vertical plotting direction (CINE mode)  
+1 +Z in horizontal plotting direction (COMIC mode) |
| IDRX | 1       | +1 Draw lines of constant X  
0 Do not |
| IDRY | 1       | +1 Draw lines of constant Y  
0 Do Not. |
| IDRZ | 0       | +1 Draw lines of constant Z (contour lines).  
0 Do Not. |
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| IUPPER     | 0     | +1 Draw upper side of surface.  
               0 Draw both sides.  
               -1 Draw lower side. |
| ISKIRT     | 0     | +1 Draw a skirt around the surface.  
               BOTTOM = HSKIRT.  
               0 Do Not. |
| NCLA       | 6     | Approximate number of levels of constant Z that are drawn if levels are not specified. 40 levels MAXIMUM. |
| THETA      | .02   | Angle in Radians between eyes for STEREO pairs. |
| HSKIRT     | 0.    | Height of SKIRT (IF ISKIRT = 1). |
| CHI        | 0.    | Highest level of constant Z. |
| CLO        | 0.    | Lowest level of constant Z. |
| CINC       | 0.    | Increment between levels.  
               [If CHI, CLO, or CINC is Zero, a nice value is generated automatically.] |
| IOFFP      | 0     | Flag to control use of special value feature. Do not have BOTH IOFFP=1 and ISKIRT=1.  
               0 = Feature not in use  
               +1= Feature in use. No lines drawn to data points in Z that are equal to SPVAL. |
| SPVAL      | 0.    | Special value used to mark unknown data when IOFFP=1. |
A sample picture follows:

DEMONSTRATION PLOT FOR EZSRFC ENTRY OF SRFACE
SAMPLE PLOT

DEMONSTRATION PLOT FOR SRFACE ENTRY OF SRFACE
SUBROUTINE STRMLN (U,V,WORK,IMAX,IPTSX,JPTSY,NSET,IER)

DIMENSION OF ARGUMENTS
U(IMAX,JPTSY),V(IMAX,JPTSY),WORK(2*IMAX*JPTSY)

WRITTEN October 1979

PURPOSE STRMLN draws a streamline representation of the flow field. The representation is independent of the flow speed.

USAGE If the following assumptions are met, use
CALL EZSTRM (U,V,WORK,IMAX,JMAX)

- The whole array is to be processed
- The arrays are dimensioned U(IMAX,JMAX), V(IMAX,JMAX), and WORK (2*IMAX*JMAX).
- SET and PERIM are to be called by STRMLN

If these assumptions are not met, use
CALL STRMLN (U,V,WORK,IMAX,IPTSX,JPTSY,NSET,IER)

The user must call FRAME in the calling routine.

Note: The user may change various internal parameters via common blocks (see below).

ARGUMENTS

On Input U,V
The two dimensional arrays containing the velocity fields to be plotted. Note: If the U and V components are, for example, defined in Cartesian coordinates and the user wishes to plot them on, say, a polar stereographic projection, then an appropriate transformation must be made to the U and V components via the functions FU and FV (located in subroutine DRWSTR). See Appendix B of the Graphics Utilities Used at NCAR for more information on coordinate transformation.

WORK User-provided work array. The dimension of this array is 2*IMAX*JPTSY. Caution: This routine does not check the size of the WORK array.
IMAX
The first dimension of U and V in the calling program (x-direction).

IPTSX
The number of points to be plotted in the first subscript direction (x-direction).

JPTSY
The number of points to be plotted in the second subscript direction (y-direction). Note: See Appendix 1 this documentation for more information on processing parts of arrays.

NSET
Flag to control scaling

>0 STRMLN assumes that SET has been called by the user in such a way as to properly scale the plotting instructions generated by STRMLN. PERIM is not called.

=0 STRMLN calls SET to properly scale the plotting instructions to the standard configuration. PERIM is called to draw the border.

<0 STRMLN calls SET in such a way as to place the streamlines within the limits of the user's last SET call. PERIM is not called.

On Output Only the IER argument may be changed. All other arguments are unchanged.

IER

=0 when no errors are detected.

=1 when the routine is called with ICYC=0 and the data are not cyclic. (ICYC is an internal parameter and is described below.) In this case the routine will draw the streamlines with the non-cyclic interpolation formulas.

ENTRY POINTS STRMLN, DRWSTR, EZSTRM, GNEWPT, CHKCYC

SPACE REQUIRED Approximately 5650 on the CDC 7600 not including the required resident routines or the system plot package. This program may be made smaller by deleting portions of GNEWPT or making the circular list arrays in DRWSTR smaller.
COMMON BLOCKS

STRO1 (20\(^3\))

Used for communication between various streamline routines.

STRO2 (6\(^3\))

This block contains plotter information and may be used by the user to alter the default size of the STRMLN plot.

STRO3 (14\(^3\))

This block is probably the most useful to the user. Many internal parameters, discussed below, may be altered by the user by inserting this block in the user's code and changing the default values.

STRO4 (2736\(^3\))

Primarily internal to DWRSTR. This block contains the circular lists used to prevent streamline crossover.

I/O

Draws streamlines.

PORTABILITY

Portable. Note: All variable names are unique to five characters. This makes it compatible with Data General FORTRAN.

PRECISION

Single

TIMING

Highly variable. It depends on the complexity of the flow field and the parameters DISPL, DISPC, CSTOP, INITA, INITB, ITERC, and IGFLG. (See below for a discussion of these parameters.) If all values are default then a simple linear flow field for a 40 x 40 grid will take slightly less than one second on the CDC 7600 and about 0.4 seconds on the CRAY-1. A fairly complex flow field will take about 4 seconds on the CDC 7600 and about 1.5 seconds on the CRAY-1. As a general rule, this routine runs about 2.5 times faster on the CRAY-1. No attempt was made to vectorize since many IF checks must be made at virtually every step.

Timing tests indicate that STRMLN runs about 20% faster than the previous ULIB streamline package (STRML2).

REQUIRED ULIB ROUTINES

None

LANGUAGE

FORTRAN

HISTORY

The techniques utilized here are described in an article by Thomas Whittaker (University of Wisconsin) which appeared in the "Notes and Correspondence" section of Monthly Weather Review, June 1977.
Wind components are normalized to the value of DISPL. Then the least significant two bits of the work array are utilized as flags for each grid box. Flag 1 indicates whether any streamline has previously passed through this box. Flag 2 indicates whether a directional arrow has already appeared in a box. Judicious use of these flags prevents overcrowding of streamlines and directional arrows. Experience indicates that a final pleasing picture is produced when streamlines are initiated in the center of a grid box. The streamlines are drawn in one direction then in the opposite direction.

**Plotting Routines Used**

STRMLN: GETSET, SET, PERIM

DRWSTR: FRSTPT, VECTOR

**Required Resident Routines**

ATAN2, COS, SIN, IFIX, SQRT, MOD, MINO, AND, OR, COMPL, AINT

Note to Data General ECLIPSE users: The AND/OR statement functions do not have a counterpart on the DG; thus, specially written routines are used in their place. These routines are ANDF and ORF; the COMPL is replaced by the DG routine NOT (see Lin Thiel for information).
### Internal Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRTX</td>
<td>1024</td>
<td>Maximum number of plotter points in X and Y</td>
</tr>
<tr>
<td>CRTY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXT</td>
<td>0.25</td>
<td>Lengths of the sides of the plot are proportional to IPTSX and JPTSY except in the case when MIN(IPTSX,JPTSY) / MAX(IPTSX,JPTSY) (\leq) EXT, in which case a square graph is plotted.</td>
</tr>
<tr>
<td>NAND</td>
<td>3</td>
<td>The complement of NAND is used to mask out the low order bits as flags.</td>
</tr>
<tr>
<td>SIDE</td>
<td>0.90</td>
<td>Length of longer edge of plot (see also EXT).</td>
</tr>
</tbody>
</table>
| XLT  | 0.05          | Left hand edge of the plot (XLT+SIDE \(\leq\) 1.)  
0.0 = left edge of frame  
1.0 = right edge of frame |
| YBT  | 0.05          | Bottom edge of the plot (YBT+SIDE \(\leq\) 1.)  
0.0 = bottom edge  
1.0 = top edge |
| INITA| 2             | Used to pre-condition grid boxes to be eligible to start a streamline; e.g., a value of 4 means that every fourth grid box is eligible; a value of 2 means that every other grid box is eligible. |
| INITB| 2             | Used to pre-condition grid boxes to be eligible for directional arrows. Note: If the user changes the default values of INITA and/or INITB, it should be done such that MOD(INITA,INITB) = 0. If the user has a reasonably dense grid, try INITA = 4, INITB = 2. This will cut down CPU time. |
**INTERNAL PARAMETERS (CON’T)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AROWL</td>
<td>0.33</td>
<td>Length of directional arrows (0.33, for example, means that each directional arrow will take up a third of a grid box).</td>
</tr>
<tr>
<td>ITERP</td>
<td>35</td>
<td>Every ITERP iterations, the streamline progress is checked.</td>
</tr>
<tr>
<td>ITERC</td>
<td>-99</td>
<td>The default value of this parameter is such that it has no effect on the code. When set to some positive value, the program will check for streamline crossover every ITERC iterations. The routine currently does this every time it enters a new grid box. Caution: When this parameter is activated, CPU time will increase.</td>
</tr>
</tbody>
</table>

**STRMLN - 6**
<table>
<thead>
<tr>
<th>Name</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGFLG</td>
<td>0</td>
<td>A value of zero means that the 16 point Bessel interpolation formula will be utilized where possible. When near the grid edges, quadratic and bi-linear interpolation will be used. This mixing of interpolation schemes can cause slight raggedness near the grid edges. This, however, does not occur very often. If IGFLG.NE.0, then the bi-linear interpolation formula (only) is to be used. This will result in slightly faster plot times. However, in general, the plots will be less pleasing.</td>
</tr>
<tr>
<td>IMSG</td>
<td>0</td>
<td>If zero, then no missing U and V components are present. If not equal to 0, then STRMLN will utilize the bi-linear interpolation scheme and if any of the points are missing the streamline will be terminated.</td>
</tr>
<tr>
<td>UVMSG</td>
<td>1.E+36</td>
<td>Value assigned to a missing point.</td>
</tr>
<tr>
<td>ICYC</td>
<td>0</td>
<td>Zero means the data are non-cyclic in the x- direction. If not equal to 0, then the cyclic interpolation formulas will be used. Note: Even if the data are cyclic in x, leaving ICYC=0 will do no harm.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>DISPL</td>
<td>0.33</td>
<td>The wind speed is normalized to this value. (See below for a discussion of DISPL, DISPC, and CSTOP parameters.)</td>
</tr>
<tr>
<td>DISPC</td>
<td>0.67</td>
<td>The critical displacement. If after ITERP iterations the streamline has not moved this distance, then the streamline will be terminated. (See below for a discussion of DISPL, DISPC, and CSTOP parameters.)</td>
</tr>
<tr>
<td>CSTOP</td>
<td>0.50</td>
<td>This parameter controls the spacing between streamlines. The checking is done when a new grid box is entered. (See below for a discussion of DISPL, DISPC, and CSTOP parameters.)</td>
</tr>
<tr>
<td>UXSML</td>
<td>1.E-50</td>
<td>See subroutine DRWSTR for a discussion of this parameter. This may need to be changed depending upon the target computer. Basically it should be some very small number.</td>
</tr>
<tr>
<td>NCHK</td>
<td>750</td>
<td>This parameter is located in DRWSTR. It specifies the length of the circular lists used for checking for STRMLN crossovers. For most plots this number may be reduced to 500 or less and the plots will not be altered.</td>
</tr>
</tbody>
</table>

**Note:** With the exception of UXSML and NCHK, any or all of the above parameters may be changed by utilizing common blocks STRO2 and/or STRO3.

**DISPL, DISPC, CSTOP:** Assume a value of 0.33 for DISPL. This means that it will take three steps to move across one grid box if the flow is all in the x-direction. If the flow is zonal, then a larger value of DISPL is in order. If, however, the flow is highly turbulent then a smaller value is in order. Note: The smaller the value of DISPL, the more the computer time.

A value of two to four times DISPL is a reasonable value for DISPC. DISPC should always be greater than DISPL.
A value of 0.33 for CSTOP would mean that a maximum of three streamlines will be drawn per grid box. This maximum will normally only occur in areas of singular points.

**IMPORTANT:** Any or all of the above parameters may be changed by utilizing common blocks STR02 and/or STR03.
SAMPLE PLOT

DEMONSTRATION PLOT FOR ROUTINE STRMLN

STRMLN - 10
SUBROUTINE SUPFMAP (JPROJ, POLAT, POLONG, BOT, PL1, PL2, PL3, PL4, JLTS, JGRID, USOUT, IDOT, IER)

DIMENSION OF
PL1(2), PL2(2), PL3(2), PL4(2)

LATEST REVISION January 1978

PURPOSE
To plot continental and/or U.S. state outlines according to one of nine projections. The origin and orientation of the projection are selected by the user. Points on the earth defined by latitude and longitude are transformed to points in the U,V plane, the plane of projection. The U and V axes are respectively parallel to the X and Y axes of the plotter. A rectangular frame parallel to the U and V axes is chosen and only material within the frame is plotted.

User's Note: This documentation is for SUPMAP on ULIB. This version runs only on NCAR’S CDC7600 mainframe. It will not run on the CRAY-1. A version of SUPMAP which does run on the CRAY-1 resides on CRAYLIB under the name of EZMAP. The CRAYLIB EZMAP should not be confused with the EZMAP entry in this documentation. For details on EZMAP for use on the CRAY-1, please refer to the EZMAP documentation in this manual.

USAGE
If the following assumptions are met, use

CALL EZMAP (JPROJ, POLAT, POLONG)

Assumptions:

The Lambert conformal conic with two standard parallels, (IABS(JPROJ) = 3), is not required.

The maximum useful area for the projection is plotted.

The origin is at the point with latitude POLAT and longitude POLONG.

At the origin "north" is parallel to the V axis.

Continental outlines are plotted as continuous lines.

Grids are plotted at 10 degree intervals.

The SUPMAP call is neither printed nor written beneath the map.

SUPMAP - 1
If these assumptions are not satisfied, use

CALL SUPMAP (JPROJ,POLAT,POLONG,ROT,PL1,PL2,PL3,PL4,
JLTS,JGRID,IUSOUT,IDOT,IER)

On Input for EZMAP
JPROJ,POLAT,POLONG
See below under "On Input for SUPMAP"

On Output for EZMAP
All arguments are unchanged.

On Input for SUPMAP
JPROJ IABS(JPROJ) defines the projection type according to the following code:

1  Stereographic
2  Orthographic
3  Lambert conformal conic with two standard parallels
4  Lambert equal area
5  Gnomonic
6  Azimuthal equidistant
7  Dummy—this code is not used
8  Cylindrical equidistant
9  Mercator
10  Mollweide type

If JPROJ is negative, the continental outlines are omitted.

POLAT,POLONG,ROT If (IABS(JPROJ)≠ 3)

- POLAT and POLONG define in degrees the latitude and longitude of the point on the globe which is to transform to the origin of the U,V plane, where:

\[-90 \leq \text{POLAT} \leq 90\]
\[-180 \leq \text{POLONG} \leq 180\]

Degrees of latitude north of the equator and degrees of longitude east of the Greenwich meridian are positive. If the origin is at the north pole, "north" is considered to be in the direction of (POLONG+180.). If the origin is at the south pole, "north" is in the direction of POLONG.

- ROT is the angle between the V axis and "north" at the origin. It is measured in degrees and is taken to be positive if the angular movement from "north" to the V axis is counter-clockwise. For the
cylindrical projections (8,9,10), the axis of the projection is parallel to the V axis.

If (IABS(JPROJ)=3) (Lambert conformal conic with two standard parallels)

- POLONG = central meridian of projection in degrees
- POLAT, ROT are the two standard parallels in degrees

JLTS, PL1, PL2, PL3, PL4, IABS(JLTS) can take the values 1 through 5 and specifies one of five options on the way in which the limits of the rectangular map are defined by the parameters PL1, PL2, PL3, PL4

IABS(JLTS) = 1
The maximum useful area produced by the projection is plotted. PL1, PL2, PL3, PL4 are not used and may be set to zero.

IABS(JLTS) = 2
In this case (PL1, PL2) and (PL3, PL4) are the latitudes and longitudes in degrees of two points which are to be at opposite corners of the map. Care must be taken when using cylindrical projections and this option.

IABS(JLTS) = 3
The minimum and maximum values of U and V are specified by PL1 through PL4. PL1 = UMIN, PL2 = UMAX, PL3 = VMIN, PL4 = VMAX. Knowledge of the transformation equations is necessary for this option to be used (see below).

IABS(JLTS) = 4
Here PL1 = AUMIN, PL2 = AUMAX, PL3 = AVMIN, PL4 = AVMAX, where

- AUMIN = angular distance from origin to left frame of map
- AUMAX = angular distance from origin to right frame of map
- AVMIN = angular distance from origin to lower frame
- AVMAX = angular distance from origin to upper frame

AUMIN, AUMAX, AVMIN, AVMAX must be positive and the origin must be within the rectangular limits of the map. This option is useful for polar projections. It is not appropriate for the Lambert conformal with two standard parallels. An error message is printed if an attempt is made to use JLTS = 4 when JPROJ = 3, (see below).

IABS(JLTS) = 5
PL1 through PL4 are two-element arrays giving the latitudes and longitudes of four points which are to be on the four sides of the rectangular frame. PL1(1), PL1(2) are respectively the latitude and longitude of a point on the left frame. Similarly PL2 lies on the right frame,
PL3 lies on the lower frame and PL4 lies on the upper frame. Note that in the calling program PL1 through PL4 will be dimensioned:

```
DIMENSION PL1(2),PL2(2),PL3(2),PL4(2)
```

If JLTS is positive, the SUPMAP call is written below the map. This is omitted if JLTS is negative.

**JGRID** IABS (JGRID) gives in degrees the interval at which lines of latitude and longitude are to be plotted. A value in the range 1 through 10 will usually be appropriate but higher values are acceptable. If JGRID

- `< 0` The border around the map is omitted
- `= 0` No grid lines are plotted
- `= -0` Both grid and border are omitted

**IUSOUT** IABS(IUSOUT)

- `= 1` U.S. state outlines are plotted
- `= 0` U.S. state outlines are not plotted

Note that if U.S. state outlines are required, it will be usual to suppress the continental outlines by making JPROJ negative.

If IUSOUT is positive, the SUPMAP call and values of UMIN, UMAX, VMIN, VMAX are printed as an aid to debugging. This is omitted if IUSOUT = -0 or -1.

**IDOT**

- `= 0` For continuous outlines
- `= 1` For dotted outlines

On Output for SUPMAP

All arguments except IER are unchanged.

**IER** Error flag with the following meaning. If IER

- `= 0` Map successfully plotted
- `= 33` Attempt to use non-existent projection
- `= 34` Map limits inappropriate
- `= 35` Angular limits too great
- `= 36` Map has zero area
- `= 37-40` Failures in ULIB data access.

If IER

- `= 37` EOF on ULIB file
- `= 39` Unsuccessful ULIB read
- `= 40` No check on last ULIB operation
ENTRY POINTS
EZMAP, MAPLOT, SUPCON, SUPCONQ, SUPFST, SUPMAP, SUPTRP,
SUPVEC, SUPVECQ, VECPLT

MAPLOT
Actually draws the map

SUPCON
Once the transformation has been set up by an initial
call to SUPMAP, the subroutine SUPCON may be called to
transform a point, (latitude, longitude) to the
corresponding point, (U,V) on the plane. Contours may
thus be readily drawn against the map background (see
SUPRST and SUPVEC below).

CALL SUPCON (RLAT,RLON,U,V)

On Input:
RLAT,RLON are the latitude and longitude of a point to
be transformed to the U,V plane. -90. \leq RLAT \leq 90. -
180 \leq RLON \leq 180.

On Output:
RLAT,RLON are unchanged. U,V are the transformed coor-
dinates of the point (RLAT,RLON).

SUPCONQ
Actually performs the above mentioned transformation

SUPFST,SUPVEC
To facilitate drawing lines on the map these routines
which act like the plotting routines FRSTPT and VECTOR
are included. They are subject to the same restrictions
as SUPCON above.

CALL SUPFST (RLAT,RLON)

CALL SUPVEC (RLAT,RLON)

SUPVECQ
This routine decides what lines are to be drawn and
where.

SUPTRP
Performs interpolation to the edges of the frame

VECPLT
Called by SUPVECQ to draw (dot) lines on the plotter

COMMON BLOCKS
SUPMP1 of length 48,10 = 608
SUPMP2 of length 1025,10 = 20018

(Occurs only in SUPMAP and MAPLOT. Blank common may be
used.)

I/O
Map plotted; outline data is read from ULIB. SUPMAP call
printed (possibly).

SUPMAP - 5
**PRECISION** Single

**REQUIRED ULIB ROUTINES** ULIBER (on system).

**LANGUAGE** FORTRAN


**ALGORITHM** The latitudes and longitudes of successive outline points are transformed to coordinates in the plane of projection and joined by a vector.

**REFERENCES**

**SPACE REQUIRED** Less than 50000 (not including plotting routines and common blocks).

**ACCURACY** The definition of the map produced is limited by two factors:

- The outline data has a resolution of 1°.
- The resolution of the dd80 is limited to 1024 units in the X and Y directions.

**TIMING** Usually less than one second per map depending upon projection, origin, and orientation. The cylindrical equidistant projection with POLAT = ROT = 0.0 is particularly fast.

**PORTABILITY** This package has several nonstandard features for optimization. A 1966 American National Standards Institute standard FORTRAN version is available from the Computing Facility. The text of the portable version is available on PLIB and may be accessed by the following:

*FORTRAN,S=PLIB,P=43410009,N=SUPMAPPORT,FL,FH*
PLOTTING ROUTINES

USED
PWRT, FRSTPT, VECTOR, POINT, DASHLN, PERIM, SET

REQUIRED RESIDENT ROUTINES
ATAN, TAN, SIN, COS, ALOG, SQRT, ATAN, ATAN2, ACOS

More History
R.L. Parker of UCSD wrote the original SUPERMAP system. This was adapted for use on the NCAR system by Lee (1968). The present SUPMAP is an improved version. The information for plotting continental and state outlines is due to Hershey (1963). The nine projections incorporated in the routine are described briefly below. For more detail refer to Steers (1962).

Mathematical Method
On ULIB there are three data files containing: the continental outlines at a resolution of about 1 degree; the United States geo-political outlines at a resolution of about 1 degree; and the disjoint union of the two other files. The latitude and longitude of successive points on the outline are transformed to coordinates in the U,V plane and joined by a vector. To improve definition, points closer than 4 plotter address units are discarded. The parameter, IDOT, enables a dotted outline to be selected. In this case, points are linearly inserted if successive data points are separated by more than 4 plotter address units. A similar procedure is used to plot the grid lines joining successive points by a dotted vector.

The subroutine, SUPMAP, calculates the limits of the rectangular map before calling subroutine MAPLOT to plot the continental outlines, grid lines, and perimeter. Both SUPMAP and MAPLOT call subroutine SUPCONQ to transform a point on the globe to a point in the U,V plane.
Azimuthal Projections (IABS(JPROJ) = 1,2,4,5,6)

Projections Used

The U,V plane is tangential to the globe at the point phi(POLAT, POLONG), which transforms to the origin, phi'(0,0), of the U,V plane (see Figure 1). ROT is the angle between the V-axis and north at phi'.

Let P be a point on the globe at an angular distance, A, from the point phi(POLAT, POLONG). Let B be the angle between the great circle phiP and the meridian at phi. Let P transform to P' in the U,V plane (see Figure 2). The program calculates SINA = SIN(A), COSA = COS(A), SINB = SIN(B), COSB = COS(B) in terms of the latitude and longitudes of phi and P. Then, if the distance phi'P' is R and SINR = SIN(ROT), COSR = COS(ROT), point P' has coordinates

\[ U = R \cdot \sin(B + ROT) = R \cdot (\sin B \cdot \cos R + \cos B \cdot \sin R) \]
\[ V = R \cdot \cos(B + ROT) = R \cdot (\cos B \cdot \cos R - \sin B \cdot \sin R) \]

It remains to define R in terms of COSA and SINA.
Description of Projections Used (Con't)

Stereographic Projection (1)

\[ R = \tan\left(\frac{A}{2}\right) = \frac{1 - \cos A}{\sin A} \]

as \( A \) approaches 180°, and \( R \) approaches infinity.

Thus the entire surface of the globe transforms to the entire \( U,V \) plane. In practice distortion becomes great beyond \( R = 2 \), or \( A \) approximately equal to 180 degrees.

Orthographic Projection (2)

\[ R = \sin A \]

This projection plots a hemisphere within radius \( R = 1 \).

The maximum possible value of \( A = 90 \) degrees.

Lambert Equal Area Projection (4)

\[ R \] is calculated by the FORTRAN statement

\[ R = 2.0 \times \sin A / \sqrt{2.0 \times (1.0 - \cos A)} \]

As \( A \) approaches 180 degrees, \( R \) approaches infinity, and the entire surface of the globe is plotted within a radius \( R = 2 \). The maximum value of \( A = 180 \) degrees.
In this projection the area between two circles, center phi', is proportional to the corresponding area on the globe.

**Gnomonic Projection** (5)

\[ R = \frac{\sin A}{\cos A} \quad \text{As } A \text{ approaches } 90^\circ, \text{ } R \text{ approaches infinity.} \]

A hemisphere is plotted over the entire U,V plane. In practice, distortion becomes great beyond \( R = 2 \), or \( A \) approximately equal to 65 degrees.

**Azimuthal Equidistance Projection** (6)

\[ R = A \text{ (in radians)} \]

\[ = \frac{\cos A}{\cos A} \quad \text{As } A \to 180 \text{ approaches, } R \text{ approaches } \pi. \]

The entire globe surface is plotted within a radius \( R = \pi \).

**Cylindrical Projections** (IBS(JPROJ) = 8,9,10)

The U,V plane must be imagined to be wrapped around the globe to form a cylinder, the U-axis touching the globe on some great circle (see Figure 3). The axis of the projection is perpendicular to this great circle and parallel to the V-axis. The point phi'(POLAT, POLONG), which transforms to the origin, phi"(0,0), of the projection, lies on the great circle. The limits of the U-axis are defined by a cut in the cylinder parallel to its axis and diametrically opposite to phi. The pole of the projection, Q, is the point 90 degrees from the great circle in the direction of +V. ROT is the angle between the V-axis and north at phi. These points and N, the north pole, are shown in Figure 3. Points on the surface of the globe are transformed to points on the U,V cylinder by the rule appropriate to the projection.

The latitude and longitude of Q are calculated in terms of the latitude and longitude of phi and the angle ROT. The angle ROT1 (see Figure 3) is also computed.

In Figure 3, P is some general point on the surface of the globe. A is the angular distance of P from Q. B is the angle between the great circles QN and QP. The quantities

\[ \sin A = \sin(A), \quad \cos A = \cos(A) \]
\[ \sin B = \sin(B), \quad \cos B = \cos(B) \]
\[ \sin \theta = \sin(\theta), \quad \cos \theta = \cos(\theta) \]

SUPMAP - 10
are computed.

Let $P'(U,V)$ be the point on the $U,V$ plane corresponding to the point $P$ on the surface of the globe. Then $U$ is proportional to the angle Alpha (see Figure 3). The value of $V$ depends upon the type of projection. The coordinates of $P'$ are given below.
Cylindrical Equidistant Projection (8)

\[ U = \text{Alpha (in degrees)} = \frac{\text{ATAN2(SIN(B+\text{ROT 1}), -COS(B+\text{ROT 1}}))}{F} = \text{ATAN2(SINB*COSR+COSB*SINR,SINB*SINR-COSB*COSR)/F} \]

Here division by \( F \) converts radians to degrees.

\[ V = 90.-\text{A (in degrees)} = 90.-\text{ACOS(COSA)/F} \]

The entire surface of the globe is transformed to a rectangle in the \( U,V \) plane.

\[-180. < U < 180. \]
\[-90. < V < 90. \]

Mercator Projection with Arbitrary Pole (9)

\[ U = \text{Alpha (in radians)} = \text{ATAN2(SINB*COSR+COSB*SINR,SINB*SINR-COSB*COSR)} \]

\[ V = \text{ALOG(COT(A/2))} = \text{ALOG(1.+COSA)/SINA)} \]

The entire surface of the globe is transformed to an infinite rectangle in the \( U,V \) plane. As \( \text{Alpha} \) approaches 0, \( V \) approaches infinity; as \( \text{Alpha} \) approaches 180 degrees, \( V \) approaches negative infinity. When \( \text{Alpha} \) equals 180 degrees, \( U=\pi \); when \( \text{Alpha} \) equals negative 180 degrees, \( U=-\pi \).

Hence

negative infinity \( \leq V \leq \) positive infinity

negative pi \( \leq U \leq \) positive pi

In practice distortion becomes great for \( \text{Alpha} < 5 \) degrees or > 175 degrees.

Mollweide-Type Projection (10)

The projection used is not a true Mollweide. The coordinates of \( P' \) are given by

\[ V = \text{COSA} \]

\( U \) is given by the two FORTRAN statements

\[ U = \text{ATAN2(SINB*COSR+COSB*SINR,SINB*SINR-COSB*COSR)*SF} \]

SUPMAP - 12
where \( SF = \frac{2}{\pi} \)

\[ U = U \times \sqrt{1 - V^2} \]

The entire surface of the globe transforms to an ellipse in the \( U,V \) plane. The major and minor axes of the ellipse are along the \( U \) and \( V \) axes respectively.

\[-2. \leq U \leq 2.\]

\[-1. \leq V \leq 1.\]

**Conical Projections \((IABS(JPROJ) = 3)\)**

In a conical projection, the meridians are represented by straight lines radiating from the apex of the flattened cone. Any two longitudes \( \phi_1 \) and \( \phi_2 \) transform to two straight lines at an angle \( n(\phi_2 - \phi_1) \), where \( n \) is the cone constant, less than unity. The parallels transform to circles centered at the apex. The radius of the circle corresponding to latitude and is given by:

\[ R = f(\theta) \]

The cone constant, \( n \), and the function \( f \) depend upon the type of conical projection in use.

In Figure 4, the longitudes \( \phi_1,\phi_2 \) are represented by \( PA,PB \). Then

\[ \angle APB = n(\phi_2 - \phi_1) \]

CD is the arc of a circle with center \( P \), radius \( R \) and represents latitude \( \theta \). Then

\[ R = f(\theta) \]

\[ \begin{align*}
\phi_1 & & A \\
\phi_2 & & B \\
C & & D \\
\theta & & \theta \\
P & & P
\end{align*} \]

Figure 4

\[ \text{SUPMAP - 13} \]
**Lambert Conformal with Two Standard Parallels (3)**

This projection has the property of preserving angles and is relatively distortion free for mid-latitude regions. The cone constant is given by

$$n = \frac{\log(\cos \theta_1) - \log(\cos \theta_2)}{\log(\tan(45 \pm \frac{\theta_1}{2})) - \log(\tan(45 \pm \frac{\theta_2}{2}))}$$

where the lower signs are for the northern hemisphere, upper signs are the southern. The relationship between $R$ and theta is

$$R = [\tan(45 \pm \frac{\theta}{2})]^n$$

The apex of the cone is the origin of the $(U,V)$ plane of projection. The central meridian, theta, is parallel to the V axis with north in the positive direction. Then:

$$U = R\sin[n(\phi - \phi_0)]$$

$$V = \pm R\cos[n(\phi - \phi_0)]$$
SAMPLE PLOT

SUPMAP DEMONSTRATION: LAMBERT CONFORMAL CONIC PROJECTION

SUPMAP (-3,45.0,-100.0,45.0,20.0,-75.0,50.0,-130.0,2,10,1,0)

SUPMAP DEMONSTRATION: LAMBERT CONFORMAL CONIC PROJECTION

SUPMAP (-3,45.0,-100.0,45.0,20.0,-75.0,50.0,-130.0,2,10,1,0)

SUPMAP - 15

SUPMAP (-3,45.0,-100.0,45.0,20.0,-75.0,50.0,-130.0,2,10,1,0)

SUPMAP - 15
SAMPLE PLOT

SUPMAP DEMONSTRATION: ORTHOGRAPHIC PROJECTION

(2, 60., -120., 0., 0., 0., 0., 0., 1, 10, -1, 0)

SUPMAP - 16
(5, 0., 0., 0., 0., 0., 0., 1, 10, -1, 0, -2)
Stereographic Projection

(1, 80., -160., 0., 0., 0., 0., 0., 1, 10, -1, 0)
SAMPLE PLOT

SUPMAP DEMONSTRATION: MOLLWEIDE TYPE PROJECTION

(10, -80., 160., 0., 0., 0., 0., 0., 1, 10, -1, 0, -2)
THREE-DIMENSIONAL LINE DRAWING PACKAGE

LATEST REVISION March 1980

PURPOSE THREED is a package of subroutines that provides line drawing capabilities in three-space. Entry points are equivalent to the line drawing entry points of the System Plot Package with additional arguments added as needed.

USAGE Usage is similar to that of the system PLOT package.

<table>
<thead>
<tr>
<th>SYSTEM PLOT PACKAGE</th>
<th>THREED EQUIVALENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET (XA, XB, YA, YB, XC, XD, YC, YD, LTYPE)</td>
<td>SET3 (XA, XB, YA, YB, UC, UD, VC, VD, WC, WD, EYE)</td>
</tr>
<tr>
<td>CURVE (X, Y, N)</td>
<td>CURVE3 (U, V, W, N)</td>
</tr>
<tr>
<td>LINE (XA, YA, XB, YB)</td>
<td>LINE3 (UA, VA, WA, UB, VB, WB)</td>
</tr>
<tr>
<td>FRSTPT (X, Y)</td>
<td>FRST3 (U, V, W)</td>
</tr>
<tr>
<td>VECTOR (X, Y)</td>
<td>VECT3 (U, V, W)</td>
</tr>
<tr>
<td>POINT (X, Y)</td>
<td>POINT3 (U, V, W)</td>
</tr>
<tr>
<td>PSYM (X, Y, IDPC, IS, IC, IPEN)</td>
<td>PSYM3 (U, V, W, IDPC, SIZE, IDIR, ITOP, IPEN)</td>
</tr>
<tr>
<td>PERIM (MAGX, MINX, MAGY, MINY)</td>
<td>PERIM3 (MAGR1, MINR1, MAGR2, MINR2, IWHICH, VAR)</td>
</tr>
<tr>
<td>TICK4 (MAGX, MINX, MAGY, MINY)</td>
<td>TICK43 (MAGU, MINU, MAGV, MINV, MAGW, MINW)</td>
</tr>
<tr>
<td>TICK (MAGR, MINR)</td>
<td>TICK3 (MAGR, MINR)</td>
</tr>
<tr>
<td>None</td>
<td>FENCE3 (U, V, W, N, IOREN, BOT)</td>
</tr>
</tbody>
</table>

NOTE: Character writing capabilities are available for use with THREED. See the PWRZT write-up.

ARGUMENTS

On Input All arguments correspond to those of the System Plot Package with the following exceptions.

- Where the System Plot Package has X and Y as arguments, the THREED routine has U, V, and W.
- U, V, and W must be floating point.
- EYE is an array three words long containing the U, V, and W coordinates of the EYE position from which the lines are viewed.
- For description of SIZE, IDIR, ITOP, and ICENT, see the write-up for PWRZT.
VAR is the value on the axis specified by IWHICH where the perimeter is drawn.

The subroutine FENCE3 is unique to THREED; its purpose is to draw a line in three space as well as a "fence" between the line and a plane normal to one of the coordinate axes (see the sample plot at the end of this write-up for an example.) The usage is:

CALL FENCE3 (U,V,W,N,IOREN,BOT)

where:

| U, V, W | are linear arrays specifying the coordinates of points in three space on the line to be drawn. |
| N      | is the number of points of the arrays U, V, and W to be used in drawing the line and the fence. |
| IOREN  | specifies the direction in which the fence lines are to be drawn (1 indicates parallel to the U axes, 2 indicates parallel to the V axes, and 3 indicates parallel to the W axes.) |
| BOT    | specifies where the bottom of the fence is to be drawn, e.g. if the fence lines are to be drawn parallel to the W axes, and BOT is given a value of 2., then the bottom of the fence would be the plane W=2. |

On Output  All arguments are unchanged.

NOTES

- Keep EYE outside of the box with corners (UC,VC,WC) and (UD,CD,WD).

- Calls to the regular plotting routines may be made at any time and do not affect this package; i.e., they are completely independent, except that one must call FRSTPT or FRST3 appropriately when switching from one package to the other and using VECTOR or VECT3.

- Routines for annotated axes, software dashed lines, smoothing, etc., will be added at a later date according to user demand.

COMMON BLOCKS  PWRZ1T, SET31, PRM31, TCK31

I/O  Plots lines

PRECISION  Single

REQUIRED ULIB ROUTINES  None

LANGUAGE  FORTRAN

HISTORY  Standardized January 1975

SPACE REQUIRED  44008

ACCURACY  +.5 plotter units per call. There is no cumulative error.

TIMING  The sample picture took about 0.1 seconds on the 7600.

PORTABILITY  DATA statements for variables in COMMON, ENTRY statements; otherwise, portable.

PLOTTING ROUTINES USED  FRSTPT, VECTOR, LINE, CURVE, POINT

REQUIRED RESIDENT ROUTINES  SQRT, ACOS, SIN, COS

SAMPLE PICTURE  The sample picture on the next page was produced by the following program:

PROGRAM MAIN
REAL EYE(3)
DIMENSION U(50), V(50), W(50)
DATA EYE/5., -10., 4./
ISIZ=36
CALL TICK43(24, 16, 24, 16, 24, 16)
CALL SET3(90, 1010, 90, 1010, 0., 2., -1., 1., 0., 1., EYE)
DO 1 I=1, 50
  U(I)=FLOAT(I)*.04
  V(I)=SIN(U(I)*6.)*FLOAT(80-I)/80.
1  W(I)=.5+SIN(U(I)*3.141592)*.5
CALL PERIM3(2, 5, 1, 5, 1, 0.)
CALL PERIM3(2, 5, 1, 5, 2, -1.)
CALL PERIM3(2, 5, 2, 5, 3, 0.)
CALL PWRZT(2.1, -1., 0., 2HU->, 2, ISIZ, 1, 3, -1)
CALL PWRZT(0., 1.1, 0., 2HV->, 2, ISIZ, 2, 3, 0)
CALL PWRZT(0., -1., 1.1, 2HW , 2, ISIZ, 3, -1, 0)
CALL FENCE3(U, V, W, 50, 3, 0.)
CALL FRAME
STOP
END
SAMPLE PLOT
SAMPLE PLOT

DEMONSTRATION PLOT FOR ROUTINE THREED

THREED - 5
SUBROUTINE VELVCT (U,LU,V,LV,M,N,FLO,HI,NSET,LENGTH,ISPV,SPV)

DIMENSION OF ARGUMENTS
U(LU,N), V(LV,N), SPV(2)

LATEST REVISION
December 1979

PURPOSE
VELVCT draws a representation of a two-dimensional velocity field by drawing arrows from each data location, with the length of the arrow proportional to the strength of the field at that location and the direction of the arrow indicating the direction of the flow at that location.

USAGE
If the following assumptions are met, use

CALL EZVEC (U,V,M,N)

Assumptions:
The whole array is to be processed.
The scale factor is chosen internally.
The perimeter is drawn by EZVEC.
FRAME is called by EZVEC.
There are no special values

If these assumptions are not met, use

Call VELVCT (U,LU,V,LV,M,N,FLO,HI,NSET,LENGTH,ISPV,SPV)

ARGUMENTS

On Input U,V
The origins of the two-dimensional arrays containing the velocity field to be plotted. The vector at the point (I,J) has magnitude SQRT(U(I,J)**2 + V(I,J)**2) and direction ATAN2(V(I,J),U(I,J)). Other representations (such as (R,THETA)) can be plotted by changing statement functions in this routine.

LU
The first dimension of U in the calling program

LV
The first dimension of V in the calling program
The number of data values to be plotted in the X-direction (the first subscript direction). When plotting the entire array, \(LU=LV=M\).

The minimum vector length to be plotted

The maximum vector length. (If \(=0\), the maximum of the array will be chosen.)

Flag to control scaling

= 0 VELVCT calls SET to properly scale the plotting instructions to the standard configuration. PERIM is called to draw a border.

> 0 If NSET is positive, VELVCT assumes that SET has been called by the user in such a way as to properly scale the plotting instructions generated by VELVCT. PERIM is not called.

< 0 If NSET is negative, VELVCT calls SET in such a way as to place the contour plot within the limits of the user's last SET call. PERIM is not called.

Length in CRT units of vector if it is HI in magnitude (or, if HI=0, length in CRT units of the longest vector). If \(LENGTH=0\), a value is chosen such that the longest vector could just reach to the tail of the next vector.

Flag to control the special value feature

= 0 Means that the feature is not in use

= 1 Means that if the value of \(U(I,J)=SPV(1)\) the vector will not be plotted.

= 2 Means that if the value of \(V(I,J)=SPV(2)\) the vector will not be plotted.

= 3 Means that if either \(U(I,J)=SPV(1)\) or \(V(I,J)=SPV(2)\), then the vector will not be plotted

= 4 Is identical to ISPV=3 with \(SPV(2)=SPV(1)\)
On Output
All arguments remain unchanged.

NOTE
The endpoints of the vector are computed by:

\[(FX(X,Y),FY(X,Y))\]

and

\[(MXF(X,Y,U,V,SFX,SFY,MX,MY),MYF(X,Y,U,V,SFX,SFY,MX,MY))\]

where \(X = I\), \(Y = J\), \(U = U(I,J)\), \(V = V(I,J)\), and \(SFX\) and \(SFY\) are scale factors. Here \(I = X\)-index, \(J = Y\)-index. \((MX,MY)\) is the location of the tail. Thus, the actual magnitude of the vector is \(\sqrt{DX^2 + DY^2}\) and the direction is \(\text{ATAN2}(DY,DX)\), where \(DX = MX - MXF(...)\) and \(DY = MY - MYF(...)\).

ENTRY POINTS
VELVCT, EZVECT, DRWVEC, VELVEC, VELDAT

COMMON BLOCKS
VEC1 168
VEC2 3

I/O
Plots the vector field

PRECISION
Single

REQUIRED ULIB ROUTINES

LANGUAGE
FORTRAN

HISTORY
Written and standardized in November 1973; revised in May 1975 to include MXF and MYF.

ALGORITHM
Each vector is examined, possibly transformed, then plotted.

SPACE REQUIRED
Approximately 1300, not including the System Plot Package

PORTABILITY
An Implementor's Writeup is available. Contact a member of the Graphics Project at NCAR.

PLOTTING ROUTINES USED
FRSTPT, GETSET, MXMY, PERIM, SET, VECTOR

REQUIRED RESIDENT ROUTINES
SQRT, ATAN2, SIN, COS, ENCD, PWRY

VELVCT - 3
### INTERNAL PARAMETERS

<table>
<thead>
<tr>
<th>NAME</th>
<th>DEFAULT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIG</td>
<td>-1.E320</td>
<td>Constant used to initialize possible search for HI</td>
</tr>
<tr>
<td>EXT</td>
<td>0.25</td>
<td>Lengths of the sides of the plot are proportional to M and N when NSET &lt; 0, except in the case when MIN(M,N)/MAX(M,N)&lt;EXT, in which case a square graph is plotted.</td>
</tr>
<tr>
<td>ICTRFG</td>
<td>1</td>
<td>Flag to control the position of the vector relative the base point</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 0  Center at (MX,MY)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt; 0  Tail at (MX,MY)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt; 0  Head at (MX,MY)</td>
</tr>
<tr>
<td>ILAB</td>
<td>0</td>
<td>Flag to control the drawing of line labels</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 0  Do not draw the labels</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≠ 0  Draw the labels</td>
</tr>
<tr>
<td>INCX</td>
<td>1</td>
<td>x-coordinate step size for less dense arrays</td>
</tr>
<tr>
<td>INCY</td>
<td>1</td>
<td>y-coordinate step size</td>
</tr>
<tr>
<td>IOFFD</td>
<td>0</td>
<td>Flag to control normalization of label numbers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 0  Means include a decimal point when possible (do not normalize)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≠ 0  Means normalize all label numbers by ASH</td>
</tr>
<tr>
<td>IOFFM</td>
<td>0</td>
<td>Flag to control plotting of the message below the plot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 0  Means plot the message</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≠ 0  Means do not plot it</td>
</tr>
<tr>
<td>RMN</td>
<td>5.0</td>
<td>Minimum size line length to scale vector (plotter units)</td>
</tr>
</tbody>
</table>
INTERNAL PARAMETERS (CONT'D)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMX</td>
<td>200.0</td>
<td>Maximum size line length to scale vector (plotter units)</td>
</tr>
<tr>
<td>SIDE</td>
<td>0.90</td>
<td>Length of longer edge of plot (See also EXT.)</td>
</tr>
<tr>
<td>SIZEP</td>
<td>1.25</td>
<td>Size of PWRY labels for vector values</td>
</tr>
<tr>
<td>XLT</td>
<td>0.05</td>
<td>Left hand edge of the plot (0.0 = left edge of the frame 1.0 = right hand edge)</td>
</tr>
<tr>
<td>YBT</td>
<td>0.05</td>
<td>Bottom edge of the plot (0.0 = bottom of frame 1.0 = top of frame)</td>
</tr>
<tr>
<td>ZMN</td>
<td>FLO</td>
<td>Minimum line length to plot (plotter units)</td>
</tr>
<tr>
<td>ZMX</td>
<td>0.0</td>
<td>Maximum line length to generate a degenerate vector (A point.)</td>
</tr>
</tbody>
</table>

Internal functions which may be modified for transformation of the data:

SCALE
 Computes a scale factor used in the determination of the length of the vector to be drawn.

DIST
 Computes the length of a vector

FX
 vector base

MXF
 Returns the x-coordinate of the vector head

FY
 Returns the y index as the y-coordinate of the vector base

MYF
 Returns the y-coordinate of the vector head

VLAB
 The value for the vector label when ILAB ≠ 0.
WINDIOWIG PACKAGE

LATEST REVISION June 1980

PURPOSE To provide a clipping capability for lines extending outside a window defined by the user's SETW call. Thus, part of a picture may be viewed without any distortion or overwriting near the edge of the picture.

USAGE First
CALL SETW(XA,XB,YA,YB,XC,XD,YC,YD,LTYPE)
then, call any of the following:
CALL LINEW(XA,YA,XB,YB)
CALL FRSTW(X,Y)
CALL VECTW(X,Y)
CALL CURVEW(X,Y,N)
CALL POINTW(X,Y)
CALL MXYW(MX,MY)
CALL PLOTIW(IX,IY)
CALL PWRIW(X,Y,ID,N,ISIZ,ITHETA,ICNT)

(The old entry point PWRYW is supported as well, but simply remaps its arguments and calls PWRIW.)

ARGUMENTS

On Input All arguments match those in the corresponding routine in the System Plot Package.
Whenever the system plot package allows either fixed or floating point arguments, the windowing plot package does also.

**On Output**

All arguments are unchanged (except MXMYW).

**Entry Points**

FRSTW, VECTW, SETW, CURVEW, LINEW, POINTW, MXMYW, PLOTIW, PWRITW, PWRYW, FTOI, WINSYS, PWWSOR, PWWGET, PWWBKD.

**COMMON BLOCKS**

WINDW1, WINDW2, WINDW3, PWWCOM.

**I/O**

Plots lines.

**PRECISION**

Single.

**REQUIRED ULIB ROUTINES**

None.

**LANGUAGE**

FORTRAN.

**HISTORY**

Originally part of SCROLL, standardized January 1975.

**ALGORITHM**

Each coordinate is located in one of nine possible areas (5 is the window):

```
        1  2  3
        ----------
        \     / 4
        \   /   5
          \ /   6
          \   7
            \ 8
              9
```

For each line segment, each of the two endpoints can be in one of the nine possible areas, making 81 possible classifications. Based on this 81-way classification, the simplest algorithm that can determine the visible part of the line segment is selected.

**ACCURACY**

Full metacode precision (32K) for floating point coordinate data, implicit virtual grid resolution (as per last SETI call, or default) for integer coordinate data.
TIMING  About twice as long as the system plot package, but the time increase is hardly noticeable in most programs, as the typical time spent at system plot package level is typically a very small fraction of total application program time.

PLOTTING ROUTINES USED  SET, PLOTIT, GETSI, FL2INT.

REQUIRED RESIDENT ROUTINES  ALOG, SIN, COS, INTT, ISHIFT, GETCHR.

NOTES  
- Calls to the regular plotting routines may be made at any time and does not affect this package; i.e., they are completely independent, except that one must call FRSTPT or FRSTW appropriately when switching from one package to the other, if using VECTOR or VECTW. Whereas PWRIT in the system plot package causes the character string itself to be output in the instruction stream, PWRITW strokes the characters, clips the strokes, and thus puts out moves and draws in the instruction stream (similar to PWRITY).

- When using integer coordinates, do not use zero or values less than zero. If the current virtual grid for integer coordinates (as per default or last SETI call) has resolution 2**N in a given coordinate direction, then any user integer coordinate input should be restricted to the range 1 to 2**N in that coordinate direction. (In fact, coordinates greater than 2**N will be handled properly, but coordinates less than zero will probably fail. The support routine INTT, used to detect integer input, is required only to distinguish positive integers from reals.)

- Coordinates returned by MXMYW are not restricted to 1 through 2**N (N as above), and could, in fact, be negative. Zero, however, is never returned.

- If integer input is being used, any SETI call to redefine the integer resolution must be followed by a SETW call before plotting proceeds.

- On computers with 16-bit integers (15 bits of significance), some loss of precision may occur if coordinates are issued that map, before window clipping according to the last SETW call, outside of the 32K metacode address space. As all internal calculations are in floating point, serious errors such as wraparound should not occur, however.
SAMPLE PLOT

DEMONSTRATION PLOT FOR WINDOW

WINDOW - 4
APPENDIX A: THE UNABRIDGED AUTOGRAPH
This document contains eighteen write-ups which together describe the new AUTOGRAPH. They are as follows:

**Table of Contents**

<table>
<thead>
<tr>
<th>TITLE OF WRITE-UP</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOGRAPH PACKAGE</td>
<td>1</td>
</tr>
<tr>
<td>HOW TO DRAW A GRAPH</td>
<td>1</td>
</tr>
<tr>
<td>A SIMPLER WAY OF DRAWING A GRAPH</td>
<td>1</td>
</tr>
<tr>
<td>THE AUTOGRAPH CONTROL PARAMETERS</td>
<td>1</td>
</tr>
<tr>
<td>CONTROL PARAMETER NAMES</td>
<td>2</td>
</tr>
<tr>
<td>ALL CONTROL PARAMETERS ARE FLOATING-POINT</td>
<td>2</td>
</tr>
<tr>
<td>USE OF THE PARAMETER-ACCESS ROUTINES</td>
<td>2</td>
</tr>
<tr>
<td>SPECIAL-VALUE PARAMETERS ‘NULL/1.’ AND ‘NULL/2.’</td>
<td>3</td>
</tr>
<tr>
<td>THE GRAPH WINDOW</td>
<td>3</td>
</tr>
<tr>
<td>THE GRID WINDOW</td>
<td>3</td>
</tr>
<tr>
<td>THE CURVE WINDOW (GRID)</td>
<td>4</td>
</tr>
<tr>
<td>THE GRID COORDINATE SYSTEM</td>
<td>4</td>
</tr>
<tr>
<td>THE USER COORDINATE SYSTEM</td>
<td>4</td>
</tr>
<tr>
<td>HOW TO GRAPH “X AS A FUNCTION OF Y”</td>
<td>5</td>
</tr>
<tr>
<td>WHAT A BACKGROUND CONSISTS OF</td>
<td>5</td>
</tr>
<tr>
<td>THE FOUR AXES</td>
<td>5</td>
</tr>
<tr>
<td>ABBREVIATED FORM OF AXIS–PARAMETER NAMES</td>
<td>5</td>
</tr>
<tr>
<td>THE PARAMETER ‘AXIS/s/CONTROL’</td>
<td>6</td>
</tr>
<tr>
<td>THE PARAMETER ‘AXIS/s/LINE.’</td>
<td>6</td>
</tr>
<tr>
<td>MOVING AN AXIS</td>
<td>6</td>
</tr>
<tr>
<td>THE “LABEL COORDINATE SYSTEM” ALONG AN AXIS</td>
<td>6</td>
</tr>
<tr>
<td>POSITIONING OF MAJOR TICK MARKS ALONG AN AXIS</td>
<td>7</td>
</tr>
<tr>
<td>APPEARANCE OF MAJOR TICK MARKS ON AN AXIS</td>
<td>7</td>
</tr>
<tr>
<td>POSITIONING AND APPEARANCE OF MINOR TICK MARKS ON AN AXIS</td>
<td>7</td>
</tr>
<tr>
<td>NUMERIC LABELS ON AN AXIS</td>
<td>8</td>
</tr>
<tr>
<td>TYPES OF NUMERIC LABELS ON AN AXIS</td>
<td>8</td>
</tr>
<tr>
<td>ORIENTATION OF NUMERIC LABELS ON AN AXIS</td>
<td>8</td>
</tr>
<tr>
<td>POSITIONING OF NUMERIC LABELS RELATIVE TO AN AXIS</td>
<td>8</td>
</tr>
<tr>
<td>CHARACTER SIZES IN NUMERIC LABELS ON AN AXIS</td>
<td>9</td>
</tr>
<tr>
<td>INFORMATIONAL LABELS</td>
<td>9</td>
</tr>
<tr>
<td>THE PREDEFINED LABELS</td>
<td>10</td>
</tr>
<tr>
<td>THE PARAMETER GROUP NAMED ‘LABEL’</td>
<td>10</td>
</tr>
<tr>
<td>THE PARAMETER GROUP NAMED ‘LINE’</td>
<td>11</td>
</tr>
<tr>
<td>HOW TO ACCESS A LABEL DEFINITION</td>
<td>11</td>
</tr>
<tr>
<td>HOW TO ACCESS A LINE DEFINITION</td>
<td>12</td>
</tr>
<tr>
<td>THE LABEL BOXES</td>
<td>12</td>
</tr>
<tr>
<td>BACKGROUND OVERLAP PROBLEMS</td>
<td>13</td>
</tr>
<tr>
<td>DASHED–LINE PATTERNS FOR CURVES</td>
<td>13</td>
</tr>
<tr>
<td>THE PRIMARY–CONTROL–PARAMETER GROUP NAMED ‘DASH’</td>
<td>14</td>
</tr>
<tr>
<td>DASHED–LINE PATTERNS USED BY EZY, EZXY, EZMY, AND EZMXY</td>
<td>14</td>
</tr>
<tr>
<td>WINDOWING OF CURVES DRAWN BY AUTOGRAPH</td>
<td>15</td>
</tr>
<tr>
<td>USE OF PWRX BY AUTOGRAPH</td>
<td>15</td>
</tr>
</tbody>
</table>
## Table of Contents

<table>
<thead>
<tr>
<th>TITLE OF WRITE-UP</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OLD ROUTINES:</strong></td>
<td></td>
</tr>
<tr>
<td>AUTOGRAPH ROUTINE EZY</td>
<td>18</td>
</tr>
<tr>
<td>AUTOGRAPH ROUTINE EZXY</td>
<td>20</td>
</tr>
<tr>
<td>AUTOGRAPH ROUTINE EZMY</td>
<td>22</td>
</tr>
<tr>
<td>AUTOGRAPH ROUTINE EZMXY</td>
<td>24</td>
</tr>
<tr>
<td>AUTOGRAPH ROUTINE ANOTAT</td>
<td>26</td>
</tr>
<tr>
<td>AUTOGRAPH ROUTINE DISPLA</td>
<td>29</td>
</tr>
<tr>
<td><strong>NEW ROUTINES:</strong></td>
<td></td>
</tr>
<tr>
<td>AUTOGRAPH ROUTINE AGSETP</td>
<td>31</td>
</tr>
<tr>
<td>AUTOGRAPH ROUTINE AGSETF</td>
<td>34</td>
</tr>
<tr>
<td>AUTOGRAPH ROUTINE AGSETI</td>
<td>35</td>
</tr>
<tr>
<td>AUTOGRAPH ROUTINE AGGETP</td>
<td>36</td>
</tr>
<tr>
<td>AUTOGRAPH ROUTINE AGGETF</td>
<td>37</td>
</tr>
<tr>
<td>AUTOGRAPH ROUTINE AGGETI</td>
<td>38</td>
</tr>
<tr>
<td>AUTOGRAPH ROUTINE AGSTUP</td>
<td>39</td>
</tr>
<tr>
<td>AUTOGRAPH ROUTINE AGBACK</td>
<td>42</td>
</tr>
<tr>
<td>AUTOGRAPH ROUTINE AGCURV</td>
<td>43</td>
</tr>
<tr>
<td>AUTOGRAPH PARAMETERS</td>
<td>45</td>
</tr>
<tr>
<td>AUTOGRAPH EXAMPLES</td>
<td>91</td>
</tr>
</tbody>
</table>

*Note: This document does not contain a write-up of the ancient routine IDIOT, although a version of this routine is included in the package.*
PURPOSE: To draw graphs — each with a labeled background and each displaying one or more curves.

HOW TO DRAW A GRAPH

To draw a graph, a user program executes a series of calls to AUTOGRAPH routines, typically as follows:

1. The routines AGSETP, AGSETF, and/or AGSETE are called to reset "primary control parameters" (described below) which have unsatisfactory values.
2. The routine AGSTUP is called to perform "set-up" tasks. Basically, it computes appropriate values for the "secondary control parameters" (described below).
3. The routine AGBACK is called to draw a background.
4. The routine AGCURV is called one or more times (once per curve) to draw the desired curves.
5. The system-plot-package routine FRAME is called to advance to a new frame.

To draw the next graph, all five steps are repeated. Step 1 may, of course, be abbreviated or omitted entirely.

A SIMPLER WAY OF DRAWING A GRAPH

Each of the routines EZY, EZXY, EZMY, and EZMXY performs a sequence of calls like that described in the preceding paragraph. A user program may call EZY to graph a single curve defined by the points (i,y(i)), for i from 1 to n, EZXY to graph a single curve defined by the points (x(i),y(i)), for i from 1 to n, EZMY to graph the m curves defined by the points (i,y(i,j)), for i from 1 to n and j from 1 to m, and EZMXY to graph the m curves defined by the points (x(i),y(i,j)) or (x(i,j),y(i,j)), for i from 1 to n and j from 1 to m. See the write-ups of these routines, below, for further information.

See examples 1 through 4 in the AUTOGRAPH examples write-up.

THE AUTOGRAPH CONTROL PARAMETERS
— PRIMARY AND SECONDARY —

The labeled common block AGCONP contains the AUTOGRAPH "control parameters", each of which controls some aspect of the package's behavior. There are two types of control parameters: "primary control parameters" and "secondary control parameters".

Each primary control parameter has a default value and is subject to change by a user program to produce a desired effect.

Each secondary control parameter is computed by AUTOGRAPH itself and is not normally subject to direct change by a user program. The values computed for some of the secondary control parameters may be of interest.
Access to all of the AUTOGRAPH control parameters is provided by the routines AGSETP, AGSETF, AGSETI, AGGETP, AGGETF, and AGGETI. (The routines ANOTAT and DISPLA provide access to a limited subset of the control parameters and are provided principally for historical reasons; they are of interest mainly to users of the routines EZY, EZXY, E2MY, and EZMXY.)

CONTROL PARAMETER NAMES

There are several groups of AUTOGRAPH control parameters. Each group has a keyword associated with it — like BACKGROUND or GRAPH or AXIS. Those groups which contain more than one control parameter are divided into subgroups, each of which also has a keyword associated with it. The subgroups may be further subdivided in the same manner.

Group keywords are used to make up names of control parameter groups and, ultimately, of individual control parameters. This is done by stringing together the group keywords, in descending order, separated by slashes and terminated by a period. For example, the name 'AXIS.' refers to a group of 92 parameters describing the four AUTOGRAPH axes, the name 'AXIS/LEFT.' to a subgroup of 23 parameters describing the left axis, the name 'AXIS/LEFT/NUMERIC.' to a further subgroup of 8 parameters describing the numeric labels on the left axis, and the name 'AXIS/LEFT/NUMERIC/TYP.' to a single parameter describing the type of numeric labels on the left axis.

Control parameter names are used as arguments in calls to the routines AGSETP, AGSETF, AGSETI, AGGETP, AGGETF, and AGGETI to identify the parameter(s) whose values a user program wishes to "set" or "get". For example, the statement

```fortran
CALL AGSETP ('AXIS/LEFT/NUMERIC/TYP.',1.)
```

or the more standard FORTRAN-66 statement

```fortran
CALL AGSETP (23HAXMS/LEFT/NUMERIC/TYP.,1.)
```

is used to set the value of the parameter specifying the type of numeric labels to be used on the left axis to '1.'.

Complete information about the parameters and their names is given in the AUTOGRAPH parameter write-up, below. In particular, that write-up describes ways in which control parameter names may be shortened considerably. For example, the name 'AXIS/LEFT/NUMERIC/TYP.' may be shortened to 'LEFT/TYP.' or even to 'LE/TY.'.

All of the variables in the labeled common block AGCONP (the AUTOGRAPH control parameters) are floating-point — even those which serve as type specifiers, control flags, item counts, list pointers, and the like — for which integer variables would normally be used. This was done because of a portability problem which arose in implementing the parameter-access routines.

Those parameters which may only have discrete integral values are referenced internally using the FORTRAN function IFIX. For example: The parameter 'X/NICE.' corresponds to a variable in the common block AGCONP named QCSEX, which may have the value "-1," "0," or "+1." The function IFIX(QCSEX) is used by AUTOGRAPH to recover an integer value.

USE OF THE PARAMETER-ACCESS ROUTINES

The routine AGSETP (AGGETP) is called by a user program to "set" ("get") the floating-point values of a specified group of related parameters. The routine AGSETF is used to store a floating-point number as the value of a single specified parameter, the routine AGGETF to retrieve the floating-point value of a single specified parameter. The routine AGSETP is used to store the floating-point equivalent of an integer as the value of a single specified parameter, the routine AGGETI to retrieve the integer equivalent of the value of a single specified parameter.

Single-word Hollerith constants should be treated as if they were floating-point. For example, to set the value of 'LABEL/NAME.' to "1HR", one should use the routine AGSETP or AGSETF; using AGSETI would result in the storing of FLOAT('1HR') as the value of 'LABEL/NAME.', with possibly disastrous results.
The primary control parameters 'NULL/1.' and 'NULL/2.' define the special values "null 1" and "null 2", which have the default values "1.E36" and "2.E36", respectively. These special values have several uses:

Certain primary control parameters may be given an "actual" value or one of the "null" values. An actual value is one which AUTOGRAPH is to use as is. The value "null 1" specifies that AUTOGRAPH is to choose an appropriate value to use, but that it is not to store that value in place of the "null 1". The value "null 2" specifies that AUTOGRAPH is to choose an appropriate actual value and store that actual value in place of the "null 2".

Example: The parameter 'Y/MINIMUM.', which specifies the minimum y coordinate, has the default value "null 1", specifying that, for each graph, AUTOGRAPH is to choose an appropriate minimum y. This parameter may be given an actual (non-null) value, thus imposing a desired minimum y, or it may be given the value "null 2", specifying that AUTOGRAPH is to choose an appropriate minimum y for the next graph and then use that value for following graphs.

The value "null 1" is used as a default value for primary control parameters whose desired default values cannot be generated at compile time.

Example: The parameter 'AXIS/LEFT/FUNCTION.' has the default value "null 1" in place of the desired default value "FLOAT(LOC(AGXTOK))".

The value "null 1" is used in x/y coordinate data to signal missing points.

Note: If your x/y coordinate data might include the values "1.E38" or "2.E36", your program's first action should be to change the values of 'NULL/1.' and 'NULL/2.' to values which cannot possibly occur in the data.

See § 9 in the AUTOGRAPH control parameters write-up.

THE GRAPH WINDOW

The primary control parameters 'GRAPH/LEFT.', 'GRAPH/RIGHT.', 'GRAPH/BOTTOM.', and 'GRAPH/TOP.' serve to locate the edges of a rectangular "graph window" within the plotter frame. The first two are stated as fractions of the frame width, the second two as fractions of the frame height. These parameters have the default values "0.", "1.", "0.", and "1., respectively, specifying a graph window which fills the entire plotter frame.

The current graph window is the area in which a graph, including labels, is to be drawn. A user program may limit a graph to any selected portion of the plotter frame. For example, changing the values of the parameters in the group 'GRAPH.' to "0.", "5.", "0.", and "5.", limits a graph to the lower left-hand quarter of the frame.

See § 10 in the AUTOGRAPH control parameters write-up. See example 8 in the AUTOGRAPH examples write-up.

THE GRID WINDOW

The primary control parameters 'GRID/LEFT.', 'GRID/RIGHT.', 'GRID/BOTTOM.', and 'GRID/TOP.' serve to locate the edges of a rectangular "grid window" within the graph window. The first two are stated as fractions of the graph-window width, the second two as fractions of the graph-window height. These parameters have the default values ".15", ".95", ".15", and ".95", respectively, specifying a grid window which is 6/10 the width of the graph window, 8/10 the height of the graph window, and placed slightly off-center (upwards and to the right) within the graph window.

The current grid window is the portion of the graph window within which the "curve window" or "grid" (see the next paragraph) is to lie.

See § 11 in the AUTOGRAPH control parameters write-up. See example 7 in the AUTOGRAPH examples write-up.
THE CURVE WINDOW (GRID)

The primary control parameter 'GRID/SHAPE.' specifies the shape of the "curve window" or "grid". This parameter has the default value "0.", specifying a curve window which completely fills the grid window.

The curve window is the portion of the grid window in which curves are to be drawn. A user program may change the value of 'GRID/SHAPE.' so as to specify a curve window of a particular fixed shape or a curve window whose shape is determined by the x and y coordinate data to be used. In any case, the curve window is centered in, and made as large as possible in, the grid window.

Various positioning parameters are stated in a "grid coordinate system" based on the curve window; curve-point coordinates are stated by the user in a "user coordinate system" which maps into the curve window (see the following two paragraphs). If the windowing control parameter (named 'WINDOW.') has the value "1.", only those curve portions which lie inside the curve window are drawn. The four AUTOGRAPH axes are positioned along the edges of the curve window. The area outside the curve window (but inside the graph window) is used for labels. Character sizes and label-offset distances are specified as fractions of the smaller dimension of the curve window, so as to be in scale with the rest of the graph.

See § 11.5 in the AUTOGRAPH control parameters write-up. See examples 5 and 8 in the AUTOGRAPH examples write-up.

THE GRID COORDINATE SYSTEM

Internally, AUTOGRAPH makes use of a "grid coordinate system"; the user may also make use of this system in setting certain parameter values. Its origin is at the lower left-hand corner of the curve window. X coordinates run linearly from "0." to "1." horizontally, and y coordinates linearly from "0." to "1." vertically, in the curve window. Note that coordinate values outside the range (0.,1.) may be used to reference points outside the curve window.

THE USER COORDINATE SYSTEM

Curve-defining points provided to AUTOGRAPH by a user program are stated in a "user coordinate system". Ten primary control parameters specify how the user coordinate system is mapped into the curve window. (The parameter 'INVERT.', described below, might be considered an eleventh.)

The first five of the ten, named 'X/MINIMUM.', 'X/MAXIMUM.', 'X/LOGARITHMIC.', 'X/ORDERING.', and 'X/NICE.', specify how user x coordinates are to be mapped onto the horizontal axis of the curve window. The default values of these parameters are such that AUTOGRAPH (the routine AGSTUP) is forced to:

Compute, from the user's x-coordinate data, minimum and maximum values Xm and XM.
Compute "nice" (rounded) values Xm' and XM' such that Xm' ≤ Xm ≤ XM ≤ XM'.
Map Xm' to the left edge, and XM' to the right edge, of the curve window. The mapping is linear.

The other five parameters, named 'Y/MINIMUM.', 'Y/MAXIMUM.', etc., specify how user y coordinates are to be mapped onto the vertical axis of the curve window. The default values specify a mapping analogous to that of x coordinates.

By changing the values of these ten parameters appropriately, a variety of desirable ends may be achieved:

Values of Xm, XM, Ym, and/or YM may be specified, thus limiting the graph to a particular range of interest and/or forcing consistent scaling of a group of graphs.
Either or both mappings may be made logarithmic. (The logarithms of coordinate values are mapped linearly onto the axis.)
Either or both mappings may be flipped end-for-end. X coordinates may be made to decrease from left to right, y coordinates to decrease from bottom to top.
The rounding process for either or both mappings may be suppressed, forcing curves to be plotted full-scale.
See §12 and §13 in the AUTOGRAPH control parameters write-up. See examples 6, 7, and 8 in the AUTOGRAPH examples write-up.

HOW TO GRAPH "X AS A FUNCTION OF Y"

The primary control parameter 'INVERT.' has the default value "0.". If it is set to "1." by a user program, the routines AGSTUP and AGCURV will behave as if their x and y arguments had been interchanged. In some sense, this provides a way of plotting "x as a function of y".

This parameter is of principal interest to the users of EZY, EZXY, EZMY, and EZMXY; those users who call the routines AGSTUP and AGCURV directly should probably leave the parameter zeroed.

See §6 in the AUTOGRAPH control parameters write-up. See example 8 in the AUTOGRAPH examples write-up.

WHAT A BACKGROUND CONSISTS OF

A background drawn by the routine AGBACK consists of four axes and up to eight informational labels, each of the latter having one or more lines of text (or none). Each of these entities is defined by a group of primary control parameters and may be modified in a variety of ways.

THE FOUR AXES

The four axes are positioned along the edges of the curve window. There is a left y axis, a right y axis, a bottom x axis, and a top x axis. Each of the axes consists of a line, major tick marks, minor tick marks, and numeric labels. Numeric labels are placed at major-tick-mark positions.

The axes are defined by the primary-control-parameter group named 'AXIS.', which has subgroups 'AXIS/LEFT.', 'AXIS/RIGHT.', 'AXIS/BOTTOM.', and 'AXIS/TOP.'. Each of these subgroups contains 23 parameters defining one of the four axes. These 23 parameters fall into six further subgroups, having the associated keywords CONTROL, LINE, INTERSECTION, FUNCTION, TICKS, and NUMERIC.

The default values of the axis parameters specify a "perimeter" background: All four axes are drawn, each has short, inward-pointing major and minor ticks, the left axis and the bottom axis have numeric labels (placed outside the curve window), the right-axis and top-axis numeric labels are suppressed. See §§8 and §14 in the AUTOGRAPH control parameters write-up. See examples 1 through 4 in the AUTOGRAPH examples write-up.

A "half-axis" background is created by suppressing the right axis and the top axis completely. A "grid" background is created by extending the left-axis and bottom-axis ticks all the way across the curve window and suppressing the ticks on the other two axes. The primary control parameter 'BACKGROUND.' allows the user to create these standard backgrounds easily; whenever its value is changed by a user-program call to AGSETP, AGSETF, or AGSETI, parameters in the group 'AXIS.' are modified to create the desired background.

See §14 in the AUTOGRAPH control parameters write-up. See examples 5, 6, and 8 in the AUTOGRAPH examples write-up.

ABBREVIATED FORM OF AXIS-PARAMETER NAMES

In the ensuing discussions of the various parameters in the group 'AXIS.', the character "s" is used to stand for any one of the keywords "LEFT", "RIGHT", "BOTTOM", and "TOP". For example, 'AXIS/s/FUNCTION.' stands for any one of 'AXIS/LEFT/FUNCTION.', 'AXIS/RIGHT/FUNCTION.', etc. This form is shorter and makes it clear that four different parameters or groups of parameters are being described at once.
THE PARAMETER 'AXIS/s/CONTROL.'

The parameter 'AXIS/s/CONTROL.' may be given any integral value from "-1." to "+4."

The value "-1." specifies that only the line portion of the axis specified by "s" is to be drawn.
The value "0." specifies that no portion of the axis is to be drawn.
A value from "1." to "4." specifies that all portions of the axis are to be drawn and tells AUTOGRAPH what liberties it may take in attempting to cope with numeric labels which will not fit along the axis without overlapping.

The precise meanings of each value are given in the AUTOGRAPH parameter write-up (§ 14.1.1).

THE PARAMETER 'AXIS/s/LINE.'

The parameter 'AXIS/s/LINE.' has the default value "0.". Setting it to a "1." causes the line portion of the axis specified by "s" to be suppressed. Tick marks and/or numeric labels may still be drawn. See §14.1.2 in the AUTOGRAPH parameters write-up.

MOVING AN AXIS

Each of the primary control parameters

'AXIS/s/INTERSECTION/GRID.'

and

'AXIS/s/INTERSECTION/USER.'

has the default value "null 1", specifying that the axis "s" is to be drawn in its normal position, along the edge of the curve window. If either parameter is given a non-null value, the axis "s" is moved away from its normal position in such a way as to intersect the sides of the grid which are perpendicular to it at a point specified by that non-null value. To move an x axis, a y coordinate is specified; to move a y axis, an x coordinate is specified.

The coordinate may be specified in the grid coordinate system or in the user coordinate system, depending on which parameter is used. If both parameters are given non-null values, the latter takes precedence.

No axis may be moved outside the graph window. Attempting to do so moves the axis as far as the edge, but no farther.
See §14.1.3.1. and §14.1.3.2. in the AUTOGRAPH control parameters write-up. See example 8 in the AUTOGRAPH examples write-up.

THE "LABEL COORDINATE SYSTEM" ALONG AN AXIS

The control parameter 'AXIS/s/FUNCTION.' specifies the memory address of a FORTRAN subroutine defining the mapping from values in the user coordinate system along the axis specified by "s" (x values or y values, as appropriate) to values in a "label coordinate system" along that axis, and vice-versa. The quantity "FLOAT(LOC(AGXTOX))", which defines the identity function "f(x)=x" and its inverse "f(x)=x", is used as a default value.

Tick marks are positioned at "nice" values in the label coordinate system, mapped to the user coordinate system, and then mapped onto the axis. Numeric labels are associated with major ticks and provide values in the label coordinate system.

A user program may change the label coordinate system for any of the four axes. For example, if the y-coordinate data is in miles/hour and it is desired that the left y axis be tick-marked and labeled in meters/second, the user program must include a subroutine — say MPHMS — which will do the conversion from miles/hour to meters/second (or vice-versa, depending on the value of its first argument) and must store "FLOAT(LOC(MPHMS))" as the value of the parameter named 'AXIS/LEFT/FUNCTION.' See §14.1.4. in the AUTOGRAPH control parameters write-up, below, for further details.
Note that the tick-marking and labeling of one x (y) axis of a graph may be completely different from that of the other x (y) axis of the graph. For example, the left y axis could be made to indicate “height in kilometers” and the right y axis “pressure in millibars”. See example 7 in the AUTOGRAPH examples write-up.

POSITIONING OF MAJOR TICK MARKS ALONG AN AXIS

The primary control parameter group named

'AXIS/s/TICKS/MAJOR/SPACING.'

contains three parameters, with associated keywords TYPE, BASE, and COUNT. These parameters are described in detail in the AUTOGRAPH parameter write-up. Major tick marks may be spaced linearly or logarithmically in the label coordinate system along the axis specified by "s", or suppressed altogether. Each of the TYPE and BASE parameters has the default value "null 1", allowing AUTOGRAPH to position major tick marks as it sees fit.

See §14.1.5.1 in the AUTOGRAPH control parameters write-up. See examples 7 and 8 and the final example in the AUTOGRAPH examples write-up.

APPEARANCE OF MAJOR TICK MARKS ON AN AXIS

The primary control parameter

'AXIS/s/TICKS/MAJOR/PATTERN.'

has an integral value from "0." to "65535." and specifies the dashed-line pattern to be used for major ticks on the axis specified by "s". Each "0" bit in the lower 18 bits of the integral value specifies a gap 3 plotter units long, each "1" bit a solid portion 3 plotter units long. The default value "65535." (2 to the 16th minus 1) specifies a solid line. The value "0." may be used to suppress major tick marks on the axis "s".

The primary control parameters

'AXIS/s/TICKS/MAJOR/LENGTH/OUTWARD.'

and

'AXIS/s/TICKS/MAJOR/LENGTH/INWARD.'

specify the lengths of the outward-pointing and inward-pointing portions of the major ticks. Each is stated as a fraction of the smaller dimension of the curve window. If either of these values is made greater than or equal to "1.", it specifies a tick-mark portion which extends to the edge of the curve window and a little beyond, the magnitude of the "little beyond" being specified by the fractional portion of the parameter value. The default values give inward-pointing major ticks of length ".015" on all axes.

See §14.1.5.1.2 and §14.1.5.1.3 in the AUTOGRAPH control parameters write-up. See example 8 in the AUTOGRAPH examples write-up.

POSITIONING AND APPEARANCE OF MINOR TICK MARKS ON AN AXIS

The primary control parameter

'AXIS/s/TICKS/MINOR/SPACING.'

specifies the number of minor ticks which are to occur between pairs of major ticks on the axis specified by "s". Minor ticks are equidistantly spaced in the label coordinate system for that axis. The default value of this parameter is "null 1", allowing AUTOGRAPH to position minor ticks as it sees fit.

The primary control parameters

'AXIS/s/TICKS/MINOR/PATTERN.',

'AXIS/s/TICKS/MINOR/LENGTH/OUTWARD.',

and

'AXIS/s/TICKS/MINOR/LENGTH/INWARD.'
specify the dashed-line pattern, outward-pointing length, and inward-pointing length of minor ticks. They are defined in the same way as the analogous major-tick parameters, except that the default inward-pointing tick length is "0.010".

See §14.1.5.2 in the AUTOGRAPH control parameters write-up. See the final example in the AUTOGRAPH examples write-up.

NUMERIC LABELS ON AN AXIS

The primary control parameter group named ‘AXIS/s/NUMERIC.’ contains eight parameters describing the numeric labels on the axis specified by "s". These parameters are described in detail in the AUTOGRAPH parameter write-up; the ensuing paragraphs describe them in a sketchy, general way.

TYPES OF NUMERIC LABELS ON AN AXIS

The primary control parameter ‘AXIS/s/NUMERIC/TYPE.’ may be given any integral value from "0." to "3." or one of the values "null 1" or "null 2".

The value "0." suppresses numeric labels on the axis specified by "s".

The values "1.", "2.", and "3." specify the use of "scientific notation", "exponential notation", and "no-exponent notation", respectively.

A null value gives AUTOGRAPH the freedom to use one of the values "1.", "2.", or "3." — whichever is most consistent with the label coordinate system along the axis.

The exact nature of the labels produced by a given value depends on the parameter

'AXIS/s/TICKS/MAJOR/SPACING/TYPE.',
described above, and two other parameters,

'AXIS/s/NUMERIC/EXPONENT.'

and

'AXIS/s/NUMERIC/FRACTION.'.

See §14.1.6. in the AUTOGRAPH control parameters write-up.
See example 7 and the final example in the AUTOGRAPH examples write-up.

ORIENTATION OF NUMERIC LABELS ON AN AXIS

The primary control parameters

'AXIS/s/NUMERIC/ANGLE/1ST.'

and

'AXIS/s/NUMERIC/ANGLE/2ND.'

may have integral values "0.", "90.", "180.", or "270.". They specify the user's first and second choices for the orientation of numeric labels on the axis specified by "s". AUTOGRAPH will attempt to use the first choice (default value — "0." for all axes); if that leads to overlap problems and shrinking the labels either doesn't help or is not permitted and rotation is permitted (see the parameter ‘AXIS/s/CONTROL.’), AUTOGRAPH may try the second choice (default value — "90." for all axes).

The values given represent angles measured in degrees counter-clockwise from horizontal.

POSITIONING OF NUMERIC LABELS RELATIVE TO AN AXIS

The primary control parameter

'AXIS/s/NUMERIC/OFFSET.'

specifies on which side of the axis specified by "s" the numeric labels are to lie and the size of the gap to be left between the axis line and the numeric labels.
A negative value specifies labels inside the curve window.
A zero value specifies labels centered on the axis, suppresses the line portion of the axis, and
moves the inward-pointing and outward-pointing portions of ticks out away from the axis so as
to leave room for the labels.
A positive value specifies labels outside the curve window.

The magnitude of the value specifies the distance from the axis to the nearest portion of the
label, stated as a fraction of the smaller side of the curve window.
The default value for all axes is ".015".

CHARACTER SIZES IN NUMERIC LABELS ON AN AXIS

The primary control parameters

'A/XIS/s/NUMERIC/WIDTH/MANTISSA.'

and

'A/XIS/s/NUMERIC/WIDTH/EXPONENT.'

specify the widths of characters in the mantissa and exponent portions of the numeric labels on the
axis specified by "s". They are stated as fractions of the smaller dimension of the curve window.
The sizes specified are those desired by the user. If an overlap problem arises and the
'A/XIS/s/CONTROL.' parameter is set so as to allow AUTOGRAPH to shrink the numeric labels, the
characters may end up smaller than desired. No character is shrunk to less than the minimum read-
able size, however.
These parameters have default values ".015" and ".010", respectively.

INFORMATIONAL LABELS

As many as m informational labels may be defined at any one time — normally, m = 8. The informational labels form a part of the background produced by a call to the routine AGBACK. Each of the informational labels is defined as follows:

Each label has a name — a short character string whose value is treated as a floating-point con-
stant — which uniquely identifies the label.
Each label has a "suppression flag", which may be set to enable or disable plotting of the label.
Each label is positioned relative to a "basepoint", whose x and y coordinates are specified in the
grid coordinate system. Normally, the basepoint lies on one edge — but not at a corner point —
of the curve window.
Emanating from the label basepoint is an "offset vector", whose x and y components are
specified as signed fractions of the smaller dimension of the curve window. Normally, the offset
vector is used to specify the size of the gap to be left between an informational label and the
define of the curve window. The presence or absence of an axis along that edge of the curve win-
dow is ignored when specifying this gap; see the paragraph "BACKGROUND OVERLAP PROB-
LEMS", below.
Emanating from the end of the offset vector is a "baseline", whose direction is specified as an
angle in degrees ("0.", "90.", "180.", or "270.", measured counter-clockwise from horizontal).
The text lines of a label are written parallel to, and in the same direction as, the baseline.
A centering option for each label determines whether the left edges, the centers, or the right
edges, of the text lines are aligned with the end of the offset vector.

Each label may contain one or more text lines (or none). The total number of text lines in all
labels must not exceed n — normally, n = 16. Each of the text lines is defined as follows:

Each text line has an integral position number which distinguishes it from every other line in
the same label. Multiples of "100." are recommended. Lines with positive position numbers are
drawn above the label baseline, lines with negative position numbers below the label baseline. A
line with position number "0." is centered on the label baseline. "Above" and "below" are
defined here from the viewpoint of a reader of the label. The position numbers of the lines in a
label specify the order in which the labels appear — strictly decreasing from top to bottom —
but do not determine the interline spacing, which is determined by AUTOGRAPH itself.

Each line has a "suppression flag", which may be set so as to enable or disable drawing of the line.

Each line has a character-width specifier, stated as a fraction of the smaller dimension of the curve window.

The text of each line is defined by a Hollerith character string — only the memory address of which is saved by AUTOGRAPH — and a count of the number of characters in the string — normally determined by AUTOGRAPH itself.

Note: The string replacements "$\text{"FLLB(10,8)"} = \text{"FLLB(10,m)"}$", where $m \geq 5$, and "$\text{"FLLN(6,16)"} = \text{"FLLN(6,n)"}$", where $n \geq 5$, may be applied to an AUTOGRAPH source file to provide for a maximum of $m$ labels and $n$ lines.

**THE PREDEFINED LABELS**

The AUTOGRAPH parameter write-up describes in detail four "predefined" labels, named 'R', 'L', 'B', and 'T'. Each of these labels lies along one of the four edges of the curve window — the left edge, the right edge, the bottom edge, or the top edge.

The predefined labels greatly simplify the task of generating labels along the edges of the curve window. For example, if you want a "header label" above the curve window, you need only specify the desired character string to define the text of line number "100." of the label named 'T'.

The default definitions of the predefined labels specify a label reading "X" below the curve window and a label reading "Y" to the left of the curve window.

See examples 5 and 7 and the final example in the AUTOGRAPH examples write-up.

**THE PARAMETER GROUP NAMED 'LABEL'**

The group of primary control parameters named 'LABEL' contains $10m+3$ parameters — normally, $m = 8$. Together with the parameters in the group named 'LINE.', they define the informational labels to be drawn by a call to the routine AGBACK. The parameters in the group 'LABEL.' are as follows:

The parameter 'LABEL/CONTROL.' may be given the value "$-1." to delete all currently-defined labels, the value "$0." to temporarily disable the drawing of labels, the value "$1." to enable the drawing of labels and prevent the shrinkage of labels when overlap problems arise, or the value "$2." to enable the drawing of labels and allow shrinkage. The default value is "$2.". See §17. in the AUTOGRAPH control parameters write-up. See examples 5 and 6 in the AUTOGRAPH examples write-up.

The parameter 'LABEL/BUFFER/LENGTH.' should not normally be set by a user program. Its value is $m$, the maximum number of labels AUTOGRAPH can handle.

The subgroup 'LABEL/BUFFER/CONTENTS.' consists of $10m$ words, in which the label definitions are stored. Normally, a user program should not attempt to store values in this block directly. See the paragraph "HOW TO ACCESS A LABEL DEFINITION", below.

The parameter 'LABEL/NAME.' is used in the process of accessing a label definition. It functions as a switch, pointing to the label definition currently being accessed.

See §16 in the AUTOGRAPH control parameters write-up. See examples 5 and 7 and the final example in the AUTOGRAPH examples write-up.

See the AUTOGRAPH parameter write-up for further information about these parameters.
THE PARAMETER GROUP NAMED 'LINE.'

The group of primary control parameters named 'LINE.' contains 6n+4 parameters - normally, n = 16. They define the lines belonging to the various labels. The parameters in the group 'LINE.' are as follows:

The parameters 'LINE/MAXIMUM.' and 'LINE/END.' define the assumed maximum line length (default = 40 characters) and the line end character (default = '$'). These parameters come into play when a user program defines the text of a line. The character string tendered by the user is assumed to be of maximum length; if it is really shorter than that, it must be followed by the line end character. See § 17 in the AUTOGRAPH control parameters write-up. See example 8 and the final example in the AUTOGRAPH examples write-up.

The parameter 'LINE/BUFFER/LENGTH.' should not normally be set by a user program. Its value is n, the maximum number of lines AUTOGRAPH can handle.

The subgroup 'LINE/BUFFER/CONTENTS.' consists of 6n words, in which the line definitions are stored. Normally, a user program should not attempt to store values in this block directly. See the paragraph "HOW TO ACCESS A LINE DEFINITION", below.

The parameter 'LINE/NUMBER.' is used in the process of accessing a line definition. It functions as a switch, pointing to the line definition currently being accessed.

See § 17 in the AUTOGRAPH control parameters write-up for further information about these parameters.

See examples 5 and 7 and the final example in the AUTOGRAPH examples write-up.

HOW TO ACCESS A LABEL DEFINITION

To access a label definition, a user program must first execute an AGSETP (or AGSETF) call to store the name of the label as the value of the parameter 'LABEL/NAME.' Such a call does not actually store the name as the value of that parameter. Instead, it causes the label buffer to be searched for the definition of the named label. If that definition is not found, a default definition is made up and inserted in the label buffer. In any case, the index of the definition is floated and stored as the value of the parameter 'LABEL/NAME.'.

Once the parameter 'LABEL/NAME.' has been set in this manner, the primary control parameter group name 'LABEL/DEFINITION.' and subgroup names of the form 'LABEL/DEFINITION/...' may be used to access the parameters defining the label. These parameters are as follows:

The parameter 'LABEL/DEFINITION/SUPPRESSION.' may be given the value "-2." to delete the label and all of its lines, the value "-1." to delete the lines of the label but leave the label itself defined, the value "0." to enable drawing of the label, and the value "1." to temporarily suppress drawing of the label. It has the default value "0.". When a label is deleted, the parameters 'LABEL/NAME.' and 'LINE/NUMBER.' become undefined; similarly, when the lines of a label are deleted, the parameter 'LINE/NUMBER.' becomes undefined.

The parameters 'LABEL/DEFINITION/BASEPOINT/X.' and '...Y' specify the coordinates, in the grid coordinate system, of the label's basepoint. The default basepoint is (.5,.5).

The parameters 'LABEL/DEFINITION/OFFSET/X.' and '...Y' specify the components of the label's offset vector, as signed fractions of the smaller dimension of the curve window. The default vector has zero components.

The parameter 'LABEL/DEFINITION/ANGLE.' specifies the angle ("0.", "90.", "180.", or "270.") at which the label's baseline emanates from the end of its offset vector. The default angle is "0."

The parameter 'LABEL/DEFINITION/CENTER.' has the value "-1." to align the left ends, the value "0." to align the centers, and the value "+1." to align the right ends, of the lines of the label with the end of its offset vector. The default value is "0.".

The parameters 'LABEL/DEFINITION/LINES.' and 'LABEL/DEFINITION/INDEX.' are not normally set by a user program; they are maintained by AUTOGRAPH. The former specifies the number of lines belonging to the label and the latter specifies the index (in the line buffer) of the definition of the first line belonging to the label. A default label has no lines - both of these parameters are zeroed.

Usage page 11
How to Access a Line Definition

To access the definition of one of the lines of a label, a user program must first access the label definition by setting the parameter 'LABEL/NAME.', as described above. Then, it must execute an AGSETP (or AGSETI) call to store the number of the desired line as the value of the primary control parameter 'LINE/NUMBER.' Such a call does not actually store the specified number as the value of that parameter. Instead, it causes the line buffer to be searched for the definition of the desired line. If that definition is not found, a default definition is made up, inserted in the line buffer, and added to the linked list of definitions of lines belonging to the label. In any case, the index of the definition is floated and stored as the value of the parameter 'LINE/NUMBER.'.

Once the parameter 'LINE/NUMBER.' has been set in this manner, the primary control parameter group name 'LINE/DEFINITION.' and subgroup names of the form 'LINE/DEFINITION/...' may be used to access the parameters defining the line. These parameters are as follows:

- The parameter 'LINE/DEFINITION/SUPPRESSION.' may be given the value "-1." to delete the line, the value "0." to enable drawing of the line, and the value "+1." to temporarily disable drawing of the line. It has the default value "0.". When a line is deleted, the parameter 'LINE/NUMBER.' becomes undefined.
- The parameter 'LINE/DEFINITION/CHARACTER-WIDTH.' specifies the desired width of each character in the line, stated as a fraction of the smaller dimension of the curve window. The default width is "0.015".
- The parameter 'LINE/DEFINITION/TEXT.' specifies the memory address of the character string comprising the text of the label. The default value is, effectively, the address of a single blank.
- The parameter 'LINE/DEFINITION/LENGTH.' specifies the length of the character string. The default value is a "1.".
- The parameter 'LINE/DEFINITION/INDEX.' is not normally set by a user program. It is maintained by AUTOGRAPH and specifies the index (in the line buffer) of the next line belonging to the label.

See §18 in the AUTOGRAPH control parameters write-up for further information about these parameters. See examples 5 and 7 and the final example in the AUTOGRAPH examples write-up.

Note: As a convenience to the user, an AGSETP call to set the parameter 'LINE/DEFINITION/TEXT.' has as its second argument a character string (or an array containing a character string), rather than the floated memory address of the string. The call sets both of the parameters defining the text of the label – '...TEXT.' and '...LENGTH.' – the latter being computed by examining the string. The string must be no longer than the length specified by the value of 'LINE/MAXIMUM.' and, if shorter, it must be followed by the 'LINE/END.' character.

The Label Boxes

Each informational label is considered to lie in one of six "label boxes", as follows:

Box 1 lies to the left of the curve window. It contains all labels which have a basepoint on the left edge of the curve window and a leftward-pointing offset vector.

Box 2 lies to the right of the curve window. It contains all labels which have a basepoint on the right edge of the curve window and a rightward-pointing offset vector.

Box 3 lies below the curve window. It contains all labels which have a basepoint on the bottom edge of the curve window and a downward-pointing offset vector.

Box 4 lies above the curve window. It contains all labels which have a basepoint on the top edge of the curve window and an upward-pointing offset vector.

Box 5 lies in the interior of the curve window. It contains all labels which have a basepoint on some edge of the curve window and an inward-pointing offset vector.
Box 6 is the entire graph window. It contains all of the remaining labels.

Three restrictions must be observed by the user: First, no label's basepoint may have coordinates (0.0), (0.1), (1.0), or (1.1); these corner points must be avoided. Second, no portion of any label in boxes 1 through 4 may lie inside the curve window. Third, no portion of any label in box 5 may lie outside the curve window.

The label-box concept is important in handling overlap problems, which are discussed in the next paragraph.

BACKGROUND OVERLAP PROBLEMS

The responsibility for avoiding background overlap problems might reasonably have been placed squarely on the shoulders of the user, except for one unpleasant fact: numeric labels are unpredictable critters. Accordingly, AUTOGRAPH accepts a part of the burden.

In attempting to keep the numeric labels on a given axis from overlapping each other, AUTOGRAPH may shrink and/or reorient them. Either or both of these actions may be suppressed by the user by resetting the parameter 'AXIS/s/CONTROL.' If a problem still exists, some of the labels may be omitted — perhaps leaving every second one, every third one, every fourth one, etc.

Informational labels are positioned by the user along the edges of the curve window as if numeric labels did not exist. AUTOGRAPH takes the following actions in attempting to prevent the information labels from overlapping the numeric labels on any axis:

1. Box 1 labels (to the left of the curve window) are moved leftward, box 2 labels (to the right of the curve window) are moved rightward, box 3 labels (below the curve window) are moved downward, box 4 labels (above the curve window) are moved upward, and box 5 labels (inside the curve window) are moved inward. Box 6 labels are not moved.

2. If, during step 1, a label is shoved outside the graph window by the numeric labels on some axis, those numeric labels may be shrunk and/or re-oriented, as allowed by the user's setting of 'AXIS/s/CONTROL'.

3. If one or more of the labels in a given box still lies partly outside the graph window, the labels in that box may be shrunk, depending on the current setting of the parameter 'LABEL/CONTROL'. Each label in the box shrinks toward the end of its offset vector.

4. If one or more of the labels in boxes 1 through 4 still lies partly outside the graph window, all of the labels in that box may be moved inward, shoving numeric labels ahead of them — onto, and perhaps across, an axis.

The algorithms used to do all of this are not perfect; if pushed too severely, they may fail to produce an esthetically pleasing or even minimally acceptable graph. In such cases, the user will have to take remedial action.

Note: None of the actions described above modify any of the primary control parameters, with the sole exception of the parameter named 'AXIS/s/NUMERIC/ANGLE/1ST.', which may be negated by subtracting a multiple of "360.". Also, no label is shrunk to less than a readable size.

DASHED-LINE PATTERNS FOR CURVES

The subroutine AGCURV draws curves (one per call). It does this by issuing calls to the routines DASHD, FRSTD, VECTD, and LASTD, in the DASHCHAR package. Each curve may thus be drawn using its own particular dashed-line pattern. One of AGCURV's arguments, called KDSH, specifies the dashed-line pattern to be used for a given curve:

If KDSH is zero, the caller is assumed to have done his own call to DASHD. AGCURV does not call it.

If KDSH is non-zero, AGCURV calls DASHD.

If KDSH is negative, its absolute value (modulo 26) specifies one of 26 "alphabetic" patterns. The curve is drawn using a solid line which is interrupted periodically by the selected letter of the alphabet. See example 8 in the AUTOGRAPH examples write-up.

If KDSH is positive, its value (modulo n) specifies one of n "user" patterns, defined by the primary control parameter group named 'DASH.'.
The nature of the "user" set of dashed-line patterns is discussed in the next paragraph.

THE PRIMARY-CONTROL-PARAMETER GROUP NAMED 'DASH.'

The following parameters, in the group 'DASH.', define the "user" set of dashed-line patterns:

The first parameter in this group, named 'DASH/ADDRESS.', specifies the memory address of an array of dashed-line patterns. Each pattern may be defined by a 16-bit integer whose 0 and 1 bits represent "pen-up" and "pen-down" segments three plotter units long or by a character string of arbitrary length in which a quote represents a "pen-up" segment, a dollar sign a "pen-down" segment, and every other character a "draw-me" segment (to be drawn as a part of the line). Note that a call to AGSETI to set the value of 'DASH/ADDRESS.' must have "LOC(ARRAY)", rather than "ARRAY", as its second argument. A call to AGSETF must have "FLOAT(LOC(ARRAY))" as its second argument. Consider also the implications of the fact that only the address of the array (rather than its contents) are saved by AUTOGRAPH. The default value of this parameter is the value "null 1"; see the note below.

The second parameter, named 'DASH/NUMBER.', has the integral floating-point value "n". It specifies the number of patterns in the array. Its default value is "1."

The third parameter, named 'DASH/INDEXING.', has an integral floating-point value specifying the index increment required to move from one pattern in the array to the next. Its default value is "1."

The fourth parameter, named 'DASH/LENGTH.', has an integral floating-point value specifying the number of characters in a character-string pattern. The default value of this parameter is "10." on the 7600, "8." on the CRAY.

The fifth parameter, named 'DASH/CHARACTER.', specifies the curve length taken up by a string character other than a dollar sign or a quote, stated as a fraction of the smaller side of the curve window. The default value of this parameter is ".010".

The sixth parameter, named 'DASH/DOLLAR.', specifies the curve length taken up by a dollar sign (solid) or a quote (gap), stated as a fraction of the smaller side of the curve window. The default value of this parameter is ".010".

Each of the last two parameters affects the appearance of the "alphabetic" set of dashed-line patterns, as well.

Note: If the parameter 'DASH/ADDRESS.' is given a "null" value, the next call to AGSTUP will reset its value to the floated address of a local variable containing the integer 65535 (sixteen one bits), the value of 'DASH/NUMBER.' to a "1.", and the value of 'DASH/INDEXING.' to a "1.", thus specifying the use of solid lines.

See § 15 in the AUTOGRAPH control parameters write-up. See example 7 in the AUTOGRAPH examples write-up.

DASHED-LINE PATTERNS USED BY EZY, EZXY, EZMY, AND EZMXY

Each of the routines EZY and EZXY, which draw one curve per call, calls AGCURV with KDSH equal to 1, specifying the use of the first of the "user-defined" set of dashed-line patterns (default - a solid line) for the single curve to be drawn.

Each of the routines EZMY and EZMXY, which draw one or more curves per call, calls AGCURV with KDSH equal to ISIGN(I,DISH), where I is the number of the curve being drawn and DISH is the integral value of the primary control parameter 'DASH/SELECTOR.'. This parameter has the default value "1.", specifying the use of the "user" set of dashed-line patterns (default - solid lines); it may be set to -1. to specify the use of the "alphabetic" set.

Note: Despite its name, the parameter 'DASH/SELECTOR.' is not a part of the group named 'DASH.' This is the single exception to the rules of naming parameters given above.
AUTOGRAPH PACKAGE

WINDOWING OF CURVES DRAWN BY AUTOGRAPH

The primary control parameter 'WINDOW.' has the default value "0.". If it is set to a "1." by a user program, curves subsequently drawn by the routine AGCURV are "windowed". This means that only those portions lying inside the curve window are drawn; the effect is as if one were viewing the curve through an actual window.

See §7 in the AUTOGRAPH control parameters write-up. See example 7 in the AUTOGRAPH examples write-up.

USE OF PWRX BY AUTOGRAPH

Normally, the routine PWRIT is used for all characters written by AUTOGRAPH. Using PWRX in its normal form poses a problem, for two reasons:

First, if "function codes" are used in the text string, the length of that string is no longer equal to the actual number of characters to be drawn.

Second, PWRX does not use the same plotter width for each character in a line of text.

The first of these problems has not yet been addressed - there is no good way to use anything but the "Principal Roman, upper" character set with AUTOGRAPH. The second problem has been solved, in a clumsy way. To use PWRX with AUTOGRAPH in this clumsy way, apply the string replacement "'PWRIT'='PWRIQ'" to an AUTOGRAPH source file (there are three calls to PWRIT in the file) and include the access cards

*FORTRAN,S=XLIB,SN=AGUSEPWRX
*FORTRAN,S=src,SN=PWRX

where "src" = "ULIB" or "XLIB", in your deck. (Check with a consultant or in the current documentation for an appropriate value of "src").

The routine PWRIQ, from the file AGUSEPWRX, calls PWRX to draw one character at a time, using the same plotter width for each. The results, while not as pleasing as one would normally expect from PWRX, are more than just acceptable.

Remember: Do not use function codes in any text string delivered to AUTOGRAPH. Also, if you use PWRX directly, always leave it in the mode "Principal Roman, upper" when AUTOGRAPH is called.

ENTRY POINTS: Except for seven routines which are included for historical reasons (EZY, EZXY, EZMY, EZXMY, IDIOT, ANOTAT, and DISPLA), the AUTOGRAPH routines have six-character names beginning with "AG". These routines (in alphabetical order) are as follows: AGAXIS, AGBACK, AGCXKO, AGCURV, AGCPYL, AGDASH, AGDFLT, AGEXAX, AGEXUS, AGFTOL, AGGETF, AGGET, AGGETP, AGINT, AGKURV, AGLBS, AGMAXI, AGMINI, AGNUMB, AGQRV, AGSCAN, AGSETF, AGSETI, AGSETP, AGSRCH, AGSTUP, and AGXTX.

Note: The routine "AGDFLT" is a block-data routine containing default values for the primary control parameters and for machine-dependent constants.

SPECIAL CONDITIONS: Under certain conditions, AUTOGRAPH may print an error message (via ULIBER) and stop. Each such error message includes the name of the routine which issued it. A description of the condition which caused the error may be found in the write-up of the named routine, under the heading "SPECIAL CONDITIONS".

For error messages issued by the utility routine AGNUMB, see the write-up of the routine AGSTUP.
COMMON BLOCKS: The AUTOGRAPH common blocks are AGCONP, AGMCHP, and AGIBSP. The first of these (AGCONP) contains the AUTOGRAPH control parameters and is 458 (decimal) words long. The second (AGMCHP) contains machine-dependent constants and is eight words long. The third (AGIBSP) is used for communication between AGSCAN and AGSRCH and is three words long.

I/O: No I/O is done directly by AUTOGRAPH. However, it calls routines in the system plot package to produce graphic output and it may call the routine ULIBER to write error messages to the printer.

REQUIRED ULIB ROUTINES: AUTOGRAPH uses the software dashed-line package DASHCHAR, for which the access card is

\*FORTRAN,S=ULIB,SN=DASHCHAR

The package DASHSMTH or the package DASHSUPR may be used instead.

SPECIALIST: David J. Kennison, NCAR Computing Facility

LANGUAGE: FORTRAN

HISTORY: Dave Robertson wrote the original routine IDIOT, which was intended to provide a simple, "quick-and-dirty" x-y graph-drawing capability. In time, as it became obvious that many users were adapting IDIOT to more sophisticated tasks, Dan Anderson wrote the first AUTOGRAPH package, based on IDIOT. It allowed the user to put more than one curve on a graph, to use more sophisticated backgrounds, to specify coordinate data in a variety of ways, and to more easily control the scaling and positioning of graphs. Eventually, this package, too, was found wanting. In 1977, Dave Kennison entirely re-wrote AUTOGRAPH, with the following goals: to maintain the ease of use for simple graphs which had been the package's principal virtue; to provide the user with as much control as possible; to incorporate desirable new features; to make the package portable.

SPACE REQUIRED: 22,412 (octal) = 9,482 (decimal) 7600 words.

PORTABILITY: AUTOGRAPH may be ported with little change to most systems having a FORTRAN compiler which meets the ANSI-66 standard. Possible problem areas are as follows:

The block-data subroutine AGDFLT contains definitions for eight machine-dependent constants, all of which are declared in the labeled common block AGMCHP. These may or may not have to be changed, depending on the target computer and/or graphics device.

The Hollerith constant "IH$" appears in two different data statements (one in AGDFLT and one in AGCURV). The first of these defines the default 'LINE/END,' character (described in the AUTOGRAPH parameter write-up) and the second defines the code for a solid-line segment in character-string dashed-line patterns. Some compilers will not allow the use of a "$".

The labeled common blocks AGCONP and AGMCHP may have to be declared in a core-resident portion of the user's program so that variables in it will maintain their values between one call to AUTOGRAPH and the next. This problem may arise when AUTOGRAPH is placed in an overlay or when some sort of memory-paging scheme is used on the target machine.

When an AGSETP call is executed to define a mapping function, a dashed-line-pattern array, or the text of a label line, AUTOGRAPH saves only the memory address of the entity defined (obtained...
using the function "LOC". During subsequent calls, out-of-array indexing is used to access the function or array. On a machine with dynamic storage allocation, this scheme would not work. It is difficult to predict what changes would need to be made in AUTOGRAPH. The variables in which addresses are saved are all in the labeled common block AGCONP and are QULM(i), for i from 1 to 4, QADP, and FLLN(4,j), for j from 1 to 18.

A few AUTOGRAPH routines are called with arguments whose type does not match that declared in the routine. This should work unless the compiler passes arguments of different types in different ways, which is unlikely. (?)

AUTOGRAPH assumes that the expression "LOC(A(2))−LOC(A(1))", where A is a real array and LOC is a system-plot-package routine, has the value 1. On systems which assign integers to a single word and reals to more than one word, this assumption may not tally with the assumption made by other portions of the plot package. All references to the function LOC in AUTOGRAPH would have to be modified.

Hollerith label names are treated by AUTOGRAPH as if they were floating-point. If this causes under- or over-flow or some other heinous problem, all user programs should use floating-point numbers for the label "names" and the names of the predefined labels "L", "R", "B", and "T" should be changed to floating-point numbers (in the block data routine AGDFLT).

REQUIRED RESIDENT ROUTINES: AUTOGRAPH uses the DASHCHAR routines DASHD, FRSTD, LASTD, LINED, and VECTD, the system-plot-package routines FLUSH, FRAME, GETSET, GETSI, LINE, OPTN, PWRIT, and SET, the system-plot-package support routines GETCHR, LOC, SETCHR, and ULIBER, and the FORTRAN-library routines ALOG10, ATAN2, COS, SIN, and SQRT.
SUBROUTINE EZY (YDRA, NPTS, GLAB)

DIMENSION OF ARGUMENTS: YDRA(NPTS); GLAB(n), where n is large enough to provide space for the character string, if any, in GLAB.


PURPOSE: To draw, in a manner determined by the current values of the AUTOGRAPH control parameters, a complete graph of a single curve through the points (I,YDRA(I)), for I from 1 to NPTS. The argument GLAB may be used to specify a "graph label", to be placed above the grid portion of the graph.

USAGE: If the default values of the AUTOGRAPH control parameters are unchanged, EZY produces a graph having the following appearance: A "perimeter" background outlines a curve window which is 8/10 the width and 8/10 the height of the plotter frame and positioned slightly above and to the right of center within it. Each edge of the perimeter has short inward-pointing major and minor tick marks, with major tick marks occurring at the ends of each edge. Numeric labels below major tick marks on the bottom edge of the perimeter, increasing in value from left to right, show the linear mapping of values of I onto the horizontal (x) axis of the graph; below them is the label "X". Numeric labels to the left of major tick marks on the left edge of the perimeter, increasing in value from bottom to top, show the linear mapping of values of YDRA(I) onto the vertical (y) axis of the graph; to the left of them is the label "Y". Above the perimeter is the label specified by "GLAB", if any. The curve itself is drawn as a solid line within the perimeter. A frame advance is done after the graph is drawn.

See example 1 in the AUTOGRAPH examples write-up.

The appearance of a graph drawn by EZY may be changed drastically by changing the values of AUTOGRAPH control parameters. See the write-ups of the routines ANOTAT, DISPLAY, and AGSETP and the AUTOGRAPH parameter write-up.

ARGUMENTS: The three arguments of EZY are as follows:

(1) YDRA is a one-dimensional array of NPTS floating-point numbers, each of which defines the user coordinates of a point (FLOAT(I),YDRA(I)) on the desired curve. The current value of the parameter 'NULL/1.' (default value "1.E36") may be used in YDRA to signal missing points; curve segments on either side of a missing point are omitted.

(2) NPTS is the number of curve points defined by the array YDRA.

(3) GLAB is either an array whose first element is "0.", specifying that no change is to be made in the current "graph label", or an array containing a character string defining a new "graph label".
A character string used must either be of the exact length specified by the current value of the parameter 'LINE/MAXIMUM.' (default - 40 characters), or shorter; if shorter, it must be terminated by the character defined by the current value of the parameter 'LINE/END.' (default - '$'). The character string is copied to an array local to AUTOGRAPH (shared by all of the "EZ..." routines) and used as a label above the curve window. (It is defined to be line number "100." of the label named "T".)

Note: On the 7600 and the CRAY, an integer "0", a floating-point "0.", or an integer array may be used for "GLAB"; this is, however, a violation of the ANSI-66 standard.

SPECIAL CONDITIONS: None.

COMMON BLOCKS: None.

SPACE REQUIRED: 77 (octal) = 63 (decimal) 7600 words.

TIMING: Variable (about 1/10 second minimum on the 7600).
SUBROUTINE EZXY (XDRA, YDRA, NPTS, GLAB)

DIMENSION OF ARGUMENTS: XDRA(NPTS); YDRA(NPTS); GLAB(n), where n is large enough to provide space for the character string, if any, in GLAB.


PURPOSE: To draw, in a manner determined by the current values of the AUTOGRAPH control parameters, a complete graph of a single curve through the points (XDRA(I),YDRA(I)), for I from 1 to NPTS. The argument GLAB may be used to specify a "graph label", to be placed above the grid portion of the graph.

USAGE: If the default values of the AUTOGRAPH control parameters are unchanged, EZXY produces a graph having the following appearance: A "perimeter" background outlines a curve window which is 8/10 the width and 8/10 the height of the plotter frame and positioned slightly above and to the right of center within it. Each edge of the perimeter has short inward-pointing major and minor tick marks, with major tick marks occurring at the ends of each edge. Numeric labels below major tick marks on the bottom edge of the perimeter, increasing in value from left to right, show the linear mapping of values of XDRA(1) onto the horizontal (x) axis of the graph; below them is the label "X". Numeric labels to the left of major tick marks on the left edge of the perimeter, increasing in value from bottom to top, show the linear mapping of values of YDRA(1) onto the vertical (y) axis of the graph; to the left of them is the label "Y". Above the perimeter is the label specified by "GLAB", if any. The curve itself is drawn as a solid line within the perimeter. A frame advance is done after the graph is drawn.

See example 2 in the AUTOGRAPH examples write-up.

The appearance of a graph drawn by EZXY may be changed drastically by changing the values of AUTOGRAPH control parameters. See the write-ups of the routines ANOTAT, DISPLA, and AGSETP and the AUTOGRAPH parameter write-up.

ARGUMENTS: The four arguments of EZXY are as follows:

(4) XDRA is a one-dimensional array of NPTS floating-point numbers, defining the x coordinates of points on the curve.

(5) YDRA is a one-dimensional array of NPTS floating-point numbers, defining the y coordinates of points on the curve.

The points on the curve have coordinates (XDRA(I),YDRA(I)), for I from 1 to NPTS. The current value of the parameter 'NULL/1.' (default value "1.E36") may be used to signal missing data in these arrays. If either coordinate of a point is missing, the point is considered to be missing; curve segments on either side of a missing point are not drawn.
(6) NPTS is the number of curve points defined by the arrays XDRA and YDRA.

(7) GLAB is either an array whose first element is "0.", specifying that no change is to be made in the current "graph label", or an array containing a character string defining a new "graph label".

A character string used must either be of the exact length specified by the current value of the parameter 'LINE/MAXIMUM.' (default - 40 characters), or shorter; if shorter, it must be terminated by the character defined by the current value of the parameter 'LINE/END.' (default - '$'). The character string is copied to an array local to AUTOGRAPH (shared by all of the "EZ..." routines) and used as a label above the curve window. (It is defined to be line number "100." of the label named "T".)

Note: On the 7600 or the CRAY, an integer "0", a floating-point "0.", or an integer array may be used for "GLAB"; this is, however, a violation of the ANSI-66 standard.

SPECIAL CONDITIONS: None.

COMMON BLOCKS: None.

SPACE REQUIRED: 100 (octal) = 64 (decimal) 7600 words.

TIMING: Variable (about 1/10 second minimum on the 7600).
**SUBROUTINE EZMY (YDRA, IDXY, MANY, NPTS, GLAB)**

**DIMENSION OF ARGUMENTS:** YDRA(IDXY,MANY), where IDXY ≥ NPTS, or YDRA(IDXY,NPTS), where IDXY ≥ MANY, depending on the current value of the AUTOGRAPH parameter named 'ROW.'; GLAB(n), where n is large enough to provide space for the character string, if any, in GLAB.

**LATEST REVISION:** April, 1978.

**PURPOSE:** To draw, in a manner determined by the current values of the AUTOGRAPH control parameters, a complete graph of one or more curves, each defined by a set of points (I,YDRA(I,J)) [or (I,YDRA(J,I)), depending on the current value of the AUTOGRAPH parameter 'ROW.'], for I from 1 to NPTS. The curve number J runs from 1 to MANY. The argument GLAB may be used to specify a "graph label", to be placed above the grid portion of the graph.

**USAGE:** If the default values of the AUTOGRAPH control parameters are unchanged, EZMY produces a graph having the following appearance: A "perimeter" background outlines a curve window which is 8/10 the width and 8/10 the height of the plotter frame and positioned slightly above and to the right of center within it. Each edge of the perimeter has short inward-pointing major and minor tick marks, with major tick marks occurring at the ends of each edge. Numeric labels below major tick marks on the bottom edge of the perimeter, increasing in value from left to right, show the linear mapping of values of I onto the horizontal (x) axis of the graph; below them is the label "X". Numeric labels to the left of major tick marks on the left edge of the perimeter, increasing in value from bottom to top, show the linear mapping of values of YDRA(I,J) onto the vertical (y) axis of the graph; to the left of them is the label "Y". Above the perimeter is the label specified by "GLAB", if any. The curves themselves are drawn as solid lines within the perimeter. A frame advance is done after the graph is drawn.

See example 3 in the AUTOGRAPH examples write-up.

The appearance of a graph drawn by EZMY may be changed drastically by changing the values of AUTOGRAPH parameters. See the write-ups of the routines ANOTAT, DISPLA, and AGSETP and the AUTOGRAPH parameter write-up.

**ARGUMENTS:** The five arguments of EZMY are as follows:

1. **YDRA** is a two-dimensional array of curve-point y coordinates. The current value of the control parameter 'NULL./1.' (default value '1.E36') may be used in YDRA to signal missing points; curve segments on either side of a missing point are omitted.

   If the control parameter 'ROW.' has a positive value (the default case), the first subscript of YDRA is a point number and the second is a curve number. If 'ROW.' has a negative value, the order of the subscripts is reversed ("row-wise", as opposed to "column-wise", storage).

2. **IDXY** is the first dimension of the array YDRA, required by EZMY in order to index the array properly.

   If the control parameter 'ROW.' has a positive value (the default case), IDXY must be greater than or equal to NPTS. If 'ROW.' has a negative value, IDXY must be greater than or equal to MANY.

3. **MANY** is the number of curves to be drawn by EZMY.
NPTS is the number of points defining each curve to be drawn by EZMY.

GLAB is either an array whose first element is "0.", specifying that no change is to be made in the current "graph label", or an array containing a character string defining a new "graph label".

A character string used must either be of the exact length specified by the current value of the parameter 'LINE/MAXIMUM.' (default - 40 characters), or shorter; if shorter, it must be terminated by the character defined by the current value of the parameter 'LINE/END.' (default - "$\). The character string is copied to an array local to AUTOGRAPH (shared by all of the "EZ..." routines) and used as a label above the curve window. (It is defined to be line number "100." of the label named "T").

Note: On the 7600 and the CRAY, an integer "0", a floating-point "0.", or an integer array may be used for "GLAB"; this is, however, a violation of the ANSI-66 standard.

SPECIAL CONDITIONS: None.

COMMON BLOCKS: None.

SPACE REQUIRED: 127 (octal) = 87 (decimal) 7600 words.

TIMING: Variable (about 1/10 second minimum on the 7600).
SUBROUTINE EZMXY (XDRA, YDRA, IDXY, MANY, NPTS, GLAB)

DIMENSION OF ARGUMENTS: XDRA(NPTS) or XDRA(IDXY,MANY), where IDXY > NPTS, or XDRA(IDXY,NPTS), where IDXY > MANY, depending on the current value of the AUTOGRAPH control parameter 'ROW.'; YDRA(IDXY,MANY), where IDXY > NPTS, or YDRA(IDXY,NPTS), where IDXY > MANY, depending on the current value of the AUTOGRAPH control parameter 'ROW.'; GLAB(n), where n is large enough to provide space for the character string, if any, in GLAB.


PURPOSE: To draw, in a manner determined by the current values of the AUTOGRAPH control parameters, a complete graph of one or more curves, each defined by a set of points (XDRA(I),YDRA(I,J)) [or (XDRA(I),YDRA(J,I)) or (XDRA(I,J),YDRA(I,J)) or (XDRA(J,I),YDRA(J,I)), depending on the current value of the AUTOGRAPH control parameter 'ROW.'], for I from 1 to NPTS. The curve number J runs from 1 to MANY. The argument GLAB may be used to specify a "graph label", to be placed above the grid portion of the graph.

USAGE: If the default values of the AUTOGRAPH control parameters are unchanged, EZMXY produces a graph having the following appearance: A perimeter background outlines a curve window which is 8/10 the width and 8/10 the height of the plotter frame and positioned slightly above and to the right of center within it. Each edge of the perimeter has short inward-pointing major and minor tick marks, with major tick marks occurring at the ends of each edge. Numeric labels below major tick marks on the bottom edge of the perimeter, increasing in value from left to right, show the linear mapping of values of XDRA(I) onto the horizontal (x) axis of the graph; below them is the label "X". Numeric labels to the left of major tick marks on the left edge of the perimeter, increasing in value from bottom to top, show the linear mapping of values of YDRA(I,J) onto the vertical (y) axis of the graph; to the left of them is the label "Y". Above the perimeter is the label specified by "GLAB", if any. The curves themselves are drawn as solid lines within the perimeter. A frame advance is done after the graph is drawn.

See example 4 in the AUTOGRAPH examples write-up.

The appearance of a graph drawn by EZMXY may be changed drastically by changing the values of AUTOGRAPH control parameters. See the write-ups of the routines ANOTAT, DISPLAY, and AGSETP, and the AUTOGRAPH parameter write-up.

ARGUMENTS: The six arguments of EZMXY are as follows:

(13) XDRA is a one-dimensional or two-dimensional array of curve-point x coordinates. The current value of the AUTOGRAPH control parameter 'NULL/1.' (default value "1.E36") may be used in XDRA to signal missing points; curve segments on either side of a missing point are not drawn.

If the AUTOGRAPH control parameter 'ROW.' has the absolute value "1." (the default case), XDRA is singly-dimensioned. It is subscripted by point number.

If 'ROW.' has the absolute value "2." or greater, XDRA is doubly-dimensioned. It is subscripted by point number and curve number, in that order if 'ROW.' is positive (the default case), in the reverse order if 'ROW.' is negative.

(14) YDRA is a two-dimensional array of curve-point y coordinates. The current value of the parameter 'NULL/1.' (default value - "1.E36") may be used in YDRA to signal missing points; curve segments on either side of a missing point are not drawn.
If the AUTOGRAPH control parameter 'ROW.' has a positive value (the default case), YDRA is subscripted by point number and curve number, in that order; if 'ROW.' has a negative value, YDRA is subscripted by curve number and point number, in that order.

(15) IDXY is the first dimension of the arrays XDRA (if it is doubly-dimensioned) and YDRA (unconditionally), required by EZMXY in order to index these arrays properly.

If the AUTOGRAPH control parameter 'ROW.' has a positive value (the default case), IDXY must be greater than or equal to NPTS. If 'ROW.' has a negative value, IDXY must be greater than or equal to MANY.

(16) MANY is the number of curves to be drawn by EZMXY.

(17) NPTS is the number of points defining each curve to be drawn by EZMXY.

(18) GLAB is either an array whose first element is "0.", specifying that no change is to be made in the current "graph label", or an array containing a character string defining a new "graph label".

A character string used must either be of the exact length specified by the current value of the parameter 'LINE/MAXIMUM.' (default - 40 characters) or shorter; if shorter, it must be terminated by the character defined by the current value of the parameter 'LINE/END.' (default - '$'). The character string is copied to an array local to AUTOGRAPH (shared by all of the "EZ..." routines) and used as a label above the curve window. (It is defined to be line number "100." of the label named 'T'.)

Note: On the 7600 and the CRAY, an integer "0", a floating-point "0.", or an integer array may be used for "GLAB"; this is, however, a violation of the ANSI-66 standard.

SPECIAL CONDITIONS: None.

COMMON BLOCKS: None.

SPACE REQUIRED: 136 (octal) = 94 (decimal) 7600 words.

TIMING: Variable (about 1/10 second minimum on the 7600).
SUBROUTINE ANOTAT (XLAB, YLAB, LBAC, LSET, NDSH, DSHL)

DIMENSION OF ARGUMENTS: XLAB(m), where m is large enough to provide space for the character string, if any, in XLAB; YLAB(n), where n is large enough to provide space for the character string, if any, in YLAB; DSHL(NDSH), if NDSH is positive (otherwise, DSHL is a dummy argument).


PURPOSE: To change the values of certain AUTOGRAPH parameters, purportedly having to do with "annotation" of a graph.

USAGE: The routine ANOTAT is provided principally for historical reasons. Each of the AUTOGRAPH parameters referenced by its argument list can be set individually by means of the routines AGSETP, AGSETF, and/or AGSETI. In fact, ANOTAT is implemented using calls to these routines.

See example 8 in the AUTOGRAPH examples write-up.

ARGUMENTS: The six arguments of ANOTAT are as follows:

(19) XLAB is either an array whose first element is "0.", specifying that no change is to be made in the current "x-axis label", or an array containing a character string defining a new "x-axis label".

A character string used must either be of the exact length specified by the current value of the parameter 'LINE/MAXIMUM.' (default = 40 characters) or shorter; if shorter, it must be terminated by the character defined by the current value of the parameter 'LINE/END.' (default = ".") The character string is copied to an array local to AUTOGRAPH (shared by all of the "EZ..." routines) and used as a label below the curve window. (It is defined to be line number "-100." of the label named 'B'.)

Note: On the 7600 and the CRAY, an integer "0", a floating-point "0.", or an integer array may be used for "XLAB"; this is, however, a violation of the ANSI-66 standard.

(20) YLAB is defined in a manner analogous to XLAB, except that the character string, if any, defines a new "y-axis label" to the left of the curve window. (It is defined to be line number "100." of the label named 'L'.)

(21) LBAC, if non-zero, has the integer value 1, 2, 3, or 4, the floating-point equivalent of which is to be stored (by a call to AGSETI) as the new value of the AUTOGRAPH control parameter 'BACKGROUND'. If LBAC is zero, no change is to be made in the current value of 'BACKGROUND'.

The value of 'BACKGROUND.' is changed to specify the use of a particular type of background. The value "1." (the default) specifies a perimeter background, the value "2." a grid background, the value "3." a half-axis background, and the value "4." no background at all.

See 'BACKGROUND.', in the AUTOGRAPH parameter write-up, for a description of the way in which AUTOGRAPH creates these backgrounds.

(22) LSET, if non-zero, is an integer having the absolute value 1, 2, 3, or 4, the floating-point equivalent of which is to be stored (by a call to AGSEFI) as the new value of the AUTOGRAPH control parameter 'SET'. If LSET is zero, no change is to be made in the current value of 'SET'.

The default value of 'SET.' is "1." Both its sign and magnitude are meaningful, as follows:
Giving 'SET.' a negative value suspends the drawing of curves by the routines EZY, EZXY, EZMY, and EZMXY. A call to one of these routines produces at most a background when 'SET.' is negative.

The absolute value of 'SET.' affects the positioning of the curve window (grid) within the plotter frame and/or the mapping of user coordinate data into the curve window, as follows:

The act of giving 'SET.' the absolute value "1." causes the parameters in the groups 'GRID.', 'X.', and 'Y.' to be reset to their default values, but implies no action beyond that. The parameter 'X/LOGARITHMIC.' is reset to its default value ('0.') only if it has the value "+1."; a value "-1." is not changed. The parameter 'Y/LOGARITHMIC.' is treated similarly.

The act of giving 'SET.' the absolute value "2." or greater causes no immediate action, but implies a particular action by subsequent calls to the routine AGSTUP, which will use values obtained from a call to the system-plot-package routine GETSET to modify parameters in the groups 'GRID.', 'X.', and/or 'Y.'. The parameters 'X/LOGARITHMIC.' and 'Y/LOGARITHMIC.' may be given values "-1." or "0.", but not "+1.", as a result.

The absolute value "2." specifies that arguments 1 through 4 of the GETSET call should determine the position and shape of the curve window (grid) and that argument 9 should determine the linear-log nature of the user-coordinate mapping.

The absolute value "3." specifies that arguments 5 through 9 of the GETSET call should determine the mapping of user x/y coordinate data into the curve window.

The absolute value "4." specifies that arguments 1 through 9 of the GETSET call should be used (a combination of the actions specified by the values "2." and "3.").

See 'SET.', in the AUTOGRAPH parameter write-up, for further details.

NDSH, if zero, specifies that no change is to be made in the AUTOGRAPH control parameters which specify the dashed-line patterns to be used for curves. If NDSH is non-zero, it specifies an integer value whose floating-point equivalent is to be stored as the new value of the parameter 'DASH/SELECTOR.' (which has the default value "1.").

If NDSH is negative, 'DASH/SELECTOR.' is set negative, forcing EZMY and EZMXY to use internally-defined "alphabetic" patterns for the MANY curves drawn ("A" for the first, "B" for the second, ..., "Z" for the 26th, "A" for the 27th, etc.). The routines EZY and EZXY are unaffected.

If NDSH is greater than zero, it must be less than or equal to 26, and the next argument, DSHL, must contain NDSH dashed-line patterns comprising the new "user" set of patterns. The parameter 'DASH/SELECTOR.' is set positive, forcing EZMY and EZMXY to use this set of patterns. (The routines EZY and EZXY always use the first pattern in this set.) The contents of the array DSHL are copied to an array local to AUTOGRAPH and the address of that array is stored as the value of the parameter 'DASH/ADDRESS.. NDSH is stored as the value of the parameter 'DASH/NUMBER.' The value of 'DASH/INDEX.' is set to "1.".

See 'DASH.', in the AUTOGRAPH parameter write-up, for further details.

DSHL is meaningful only when NDSH is greater than zero. In this case, it must be an array of dashed-line patterns, stored one per floating-point word. Each of these patterns is either a 16-bit integer in which "0" bits are "pen-up" signals and "1" bits are "pen-down" signals or a character string in which dollar signs are "pen-down" signals, quotes are "pen-up" signals, and other characters are "draw-me" signals.

See 'DASH.', in the AUTOGRAPH parameter write-up, for further details.
SPECIAL CONDITIONS: None.

COMMON BLOCKS: None.

SPACE REQUIRED: 403 (octal) = 259 (decimal) 7600 words.

TIMING: Negligible.
SUBROUTINE DISPLA (LFRA, LROW, LTYP)

DIMENSION OF ARGUMENTS: None of the arguments are dimensioned.


PURPOSE: To change the values of certain AUTOGRAPH control parameters, purportedly having to
          do with the "display" of a graph.

USAGE: The routine DISPLA is provided principally for historical reasons. Each of the AUTOGRAPH
          control parameters referenced by its argument list can be set individually by means of the
          routines AGSETP and/or AGSETI. In fact, DISPLA is implemented using calls to these rou-
          tines.

          See the final example in the AUTOGRAPH examples write-up.

ARGUMENTS: The three arguments of DISPLA are as follows:

(25) LFRA, if non-zero, has an integer value, the floating-point equivalent of which is to become
        the new value of the parameter 'FRAME.'. If LFRA is zero, no change is to be made in the
        current value of 'FRAME'.

        The default value of 'FRAME.' is "1.", specifying that each of the routines EZY, EZXY,
        EZMY, and EZMXY is to do a frame advance after drawing a graph. The value "2." specifies
        that these routines should do no frame advance, the value "3." that they should do a
        frame advance before drawing a graph.

(26) LROW, if non-zero, has an integer value, the floating-point equivalent of which is to
        become the new value of the parameter 'ROW.'. If LROW is zero, no change is to be made
        in the current value of 'ROW'.

        This parameter affects the way in which the routines EZMY and EZMXY interpret the argu-
        ments XDRA and YDRA, as follows:

        If 'ROW.' is positive, the first subscript of YDRA is a point number and the second sub-
        script is a curve number. If 'ROW.' is negative, the order of the subscripts is reversed
        (row-wise, rather than column-wise, storage).

        If the absolute value of 'ROW.' is "1.", XDRA is singly-subscripted; its subscript is a
        point number. If the absolute value of 'ROW.' is "2." or greater, XDRA is doubly-
        subscripted; the order of the subscripts is the same as for YDRA.

        The default value of 'ROW.' is "1.", specifying that XDRA is singly-subscripted by point
        number and that YDRA is doubly-subscripted by point number and curve number, in that
        order.

(27) LTYP, if non-zero, is an integer specifying new values for the AUTOGRAPH control parame-
        ters 'X/LOGARITHMIC.' and 'Y/LOGARITHMIC.'. If LTYP is zero, no change is to be made in
        the current values of these two parameters.

        The parameter 'X/LOGARITHMIC.' has the default value "0.", specifying a linear mapping of
        user x coordinates onto the horizontal axis of the curve window; it may be given either of
        the values "-1." or "+1." to specify a logarithmic mapping, instead. The value "-1." pro-
        tects the parameter from being reset by the action of setting 'SET.' to "±1.".

        The parameter 'Y/LOGARITHMIC.' is defined similarly and affects the mapping of user y
        coordinates onto the vertical axis of the curve window.

        A non-zero LTYP resets these two parameters as follows:
AUTOGRAPH PACKAGE

LTYP 'X/LOGARITHMIC.' 'Y/LOGARITHMIC.'

<table>
<thead>
<tr>
<th></th>
<th>0. (linear)</th>
<th></th>
<th>0. (linear)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0. (linear)</td>
<td>2</td>
<td>-1. (logarithmic)</td>
</tr>
<tr>
<td>3</td>
<td>-1. (logarithmic)</td>
<td>4</td>
<td>-1. (logarithmic)</td>
</tr>
</tbody>
</table>

SPECIAL CONDITIONS: None.

COMMON BLOCKS: None.

SPACE REQUIRED: 73 (octal) = 59 (decimal) 7600 words.

TIMING: Negligible.
SUBROUTINE AGSETP (TPGN, FURA, LURA)

DIMENSION OF ARGUMENTS: TPGN(m), where m is large enough to provide space for the character string in TPGN; FURA(LURA).


PURPOSE: To allow a user program to set the values of a group of AUTOGRAPH control parameters. The group might contain only one element.

USAGE:

The routine AGSETP is called with a character string identifying a group of AUTOGRAPH parameters (possibly a single parameter), an array containing new values of those parameters, and the length of the array, as arguments. It transfers the new values into the appropriate locations in the labeled common block AGCONP, thus modifying the effect of subsequent calls to other AUTOGRAPH routines.

If a single parameter is being set, the routine AGSETF or AGSETI (which see, below) may be used instead. There is one exception: the parameter 'LINE/DEFINITION/TEXT.' must be set using AGSETP.

When certain control parameters are set individually, AGSETP takes further "special" action. For example, if the parameter 'BACKGROUND.' is set, thereby requesting one of the standard types of backgrounds, AGSETP changes a number of other parameters in order to achieve the desired effect.

The parameters which imply such special action are as follows:

'SET.'
'BACKGROUN.
'NULL/1.' and 'NULL/2.'
'LABEL/CONTROL.'
'LABEL/NAME.'
'LABEL/DEFINITION/SUPPRESSION.'
'LINE/NUMBER.'
'LINE/DEFINITION/SUPPRESSION.'

ARGUMENTS: The three arguments of AGSETP are as follows:

1) TPGN is a character string of the form 'k(1)/k(2)/... k(n).', where each of the k(i)'s is a keyword. The keyword k(1) specifies a group of AUTOGRAPH control parameters, k(2) a subgroup of that group, k(3) a subgroup of that subgroup, etc. The whole string is the name of some group of AUTOGRAPH parameters the user wishes to set. For example, 'AXIS.' is the name of a 62-word group of parameters describing the four AUTOGRAPH axes, 'AXIS/RIGHT.' is the name of a 23-word subgroup describing the right y axis, 'AXIS/RIGHT/INTERSECTION.' is the name of a 2-word further subgroup describing the intersection of the right y axis with the bottom of the curve window, and 'AXIS/RIGHT/INTERSECTION/USER.' is the name of a single parameter specifying the point of intersection of the right y axis with the bottom of the curve window as an x coordinate in the user coordinate system.

Obviously, these names can sometimes become rather long. There are various ways in which they may be shortened. First, since the fifth and following characters of each keyword are ignored, they may be omitted; this would shorten 'AXIS/RIGHT/INTERSECTION/USER.' to 'AXIS/RH/INT/E/USER.'. Even fewer characters may be used, as long as no ambiguity of interpretation arises. To be completely safe, use at least the first three characters of the group keyword and at least the first two characters of each subgroup keyword; this would shorten 'AXIS/RH/INT/E/USER.' to 'AX/RI/IN/US.'. Moreover, certain group and subgroup keywords may be omitted entirely; for example, 'AX/RI/IN/US.' may be shortened to 'RI/IN/US.'.
Names may also be lengthened in various ways in order to improve their readability. Blanks may be used as desired on either side of a keyword. Any sequence of characters not including a slash or a period may be inserted after a keyword, separated from it by at least one blank. For example, the name

'DASH PATTERN / CHARACTER WIDTH .'

is equivalent to, and considerably more meaningful than, 'DAS/CH.' or even 'DASH/CHARACTER.'.

A complete list of the AUTOGRAPH control parameters may be found in the AUTOGRAPH parameter write-up, below.

(2) FURA is a user array (of length LURA) containing the new values of the AUTOGRAPH control parameters in the group specified by TPGN, in the same order as they appear in the group.

All of the AUTOGRAPH control parameters have floating-point values (because of a portability problem which arose in implementing this routine). This leads to a few peculiarities. The parameters with names of the form 'AXIS/s/FUNCTION.', where "s" represents one of the keywords "LEFT", "RIGHT", "BOTTOM", or "TOP", the parameter 'DASH/ADDRESS.', and the parameter 'LINE/DEFINITION/TEXT.' all have values of the form "FLOAT(LOC(X))", where "X" is either an array name or a subroutine name. The parameters in the group 'LABEL/BUFFER/NAMES.' are single-word Hollerith strings which are used as (floating-point) label names. The parameter 'LINE/END.' is a single-word Hollerith string, of which only the first character is meaningful. The parameter names 'LABEL/NAME.' and 'LINE/DEFINITION/TEXT.' are used in AGSETP calls with a character string as the second argument, but the effect of the call is to store a floating-point number.

(3) LURA is the length of FURA (the number of floating-point elements in it). Its value may be less than, equal to, or greater than, the length of the AUTOGRAPH control parameter group specified by TPGN. The number of parameter values transferred from FURA is the minimum of the two (but not less than one). This means that if, for example, you only wish to set the first two parameters of a 100-parameter group, you may do so by using LURA = 2.

SPECIAL CONDITIONS:

A call to AGSETP may result in the printing of an error message and the execution of a "STOP". These error messages are as follows:

AGSETP - LABEL LIST OVERFLOW - SEE AUTOGRAPH SPECIALIST

The user has attempted to define more labels than AUTOGRAPH can handle. The paragraph "INFORMATIONAL LABELS" in the "USAGE" section of the AUTOGRAPH package write-up, above, gives a string replacement which, when applied to an AUTOGRAPH source file, increases the maximum number of labels one can define.

AGSETP - ATTEMPT TO DEFINE LINE OF NON-EXISTENT LABEL

The user has attempted to define a line of a label without first specifying the label. An AGSETP (or AGSETF) call setting the parameter 'LABEL/NAME.' must be inserted prior to the AGSETP call which gave the error message.

AGSETP - LINE LIST OVERFLOW - SEE AUTOGRAPH SPECIALIST

The user has attempted to define more label lines than AUTOGRAPH can handle. The paragraph "INFORMATIONAL LABELS" in the "USAGE" section of the AUTOGRAPH package write-up, above, gives a string replacement which, when applied to an AUTOGRAPH source file, increases the maximum number of lines one can define.
AGGETP OR AGSETP – ILLEGAL KEYWORD USED IN PARAMETER IDENTIFIER

A parameter-group name contains an unrecognizable keyword.

AGGETP OR AGSETP – LABEL BUFFER TOO SMALL

The string replacement used to change the maximum number of labels AUTOGRAPH can handle must not be used to reduce that number to fewer than five without also making further changes in the AUTOGRAPH source.

AGGETP OR AGSETP – ATTEMPT TO ACCESS LABEL ATTRIBUTES BEFORE SETTING LABEL NAME

An AGSETP (or AGSETF) call setting the parameter 'LABEL/DNAME.' must be inserted prior to the call which gave the error message, specifying which label's attributes are being referenced.

AGGETP OR AGSETP – LINE BUFFER TOO SMALL

The string replacement used to change the maximum number of label lines AUTOGRAPH can handle must not be used to reduce that number to fewer than five without also making further changes in the AUTOGRAPH source.

AGGETP OR AGSETP – ATTEMPT TO ACCESS LINE ATTRIBUTES BEFORE SETTING LINE NUMBER

An AGSETP (or AGSETI) call setting the parameter 'LINE/DNUMBER.' must be inserted prior to the AGSETP call which gave the error message, specifying which label line's attributes are being referenced.

COMMON BLOCKS: AGCONP.

SPACE REQUIRED: 730 (octal) = 472 (decimal) 7600 words.

TIMING: Negligible.
SUBROUTINE AGSETF (TPGN, FUSR)

DIMENSION OF ARGUMENTS: TPGN(m), where m is large enough to provide space for the character string in TPGN.


PURPOSE: To allow a user program to store a floating-point number as the value of a single AUTOGRAPH control parameter.

USAGE: This routine transfers the contents of FUSR to a local array FURA, dimensioned 1, and executes the statement

CALL AGSETF (TPGN, FURA, 1)

This routine must not be used to set the value of the parameter 'LINE/DEFINITION/TEXT.'

for which AGSETF must be used. See the AGSETF write-up, above.

COMMON BLOCKS: None.

SPACE REQUIRED: 24 (octal) = 20 (decimal) 7600 words.

TIMING: Negligible.
SUBROUTINE AGSETI (TPGN, IUSR)

DIMENSION OF ARGUMENTS: TPGN(m), where m is large enough to provide space for the character string in TPGN.


PURPOSE: To allow a user program to store the floating-point equivalent of an integer as the value of a single AUTOGRAPH control parameter.

USAGE: This routine stores the value FLOAT(IUSR) in a local array FURA, dimensioned 1, and then executes the statement

CALL AGSETP (TPGN, FURA, 1)

See the write-up of AGSETP for further details.

COMMON BLOCKS: None.

SPACE REQUIRED: 24 (octal) = 20 (decimal) 7600 words.

TIMING: Negligible.
SUBROUTINE AGGETP (TPGN, FURA, LURA)

DIMENSION OF ARGUMENTS: TPGN(m), where m is large enough to provide space for the character string in TPGN; FURA(LURA).


PURPOSE: To allow a user program to retrieve the current values of a group of AUTOGRAPH control parameters. The group might have only one element.

USAGE: The routine AGGETP is called in exactly the same way as is AGSETP, the object being to "get" parameter values, rather than to "set" them. The routines AGGETP and AGGETI may be used to retrieve the value of any single parameter. See their write-ups, below.

No "special" action is implied for any single parameter.

See the AUTOGRAPH parameter write-up for descriptions of the parameters whose values may be retrieved.

ARGUMENTS: Same as for AGSETP. See the write-up of AGSETP, above.

SPECIAL CONDITIONS: See the AGSETP write-up. Those error messages beginning "AGGETP OR AGSETP - " apply to AGGETP.

COMMON BLOCKS: AGCONP.

SPACE REQUIRED: 50 (octal) = 40 (decimal) 7600 words.

TIMING: Negligible.
SUBROUTINE AGGETF (TPGN,FUSR)

DIMENSION OF ARGUMENTS: TPGN(m), where m is large enough to provide space for the character string in TPGN.


PURPOSE: To allow a user program to retrieve the floating-point value of a single AUTOGRAPH control parameter.

USAGE: This routine executes the statement

CALL AGGETP (TPGN,FURA,1)

where FURA is a local array, dimensioned 1, and then sets FUSR equal to FURA(1). See the write-up of AGGETP for further details.

COMMON BLOCKS: None.

SPACE REQUIRED: 24 (octal) = 20 (decimal) 7600 words.

TIMING: Negligible.
SUBROUTINE AGGETI (TPGN,IUSR)

DIMENSION OF ARGUMENTS: TPGN(m), where m is large enough to provide space for the character string in TPGN.


PURPOSE: To allow a user program to retrieve the integer equivalent of the floating-point value of a single AUTOGRAPH control parameter.

USAGE: This routine executes the statement

CALL AGGETP (TPGN,FURA,1)

where FURA is a local array, dimensioned 1, and then sets IUSR equal to IFIX(FURA(1)). See the write-up of AGGETP for further details.

COMMON BLOCKS: None.

SPACE REQUIRED: 24 (octal) = 20 (decimal) 7600 words.

TIMING: Negligible.
SUBROUTINE AGSTUP (XDRA, NV, IVX, NEVX, IEVX, YDRA, NVY, IVY, NEVY, NEY)

DIMENSION OF ARGUMENTS: XDRA(1+(NV-1)*IVX+(NEVX-1)*IEVX);
                         YDRA(1+(NVY-1)*IVY+(NEVY-1)*NEY).


PURPOSE: To perform "set-up" tasks required before AGBACK and AGCURV may be called. Basically, AGSTUP examines the current values of the "primary" AUTOGRAPH control parameters for errors and computes from them and from its arguments the values of "secondary" AUTOGRAPH control parameters. The primary and secondary control parameters together determine how the routines AGBACK and AGCURV will behave. See the AUTOGRAPH parameter write-up for descriptions of all of the control parameters.

USAGE: The routine AGSTUP is normally called once per graph, just prior to the sequence of calls to AGBACK and/or AGCURV which actually draws the graph.

The call to AGSTUP may be omitted only if (1) no primary AUTOGRAPH control parameters have been changed since the last time AGSTUP was called and (2) the position of the curve window (grid) and the mapping of user x/y coordinates into the curve window would be unaffected by the AGSTUP call. The routine must be called at least once – for the first graph; for succeeding graphs, if the call can be omitted, it should be, since the routine is rather time-consuming.

Note that each of the routines EZY, EZXY, EZMY, and EZMXY unconditionally executes a call to AGSTUP (via a routine called AGINIT) before calling AGBACK and/or AGCURV.

ARGUMENTS:

The first five arguments of AGSTUP are meaningful only if at least one of the primary AUTOGRAPH control parameters 'X/Minum.' and 'X/Maximum.' has the value "null 1" or "null 2", specifying that AUTOGRAPH is to determine for itself the minimum and/or maximum x coordinate in the user's data. Similarly, the second five arguments are meaningful only if at least one of the parameters 'Y/Minimum.' and 'Y/Maximum.' has the value "null 1" or "null 2". The first five arguments are as follows:

1. XDRA is an array of user x coordinates.
2. NVX is the number of "vectors" of data from XDRA to be considered in computing the minimum and/or maximum x values.
3. IVX is the index increment between two "vectors" in XDRA. The first element of the first vector is XDRA(1), the first element of the second vector is XDRA(1+IVX), the first element of the third vector is XDRA(1+IVX*2), etc.
4. NEVX is the number of elements of each vector in XDRA to be considered in computing the minimum and/or maximum x values.
5. IEX is the index increment between two consecutive elements of a vector in XDRA. The second element of the first vector is XDRA(1+IEX), the third element of the first vector is XDRA(1+IEX*2), etc.

If IEX has the value 0, the contents of XDRA are ignored completely; the minimum and maximum x values are considered to be "1." and FLOAT(NEVX), respectively.
Some examples showing how to set up these arguments:

<table>
<thead>
<tr>
<th>x array</th>
<th>data to be used</th>
<th>XDRA</th>
<th>NVIX</th>
<th>IIVX</th>
<th>NEVX</th>
<th>IEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>X(100) all</td>
<td>X(1)</td>
<td>1</td>
<td>-</td>
<td>100</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(X(I),I=1,99,2)</td>
<td>X(1)</td>
<td>1</td>
<td>-</td>
<td>50</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>(X(I),I=51,99,2)</td>
<td>X(51)</td>
<td>1</td>
<td>-</td>
<td>25</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>X(10,10) all</td>
<td>X(1,1)</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(X(I),I=1,10)</td>
<td>X(1,1)</td>
<td>1</td>
<td>-</td>
<td>100</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(X(I,J),I=1,10,J=1,6)</td>
<td>X(1,1)</td>
<td>6</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(X(I,J),I=3,7,J=3,9)</td>
<td>X(3,3)</td>
<td>7</td>
<td>10</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(X(I,J),I=3,7,4,J=3,9,3)</td>
<td>X(3,3)</td>
<td>3</td>
<td>30</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Note: The character "-" is used above to indicate an argument which is ignored and may be given a dummy value.

The arguments XDRA, NVIX, IIVX, NEVX, and IEX are similar, but define the y-coordinate data to be considered in computing minimum and/or maximum y values.

Normally, the x and y coordinate data tendered to AGSTUP is the same data which will later be used to draw curves. However, this need not be the case. For example, one could call AGSTUP with a two-word XDRA (setting NVIX=1, IIVX=1, NEVX=2, and IEX=1) containing a desired minimum and maximum to be used, disregarding the real data.

If the AUTOGRAPH control parameter 'INVERT.' is given the value "1." (in place of its default value "0."), AGSTUP will behave as if its first five arguments had been interchanged with the last five, so that x-coordinate values refer to vertical distances, and y-coordinate values to horizontal distances, on the graph. This parameter affects AGCURV in a similar manner, thus allowing one to plot "x as a function of y".

SPECIAL CONDITIONS:

A call to AGSTUP may result in the printing of an error message and the execution of a "STOP". These error messages are as follows:

AGSTUP - GRAPH WINDOW IMPROPERLY SPECIFIED

The primary AUTOGRAPH control parameters in the group named 'GRAPH.' have improper values.

AGSTUP - GRID WINDOW IMPROPERLY SPECIFIED

The primary AUTOGRAPH control parameters in the group named 'GRID.' have improper values. This is most likely to occur when the control parameter 'SET.' has the value "2." or "4.", specifying that the edges of the curve window (grid) are to be as implied by the last call to the plot package routine SET. Check to make sure that the portion of the plotter frame specified by the last SET call is within the current graph window.

AGSTUP - s LABELS IMPROPERLY SPECIFIED

where "s" stands for "LEFT", "RIGHT", "BOTTOM", "TOP", or "INTERIOR".

Re-read the paragraph entitled "THE LABEL BOXES" in the USAGE section of the AUTOGRAPH package write-up, above. You have defined a label with a basepoint on one edge of the curve window (grid) and an offset vector pointing outward, some part of which extends inside the curve window (or vice-versa). This is not allowed.
AGNUMB - MANTISSA TOO LONG
AGNUMB - EXPONENT TOO LARGE
AGNUMB - ZERO-LENGTH MANTISSA

The routine AGNUMB is a utility routine used by AGSTUP (and AGBACK) in the generation of numeric labels on axes. You should not normally be able to get any of the above error messages. If you do, see the AUTOGRAPH specialist. Additional information about these error messages will be added to the write-up as more is learned about the quaint and curious things one has to do in order to get them.

COMMON BLOCKS: AGCONP, AGMCHP.

SPACE REQUIRED: 2267 (octal) = 1207 (decimal) 7600 words.

TIMING: At least half of the time required to draw a complete graph is spent in AGSTUP.
SUBROUTINE AGBACK

DIMENSION OF ARGUMENTS: There are no arguments.


PURPOSE: To draw the background specified by the current values of the AUTOGRAPH control parameters - the primary parameters with default values or with values supplied by the user, and the secondary parameters with values computed by AGSTUP.

USAGE: Call it. (You'll like it.) See the AUTOGRAPH parameter write-up, below, for descriptions of the parameters which affect the appearance of a background drawn by AGBACK. See also the write-up of AGSTUP, above.

ARGUMENTS: None.

SPECIAL CONDITIONS: See the write-up of AGSTUP, above. Those error messages beginning with "AGNUMB - " apply to AGBACK as well.

COMMON BLOCKS: AGCONP.

SPACE REQUIRED: 406 (octal) = 264 (decimal) 7600 words.

TIMING: About half the time required to draw a complete graph is spent in AGBACK.
SUBROUTINE AGCURV (XVEC,IX,YVEC,IEY,NEX,KDSH)

DIMENSION OF ARGUMENTS: XVEC(1+(NEXY-1)*IEX) if IEX non-zero; YVEC(1+(NEXY-1)*IEY) if IEY non-zero.


PURPOSE: To draw a curve in a manner specified by the current values of the AUTOGRAPH control parameters – the primary parameters with default values or with values supplied by the user, and the secondary parameters with values computed by AGSTUP.

USAGE: The routine AGCURV, given the x and y coordinates of a set of data points, draws the curve defined by those points, using a dashed-line pattern selected by the final argument.

See the AUTOGRAPH parameter write-up, below, for a description of the parameters which affect the behavior of AGCURV.

One parameter of particular interest is the one named 'WINDOW.', which has the default value "0.". If 'WINDOW.' is given the value "1.", any portion of a curve which lies outside the curve window is omitted. No distortion of any curve segment results; the effect is exactly as if the curve were viewed through a window. There is an additional advantage in setting 'WINDOW.' to "1.": if either the x coordinates, or the y coordinates, or both, are mapped logarithmically into the curve window and zero or negative values occur in the data, AGCURV treats those values as missing-point signals, rather than bombing with an ALOG10 error.

ARGUMENTS: The six arguments of AGCURV are as follows:

(1) XVEC, when IEX is non-zero, is a singly-subscripted array containing NEXY x-coordinate data – curve point 1 has x coordinate XVEC(1), curve point 2 has x coordinate XVEC(1+IEX), curve point 3 has x coordinate XVEC(1+IEX*2), etc. When IEX is zero, the array XVEC is ignored – curve point 1 has x coordinate "1.", curve point 2 has x coordinate "2.", etc.

If the value of any x coordinate matches the current value of the AUTOGRAPH control parameter 'NULL/1.' (default – "1.E36"), the corresponding point is considered to be missing – curve segments on either side of that point are not drawn.

(2) IEX, if non-zero, is the index increment between one x coordinate in XVEC and the next. If IEX is zero, the array XVEC is ignored, as described above.

(3) YVEC is just like XVEC, but provides y coordinate data.

(4) IEY is just like IEX, but describes the use (or non-use) of YVEC.

(5) NEXY is the number of curve points – the number of x/y coordinate pairs to be used.

(6) KDSH specifies the dashed-line pattern to be used in drawing the curve. (Since the routines DASHD, FRSTD, VECTD, and LASTD, in the package DASHCHAR, are used to draw the curve, it may have its own particular dashed-line pattern.)

If KDSH is zero, the user is assumed to have done his own call to DASHD; AGCURV will do no such call.

If KDSH is non-zero and negative, the function MOD(-KDSH-1,26)+1 determines which of 26 "alphabetic" patterns is to be used; each of these generates a solid line interrupted by one of the letters of the alphabet (1-A, 2-B, 3-C, etc.).
If KDSH is non-zero and positive, the function MOD(KDSH - 1, n) + 1 determines which of n "user" patterns is to be used; these n patterns are defined by the AUTOGRAPH control parameters in the group named 'DASH.' - the default values specify one solid-line pattern.

Note: The routines EZY and EZXY, which draw one curve per call, always call AGCURV with KDSH = 1. The routines EZMY and EZMXY, which draw one or more curves per call, call AGCURV with KDSH = ISIGN(p, q), where p is the number of the curve being drawn (1 ≤ p ≤ MANY) and q is the current integral value of the AUTOGRAPH control parameter 'DASH/SELECTOR'.

If the AUTOGRAPH control parameter 'INVERT.' is given the value "1." (in place of its default value "0."), AGCURV will behave as if the arguments XVEC and XEX had been interchanged with the arguments YVEC and YEX, so that x-coordinate values refer to vertical distances, and y-coordinate values to horizontal distances, on the graph. This parameter affects AGSTUP in a similar manner, thus allowing one to plot "x as a function of y".

SPECIAL CONDITIONS: If NEXY is less than or equal to zero, an error stop will occur. The message printed will read

AGKURV - NUMBER OF POINTS IS .LE. 0
or
AGKURV - NUMBER OF POINTS IS .LE. 0

depending on the current value of the parameter 'WINDOW.' (The routine AGKURV draws un-windowed curves, the routine AGQURV windowed curves.)

COMMON BLOCKS: AGCONP.

SPACE REQUIRED: 312 (octal) = 202 (decimal) 7600 words.

TIMING: Negligible compared to the time required by AGSTUP and AGBACK.
INTRODUCTION

The AUTOGRAPH control parameters reside in the labeled common block AGCONP. There are currently 458 parameters, all of which are floating-point. The first 309 are "primary" control parameters. The primary control parameters have default values — each is subject to change by a user program to produce some desired effect on the behavior of the AUTOGRAPH routines and/or the nature of a graph being drawn. The last 149 are "secondary" control parameters — these are computed by AUTOGRAPH itself and are not normally subject to change by a user program.

User access to the AUTOGRAPH control parameters is provided by the routines AGSETP, AGSETF, AGSETI, AGGETP, AGGETF, and AGGETI. See the write-ups of these routines, above. Each of these routines has as its first argument a character string identifying a group of control parameters (perhaps containing only a single parameter) which the user wishes to "set" or "get". These character strings are called "control parameter names". Each has the form 'k(1)/k(2)/...k(n).', where k(1) is a keyword identifying a major group of parameters, k(2) is a keyword identifying a subgroup of that major group, k(3) is a keyword identifying a further subgroup of that subgroup, etc. Only the first three characters of k(1) and the first two characters of k(2), k(3), ... need be used; in addition, certain keywords may be omitted.

Following is a list of all the possible control parameter names. The notation 'k(1)/k(2)/...[k(i)/]...k(n).' indicates that the keyword k(i) and the following slash may be omitted. The text following each name describes the group of parameters identified by the name. The names of the AUTOGRAPH variables referenced (in the common block AGCONP) are given and may be found in the listing of AGCONP which appears at the end of this write-up.

Note: All of the AUTOGRAPH control parameters have floating-point values because of certain portability considerations. Some of them, however, may only take on discrete integral values (like "0.", "1.", "-6.", or "65535.") and are used in roles for which integers would normally be used. These will be pointed out.
# List of the AutoGraph Control Parameters

1. 'PRIMARY.' ................................................................. 49
2. 'DASH/SELECTOR.' ....................................................... 49
3. 'FRAME.' ................................................................. 50
4. 'SET.' ......................................................................... 50
5. 'ROW.' ........................................................................ 51
6. 'INVERT.' .................................................................... 51
7. 'WINDOW.' .................................................................... 52
8. 'BACKGROUND.' .................................................................... 52
9. 'NULL.' ......................................................................... 53
9.1. 'NULL/1.' ................................................................. 53
9.2. 'NULL/2.' ................................................................. 54
10. 'GRAPH.' ..................................................................... 54
10.1. 'GRAPH/LEFT.' ........................................................ 54
10.2. 'GRAPH/RIGHT.' ....................................................... 54
10.3. 'GRAPH/BOTTOM.' ..................................................... 55
10.4. 'GRAPH/TOP.' ........................................................ 55
11. 'GRID.' ......................................................................... 55
11.1. 'GRID/LEFT.' ............................................................ 55
11.2. 'GRID/RIGHT.' .......................................................... 56
11.3. 'GRID/BOTTOM.' ......................................................... 56
11.4. 'GRID/TOP.' .............................................................. 56
11.5. 'GRID/SHAPE.' .......................................................... 56
12. 'X.' .............................................................................. 57
12.1. 'X/MINIMUM.' .......................................................... 57
12.2. 'X/MAXIMUM.' .......................................................... 58
12.3. 'X/LOGARITHMIC.' ................................................... 58
12.4. 'X/ORDER.' ............................................................... 58
12.5. 'X/NICE.' ................................................................... 59
13. 'Y.' .............................................................................. 59
13.1. 'Y/MINIMUM.' .......................................................... 59
13.2. 'Y/MAXIMUM.' .......................................................... 59
13.3. 'Y/LOGARITHMIC.' ................................................... 60
13.4. 'Y/ORDER.' ............................................................... 60
13.5. 'Y/NICE.' ................................................................... 60
14. 'AXIS.' ......................................................................... 60
14.1. '[AXIS/]/s.' ............................................................... 60
14.1.1. '[AXIS/]/s/CONTROL.' ........................................... 61
14.1.2. '[AXIS/]/s/LINE.' .................................................... 61
14.1.3. '[AXIS/]/s/INTERSECTION.' ..................................... 62
14.1.3.1. '[AXIS/]/s/INTERSECTION/GRTD.' ................. 62
14.1.3.2. '[AXIS/]/s/INTERSECTI0N/USER.' .................... 62
<p>| 14.1.4. | '[AXIS]/s/FUNCTION.' .................................................................................. 62 |
| 14.1.5. | '[AXIS]/s/TICKS.' ....................................................................................... 63 |
| 14.1.5.1. | '[AXIS]/s/[TICKS]/MAJOR.' ......................................................................... 63 |
| 14.1.5.1.1. | '[AXIS]/s/[TICKS]/MAJOR/[SPACING/TYPE/]' ............................................. 64 |
| 14.1.5.1.1.2. | '[AXIS]/s/[TICKS]/MAJOR/[SPACING]/BASE.' ............................................ 64 |
| 14.1.5.1.1.3. | '[AXIS]/s/[TICKS]/MAJOR/[SPACING]/COUNT.' ........................................... 65 |
| 14.1.5.1.1.3.2. | '[AXIS]/s/[TICKS]/MAJOR/[LENGTH/OUTWARD/]' ....................................... 65 |
| 14.1.5.1.1.3.2.1. | '[AXIS]/s/[TICKS]/MAJOR/[LENGTH//INWARD/]' ......................................... 66 |
| 14.1.5.1.2. | '[AXIS]/s/[TICKS]/MINOR.' .......................................................................... 66 |
| 14.1.5.1.2.1. | '[AXIS]/s/[TICKS]/MINOR/[SPACING/]' ....................................................... 66 |
| 14.1.5.1.2.2. | '[AXIS]/s/[TICKS]/MINOR/[PATTERN/]' ....................................................... 67 |
| 14.1.5.1.2.3. | '[AXIS]/s/[TICKS]/MINOR/[LENGTH/]' .......................................................... 67 |
| 14.1.5.1.2.3.1. | '[AXIS]/s/[TICKS]/MINOR/[LENGTH/OUTWARD/]' ......................................... 67 |
| 14.1.5.1.2.3.2. | '[AXIS]/s/[TICKS]/MINOR/[LENGTH//INWARD/]' ......................................... 68 |
| 14.1.6. | '[AXIS]/s/[NUMERIC/]' ................................................................................ 68 |
| 14.1.6.1. | '[AXIS]/s/[NUMERIC]/TYPE/]' ..................................................................... 68 |
| 14.1.6.2. | '[AXIS]/s/[NUMERIC]/EXponent.' ................................................................. 68 |
| 14.1.6.3. | '[AXIS]/s/[NUMERIC]/FRACTION/' ................................................................ 68 |
| 14.1.6.4. | '[AXIS]/s/[NUMERIC]/ANGLE/' .................................................................... 71 |
| 14.1.6.4.1. | '[AXIS]/s/[NUMERIC]/ANGLE/1ST/]' ............................................................. 71 |
| 14.1.6.4.2. | '[AXIS]/s/[NUMERIC]/ANGLE/2ND/]' ............................................................. 72 |
| 14.1.6.5. | '[AXIS]/s/[NUMERIC]/OFFSET/' .................................................................. 72 |
| 14.1.6.6. | '[AXIS]/s/[NUMERIC]/WIDTH/' ................................................................... 72 |
| 14.1.6.6.1. | '[AXIS]/s/[NUMERIC]/WIDTH/MANTISSA/' ................................................... 72 |
| 14.1.6.6.2. | '[AXIS]/s/[NUMERIC]/WIDTH/EXPONENT.' ................................................... 73 |
| 15. | 'DASH/' ........................................................................................................... 73 |
| 15.1. | 'DASH/ADDRESS/' ........................................................................................ 73 |
| 15.2. | 'DASH/NUMBER/' .......................................................................................... 73 |
| 15.3. | 'DASH/INDEX/' ............................................................................................. 74 |
| 15.4. | 'DASH/LENGTH/' ............................................................................................ 74 |
| 15.5. | 'DASH/CHARACTER/' ..................................................................................... 74 |
| 15.6. | 'DASH/DOLLAR-QUOTE/' ................................................................................. 75 |
| 16. | 'LABEL/' ......................................................................................................... 75 |
| 16.1. | 'LABEL/CONTROL/' ......................................................................................... 75 |
| 16.2. | 'LABEL/NUMBER/' ........................................................................................ 76 |
| 16.2.1. | 'LABEL/NUMBER/LENGTH/' ......................................................................... 76 |
| 16.2.2. | 'LABEL/NUMBER/CONTENTS/' ....................................................................... 76 |
| 16.2.3. | 'LABEL/NUMBER/NAMES/' ............................................................................. 77 |
| 16.3. | 'LABEL/NAME/' ............................................................................................. 77 |
| 16.4. | 'LABEL/DEFINITION/' .................................................................................... 78 |
| 16.4.1. | 'LABEL/[DEFINITION]/SUPPRESSION.' ........................................................... 78 |</p>
<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.4.2</td>
<td>'LABEL/[DEFINITION/]BASEPOINT.'</td>
</tr>
<tr>
<td>16.4.2.1</td>
<td>'LABEL/[DEFINITION/]BASEPOINT/X.'</td>
</tr>
<tr>
<td>16.4.2.2</td>
<td>'LABEL/[DEFINITION/]BASEPOINT/Y.'</td>
</tr>
<tr>
<td>16.4.3</td>
<td>'LABEL/[DEFINITION/]OFFSET.'</td>
</tr>
<tr>
<td>16.4.3.1</td>
<td>'LABEL/[DEFINITION/]OFFSET/X.'</td>
</tr>
<tr>
<td>16.4.3.2</td>
<td>'LABEL/[DEFINITION/]OFFSET/Y.'</td>
</tr>
<tr>
<td>16.4.4</td>
<td>'LABEL/[DEFINITION/]ANGLE.'</td>
</tr>
<tr>
<td>16.4.5</td>
<td>'LABEL/[DEFINITION/]CENTERING.'</td>
</tr>
<tr>
<td>16.4.6</td>
<td>'LABEL/[DEFINITION/]LINES.'</td>
</tr>
<tr>
<td>16.4.7</td>
<td>'LABEL/[DEFINITION/]INDEX.'</td>
</tr>
<tr>
<td>17</td>
<td>'LINE.'</td>
</tr>
<tr>
<td>17.1</td>
<td>'LINE/MAXIMUM.'</td>
</tr>
<tr>
<td>17.2</td>
<td>'LINE/END.'</td>
</tr>
<tr>
<td>17.3</td>
<td>'LINE/BUFFER.'</td>
</tr>
<tr>
<td>17.3.1</td>
<td>'LINE/BUFFER/LENGTH.'</td>
</tr>
<tr>
<td>17.3.2</td>
<td>'LINE/BUFFER/CONTENTS.'</td>
</tr>
<tr>
<td>17.4</td>
<td>'LINE/NUMBER.'</td>
</tr>
<tr>
<td>17.5</td>
<td>'LINE/DEFINITION.'</td>
</tr>
<tr>
<td>17.5.1</td>
<td>'LINE/[DEFINITION/]SUPPRESSION.'</td>
</tr>
<tr>
<td>17.5.2</td>
<td>'LINE/[DEFINITION/]CHARACTER.'</td>
</tr>
<tr>
<td>17.5.3</td>
<td>'LINE/[DEFINITION/]TEXT.'</td>
</tr>
<tr>
<td>17.5.4</td>
<td>'LINE/[DEFINITION/]LENGTH.'</td>
</tr>
<tr>
<td>17.5.5</td>
<td>'LINE/[DEFINITION/]INDEX.'</td>
</tr>
<tr>
<td>18</td>
<td>'SECONDARY.'</td>
</tr>
<tr>
<td>18.1</td>
<td>'SECONDARY/GRAPH.'</td>
</tr>
<tr>
<td>18.2</td>
<td>'SECONDARY/USER.'</td>
</tr>
<tr>
<td>18.3</td>
<td>'SECONDARY/CURVE.'</td>
</tr>
<tr>
<td>18.4</td>
<td>'SECONDARY/DIMENSIONS.'</td>
</tr>
<tr>
<td>18.5</td>
<td>'SECONDARY/AXIS.'</td>
</tr>
<tr>
<td>18.5.1</td>
<td>'SECONDARY/[AXIS/]s.'</td>
</tr>
<tr>
<td>18.5.1.1</td>
<td>'SECONDARY/[AXIS/]s/POSITION.'</td>
</tr>
<tr>
<td>18.5.1.2</td>
<td>'SECONDARY/[AXIS/]s/TICKS.'</td>
</tr>
<tr>
<td>18.5.1.3</td>
<td>'SECONDARY/[AXIS/]s/NUMERIC.'</td>
</tr>
<tr>
<td>18.6</td>
<td>'SECONDARY/LABEL.'</td>
</tr>
<tr>
<td>18.6.1</td>
<td>'SECONDARY/LABEL/b.'</td>
</tr>
</tbody>
</table>
DESCRIPTION OF THE AUTOGRAPH CONTROL PARAMETERS

1. 'PRIMARY.'

Simplest form of name: 'PRI.'

Variables referenced: Every variable from QDSH through QNAN. (Normally, 309 in all, unless the string replacements given in the paragraph "INFORMATIONAL LABELS" in the section "USAGE" of the AUTOGRAPH package write-up are used to increase the length of the label buffer and/or the label-line buffer.)

Description: This group name gives one the capability of saving and restoring all of the primary control parameters at once. As a hedge against future expansion of AUTOGRAPH's primary control parameter list, you should use an array of more than 309 words (say, 350) in calls to AGGETP and AGSETP which do this. (Those routines are set up to transfer only the current actual number of words in the list.) See the routines EXMPL6, EXMPL7, EXMPL8, and EXMPLF in the AUTOGRAPH examples write-up.

2. 'DASH/SELECTOR.'

Simplest form of name: 'DAS/SE.'

Variable referenced: QDSH (integer equivalent – IDSH).

Description: Despite its name, this parameter is not one of those in the group 'DASH.', described below. It has been placed in a "group" by itself because it is of interest only to users of the routines EZMY and EZMXY, but retains its former name because I couldn't think of a better one. This is the single exception to the rules of name formation stated in the introduction above.

The parameter 'DASH/SELECTOR.' is given a negative integral value to specify that the routines EZMY and EZMXY should use the "alphabetic" set of dashed-line patterns for the curves they draw, a positive integral value to specify that these routines should use the "user" set of dashed-line patterns, defined by the current values of the control parameters in the group 'DASH.'.

Note that the routines EZY and EZXY, which draw but one curve per call, always use the first of the dashed-line patterns in the "user" set; they are unaffected by the value of 'DASH/SELECTOR.'.

Each of the dashed-line patterns in the "alphabetic" set specifies a solid line interrupted periodically by a letter of the alphabet.

Each of the dashed-line patterns in the "user" set is as defined by the user. The default "user" set produces all solid lines.

Default value: "+1." (The "user" set of dashed-line patterns is to be used by EZMY and EZMXY.)
3. 'FRAME.'

Simplest form of name: 'FRA.'

Variable referenced: QFRA (integer equivalent - IFRA).

Description: An integral floating-point number specifying when a frame advance is to be performed by the routines EZY, EZXY, EZMY, and EZMXY and having one of three possible values (out-of-range values are reset by AUTOGRAPH to the value at the nearest end of the range):

   The value "1." specifies a frame advance after drawing a graph.
   The value "2." specifies no frame advance at all.
   The value "3." specifies a frame advance before drawing a graph.

Default value: "1." (frame advance after drawing graph).

4. 'SET.'

Simplest form of name: 'SET.'

Variable referenced: QSET (integer equivalent − ISET).

Warning: This parameter is supported because it was in the old AUTOGRAPH. In fact, many of the problems with the old AUTOGRAPH arose from the difficulty of using it. I do not recommend using 'SET.' It is obsolete. The parameters in the groups 'GRID.', 'X.', and 'Y.' may be used directly to accomplish the ends for which 'SET.' was used. The resulting code is sometimes more verbose, but it is clearer.

Description: An integral floating-point number specifying whether or not AUTOGRAPH (the routine AGSTUP) is to use the arguments of the last "SET" call to determine the linear/log nature of the current graph, the position of the curve window (grid) and/or the x/y minimum/maximum values.

Note: The routine SET is a part of the system plot package. Its first four arguments specify a portion of the plotter frame, its next four arguments specify the minimum and maximum x and y coordinate values to be mapped to that portion, and its ninth argument specifies the linear/log nature of the mapping. The routine GETSET, also a part of the system plot package, is used to retrieve the arguments of the last SET call.

There are eight possible values of 'SET.', four of which are just the negatives of the other four. Using a negated value suppresses the drawing of curves by the routines EZY, EZXY, EZMY, and EZMXY. The possible absolute values of 'SET.' are as follows (out-of-range values are reset by AUTOGRAPH to the value at the nearest end of the range):

   The value "1." means that the arguments of the last SET call are not to be used. However, giving 'SET.' the value "1." implies "special" action by the routine AGSETP; see the note below.
   The value "2." means that the control parameters 'X/LOGARITHMIC.' and 'Y/LOGARITHMIC.' are to be given values ('0.' or "-1." ) consistent with the ninth argument of the last SET call and that the control parameters in the group named 'GRID.' are to be given values consistent with the first four arguments of the last SET call.
   The value "3." means that the control parameters 'X/LOGARITHMIC.' and 'Y/LOGARITHMIC.' are to be given values ('0.' or "-1." ) consistent with the ninth argument of the last SET call and that the other control parameters in the groups named 'X.' and 'Y.' are to be given values consistent with the fifth
through eighth arguments of the last \texttt{SET} call. The value "4." means a combination of the actions specified by the values "2." and "3."

Default value: "1." (no arguments of last \texttt{SET} call used).

Special action by \texttt{AGSETP}: If the parameter \texttt{"SET."} is set (individually, rather than as part of a group) to "1," by an \texttt{AGSETP} call, the parameters in the groups named \texttt{"GRID."}, \texttt{"X."}, and \texttt{"Y."} are reset to their default values. The parameter \texttt{"X/LOGARITHMIC."} is reset to its default value ("0.") only if it has the value "+1."; a value "-1." is not changed. The parameter \texttt{"Y/LOGARITHMIC."} is treated similarly.

5. \texttt{"ROW."}

Simplest form of name: \texttt{"ROW."}

Variable referenced: \texttt{QROW} (integer equivalent -- \texttt{1ROW}).

Description: An integral floating-point number specifying the assumed dimensioning of \texttt{x} and \texttt{y} coordinate data arrays used in calls to the routines \texttt{EZMY} and \texttt{EZMXY}. There are five possible values (out-of-range values are reset by \texttt{AUTOGRAPH} to the value at the nearest end of the range):

- The value "-2." means that both \texttt{x} and \texttt{y} arrays are subscripted by curve number and point number, in that order.
- The value "-1." means that \texttt{y} arrays are subscripted by curve number and point number, in that order, but that \texttt{x} arrays are subscripted by point number only. (The same \texttt{x}-coordinate data is used for all the curves.)
- The value "0." or "1." means that \texttt{y} arrays are subscripted by point number and curve number, in that order, but that \texttt{x} arrays are subscripted by point number only. (The same \texttt{x}-coordinate data is used for all of the curves.)
- The value "+2." means that both \texttt{x} and \texttt{y} arrays are subscripted by point number and curve number, in that order.

Default value: "1." (\texttt{y} by point and curve number, \texttt{x} by point number only).

6. \texttt{"INVERT."}

Simplest form of name: \texttt{"INV."}

Variable referenced: \texttt{QIXY} (integer equivalent -- \texttt{IXY}).

Description: An integral floating-point number having the value "0." or "1.". Out-of-range values are reset by \texttt{AUTOGRAPH} to the value at the nearest end of the range. Giving this parameter the value "1." causes the routines \texttt{AGSTUP} and \texttt{AGCURV} to behave as if arguments defining \texttt{x}-coordinate data had been interchanged with arguments defining \texttt{y}-coordinate data, thus, in some sense, allowing one to graph \texttt{x} as a function of \texttt{y}. This parameter is principally intended for the users of the routines \texttt{EZY}, \texttt{EZXY}, \texttt{EZMY}, and \texttt{EZMXY}.

Default value: "0." (no inversion of \texttt{x} and \texttt{y} arguments).
7. 'WINDOW.'

Simplest form of name: 'WIN.'

Variable referenced: QWND (integer equivalent - IWND).

Description: An integral floating-point number having the value "0." or "1." Out-of-range values are reset by AUTOGRAPH to the value at the nearest end of the range. Giving this parameter the value "1." causes the routine AGCURV to use the subroutine AGQURV, rather than AGKURV, for drawing curves. The result is that curve portions falling outside the curve window (grid) are omitted. See the write-up of AGCURV, above.

Default value: "0." (no windowing of curves).

8. 'BACKGROUND.'

Simplest form of name: 'BAC.'

Variable referenced: QBAC (integer equivalent - IBAC).

Description: An integral floating-point number specifying the type of background to be drawn by AGBACK. There are four possible values (out-of-range values are reset by AUTOGRAPH to the value at the nearest end of the range):

- The value "1." specifies a "perimeter" background.
- The value "2." specifies a "grid" background.
- The value "3." specifies a "half-axis" background.
- The value "4." specifies no background at all.

Default value: "1." (a "perimeter" background).

Special action by AGSETP:

If the parameter 'BACKGROUND.' is set (individually, rather than as part of a group) by a call to AGSETP, the desired background is created by changing the following AUTOGRAPH control parameters:

```
"[AXIS]/s/CONTROL.
"[AXIS]/s/[TICKS]/MAJOR/[LENGTH]/INWARD.
"[AXIS]/s/[TICKS]/MINOR/[LENGTH]/INWARD.
"LABEL/CONTROL.
```

where "s" stands for "LEFT", "RIGHT", "BOTTOM", and "TOP". This determines which of the axes are plotted, how long the inward-pointing portions of major and minor tick marks are to be, and whether or not informational labels are to be plotted. Values used are as follows:

The value "1." sets:
- 's/CONTROL.' to "4." for all s;
- 's/MAJOR/INWARD.' to ".015" for all s;
- 's/MINOR/INWARD.' to ".010" for all s;
- 'LABEL/CONTROL.' to "2.".
The value "2." sets:
's/CONTROL.' to "4." for "s" = "LEFT" and "BOTTOM",
to "-1." for "s" = "RIGHT" and "TOP";
's/MAJOR/INWARD.' to "1." for all s;
's/MINOR/INWARD.' to "1." for all s;
'LABEL/CONTROL.' to "2.".

The value "3." sets:
's/CONTROL.' to "4." for "s" = "LEFT" and "BOTTOM",
to "-1." for "s" = "RIGHT" and "TOP";
's/MAJOR/INWARD.' to ".015" for all s;
's/MINOR/INWARD.' to ".010" for all s;
'LABEL/CONTROL.' to "2.".

The value "4." sets:
's/CONTROL.' to "0." for all s;
's/MAJOR/INWARD.' to ".015" for all s;
's/MINOR/INWARD.' to ".010" for all s;
'LABEL/CONTROL.' to "0.".

The default values of these thirteen parameters correspond to the default value of 'BACKGROUND.' Note that, if the thirteen parameters are changed directly, the value of the parameter 'BACKGROUND.' may not reflect the actual nature of the background defined by them.

9. 'NULL.'

Simplest form of name: 'NUL.'

Variables referenced: SVAL(1) and SVAL(2).

Description: These are the AUTOGRAPH "nulls" or "special values". See below.

9.1. 'NULL/1.'

Simplest form of name: 'NUL/1.'

Variable referenced: SVAL(1).

Description: A floating-point number "null 1", used in several ways by AUTOGRAPH:

Certain AUTOGRAPH control parameters have by default, or may be given, the value "null 1", specifying that AUTOGRAPH (the routine AGSTUP) is to choose values for the parameters. The value chosen for a given parameter is not back-stored in place of the "null 1"; thus, a unique value will be chosen for each graph drawn.

If a curve point specified by the user has an x coordinate or a y coordinate equal to "null 1", that curve point is ignored. It is not used in computing minimum and maximum values. Curve segments on either side of it are not drawn.

Those primary control parameters whose default values cannot be placed in a data statement, like the value "FLOAT(LOC(AGXTOX))" for the parameter 'AXIS/s/FUNCTION.', are given the default value "null 1" as a "default-value signal".

Default value: "1.E36" ('Helgasonian infinity' on the IBM 7094).

Special action by AGSETP: If the parameter 'NULL/1.' is changed (individually, rather than as part of a group) by an AGSETP call, the entire list of primary control parameters is scanned — any value equal to the old "null 1" is replaced by the new one.
9.2. 'NULL/2.'

Simplest form of name: 'NUL/2.'

Variable referenced: SVAL(2).

Description: A floating-point number "null 2". Certain AUTOGRAPH control parameters may be given the value "null 2", specifying that AUTOGRAPH (the routine AGSTUP) is to choose values for the parameters. The value chosen for a given parameter is back-stored in place of the "null 2"; thus, a unique value may be chosen for the first graph of a series and then used for all remaining graphs in the series.

Default value: "2.E36" (twice the "Helgasonian infinity" on the IBM 7094).

Special action by AGSETP: If the parameter 'NULL/2.' is changed (individually, rather than as part of a group) by an AGSETP call, the entire list of primary control parameters is scanned – any value equal to the old "null 2" is replaced by the new one.

10. 'GRAPH.'

Simplest form of name: 'GRA.'

Variables referenced: XLGF, XRGF, YBGF, and YTGF.

Description: These four parameters describe the position of the "graph window" within the plotter frame. A graph drawn by AUTOGRAPH (including labels) is forced to lie entirely within this window.

10.1. 'GRAPH/LEFT.'

Simplest form of name: 'GRA/LE.'

Variable referenced: XLGF.

Description: A floating-point number between "0." and "1." (out-of-range values are reset by AUTOGRAPH to the value at the nearest end of the range) specifying the position of the left edge of the graph window as a fraction of the distance from the left edge of the plotter frame to the right edge of the plotter frame.

Default value: "0." (left edge of plotter frame).

10.2. 'GRAPH/RIGHT.'

Simplest form of name: 'GRA/RL.'

Variable referenced: XRGF.

Description: A floating-point number between "0." and "1." (out-of-range values are reset by AUTOGRAPH to the value at the nearest end of the range) specifying the position of the right edge of the graph window as a fraction of the distance from the left edge of the plotter frame to the right edge of the plotter frame.
10.3. 'GRAPH/BOTTOM.'

Simplest form of name: 'GRA/BO.'

Variable referenced: YBGF.

Description: A floating-point number between "0." and "1." (out-of-range values are reset by AUTOGRAPH to the value at the nearest end of the range) specifying the position of the bottom edge of the graph window as a fraction of the distance from the bottom edge of the plotter frame to the top edge of the plotter frame.

Default value: "0." (bottom edge of plotter frame).

10.4. 'GRAPH/TOP.'

Simplest form of name: 'GRA/TO.'

Variable referenced: YTGF.

Description: A floating-point number between "0." and "1." (out-of-range values are reset by AUTOGRAPH to the value at the nearest end of the range) specifying the position of the top edge of the graph window as a fraction of the distance from the bottom edge of the plotter frame to the top edge of the plotter frame.

Default value: "1." (top edge of plotter frame).

11. 'GRID.'

Simplest form of name: 'GRI.'

Variables referenced: XLGD, XRGD, YBGD, YTGD, and SOGD.

Description: These five parameters describe the position and shape of the "grid window" within the graph window and the position and shape of the "curve window" ("grid") within the grid window.

11.1. 'GRID/LEFT.'

Simplest form of name: 'GRI/LE.'

Variable referenced: XLGD.

Description: A floating-point number between "0." and "1." (out-of-range values are reset by AUTOGRAPH to the value at the nearest end of the range) specifying the position of the left edge of the grid window as a fraction of the distance from the left edge of the graph window to the right edge of the graph window.

Default value: ".15" (a value from the old AUTOGRAPH).
11.2. 'GRID/RIGHT.'

Simplest form of name: 'GRI/RL.'

Variable referenced: XRGD.

Description: A floating-point number between "0." and "1." (out-of-range values are reset by AUTOGRAPH to the value at the nearest end of the range) specifying the position of the right edge of the grid window as a fraction of the distance from the left edge of the graph window to the right edge of the graph window.

Default value: ".95" (a value from the old AUTOGRAPH).

11.3. 'GRID/BOTTOM.'

Simplest form of name: 'GRI/BO.'

Variable referenced: YBGD.

Description: A floating-point number between "0." and "1." (out-of-range values are reset by AUTOGRAPH to the value at the nearest end of the range) specifying the position of the bottom edge of the grid window as a fraction of the distance from the bottom edge of the graph window to the top edge of the graph window.

Default value: ".15" (a value from the old AUTOGRAPH).

11.4. 'GRID/TOP.'

Simplest form of name: 'GRI/TO.'

Variable referenced: YTGD.

Description: A floating-point number between "0." and "1." (out-of-range values are reset by AUTOGRAPH to the value at the nearest end of the range) specifying the position of the top edge of the grid window as a fraction of the distance from the bottom edge of the graph window to the top edge of the graph window.

Default value: ".95" (a value from the old AUTOGRAPH).

11.5. 'GRID/SHAPE.'

Simplest form of name: 'GRI/SH.'

Variable referenced: SOGD.

Description: A floating-point number specifying the shape of the curve window (grid) to be placed inside the grid window. (The curve window, of whatever shape, is centered in, and made as large as possible in, the grid window.) The value of the parameter 'GRID/SHAPE.' falls in one of four possible ranges, as follows:

A value less than "0." specifies the negative of the desired ratio of the curve window's width to its height. For example, the value "-2." specifies a curve window which is twice as wide as it is high.
The value "0." specifies a curve window of exactly the same shape as the grid window. The curve window fills the grid window completely.

A value "s" between "0." and "1." specifies a curve window whose shape is determined by the range of the user's coordinate data, reverting to the shape of the grid window if the ratio of the shorter side of the curve window to the longer side of the curve window would thereby be made less than "s". For example, if "s" were given the value "0.5" and the user x coordinate data ranged in value from "0." to "10." and the user y coordinate data ranged in value from "0." to "15.", the curve window would be made two-thirds as wide as it was high; however, if the y coordinate data ranged in value from "0." to "100.", the curve window would revert to the shape of the grid window, rather than being made one-tenth as wide as it was high.

A value "s" greater than or equal to "1." specifies a curve window whose shape is determined by the range of the user's coordinate data, reverting to a square if the ratio of the longer side of the curve window to the shorter side of the curve window would thereby be made greater than "s".

Note that, if 'GRID/SHAPE.' is given a value greater than "0.", AUTOGRAPH assumes that the user's x and y coordinate data have the same units (both in inches, for example) and that the outline of a real two-dimensional object is to be graphed without distortion. The curve window is shaped in such a way as to accomplish this. This feature should not be used when either of the parameters 'X/LOGARITHMIC.' or 'Y/LOGARITHMIC.' has a non-zero value; doing so will yield strange results.

Note that either of the values "-1." or "+1." produces a square and (for the classicists among us) that the value "-1.61803398874989" produces a golden rectangle.

Default value: "0." (curve window fills grid window).

12. 'X.'

Simplest form of name: 'X.'

Variables referenced: XMIN, XMAX, QLUX, QOVX, and QCEX.

Description: These five parameters specify the mapping of the user's x-coordinate data onto the horizontal axis of the curve window (grid). See also the parameters 'SET.' and 'INVERT.', above.

12.1. 'X/MINIMUM.'

Simplest form of name: 'X/ML'

Variable referenced: XMIN.

Description: A floating-point number specifying the minimum user x coordinate to be considered. This parameter normally has the value "null 1", specifying that AUTOGRAPH (the routine AGSTJUP) should examine the user's x-coordinate data and find the minimum value for itself.

This parameter may be given the value "null 2"; the "null 2" will be replaced the next time AGSTJUP is called by the actual minimum value that AGSTUP computes.

The parameter may also be given an "actual" value; using such a value establishes a cut-off point, below which x-coordinate data are to be ignored. (This can lead to all of the data being ignored, of course, but AUTOGRAPH won't bomb because of that.)

If both of the parameters 'X/MINIMUM.' and 'X/MAXIMUM.' are given actual values, the former should have a lesser value than the latter.
Default value: "1.0E-36" ("null 1").

12.2. 'X/maximum.'

Simplest form of name: 'X/MA.'
Variable referenced: XMAX.

Description: Analogous to 'X/minimum.' above; it specifies the way in which the maximum x coordinate is to be determined.

Default value: "1.0E-36" ("null 1").

12.3. 'X/logarithmic.'

Simplest form of name: 'X/LO.'
Variable referenced: QLUX (integer equivalent - LLUX).

Description: An integral floating-point number having the value "-1.", "0.", or "1.". If this parameter is given a value outside the range [-1., 1.], it is reset by AUTOGRAPH to the value at nearest end of the range.

The value "0." specifies that the mapping of user x coordinates onto the horizontal axis of the curve window is to be linear.

The values "-1." and "1." specify that the mapping is to be logarithmic, in which case: 1) all user x-coordinate data must be greater than zero, or 2) 'X/minimum.' must be given an "actual" value and 'WINDOW.' must be given the value "1.". The value "-1." is immune to change when 'SET.' (which see, above) is given the value "±1."; the value "+1." is not.

Default value: "0." (linear x mapping).

12.4. 'X/order.'

Simplest form of name: 'X/OR.'
Variable referenced: QOVX (integer equivalent - IOVX).

Description: An integral floating-point number having the value "0." or "1.". If this parameter is given a value outside the range [0., 1.], it is reset by AUTOGRAPH to the value at the nearest end of the range.

The value "0." specifies that the values of user x-coordinates mapped to the horizontal axis of the curve window (grid) should increase from left to right.

The value "1." specifies that user x coordinates should decrease from left to right.

Default value: "0." (increase from left to right).
12.5. 'X/NICE.'

Simplest form of name: 'X/NL.'

Variable referenced: QCEX (integer equivalent = NCEX).

Description: An integral floating-point number having the value "-1."", "0.", or "+1.". If this parameter is given a value outside the range [-1..+1.], it is reset by AUTOGRAPH to the value at the nearest end of the range.

The value "-1." specifies that user x-coordinate data are to be mapped onto the horizontal axis of the curve window (grid) in such a way as to force major-tick positions at the endpoints of the bottom x axis.

The value "+1." specifies that user x-coordinate data are to be mapped onto the horizontal axis of the curve window (grid) in such a way as to force major-tick positions at the endpoints of the top x axis.

The value "0." specifies that the x-coordinate data are to be mapped so as to range from the left edge of the curve window to the right edge of the curve window; major-tick positions are not forced at the ends of either x axis.

Default value: "-1." (bottom axis "nice").

13. 'Y.'

Simplest form of name: 'Y.'

Variables referenced: YMIN, YMAX, QLUM, QOVY, and QCEY.

Description: These five parameters specify the mapping of the user's y-coordinate data onto the vertical axis of the curve window (grid). See also the parameters 'SET.' and 'INVERT.', above.

13.1. 'Y/MINIMUM.'

Simplest form of name: 'Y/MI.'

Variable referenced: YMIN.

Description: Analogous to 'X/MINIMUM.', above; it specifies the way in which the minimum y coordinate is to be determined.

Default value: "1.E36" ("null 1").

13.2. 'Y/MAXIMUM.'

Simplest form of name: 'Y/MA.'

Variable referenced: YMAX.

Description: Analogous to 'X/MAXIMUM.', above; it specifies the way in which the maximum y coordinate is to be determined.
Default value: "1.E36" ("null 1").

13.3. 'Y/LOGARITHMIC.'

Simplest form of name: 'Y/ILO.'

Variable referenced: QLUY (integer equivalent -- LLUY).

Description: Analogous to 'X/LOGARITHMIC.', above; it specifies whether the mapping of y coordinates is linear or logarithmic.

Default value: "0." (linear y).

13.4. 'Y/ORDER.'

Simplest form of name: 'Y/OR.'

Variable referenced: QOVY (integer equivalent -- IOVY).

Description: Analogous to 'X/ORDER.', above; it specifies whether y-coordinates increase or decrease from bottom to top.

Default value: "0." (increase from bottom to top).

13.5. 'Y/NICE.'

Simplest form of name: 'Y/NL.'

Variable referenced: QCEY (integer equivalent -- NCEY).

Description: Analogous to 'X/NICE.', above; it specifies whether the left y axis, the right y axis, or neither, is to be "nice".

Default value: "-1." (left axis "nice").

14. 'AXIS.'

Simplest form of name: 'AXI.'

Variables referenced: QDAX, QSPA, PING, ..., WCLE, each dimensioned 4; 92 parameters in all.

Description: These 92 parameters describe four axes: the left axis, the right axis, the bottom axis, and the top axis. In the following, I shall use "s" to mean "any one of the keywords LEFT, RIGHT, BOTTOM, or TOP" and "i(s)" to mean "the integer 1, 2, 3, or 4, depending on the value of s".
14.1. '[AXIS/] s.'

("s" = any one of LEFT, RIGHT, BOTTOM, or TOP.)

Simplest form of name: 's.'

Variables referenced: QDAX(i(s)), QSPA(i(s)), ..., WCLE(i(s)); 23 parameters in all.

Description: These 23 parameters describe the axis specified by "s".

14.1.1. '[AXIS/] s/CONTROL.'

Simplest form of name: 's/CO.'

Variable referenced: QDAX(i(s)) (integer equivalent - IDAX(i(s))).

Description: An integral floating-point number having the value "-1.," "0.," "1.," "2.," "3.," or "4."

Out-of-range values are reset by AUTOGRAPH to the value at the nearest end of the range. This parameter controls certain aspects of the drawing of the axis specified by "s", as follows:

- The value "-1." specifies that only the line portion of the axis may be drawn; tick marks and numeric labels are suppressed.
- The value "0." specifies that no portion of the axis may be drawn.
- A positive value specifies that all portions of the axis may be drawn and specifies what actions AUTOGRAPH may take to prevent numeric-label overlap problems, as follows:
  - The value "1." specifies that numeric labels may not be shrunk or rotated.
  - The value "2." specifies that numeric labels may be shrunk, but not rotated.
  - The value "3." specifies that numeric labels may be rotated, but not shrunk.
  - The value "4." specifies that numeric labels may be both shrunk and/or rotated.

Default value: "4." for all s (all axes drawn, numeric labels may be shrunk and/or rotated).

14.1.2. '[AXIS/] s/LINE.'

Simplest form of name: 's/LI.'

Variable referenced: QSPA(i(s)) (integer equivalent - ISPA(i(s))).

Description: An integral floating-point number having the value "0." or "1." A value outside the range [0., 1.] will be reset by AUTOGRAPH to the value at the nearest end of the range.

- The value "0." specifies that the line portion of the axis specified by "s" may be drawn.
- The value "1." suppresses the line portion of the axis specified by "s".

Default value: "0." for all s (line portions of all axes may be drawn).
14.1.3. ‘[AXIS/] s/INTERSECTION.’

Simplest form of name: ‘s/IN.’

Variables referenced: PING(i(s)) and PINU(i(s)).

Description: Each of these two parameters has the default value “1.E36” (“null 1”). Giving either of them a non-null value causes the axis specified by “s” to be moved away from its normal position on one edge of the curve window. If both are given non-null values, the latter takes precedence over the former.

If the left y axis or the right y axis is moved, it remains vertical, but intersects the bottom of the curve window at a new specified x coordinate. Similarly, if the bottom x axis or the top x axis is moved, it remains horizontal, but intersects the left edge of the grid at a specified y coordinate.

No axis may be moved outside the current graph window; if an attempt is made to do so, the axis is moved as far as the edge and no farther.

14.1.3.1. ‘[AXIS/] s/INTERSECTION/GRID.’

Simplest form of name: ‘s/IN/GR.’

Variable referenced: PING(i(s)).

Description: A floating-point number which, if not equal to the current “null 1”, specifies, in the grid coordinate system, the x coordinate (if “s” = LEFT or RIGHT) or the y coordinate (if “s” = BOTTOM or TOP) of the point of intersection of the axis specified by “s” with the perpendicular sides of the curve window.

Default value: “1.E36” (“null 1”) for all s (axes lie on the edges of the curve window).

14.1.3.2. ‘[AXIS/] s/INTERSECTION/USER.’

Simplest form of name: ‘s/IN/US.’

Variable referenced: PINU(i(s)).

Description: A floating-point number which, if not equal to the current “null 1”, specifies, in the user coordinate system, the x coordinate (if “s” = LEFT or RIGHT) or the y coordinate (if “s” = BOTTOM or TOP) of the point of intersection of the axis specified by “s” with the perpendicular sides of the curve window.

Default value: “1.E36” (“null 1”) for all s (axes lie on the edges of the curve window).

14.1.4. ‘[AXIS/] s/FUNCTION.’

Simplest form of name: ‘s/FU.’

Variable referenced: QULM(i(s)).

Description: A floating-point number whose integer equivalent is the memory address of a FORTRAN subroutine defining the mapping from the “user coordinate system” along the axis specified by “s” to the “label coordinate system” along that axis and its inverse. This subroutine must have three arguments – IDMA, VINP, and VOTP – defined as follows:
IDMA specifies the direction of the mapping. If IDMA is positive, the subroutine must map the user-system value VINP to a label-system value VOTP. If IDMA is negative, the subroutine must map the label-system value VINP to a user-system value VOTP.

VINP is an input value in one of the two coordinate systems.

VOTP is an output value in the other coordinate system.

The two mappings defined by the subroutine must be continuous and monotonic. Each must be the mathematical inverse of the other or AUTOGRAPH will probably go bananas.

Tick marks on the axis specified by "s" are positioned in the label coordinate system. Numeric labels on the axis give values in the label coordinate system.

Whenever this parameter is given the value "null 1", AUTOGRAPH (the routine AGSTUP) replaces that value by the floating-point equivalent of the address of the subroutine AGXTOX, which defines the identity mapping, making the label coordinate system the same as the user coordinate system.

Default value: "1.E36" ("null 1") for all s (identity mapping for all axes).

Special action by AGSETP: None. Note, however, that the second argument of an AGSETP call to set this parameter must be "LOC(sub)", that the second argument of an AGSETP call to set this parameter must be "FLOAT(LOC(sub))", and that the routine name "sub" must appear in an "EXTERNAL" statement.

14.1.5. ‘[AXIS/] s/TICKS.’

Simplest form of name: ‘s/TL.’

Variables referenced: QBTD(i(s)), BASD(i(s)), QMJD(i(s)), QJDP(i(s)), WMJL(i(s)), WMIJR(i(s)), QMND(i(s)), QNDP(i(s)), WMNL(i(s)), and WMNR(i(s)).

Description: These parameters describe the tick marks, if any, which are to be a part of the axis specified by "s". See below.

14.1.5.1. ‘[AXIS/] s/ [TICKS/] MAJOR.’

Simplest form of name: ‘s/MA.’

Variables referenced: QBTD(i(s)), BASD(i(s)), QMJD(i(s)), QJDP(i(s)), WMJL(i(s)), and WMIJR(i(s)).

Description: These parameters describe the major tick marks, if any, which are to be a part of the axis specified by "s". See below.

14.1.5.1.1. ‘[AXIS/] s/ [TICKS/] MAJOR/SPACING.’

Simplest form of name: ‘s/MA/SP.’

Variables referenced: QBTD(i(s)), BASD(i(s)), and QMJD(i(s)).

Description: These parameters describe the way in which major tick marks, if any, are to be spaced along the axis specified by "s". See below.
14.1.5.1.1.1. ‘[AXIS/] s/ [TICKS/] MAJOR/ [SPACING/] TYPE.’

Simplest form of name: ‘s/MA/TY.’

Variable referenced: QBTD(i(s)) (integer equivalent – NBT(i(s))).

Description: A floating-point number specifying where major tick marks are to be placed along the axis specified by “s” (at what values in the label coordinate system along that axis).

There are six possible values:

- The value “0.” specifies that no major tick marks are to be drawn on the axis.
- The value “1.” specifies major tick marks at values of the form ±b times k, where b is the base value specified by ‘[AXIS/] s/ [TICKS/] MAJOR/ [SPACING/] BASE.’ (described below) and k is integral.
- The value “2.” specifies major tick marks at values of the form ±b times 10 to the power k, where b is the base value specified by ‘[AXIS/] s/ [TICKS/] MAJOR/ [SPACING/] BASE.’ (described below) and k is integral.
- The value “3.” specifies major tick marks at values of the form ±b to the power k, where b is the base value specified by ‘[AXIS/] s/ [TICKS/] MAJOR/ [SPACING/] BASE.’ (described below) and k is integral.
- The value “null 1” specifies that AUTOGRAPH (the routine AGSTUP) should use a value “1.”, “2.”, or “3.” – whichever it considers best.
- The value “null 2” specifies that AUTOGRAPH (the routine AGSTUP) should use a value “1.”, “2.”, or “3.” – whichever it considers best – and replace the “null 2” by that value.

Notice that major tick marks on a linear axis may be spaced logarithmically and that major tick marks on a logarithmic axis may be spaced linearly; this is sometimes useful.

Default value: “1.E36” (“null 1”) for all s (AUTOGRAPH spaces major tick marks as it sees fit).

14.1.5.1.1.2. ‘[AXIS/] s/ [TICKS/] MAJOR/ [SPACING/] BASE.’

Simplest form of name: ‘s/MA/BA.’

Variable referenced: BASD(i(s)).

Description: A floating-point number which, if greater than zero and non-null, specifies the base value (“b”, in the preceding parameter description) used in spacing major tick marks in the label coordinate system along the axis specified by “s”. A negative or zero value suppresses major tick marks on the axis. A null value causes AUTOGRAPH to pick an appropriate base value and, if the null was a “null 2”, to backstore that value in place of the “null 2”.

Default value: “1.E36” (“null 1”) for all s (AUTOGRAPH picks the base values).
14.1.5.1.3. \([\text{AXIS/}] \ s/ \ [\text{TICKS/}] \text{MAJOR/} \ [\text{SPACING/}] \text{COUNT.}^*\)

Simplest form of name: 's/MA/CO.'

Variable referenced: QMJD(i(s)) (integer equivalent - NMJD(i(s))).

Description: A floating-point number, having an integral value "n" greater than or equal to 0. A negative value is treated as if it were a zero. The value n is only used when major tick marks are to be spaced linearly and the base value ("b", in the preceding parameter descriptions) is to be chosen by AUTOGRAPH. In this case, n is a rough estimate of the minimum number of major tick marks to be placed on the axis specified by "s". The actual number used may vary between "n+2" and "5n/2+4" (approximately).

Default value: "6." for all s (somewhere between 8 and 19 major tick marks per linear axis).

14.1.5.1.2. \([\text{AXIS/}] \ s/ \ [\text{TICKS/}] \text{MAJOR/PATTERN.}^*\)

Simplest form of name: 's/MA/PA.'

Variable referenced: QJDP(i(s)) (integer equivalent - MJDP(i(s))).

Description: A floating-point number specifying the dashed-line pattern to be used for major tick marks on the axis specified by "s". Normally, its integer equivalent is a 16-bit integer in which "0" bits specify "pen-up" segments (gaps) 3 plotter units long and "1" bits specify "pen-down" segments (solids) 3 plotter units long. The value "0." turns off the major tick marks, the value "65535." (decimal) = "177777." (octal) makes them solid lines. If this parameter is given the value "null 1", the next call to AGSTUP resets it to "65535." (decimal).

Default value: "1.E36" ("null 1") for all s (solid-line patterns).

14.1.5.1.3. \([\text{AXIS/}] \ s/ \ [\text{TICKS/}] \text{MAJOR/LENGTH.}^*\)

Simplest form of name: 's/MA/LE.'

Variables referenced: WMJL(i(s)) and WMJR(i(s)).

Description: Two parameters determining the length of the outward-pointing and inward-pointing portions of the major tick marks on the axis specified by "s". See below.

14.1.5.1.3.1. \([\text{AXIS/}] \ s/ \ [\text{TICKS/}] \text{MAJOR/LENGTH/OUTWARD.}^*\)

Simplest form of name: 's/MA/OU.'

Variable referenced: WMJL(i(s)).

Description: A floating-point number specifying the length of the outward-pointing portion of each major tick mark on the axis specified by "s". The value of this parameter must be of the form "e", "1.+e", or "-e", where "e" is greater than or equal to "0." and less than "1." and represents a fraction of the smaller dimension of the curve window.

When a value "e" is used, each major tick mark extends outward "e" units from the axis.
When a value "1.+e" is used, each major tick mark extends outward to the far edge of the grid and then "e" units beyond the edge. If the axis is not moved away from its normal position, "1.+e" has the same effect as "e".

When a value "-e" is used, the first "e" units of the inward-pointing portion of each major tick mark are erased. This can be used to create off-axis major tick marks (for whatever that may be worth).

Default value: "0." for all s (all major ticks point inward).

14.1.5.1.3.2. \('[AXIS/] s/ [TICKS/] MAJOR/ [LENGTH/] INWARD.'\)

Simplest form of name: 's/MA/IN.'

Variable referenced: WMJR(i(s)).

Description: A floating-point number specifying the length of the inward-pointing portion of each tick mark on the axis specified by "s". The value of this parameter must be of the form "e", "1.+e", or "-e", where e is greater than or equal to "0.” and less than "1.” and represents a fraction of the smaller dimension of the curve window.

When a value "e" is used, each major tick mark extends inward "e" units from the axis.

When a value "1.+e" is used, each major tick mark extends inward to the far edge of the grid and then "e" units beyond the edge. This feature is used to create grid backgrounds.

When a value "-e" is used, the first "e" units of the outward-pointing portion of each major tick mark are erased.

Default value: ".015" for all s (all major ticks point inward).

14.1.5.2. \('[AXIS/] s/ [TICKS/] MINOR.'\)

Simplest form of name: 's/MI.'

Variables referenced: QMND(i(s)), QNDP(i(s)), WMNL(i(s)), and WMNR(i(s)).

Description: Four parameters describing the minor tick marks, if any, which are to be a part of the axis specified by "s". See below.

14.1.5.2.1. \('[AXIS/] s/ [TICKS/] MINOR/SPACING.'\)

Simplest form of name: 's/MI/SP.'

Variable referenced: QMND(i(s)) (integer equivalent - NMND(i(s))).

Description: A floating-point number specifying the desired number of minor tick marks to be distributed between each pair of major tick marks on the axis specified by "s". Possible values are as follows:

A value less than "1." suppresses minor tick marks completely.

A value greater than or equal to "1." which is non-null should be integral; it specifies the number of minor tick marks directly.
The values "null 1" and "null 2" specify that AUTOGRAPH is to choose a reasonable integral value; if a "null 2" is specified, it is replaced by the integral value chosen by AUTOGRAPH.

The minor tick marks, if any, are spaced linearly in the label coordinate system along the axis specified by "s". Note that the appropriate value of this parameter for the usual sort of logarithmic axis is "6."; this causes the minor tick marks between two major tick marks at label-system values 10**n and 10**(n+1) to be placed at the label-system values 2*10**n, 3*10**n, 4*10**n, ..., 9*10**n.

Default value: "1.E36" ("null 1") for all s (AUTOGRAPH chooses appropriate values).

14.1.5.2.2. ['AXIS/] s/ [TICKS/] MINOR/PATTERN."

Simplest form of name: 's/MI/PA.'

Variables referenced: QNDP(i(s)) (integer equivalent - MNDP(i(s))).

Description: A floating-point number specifying the dashed-line pattern to be used for minor tick marks on the axis specified by "s"; analogous to the parameter 'AXIS/] s/ [TICKS/] MAJOR/PATTERN. ', described above.

Default value: "1.E36" ("null 1") for all s (solid-line patterns).

14.1.5.2.3. ['AXIS/] s/ [TICKS/] MINOR/LENGTH."

Simplest form of name: 's/MI/LE.'

Variables referenced: WMNL(i(s)) and WMNR(i(s)).

Description: Two parameters determining the length of the outward-pointing and inward-pointing portions of the minor tick marks on the axis specified by "s". See below.

14.1.5.2.3.1. ['AXIS/] s/ [TICKS/] MINOR/ [LENGTH/] OUTWARD."

Simplest form of name: 's/MI/OU.'

Variable referenced: WMNL(i(s)).

Description: A floating-point number specifying the length of the outward-pointing portion of each minor tick mark on the axis specified by "s"; analogous to the parameter 'AXIS/] s/ [TICKS/] MAJOR/ [LENGTH/] OUTWARD. ', described above.

Default value: "0." for all s (all minor ticks point inward).
14.1.5.2.3.2. \([\text{AXIS/}] \text{s/ [TICKS/]} \text{MINOR/ [LENGTH/]} \text{INWARD.}\)

Simplest form of name: 's/MI/IN.'

Variable referenced: WMNR(i(s)).

Description: A floating-point number specifying the length of the inward-pointing portion of each
minor tick mark on the axis specified by "s"; analogous to the parameter \([\text{AXIS/}] \text{s/ [TICKS/]} \text{MAJOR/ [LENGTH/]} \text{INWARD.}\), described above.

Default value: ".010" for all s (all minor ticks point inward).

14.1.6. \([\text{AXIS/}] \text{s/NUMERIC.}\)

Simplest form of name: 's/NU.'

Variables referenced: QLTD(i(s)), QLED(i(s)), QLFD(i(s)), QLOF(i(s)), QLOS(i(s)), DNLA(i(s)),
WCLM(i(s)), and WCLE(i(s)).

Description: Eight parameters describing the numeric labels, if any, which are to be a part of the
axis specified by "s". The first three parameters in this group are described together,
because they are so closely interdependent.

14.1.6.1. \([\text{AXIS/}] \text{s/ [NUMERIC/]} \text{TYPE.}\)

14.1.6.2. \([\text{AXIS/}] \text{s/ [NUMERIC/]} \text{EXPONENT.}\)

14.1.6.3. \([\text{AXIS/}] \text{s/ [NUMERIC/]} \text{FRACTION.}\)

Simplest forms of names: 's/TY.', 's/EX.', and 's/FR.'

Variables referenced: QLTD(i(s)), QLED(i(s)), and QLFD(i(s)) (integer equivalents - NLTDE(i(s)),
NLED(i(s)), and NLFDE(i(s))).

Description:

These three parameters specify the type of numeric labels to be used (at major-tick positions)
on the axis specified by "s". A fourth parameter, \([\text{AXIS/}] \text{s/ [TICKS/]} \text{MAJOR/ [SPACING/]} \text{TYPE.}\),
described above, also affects the type of numeric labels to be used. I shall refer to these four param-
eters in the ensuing discussion using short forms of their names ('s/TYPE.', 's/EXPO.', 's/FRAC.', and
's/MAJOR/TYPE.', respectively).

All four of these parameters have the default value "null 1" (except for the first, which has
default value "0." for "s" = RIGHT and TOP), leaving AUTOGRAPH free to choose values which are con-
sistent with each other and with other parameters describing the axis specified by "s". Any one or
more of them may be given the value "null 2" (in which case an actual value chosen by AUTOGRAPH is
backstored over the "null 2") or an actual integral floating-point value. Actual values are as follows:

- Setting 's/TYPE.' to "0." turns off the numeric labels on the axis specified by "s". The other
three parameters are ignored.

- Setting 's/TYPE.' to "1." selects "scientific" notation. Each numeric label is written in the form

\[-\] [1] [.1 [7]] \times 10^e

where brackets enclose portions which may be independently present or absent and "e" is a
superscript exponent.
The parameter 's/EXPO.' specifies the length of "i" (the number of characters), thus also specifying the value of the exponent "e". If 's/EXPO.' has a value less than or equal to zero, "i" is omitted. If 's/EXPO.' is less than zero and has the integral absolute value "n", the fraction "f" is forced to have "n" leading zeroes.

The parameter 's/FRAC.' specifies the length of "f" (the number of characters). If 's/FRAC.' is less than or equal to zero, "f" is omitted. If 's/FRAC.' is less than zero, the decimal point is omitted.

If "[i] [.] [f]" has the value "1.", the first part of the label is omitted, leaving only "10 e". If the entire label has the value "0.", the single character "0" is used.

The value of 's/MAJOR/TYP.' is immaterial.

Setting 's/TYP.' to "2." selects "exponential" notation, the exact nature of which depends on the value of 's/MAJOR/TYP.', as follows:

- If 's/MAJOR/TYP.' has the value 1. (all major ticks at values of the form ±b times k), each numeric label is written in the form

  \[ [a] [b] [c] \times 10^e \]

  where brackets enclose portions which may be independently present or absent and "e" is a superscript exponent.

  The parameter 's/EXPO.' specifies the integral value of the exponent "e".

  The parameter 's/FRAC.' specifies the length of "f" (the number of characters). If 's/FRAC.' is less than or equal to zero, "f" is omitted. If 's/FRAC.' is less than zero, the decimal point is omitted.

  If the label value is exactly zero, the single character "0" is used.

- If 's/MAJOR/TYP.' has the value "2." (all major ticks at values of the form ±b times 10 to the power k), each numeric label is written in the form

  \[ [a] [b] [c] \times 10^e \]

  where brackets enclose portions which may be independently present or absent and "e" is a superscript exponent.

  The parameter 's/EXPO.' specifies the integral value of the exponent "e" when "k" equals 0. The value of "e" is 's/EXPO.' plus "k".

  The parameter 's/FRAC.' specifies the length of "f" (the number of characters). If 's/FRAC.' is less than or equal to zero, "f" is omitted. If 's/FRAC.' is less than zero, the decimal point is omitted.

  If the label value is exactly zero, the single character "0" is used.

- If 's/MAJOR/TYP.' has the value "3." (all major ticks at values of the form ±b to the power k), each numeric label is written in the form

  \[ [a] [b] [c] \]

  where brackets enclose portions which may be independently present or absent and "e" is a superscript exponent.

  The parameter 's/EXPO.' is ignored. The value of "e" is "k".

  The parameter 's/FRAC.' specifies the length of "f" (the number of characters). If 's/FRAC.' is less than or equal to zero, "f" is omitted. If 's/FRAC.' is less than zero, the decimal point is omitted.

  Note that "[i] [.] [f]" expresses the value of "b".
Setting `s/TYPE.` to "3." selects "no-exponent" notation, the exact nature of which depends on the value of `s/MAJOR/TYPE.`, as follows:

- If `s/MAJOR/TYPE.` has the value "1." (all major ticks at values of the form ±b times \(k\)), each numeric label is written in the form

\[-[ l ][ . ] [ f] \]

where brackets enclose portions which may be independently present or absent.

The parameter `s/EXPO.` is ignored.

The parameter `s/FRAC.` specifies the length of "f" (the number of characters). If `s/FRAC.` is less than or equal to zero, "f" is omitted. If `s/FRAC.` is less than zero, the decimal point is omitted.

If the label value is exactly zero, the single character "0" is used.

- If `s/MAJOR/TYPE.` has the value "2." (all major ticks at values of the form ±b times \(10^k\)), each numeric label is written in the form

\[-[ l ][ . ] [ f] \]

where brackets enclose portions which may be independently present or absent.

The parameter `s/EXPO.` is ignored.

The length of "f" (the number of characters) is specified by the function

\[\max(s/FRAC., 0) - k\]

if this quantity is greater than zero, and

\[\min(s/FRAC., 0)\]

otherwise. This may appear somewhat formidable, but it produces a simple, desirable result. Suppose, for example, that `s/FRAC.` = "1.", "b" = "3.6", and "k" ranges from -3 to +3; the labels produced are

.0036, .036, 3.6, 36., 360., and 3600.

The parameter `s/FRAC.` may be viewed as specifying the length of "f" when "k" is zero. If the function value is less than or equal to zero, "f" is omitted; if it is less than zero, the decimal point is omitted.

- If `s/MAJOR/TYPE.` has the value "3." (all major ticks at values of the form ±b to the power \(k\)), each numeric label is written in the form

\[-[ l ] [ . ] [ f] \]

if "k" is greater than or equal to zero, and in the form

\[-1/[ l ] [ . ] [ f] \]

if "k" is less than zero. Brackets enclose portions which may be independently present or absent.

The parameter `s/EXPO.` is ignored.

The length of "f" (the number of characters) is specified by the function

\[s/FRAC. * \text{abs}(k)\]

if "k" is non-zero, or

\[\min(s/FRAC., 0)\]

if "k" is zero. Again, this function produces a simple result. Suppose that `s/FRAC.` =
"1.\textsuperscript{a}, "b" = "1.1", and "k" ranges from -3 to +3; the labels produced are

\begin{align*}
1/1.331, \ 1/1.21, \ 1/1.1, \ 1., \ 1.1, \ 1.21, \text{ and } 1.331
\end{align*}

The parameter 's/FRAC.' may be viewed as specifying the length of "f" when "k" is equal to 1. If the function value is less than or equal to zero, "f" is omitted; if it is less than zero, the decimal point is omitted.

Another example: Suppose 's/FRAC.' = "-1.\textsuperscript{a}, "b" = "2.\textsuperscript{a}, and "k" ranges from -4 to +4; the labels produced are

\begin{align*}
1/16, \ 1/8, \ 1/4, \ 1/2, \ 1, \ 2, \ 4, \ 8, \text{ and } 16
\end{align*}

Default value: "1.E36" ("null 1") for all three for all s (AUTOGRAPH chooses values to use), except for

\begin{align*}
'\text{RIGHT}/ [\text{NUMERIC/}] \text{ TYPE.}'
\end{align*}

and

\begin{align*}
'\text{TOP}/ [\text{NUMERIC/}] \text{ TYPE.}'
\end{align*}

which are zeroed to suppress the numeric labels on the right and top axes.

14.1.6.4. '[\text{AXIS/}] s/ [\text{NUMERIC/}] \text{ ANGLE.}'

Simplest form of name: 's/AN.'

Variables referenced: QLOF(i(s)), QLOS(i(s)).

Description: Two integral floating-point numbers specifying the orientation angle of numeric labels on the axis specified by "s". See below.

14.1.6.4.1. '[\text{AXIS/}] s/ [\text{NUMERIC/}] \text{ ANGLE/1ST.}'

Simplest form of name: 's/AN/1S.'

Variable referenced: QLOF(i(s)) (integer equivalent = NLOF(i(s))).

Description: An integral floating-point number having one of the values "0.\textsuperscript{a}, "90.\textsuperscript{a}, "180.\textsuperscript{a}, or "270.\textsuperscript{a}"
(plus or minus a small multiple of "360.\textsuperscript{a}"), specifying the user's first choice for the orientation angle of numeric labels on the axis specified by "s". The value is stated in degrees counter-clockwise from a left-to-right horizontal vector.

The AUTOGRAPH routine AGSTUP decides whether the first choice or the second choice is to be used. The second choice is used only when the first choice leads to overlap problems and the current value of the parameter '[\text{AXIS/}] s/\text{CONTROL}' is a "3.\textsuperscript{a} or a "4.\textsuperscript{a} and the second choice works out better than the first. If AGSTUP decides to use the first choice, it leaves the first-choice parameter with a positive value; if it decides to use the second choice, it leaves the first-choice parameter with a negative value.

Values are made positive or negative by adding and subtracting multiples of "360.\textsuperscript{a}".

Default value: "0.\textsuperscript{a}" for all s (horizontal labels preferred on all axes).
14.1.6.4.2. '[AXIS/] s/ [NUMERIC/] ANGLE/2ND.'

Simplest form of name: 's/AN/2N.'

Variable referenced: QLOS(i(s)) (integer equivalent – NLOS(i(s))).

Description: An integral floating-point number having one of the values "0.", "90.", "180.", or "270." (plus or minus a small multiple of "360.") specifying the user's second choice for the orientation angle of numeric labels on the axis specified by 's'. The value is stated in degrees counter-clockwise from a left-to-right horizontal vector. See the description of the preceding parameter.

Default value: "90." for all s (vertical labels, readable from the right, on all axes).

14.1.6.5. '[AXIS/] s/ [NUMERIC/] OFFSET.'

Simplest form of name: 's/OF.'

Variable referenced: DNLA(i(s)).

Description: A floating-point number specifying the desired position of numeric labels relative to the axis specified by 's'.

If the value is positive, numeric labels are to be placed toward the outside of the grid. If the value is negative, numeric labels are to be placed toward the inside of the grid. In either of these two cases, the magnitude of the value specifies the distance from the line portion of the axis to the nearest part of any numeric label, stated as a fraction of the smaller dimension of the curve window.

If the value is exactly zero, each numeric label is centered on the axis. In this case, the line portion of the axis is suppressed and major and minor tick marks are moved outward so as not to overlap the numeric labels.

Default value: ".015" for all s (all labels outside the grid).

14.1.6.6. '[AXIS/] s/ [NUMERIC/] WIDTH.'

Simplest form of name: 's/WI.'

Variables referenced: WCLM(i(s)) and WCLE(i(s)).

Description: Two floating-point parameters specifying the widths of characters to be used in numeric labels on the axis specified by 's'. See below.

14.1.6.6.1. '[AXIS/] s/ [NUMERIC/] WIDTH/MANTISSA.'

Simplest form of name: 's/WI/MA.'

Variable referenced: WCLM(i(s)).

Description: A floating-point number specifying the width of characters to be used in the "mantissa" of each numeric label on the axis specified by 's', expressed as a fraction of the smaller dimension of the curve window.
Default value: ".015" for all s.

**14.1.6.6.2. \[AXIS/\] s/ \[NUMERIC/\] WIDTH/EXPONENT.**

Simplest form of name: 's/WI/EX.'

Variable referenced: WCLE(i(s)).

Description: A floating-point number specifying the width of characters to be used in the exponent of each numeric label on the axis specified by 's', expressed as a fraction of the smaller dimension of the curve window.

Default value: "".010"" for all s.

**15. 'DASH.'**

Simplest form of name: 'DAS.'

Variables referenced: QADP, QODP, QIDP, QCDP, WOCD, and WODQ.

Description: A group of six parameters describing the "user" set of dashed-line patterns (as opposed to the "alphabetic" set, which is defined by code in the subroutine AGCURV and is not subject to change by the user). Note that the parameter 'DASH/SELECTOR.', described above, is not, despite its name, a part of this group.

**15.1. 'DASH/ADDRESS.'**

Simplest form of name: 'DAS/AD.'

Variable referenced: QADP (integer equivalent – IADP).

Description: A floating-point number, normally one whose integer equivalent is the memory address of an array of dashed-line patterns. This parameter may be given the floating-point value "null 1"; this is, in fact, its default value. When the routine AGSTUP is called next, it replaces the "null 1" by the value "FLOAT(LOC(JSLD))", where the integer variable "JSLD" has the value 65535 (decimal) = 177777 (octal), which is an integer dashed-line pattern specifying a solid line. It also resets the parameter 'DASH/NUMBER.' to a "1." and the parameter 'DASH/INDEX.' to a "1.". Together, these values define the default "user" set of dashed-line patterns: a single pattern defining a solid line.

Default value: "1.E36", which AUTOGRAPH replaces by "FLOAT(LOC(177777 octal))".

**15.2. 'DASH/NUMBER.'**

Simplest form of name: 'DAS/NU.'

Variable referenced: QODP (integer equivalent – NODP).

Description: An integral floating-point number specifying the number of dashed-line patterns in the array whose address is specified by 'DASH/ADDRESS.' If this parameter is given a value less than "1." it is reset to a "1.".
The dashed-line patterns are used in a circular fashion. For example, if 'DASH/NUMBER.' has the value "3," and 'DASH/SELECTOR.' is positive and EZMY is used to draw nine curves, pattern 1 will be used for curves 1, 4, and 7, pattern 2 for curves 2, 5, and 8, and pattern 3 for curves 3, 6, and 9.

Default value: "1." (one dashed-line pattern in the array).

15.3. 'DASH/INDEX.'

Simplest form of name: 'DAS/IN.'

Variable referenced: QIXY (integer equivalent = IXY).

Description: An integral floating-point number specifying the index increment which is required to move from one dashed-line pattern to the next in the array whose address is specified by 'DASH/ADDRESS.' If this parameter is given a value less than "0.0," it is reset to a "0.0." Out-of-array subscripting of a dummy array of type REAL is used to access the dashed-line patterns in the actual user array. Assuming the user's array is called "ARRAY," the dashed-line patterns are assumed to start in array elements ARRAY(1), ARRAY(1+i), ARRAY(1+2i), ..., etc., where "i" is the integer equivalent of 'DASH/INDEX.'.

Each dashed-line pattern may be a 16-bit integer (in the range 0 to 65535 decimal) in which each "0" bit specifies a "pen-up" gap segment 3 plotter units long and each "1" bit specifies a "pen-down" solid segment 3 plotter units long, or a character string in which each single quote specifies a "pen-up" gap segment, each dollar sign specifies a "pen-down" solid segment, and each other character is simply to be drawn as a part of the line. The remaining parameters in the group 'DASH.' are meaningful only when character-string dashed-line patterns are used.

Default value: "1." (patterns are stored one per floating-point word).

15.4. 'DASH/LENGTH.'

Simplest form of name: 'DAS/LE.'

Variable referenced: QCDP (integer equivalent = NCDP).

Description: An integral floating-point number specifying the number of characters in each character-string dashed-line pattern, if any, in the array whose address is specified by 'DASH/ADDRESS.' If this parameter is given a value less than "1.0," it is reset to a "1.".

Default value: "10." on the 7800, "8." on the CRAY (one full word of characters).

15.5. 'DASH/CHARACTER.'

Simplest form of name: 'DAS/CH.'

Variable referenced: WOCD.

Description: A floating-point number specifying the width of each character (other than a dollar sign or a quote) which is drawn along a curve as directed by a character-string dashed-line pattern (whether from the "alphabetic" set or from the "user" set). This width is expressed as a fraction of the smaller dimension of the curve window.
15.6. 'DASH/DOLLAR-QUOTE.'

Simplest form of name: 'DAS/DO.'

Variable referenced: WODQ.

Description: A floating-point number specifying the line length corresponding to a dollar sign (solid) or a quote (gap) in a character-string dashed-line pattern, expressed as a fraction of the smaller dimension of the curve window.

Default value: "0.010"

16. 'LABEL.'

Simplest form of name: 'LAB.'

Variables referenced: QDLB, QBIM, ((FLLB(i,j), i=1,10), j=1,1LBIM), where LBIM is the integer equivalent of 'LABEL/BUFFER/LENGTH.', below, and QBAN.

Description: A set of parameters describing up to eight informational labels which form a part of the background drawn by a call to the routine AGBACK.

16.1. 'LABEL/CONTROL.'

Simplest form of name: 'LAB/CO.'

Variable referenced: QDLB (integer equivalent - IDLB).

Description: An integral floating-point number having the value "0.", "1.", or "2.". Values greater than "2." are changed to a "2." by the next AGSTUP call. Values less than "0." are changed to a "0." by the next AGSTUP call; negative values have a special use, however (see below).

The value "0." disables the drawing of informational labels. They remain defined, however.

The value "1." enables the drawing of informational labels and specifies that they may not be shrunk in response to overlap problems.

The value "2." enables the drawing of informational labels and specifies that they may be shrunk in response to overlap problems.

Default value: "2." (labels enabled, shrinkable).

Special action by AGSETP: An AGSETP call which sets this parameter (individually, rather than as part of a group) to a negative value results in the deletion of all currently-defined labels. Note that the negative value is changed to a zero by the next AGSTUP call; thus, the drawing of informational labels is disabled until re-enabled by the user.
16.2. 'LABEL/BUFFER.'

Simplest form of name: 'LAB/BU.'

Variables referenced: QBIM, ((FLLB(i,j),i=1,10),j=1,LBIM), where LBIM is the integer equivalent of 'LABEL/BUFFER/LENGTH.', below.

Description: The length of the label buffer, plus the label buffer itself. See below.

16.2.1. 'LABEL/BUFFER/LENGTH.'

Simplest form of name: 'LAB/BU/LE.'

Variable referenced: QBIM (integer equivalent – LBIM).

Description: An integral floating-point number specifying the number of 10-word label definitions the label buffer will hold. This parameter is computed by AUTOGRAPH. It has the value "FLOAT((LOC(QBAN)-LOC(FLLB))/10)", and should not be changed by a user program.

See the paragraph entitled "INFORMATIONAL LABELS" in the "USAGE" section of the AUTOGRAPH package write-up, above, for a string replacement which may be applied to the AUTOGRAPH source file to increase the size of the label buffer.

Default value: "4.", reset to actual value (normally "8.") the first time any parameter in the group 'LABEL.' is referenced.

16.2.2. 'LABEL/BUFFER/CONTENTS.'

Simplest form of name: 'LAB/BU/CO.'

Variables referenced: ((FLLB(i,j),i=1,10),j=1,LBIM), where LBIM is the integer equivalent of 'LABEL/BUFFER/LENGTH.', above.

Description: An array containing up to LBIM 10-word label definitions. For a second subscript j,

FLLB(1,j) is either a floating-point "0.", saying that no label is defined by this 10-word block, or a single-word Hollerith label name, saying that a label is defined, in which case:
FLLB(2,j) is either a "0.", to enable drawing of the label, or a "1.", to disable drawing of the label,
FLLB(3,j) and FLLB(4,j) are the x and y coordinates of the label's "basepoint", in the grid coordinate system,
FLLB(5,j) and FLLB(6,j) are the x and y components of the label's "offset vector", stated as signed fractions of the smaller dimension of the curve window,
FLLB(7,j) is an integral floating-point number "0.", "90.", "180.", or "270.", specifying the angle at which the label's "baseline" emanates from the end of its offset vector,
FLLB(8,j) is an integral floating-point number specifying how the lines of the label are to be positioned relative to the end of the offset vector ("-1." to line up the left ends, "0." to line up the centers, "+1." to line up the right ends),
FLLB(9,j) is an integral floating-point count of the number of lines belonging to
the label, and

\[ FLLB(10,j) \] is an integral floating-point pointer specifying the second subscript
(in the line buffer) of the first line of the label (the one having the largest line
number), or, if no lines belong to the label, a "0.".

It is not recommended that a user program change the contents of this buffer
directly. Label definitions should be accessed indirectly by means of the parameters
'LABEL/NAME.' and 'LABEL/[DEFINmON/]...'.

Default value: The label buffer contains four pre-defined labels, corresponding to the four edges of
the curve window. They are as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>'L'</th>
<th>'R'</th>
<th>'B'</th>
<th>'T'</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suppression flag:</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Basepoint x:</td>
<td>0.0</td>
<td>1.0</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Basepoint y:</td>
<td>0.5</td>
<td>0.5</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Offset x:</td>
<td>-0.015</td>
<td>+0.015</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Offset y:</td>
<td>0.0</td>
<td>0.0</td>
<td>-0.015</td>
<td>+0.015</td>
</tr>
<tr>
<td>Baseline angle:</td>
<td>90.0</td>
<td>90.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Centering option:</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Line count:</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>First-line index:</td>
<td>1.0</td>
<td>2.0</td>
<td>3.0</td>
<td>4.0</td>
</tr>
</tbody>
</table>

The description of the parameter 'LINE/BUFFER/CONTENTS.', below, gives the
default values for the definitions of the lines which belong to these labels.

16.2.3. 'LABEL/BUFFER/NAMES.'

Simplest form of name: 'LAB/BU/NA.'

Variables referenced: \( FLLB(1.2, j = 1, LBIM) \), where \( LBIM \) is the integer equivalent of
'LABEL/BUFFER/LENGTH.', above.

Description: This parameter group is a subset of the previous one. It provides a way of retrieving
the names of all currently-defined labels.

16.3. 'LABEL/NAME.'

Simplest form of name: 'LAB/NA.'

Variable referenced: QBAN (aka LBAN).

Description: An integral floating-point pointer which, if non-zero, specifies the second subscript of a
label definition in the label buffer – the one which is to be referenced by the parameter
group named 'LABEL/DEFINITION.' (which see, below).

Setting this parameter is the required first step in accessing a particular label
definition.

Default value: "0." (undefined).

Special action by AGSETP: To access the definition of a particular label, one must first call AGSETP
(or AGSETF) with 'LABEL/NAME.' as the first argument and the name of
the label one wishes to access as the second argument. This causes
AGSETP to search for the definition of the desired label in the label
buffer. If that definition is not found, a new one is made up and inserted
in the label buffer. In either case, the parameter 'LABEL/NAME.' is given a floating-point value whose integer equivalent specifies the second subscript of the label definition in the label buffer.

The definition of a new label has the name specified by the user, a suppression flag "0.", a basepoint (.5,.5), an offset vector (0.,0.), a baseline angle "0.", a centering option "0.", a line count "0.", and a first-line index "0.".

16.4. 'LABEL/DEFINITION.'

Simplest form of name: 'LAB/DE.'

Variables referenced: (FLLB(1,LBAN),i=2,10), where LBAN is the integer equivalent of 'LABEL/NAME.', above.

Description: The parameters defining the label specified by the current value of 'LABEL/NAME.'. If 'LABEL/NAME.' has the value "0.", referencing this group or a parameter in it causes an error exit.

16.4.1. 'LABEL/ [DEFINITION/] SUPPRESSION.'

Simplest form of name: 'LAB/SU.'

Variable referenced: FLLB(2,LBAN), where LBAN is the integer equivalent of 'LABEL/NAME.', above.

Description: An integral floating-point "suppression flag" having the value "0." or "1." and specifying whether drawing of the label specified by 'LABEL/NAME.' is enabled ('0.') or disabled ('1.')

Default value: "0." (label enabled).

Special action by AGSETP: If a user program attempts to set this parameter (individually, rather than as part of a group) to a negative value, the lines of the label specified by 'LABEL/NAME.' are deleted and the parameter 'LINE/NUMBER.' is zeroed. If the negative value is less than "-1.", the label is deleted as well and the parameter 'LABEL/NAME.' is zeroed. (Deleting a label means that its name cell is set to "0.").

16.4.2. 'LABEL/ [DEFINITION/] BASEPOINT.'

Simplest form of name: 'LAB/BA.'

Variables referenced: FLLB(3,LBAN) and FLLB(4,LBAN), where LBAN is the integer equivalent of 'LABEL/NAME.', above.

Description: Two parameters specifying the x and y coordinates of the basepoint of the label specified by 'LABEL/NAME.', in the grid coordinate system. The label is positioned relative to this basepoint.
16.4.2.1. 'LABEL/ [DEFINITION/] BASEPOINT/X.'

Simplest form of name: 'LAB/BA/X.'

Variable referenced: FLLB(3,LBAN), where LBAN is the integer equivalent of 'LABEL/NAME.', above.

Description: The x coordinate of the basepoint of the label specified by 'LABEL/NAME.'. The value "0." refers to the left edge of the curve window, the value "1." to the right edge of the curve window.

Default value: ".5" (the center of the curve window).

16.4.2.2. 'LABEL/ [DEFINITION/] BASEPOINT/Y.'

Simplest form of name: 'LAB/BA/Y.'

Variable referenced: FLLB(4,LBAN), where LBAN is the integer equivalent of 'LABEL/NAME.', above.

Description: The y coordinate of the basepoint of the label specified by 'LABEL/NAME.'. The value "0." refers to the bottom edge of the curve window, the value "1." to the top edge of the curve window.

Default value: ".5" (the center of the curve window).

16.4.3. 'LABEL/ [DEFINITION/] OFFSET.'

Simplest form of name: 'LAB/OF.'

Variables referenced: FLLB(5,LBAN) and FLLB(6,LBAN), where LBAN is the integer equivalent of 'LABEL/NAME.', above.

Description: Two parameters specifying the x and y components of the offset vector of the label specified by 'LABEL/NAME.', as signed fractions of the smaller dimension of the curve window. The offset vector has its basepoint at the label basepoint.

16.4.3.1. 'LABEL/ [DEFINITION/] OFFSET/X.'

Simplest form of name: 'LAB/OF/X.'

Variable referenced: FLLB(5,LBAN), where LBAN is the integer equivalent of 'LABEL/NAME.', above.

Description: The x component of the offset vector of the label specified by 'LABEL/NAME.' — negative toward the left edge, positive toward the right edge, of the curve window. The magnitude represents a fraction of the smaller dimension of the curve window.

Default value: "0." (zero-length vector).
16.4.3.2. 'LABEL/[DEFINITION/] OFFSET/Y.'

Simplest form of name: 'LAB/OF/Y.'

Variable referenced: FLLB(6, LBAN), where LBAN is the integer equivalent of 'LABEL/NAME.', above.

Description: The y component of the offset vector of the label specified by 'LABEL/NAME.' — negative toward the bottom edge, positive toward the top edge, of the curve window. The magnitude represents a fraction of the smaller dimension of the curve window.

Default value: "O." (zero-length vector).

16.4.4. 'LABEL/[DEFINITION/] ANGLE.'

Simplest form of name: 'LAB/AN.'

Variable referenced: FLLB(7, LBAN), where LBAN is the integer equivalent of 'LABEL/NAME.', above.

Description: An integral floating-point number having the value "0.," "90.," "180.," or "270.," and specifying the direction in which the baseline of the label specified by 'LABEL/NAME.' emanates from the end of its offset vector, measured counter-clockwise from a left-to-right horizontal vector. All the lines of a label are written parallel to its baseline and in the direction of the baseline.

Default value: "0." (horizontal, to the right).

16.4.5. 'LABEL/[DEFINITION/] CENTERING.'

Simplest form of name: 'LAB/CE.'

Variable referenced: FLLB(8, LBAN), where LBAN is the integer equivalent of 'LABEL/NAME.', above.

Description: An integral floating-point number specifying the alignment of the lines of the label specified by 'LABEL/NAME.' with the end of its offset vector. A negative value aligns the left ends, a zero value the centers, and a positive value the right ends, of the lines.

Default value: "O." (centers aligned).

16.4.6. 'LABEL/[DEFINITION/] LINES.'

Simplest form of name: 'LAB/LI.'

Variable referenced: FLLB(9, LBAN), where LBAN is the integer equivalent of 'LABEL/NAME.', above.

Description: An integral floating-point number specifying the number of lines in the label specified by 'LABEL/NAME.'.

This parameter is updated by AUTOGRAPH as lines are added to or deleted from the label and is not normally to be set by a user program.

Default value: "O." (no lines).
16.4.7. 'LABEL/[DEFINITION/] INDEX.'

Simplest form of name: 'LAB/IN.'

Variable referenced: FLLB(10,LBAN), where LBAN is the integer equivalent of 'LABEL/NAME.', above.

Description: An integral floating-point number specifying the second subscript (in the line buffer) of the first line belonging to the label specified by 'LABEL/NAME.' — a zero if no line belongs to the label.

This parameter is updated by AUTOGRAPH as lines are added to or deleted from the label and is not normally set by a user program.

Default value: "0." (no lines).

17. 'LINE.'

Simplest form of name: 'LIN.'

Variables referenced: QLLN, TCLN, QNIM, ((FLLN(i,j),i=1,6),j=1,LNIM), where LNIM is the integer equivalent of 'LINE/BUFFER/LENGTH.', below, and QNAN.

Description: A set of parameters describing up to 16 lines belonging to the informational labels defined by the parameter group 'LABEL', above.

17.1. 'LINE/MAXIMUM.'

Simplest form of name: 'LIN/MA.'

Variable referenced: QLLN (integer equivalent — MLLN).

Description: An integral floating-point number specifying the assumed maximum length of a character string delivered to AUTOGRAPH for use as the text of a label line.

Such a character string may occur as the first argument of a call to ANOTAT (defining the text of line "100." in the label named 'L'), as the second argument of a call to ANOTAT (defining the text of line "-100." in the label named 'B'), as the last argument of a call to one of the routines EZY, EZXY, EZMY, or EZMXY (defining the text of line "100." in the label named 'T') or as the second argument of a call to AGSETP whose first argument is 'LINE/[DEFINITION/] TEXT.' (defining the text of any line). In each of these cases, the character string must be of the length specified by 'LINE/MAXIMUM.' or shorter. If it is shorter, its last character must be the character specified by the parameter 'LINE/END.', described below.

The parameter 'LINE/MAXIMUM.' may be given any desired non-negative integral value. However, if it is given a value greater than "80.», its effective value for character strings in calls to ANOTAT, EZY, EZXY, EZMY, and EZMXY will be only "80.". This is because those routines copy the character string to an internal buffer which is just large enough to hold 80 characters. This restriction does not apply to the routine AGSETP, which only saves the memory address of the character string.

Default value: "40." (as in the old AUTOGRAPH).
17.2. 'LINE/END.'

Simplest form of name: 'LIN/EN.'

Variable referenced: TLCN.

Description: A one-word character string whose first character is the one used to mark the end of a character string defining the text of a label line (in calls to ANOTAT, EZY, EZXY, EZMY, EZMXY, and AGSETP), when that character string is shorter than the current maximum specified by the parameter 'LINE/MAXIMUM.', described above. The terminator character does not become a part of the text of the line. It is stripped off, so that only the preceding characters constitute the text of the line.

Default value: "1H$" (as in the old AUTOGRAPH).

17.3. 'LINE/BUFFER.'

Simplest form of name: 'LIN/BU.'

Variables referenced: QNIM, ((FLLN(i,j),i=1,6),j=1,LNIM), where LNIM is the integer equivalent of 'LINE/BUFFER/LENGTH.', below.

Description: The length of the line buffer, plus the line buffer itself. See below.

17.3.1. 'LINE/BUFFER/LENGTH.'

Simplest form of name: 'LIN/BU/LE.'

Variable referenced: QNIM (integer equivalent = LNIM).

Description: An integral floating-point number specifying the number of 6-word line definitions the line buffer will hold. This parameter is computed by AUTOGRAPH. It has the value "FLOAT((LOC(QNAN)-LOC(FLLN))/6)", and should not be changed by a user program.

See the paragraph entitled "INFORMATIONAL LABELS" in the "USAGE" section of the AUTOGRAPH package write-up, above, for a string replacement which may be applied to the AUTOGRAPH source file to increase the size of the line buffer.

Default value: "4.", reset to actual value (normally "16.") when any parameter in the group 'LINE.' is referenced.

17.3.2. 'LINE/BUFFER/CONTENTS.'

Simplest form of name: 'LIN/BU/CO.'

Variables referenced: ((FLLN(i,j),i=1,6),j=1,LNIM), where LNIM is the integer equivalent of 'LINE/BUFFER/LENGTH.', above.

Description: An array containing up to LNIM 6-word label-line definitions. For a second subscript j,

\[ FLLN(i,j) \]

is either a floating-point "null 1", saying that no label line is defined by this 6-word block, or an integral floating-point "line number", saying that it does define a label line, in which case:
FLLN(2,j) is either "0.", to enable drawing of the line, or "1.", to disable drawing of the line.

FLLN(3,j) is the floating-point width of each character of the line, stated as a fraction of the smaller dimension of the curve window.

FLLN(4,j) is an integral floating-point number specifying the memory address of the character string defining the text of the line.

FLLN(5,j) is an integral floating-point count of the number of characters in the text of the line.

FLLN(6,j) is an integral floating-point number specifying the second subscript (in the line buffer) of the next line of the label to which this line belongs (that one of the remaining lines in the chain with the largest line number) or, if there is no next line, a "0."

It is not recommended that a user program change the contents of this buffer directly. Line definitions should be accessed indirectly by means of the parameters 'LINE/NUMBER.' and 'LINE/[DEFINITION/]...'.

Default value: The line buffer contains four pre-defined lines, each of which belongs to one of the four pre-defined labels. They are as follows:

<table>
<thead>
<tr>
<th>Label name</th>
<th>'L'</th>
<th>'R'</th>
<th>'B'</th>
<th>'T'</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line number:</td>
<td>+100.0</td>
<td>-100.0</td>
<td>-100.0</td>
<td>+100.0</td>
</tr>
<tr>
<td>Suppression flag:</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Character width:</td>
<td>0.015</td>
<td>0.015</td>
<td>0.015</td>
<td>0.020</td>
</tr>
<tr>
<td>Text address:</td>
<td>-2.0</td>
<td>-3.0</td>
<td>-1.0</td>
<td>-3.0</td>
</tr>
<tr>
<td>Text length:</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Next-line index:</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

The values shown for text addresses are special values: "−1." is equivalent to "FLOAT(LOC(1HX))", "−2." is equivalent to "FLOAT(LOC(1HY))", and "−3." is equivalent to "FLOAT(LOC(1H))".

The description of the parameter 'LABEL/BUFFER/CONTENTS.', above, gives default values for the definitions of the four labels which contain these lines.

17.4. 'LINE/NUMBER.'

Simplest form of name: 'LIN/NU.'

Variable referenced: QNAN (integer equivalent — LNAN).

Description: An integral floating-point number which, if non-zero, specifies the second subscript of a line definition in the line buffer — the one which is to be referenced by the parameter name 'LINE/DEFINITION.' (which see, below).

Setting the value of this parameter is the required first step in accessing a line definition.

Default value: "0." (undefined).

Special action by AGSETP: To access the definition of a particular line of a particular label, one must ensure that the parameter 'LABEL/NAME.' (which see, above) is set. Then, one must call AGSETP (or AGSETI) with 'LINE/NUMBER.' as the first argument and the number of the line one wishes to access as the second argument. This causes AGSETP to search the line buffer for the definition of a line belonging to the label specified by the current value of 'LABEL/NAME.' and having the desired line number. If no such definition is found, a new one is made up, inserted in the line buffer, and linked into the proper place in the chain of lines belonging to the label.
In either case, the parameter 'LINE/NUMBER.' is given an integral floating-point value specifying the second subscript of the line definition in the line buffer.

The definition of a new line has the number specified by the user, a suppression flag "0.", a character width "0.015", a text address "-3." (a special value which is equivalent to "FLOAT(LOC(1H))"), and a text length "1.".

Note: The "line numbers" are used to identify the lines of a label and to specify their positions relative to each other and to the baseline of the label. Lines having positive line numbers are drawn above the label baseline, lines having zero line numbers are drawn along the label baseline, and lines having negative line numbers are drawn below the label baseline. A line having a greater line number than another line is drawn above that line. "Above" and "below" are used here from the viewpoint of someone reading the label. The magnitudes of the line numbers in no way affect inter-line spacing, which is determined by AUTOGRAPH itself.

17.5. 'LINE/DEFINITION.'

Simplest form of name: 'LIN/DE.'

Variables referenced: \( \text{FLLN}(i, \text{LNAN}), i=2,6 \), where LNAN is the integer equivalent of 'LINE/NUMBER.', above.

Description: The parameters defining the line specified by 'LINE/NUMBER.'. If 'LINE/NUMBER.' has the value "0.", referencing a parameter in this group causes an error exit.

17.5.1. 'LINE/ [DEFINITION/] SUPPRESSION.'

Simplest form of name: 'LIN/SU.'

Variable referenced: FLLN(2, LNAN), where LNAN is the integer equivalent of 'LINE/NUMBER.', above.

Description: An integral floating-point number having the value "0." or "1." and specifying whether drawing of the line specified by 'LINE/NUMBER.' is enabled ("0."), or disabled ("1.").

Default value for a new line: "0." (line enabled).

Special action by AGSETP: If a user program attempts to set this parameter (individually, rather than as a part of a group) to a negative value, the line specified by 'LINE/NUMBER.' is deleted and the parameter 'LINE/NUMBER.' is reset to "0.". (Deleting a line means that it is unlinked from the chain of lines belonging to its label and that its number cell is set to "null 1").
17.5.2. 'LINE/[DEFINITION/] CHARACTER.'

Simplest form of name: 'IUN/CH.'

Variable referenced: FLLN(3,LNAN), where LNAN is the integer equivalent of 'LINE/NUMBER.', above.

Description: A floating-point number specifying the desired width of each character of the line specified by 'LINE/NUMBER.', stated as a fraction of the smaller dimension of the curve window.

Default value for a new line: "0.015".

17.5.3. 'LINE/[DEFINITION/] TEXT.'

Simplest form of name: 'IUN/TE.'

Variables referenced: FLLN(4,LNAN), where LNAN is the integer equivalent of 'LINE/NUMBER.', above.

Description: An integral floating-point number specifying the memory address of a character string defining the text of the line specified by 'LINE/NUMBER.'.

The special values "-1.", "-2.", and "-3." are assumed to be distinguishable from memory-address values and are used to specify the text strings 'X', 'Y', and ' ', respectively.

Default value for a new line: "-3.", specifying the text string ' '.

Special action by AGSETP: When this parameter is set by a call to AGSETP (individually, rather than as part of a group), AGSETP's second argument must be the character string itself (or an array containing the character string). AGSETP uses the function "FLOAT(LOC(X))" to obtain the proper value for the parameter. At the same time, it also examines the character string, determining its length and storing that as the value of the following parameter ('LINE/[DEFINITION/] LENGTH.'). See the descriptions of 'LINE/MAXIMUM.' and 'LINE/TERMINATOR.', above.

17.5.4. 'LINE/[DEFINITION/] LENGTH.'

Simplest form of name: 'IUN/LE.'

Variables referenced: FLLN(5,LNAN), where LNAN is the integer equivalent of 'LINE/NUMBER.', above.

Description: An integral floating-point count of the number of characters in the text of the line specified by 'LINE/NUMBER.'. Setting this parameter less than or equal to zero suppresses the drawing of the line. See also the description of the previous parameter ('LINE/[DEFINITION/] TEXT.').

Default value for a new line: "1." (one character — a blank).
17.5.5. 'LINE/ [DEFINITION/] INDEX.'

Simplest form of name: 'LIN/IN.'

Variables referenced: FLLN(6, LNAN), where LNAN is the integer equivalent of 'LINE/NUMBER.', above.

Description: An integral floating-point number specifying the second subscript (in the line buffer) of the next line of the label — a zero if there is no next line.

This parameter is updated by AUTOGRAPH as lines are added to or deleted from the label and is not normally set by a user program.

18. 'SECONDARY.'

Simplest form of name: 'SEC.'

Variables referenced: XLGW through SBOX(6,4) — 149 in all.

Description: All of the secondary AUTOGRAPH control parameters. These parameters are not normally set by a user program, but are computed by AUTOGRAPH itself (the routine AGSTUP). Their values may be of use in some applications.

18.1. 'SECONDARY/GRAPH.'

Simplest form of name: 'SEC/GR.'

Variables referenced: XLGW, XRGW, YBGW, and YTGW.

Description: Four floating-point numbers specifying the x coordinates of the left and right edges of the graph window and the y coordinates of the bottom and top edges of the graph window, in the grid coordinate system. These values are used by AUTOGRAPH to determine whether a point whose coordinates are expressed in the grid coordinate system lies inside or outside the graph window.

If the primary control parameters in the group named 'GRID.' have their default values ('.15', '.95', '.15', '.95', and '0.'), these four parameters will be given the values '−.1875', '1.0625', '−.1875', and '1.0625'. Note that '−.1875 = (0.−.15)/(.95−.15) and that '1.0625 = (1.−.15)/(.95−.15).

18.2. 'SECONDARY/USER.'

Simplest form of name: 'SEC/US.'

Variables referenced: XLUW, XRUW, YBUW, and YTUW.

Description: Four floating-point numbers specifying the x coordinates of the left and right edges of the curve window (grid) and the y coordinates of the bottom and top edges of the curve window (grid), in the user coordinate system. These values are used in mapping user curve points into the curve window. The routines AGSTUP, AGBACK, and AGCURV use these four numbers as arguments 5 through 8 in calls to the system-plot-package routine SET.
18.3. 'SECONDARY/CURVE.'

Simplest form of name: 'SEC/CU.'

Variables referenced: XLCW, XRCW, YBCW, and YTCW.

Description: Four floating-point numbers specifying the x coordinates of the left and right edges of the curve window (grid) and the y coordinates of the bottom and top edges of the curve window (grid). The x coordinates are stated as fractions of the distance from left to right, and the y coordinates as fractions of the distance from bottom to top, in the plotter frame. The routines AGSTUP, AGBACK, and AGCURV use these four numbers as arguments 1 through 4 in calls to the system-plot-package routine SET. If the primary control parameters in the groups named 'GRAPH.' and 'GRID.' have their default values, these four parameters are given the values "15", "95", "15", and "95".

18.4. 'SECONDARY/DIMENSIONS.'

Simplest form of name: 'SEC/DI.'

Variables referenced: WCWP, HCWP, and SCWP.

Description: Three floating-point numbers, the first two of which specify the width and height of the curve window (grid) and the third of which is equal to the smaller of the first two. Each is stated as a number of plotter units. If the primary control parameters in the groups named 'GRAPH.' and 'GRID.' have their default values and the plotter being used has 1024x1024 addressable positions, then each of these three parameters will be given the value "818.4" = (.95-.15) * 1023.

18.5. 'SECONDARY/AXIS.'

Simplest form of name: 'SEC/AX.'

Variables referenced: XBGA, YBGA, ..., WNLE, each dimensioned 4 — a total of 4 * 19 = 76 parameters.

Description: Seventy-six parameters having to do with the drawing of the four axes. See below.

18.5.1. 'SECONDARY/[AXIS/]s.'

Simplest form of name: 'SEC/s.'

Variables referenced: XBGA(i(s)), YBGA(i(s)), ..., WNLE(i(s)), for i(s) = 1, 2, 3, or 4, depending on "s" — 19 parameters in all.

Description: Nineteen parameters having to do with the drawing of the axis specified by "s", where "s" is one of the keywords LEFT, RIGHT, BOTTOM, or TOP. See below.
18.5.1.1. 'SECONDARY/[AXIS/] s/POSITION.'

Simplest form of name: 'SEC/s/PO.'

Variables referenced: XEGA(i(s)), YBGA(i(s)), UBGA(i(s)), XNDA(i(s)), YNDA(i(s)), and UNDA(i(s)), for i(s) = 1, 2, 3, or 4, depending on "s" – six parameters in all.

Description: Six floating-point numbers, the first three of which describe a point at the beginning of axis "s" and the last three of which describe a point at the end of axis "s". The first two numbers of each triplet are the x and y coordinates of the point, in the grid coordinate system. The third number of each triplet is a user-system x or y coordinate (an x coordinate for a horizontal axis, a y coordinate for a vertical axis) of the point.

18.5.1.2. 'SECONDARY/[AXIS/] s/TICKS.'

Simplest form of name: 'SEC/s/TL.'

Variables referenced: QBTIP(i(s)), BASE(i(s)), and QMNT(i(s)), for i(s)) = 1, 2, 3, or 4, depending on "s" – three parameters in all.

Description: Three floating-point numbers, specifying the values AUTOGRAPH has chosen to use for the primary control parameters

'[AXIS/] s/[TICKS/] MAJOR/ [SPACING/] TYPE.'
'[AXIS/] s/[TICKS/] MAJOR/ [SPACING/] BASE.'
'[AXIS/] s/[TICKS/] MINOR/SPACING.'

(which see, above). These secondary parameters are used to hold the values AUTOGRAPH chooses for the corresponding primary parameters so as not to disturb "null 1" values of those primary parameters.

18.5.1.3. 'SECONDARY/[AXIS/] s/NUMERIC.'

Simplest form of name: 'SEC/s/NU.'

Variables referenced: QLTP(i(s)), QLEX(i(s)), QLFL(i(s)), QCIM(i(s)), QCIE(i(s)), RFNL(i(s)), WNLL(i(s)), WNLLE(i(s)), and WNLR(i(s)), for i(s)) = 1, 2, 3, or 4, depending on "s" – ten parameters in all.

Description: Ten floating-point numbers having to do with the generation of numeric labels on the axis specified by "s". The first three of these specify the values AUTOGRAPH has chosen to use for the primary control parameters

'[AXIS/] s/[NUMERIC/] TYPE.'
'[AXIS/] s/[NUMERIC/] EXPONENT.'
'[AXIS/] s/[NUMERIC/] FRACTION.'

(which see, above). The secondary parameters are used so as not to disturb "null 1" values of the primary parameters. The fourth parameter is an integral floating-point count of the number of characters in the longest numeric-label mantissa on the axis "s". The fifth parameter is an integral floating-point count of the number of characters in the longest numeric-label exponent on the axis "s". The sixth parameter is the necessary multiplicative "reduction factor" (between "0." and "1." ) to be applied to the sizes of numeric labels on the axis "s" in order to make them fit without overlap problems. The seventh, eighth, ninth, and tenth parameters are floating-point numbers specifying the width of the space required by numeric labels to the left (outward), to the right (inward), at the beginning and at the end of the axis "s" – each is
stated as a fraction of the width or height of the curve window, depending on the orientation of the axis "s".

18.6. 'SECONDARY/LABEL.'

Simplest form of name: 'SEC/LA.'

Variables referenced: (RBOX(ib),ib=1,6), ((DBOX(ib,js),ib=1,6),js=1,4), ((SBOX(ib,js),ib=1,6),js=1,4) – a total of 54 parameters.

Description: A set of parameters describing the six "label boxes", each of which provides a mechanism for moving and/or shrinking a particular group of labels in attempting to keep any label in that group from overlapping an axis or extending outside the current graph window. See below.

18.6.1. 'SECONDARY/LABEL/b.'

Simplest form of name: 'SEC/LA/b.'

Variables referenced: RBOX(ib), (DBOX(ib,js),js=1,4), (SBOX(ib,js),js=1,4), for ib = 1, 2, 3, 4, 5, or 6, depending on "b" – a total of nine parameters.

Description: The keyword "b" specifies the label box, as follows:

If "b" = "LEFT", label box 1 is specified. It contains all labels having a basepoint on the left edge of the curve window and a leftward-pointing offset vector. These labels are to be moved leftward as required to avoid overlapping any numeric labels on either y axis.

If "b" = "RIGHT", label box 2 is specified. It contains all labels having a basepoint on the right edge of the curve window and a rightward-pointing offset vector. These labels are to be moved rightward as required to avoid overlapping any numeric labels on either y axis.

If "b" = "BOTTOM", label box 3 is specified. It contains all labels having a basepoint on the bottom edge of the curve window and a downward-pointing offset vector. These labels are to be moved downward as required to avoid overlapping any numeric labels on either x axis.

If "b" = "TOP", label box 4 is specified. It contains all labels having a basepoint on the top edge of the curve window and an upward-pointing offset vector. These labels are to be moved upward as required to avoid overlapping any numeric labels on either x axis.

If "b" = "CENTER", label box 5 is specified. It contains all labels having a basepoint on some edge of the curve window and an inward-pointing offset vector. These labels are to be moved inward as required to avoid overlapping numeric labels on any axis.

If "b" = "GRAPH", label box 6 is specified. It contains all labels not specifically contained in one of the other boxes. These labels are not moved, but may still be shrunk as required to avoid their running outside the curve window.

Prior to a call to AGSTUP, the nine parameters in this group are undefined. Following an AGSTUP call, but preceding an AGBACK call, they have what I shall call "interim" values. Following an AGBACK call, they have what I shall call "final" values.

The first parameter in the group is a "reduction factor" for the widths of characters in the labels in box "b". This parameter may have the interim value "0.", specifying that no actual value has yet been computed, or "1.", specifying that the user has prohibited shrinkage of labels in box "b" (by giving the parameter "LABEL/CONTROL." the value "1."). The final value of the reduction factor may be "-1.", specifying that minimum-sized labels were used, but even they led to overlap problems, or a value between "0.")
and "1.", specifying the actual reduction factor applied when the labels were drawn.

The next four parameters in the group specify the grid-system x coordinates of the left and right edges, and the grid-system y coordinates of the bottom and top edges, of label box "b". The interim values specify the box in which the labels must be made to fit in order to avoid overlap, the final values the box in which the labels were actually made to fit.

The last four parameters in the group specify the grid-system x coordinates of the left and right edges, and the grid-system y coordinates of the bottom and top edges, of the label box "b" which would result if all the labels were reduced to minimum size. The interim values specify an unmoved box, the final values a (possibly) moved box.

The AUTOGRAPH Labeled Common Block AGCONP

The labeled common block AGCONP, which contains all of the AUTOGRAPH control parameters defined above, has the following definition:

```fortran
COMMON /AGCONP/ QDSHQFRA,QSET,QROW,QIXY,QWND,QBAC , SVAL(2) ,
+  XLGF,XRGF,YBGF,YTGF , XLGD,XRGD,YBGD,YTGD , SOGD ,
+  XMIN,XMAX,QLUX,QOVX,QCEX ,
+  YMIN,YMAX,QLUY,QOVY,QCEY ,
+  QDAX(4),QSPA(4),PING(4),PINU(4),QULM(4),QBTD(4),
+  BASD(4),QMD(4),QDP(4),WML(4),WMR(4),QMDND(4),
+  QNDP(4),WMNL(4),WNR(4),QTD(4),QLED(4),QLFD(4),
+  QLOF(4),QLOS(4),DNLA(4),WCLM(4),WCLE(4),
+  QDQP,QDQP,QCDP,QDQP,QODP,QDQP,QCDP,WQCD,WQDP,
+  QDLB,QBM,FLLB(10,8),QBDN ,
+  QLEQ,TCLN,QNIM,FLIN(6,16),QNA ,
+  +
+  XLGW,XRGW,YBGW,YTGW , XLUW,XRUW,YBUIW,YTUIW ,
+  XLCW,XRCW,YBUCW,YTCW , WCWP,HCWP,SCWP ,
+  XBGA(4),YBGA(4),YBGA(4),XNDA(4),YNDA(4),UNDA(4),
+  QBITP(4),BASE(4),QMN(4),QTP(4),QLE(4),QLFL(4),
+  QCM(4),QCIE(4),RFL(4),WNLL(4),WNLR(4),WNLB(4),
+  WNLB(4),QLUA(4),
+  +
+  RBOX(6),DBOX(6,4),SBOX(6,4)
```

Variables whose names begin with the letter "Q" normally have integral values. Some elements of the arrays "FLLB" and "FLIN" also have integral values.

The common declaration is published here for two reasons: First, it is intended as a convenience for those who wish to by-pass the parameter access routines and declare the common block in their own program. I do not recommend this course of action very highly; be forewarned that the common declaration given is subject to change. Second, it will be useful in determining exactly what parameters are included in each parameter group accessed by an AGGETP call or an AGSETP call.
AUTOGRAH EXAMPLES

This write-up contains the FORTRAN source and the dd80 output from nine CRAY-1 jobs using AUTOGRAPH.

Example 1 illustrates a simple use of EXY. Note that AUTOGRAPH chose to use exponential left-axis numeric labels, since non-exponential labels would have required more characters.

Example 2 illustrates a simple use of EZXY. Note that X coordinates used need not be monotonically increasing.

Example 3 illustrates a simple use of EZMY -- a remarkably uninteresting graph.

Example 4 illustrates a simple use of EZMXY -- nested ellipses.

Example 5 illustrates a relatively simple use of EZMXY, although the data involves the use of some truly fascinating numeric constants. Essentially, six circles are drawn: two small portions of each are blanked out by salting in some "null 1's" (1.E36's). The result is a possibly recognizable commercial logo.

Example 6 illustrates the four principal types of backgrounds one can use: perimeter, grid, half-axis, and none. Note that I have cheated a bit and turned the labels back on for the last of these. This example also illustrates the use of linear and logarithmic X and Y mappings.

Example 7 illustrates several features. It shows how to define informational labels. It provides two examples of the way in which a label coordinate system along an axis is defined. Windowing is used to prevent the curves from running wild. The "user" set of dashed-line patterns is employed. ("Incrudesence" is a word that I invented).

Example 8 is a somewhat ugly graph demonstrating that one can plot "X as a function of Y", that label values can run "backwards" along an axis, and that axes have no God-given right to have major tick marks at their ends. The "alphabetic" set of dashed-line patterns is used. Major tick marks on the X axis are extended in both directions.

The final example conveys an appropriate message. It demonstrates two of the peculiar types of numeric labels which may be generated. There are others. It also demonstrates the use of dashed minor ticks (a hellishly time-consuming business which sometimes produces a nice effect).

Additions to this set of examples by AUTOGRAPH users will be gratefully accepted. Of particular interest are examples illustrating pitfalls of AUTOGRAPH and the manner in which they may be avoided, examples produced by "real" programs of scientific value, examples showing creative solutions to graphing problems which are apt to be encountered frequently by others, and last but not least, examples of graphs which just "look neat".

EXAMPLES

-91-

AUTOGRAPH
PROGRAM EXMPL1
C
C DEFINE THE DATA ARRAY.
C
REAL YDRA(51)
C
FILL THE DATA ARRAY.
C
DO 101 I=1,51
   X=FLOAT(I)
   YDRA(I)=10.**(X-1.)*(X-11.)*(X-21.)*(X-31.)*(X-41.)*(X-51.)
       +2.E7*(FRAN()-0.5)
101 CONTINUE
C
DRAW A BOUNDARY AROUND THE EDGE OF THE PLOTTER FRAME.
C
CALL BNDARY
C
DRAW THE GRAPH, USING EZY.
C
CALL EZY (YDRA,51,'EXAMPLE 1 (EZY)$')
C
STOP
C
END

FUNCTION FRAN()
COMMON /FRANCM/ X
DATA X / 2.7182818 /
X=AMOD(9821.*X+.211327,1.)
FRAN=X
RETURN
END

SUBROUTINE BNDARY
C
C ROUTINE TO DRAW THE PLOTTER-FRAME EDGE.
C
CALL PLOTIT ( 0, 0,0)
CALL PLOTIT (32767, 0,1)
CALL PLOTIT (32767,32767,1)
CALL PLOTIT ( 0,32767,1)
CALL PLOTIT ( 0, 0,1)
RETURN
END
EXAMPLE 1 (EZY)
PROGRAM EXMPL2

DEFINE THE DATA ARRAYS.
REAL XDRA(401), YDRA(401)

FILL THE DATA ARRAYS.
DO 101 I=1,401
   THETA=.015707963267949*FLOAT(I-1)
   RHO=SIN(2.*THETA)+.05*SIN(20.*THETA)+.05*SIN(120.*THETA)
   XDRA(I)=RHO*COS(THETA)
   YDRA(I)=RHO*SIN(THETA)
101 CONTINUE

DRAW A BOUNDARY AROUND THE EDGE OF THE PLOTTER FRAME.
CALL BNDARY

DRAW THE GRAPH, USING EZXY.
CALL EZXY(XDRA,YDRA,401,'EXAMPLE 2 (EZXY)$')
STOP
END

SUBROUTINE BNDARY
ROUTINE TO DRAW THE PLOTTER-FRAME EDGE.

CALL PLOTIT(0,0,0)
CALL PLOTIT(32767,0,1)
CALL PLOTIT(32767,32767,1)
CALL PLOTIT(0,32767,1)
RETURN
END
PROGRAM EXMPL3

DEFINE THE DATA ARRAY.

REAL YDRA(100,2)

FILL THE DATA ARRAY.

DO 101 I=1,100

YDRA(I, 1)=COS(3.14159265358979*FLOAT(I)/25.)*FLOAT(I)**2
YDRA(I, 2)=COS(3.14159265358979*FLOAT(I)/25.)*10.**(.04*FLOAT(I))

101 CONTINUE

DRAW A BOUNDARY AROUND THE EDGE OF THE PLOTTER FRAME.

CALL BNDARY

CALL EZMY (YDRA, 100 ,2,100, 'EXAMPLE 3 (EZMY)$')

STOP

END

SUBROUTINE BNDARY

ROUTINE TO DRAW THE PLOTTER-FRAME EDGE.

CALL PLOTIT ( 0, 0,0)
CALL PLOTIT (32767, 0,1)
CALL PLOTIT (32767,32767,1)
CALL PLOTIT ( 0,32767,1)
CALL PLOTIT ( 0, 0,1)
RETURN
END

EXAMPLES -96-
PROGRAM EXMPL4

DEFINE THE DATA ARRAYS.
REAL XDRA(201), YDRA(201,10)

FILL THE DATA ARRAYS.
DO 102 I=1,201
XDRA(I)=-1. +. 02*FLOAT(I-1)
IF (I.GT.101) XDRA(I)=2.-XDRA(I)
   DO 101 J=1,10
   YDRA(I,J)=FLOAT(J)*SQRT(1.000000000001-XDRA(I)**2)/10.
   IF (I.GT.101) YDRA(I,J)=-YDRA(I,J)
   101 CONTINUE
102 CONTINUE

DRAW A BOUNDARY AROUND THE EDGE OF THE PLOTTER FRAME.
CALL BNDARY

DRAW THE GRAPH, USING EZMXY.
CALL EZMXY (XDRA,YDRA,201,10,201,'EXAMPLE 4 (EZMXY)$')
STOP
END

SUBROUTINE BNDARY

ROUTINE TO DRAW THE PLOTTER-FRAME EDGE.
CALL PLOTIT ( 0, 0,0)
CALL PLOTIT (32767, 0,1)
CALL PLOTIT (32767,32767,1)
CALL PLOTIT ( 0,32767,1)
RETURN
END
EXAMPLE 4 (EZMXY)
PROGRAM EXMPL5

C DEFINE THE DATA ARRAYS.
C
REAL XDRA(201,6), YDRA(201,6)
C
C COMPUTE REQUIRED CONSTANTS.
C
PI=3.14159265358979
PI100=PI/100.
PITWO=2.*PI
PIT2D3=2.*PI/3.
PIT4D3=4.*PI/3.
RADOSC=SQRT(3.)/3.
RADOLC=SQRT(3.)/2.
BSSCLL=ATAN(SQRT(12.)/6.)
BSSCUL=ATAN(SQRT(143.)/7.)
BSLCLL=ATAN(SQRT(143.)/17.)
BSLCUL=ATAN(SQRT(2.0))
C
FILL THE DATA ARRAYS.
C
DO 101 I=1,201
THETA=PI100*FLOAT(I-1)
XDRA(I,1)= -.5+RADOSC*COS(THETA)
YDRA(I,1)= RADOSC*SIN(THETA)
IF (ABS(THETA).GE.BSSCLL.AND. ABS(THETA).LE.BSSCUL) XDRA(I,1)=1.E36
IF (ABS(THETA-PITWO).GE.BSSCLL.AND. +ABS(THETA-PITWO).LE.BSSCUL) XDRA(I,1)=1.E36
XDRA(I,2)= .5+RADOSC*COS(THETA)
YDRA(I,2)= RADOSC*SIN(THETA)
IF (ABS(THETA-PIT2D3).GE.BSSCLL.AND. +ABS(THETA-PIT2D3).LE.BSSCUL) XDRA(I,2)=1.E36
XDRA(I,3)= RADOSC*COS(THETA)
YDRA(I,3)=RADOLC+RADOSC*SIN(THETA)
IF (ABS(THETA-PIT4D3).GE.BSSCLL.AND. +ABS(THETA-PIT4D3).LE.BSSCUL) XDRA(I,3)=1.E36
XDRA(I,4)= -.5+RADOLC*COS(THETA)
YDRA(I,4)=RADOLC*SIN(THETA)
IF (ABS(THETA).GE.BSLCLL.AND. ABS(THETA).LE.BSLCUL) XDRA(I,4)=1.E36
IF (ABS(THETA-PITWO).GE.BSLCLL.AND. +ABS(THETA-PITWO).LE.BSLCUL) XDRA(I,4)=1.E36
XDRA(I,5)= .5+RADOLC*COS(THETA)
YDRA(I,5)=RADOLC*SIN(THETA)
IF (ABS(THETA-PIT2D3).GE.BSLCLL.AND. +ABS(THETA-PIT2D3).LE.BSLCUL) XDRA(I,5)=1.E36
XDRA(I,6)= RADOLC*COS(THETA)
YDRA(I,6)=RADOLC+RADOLC*SIN(THETA)
IF (ABS(THETA-PIT4D3).GE.BSLCLL.AND. +ABS(THETA-PIT4D3).LE.BSLCUL) XDRA(I,6)=1.E36
101 CONTINUE
C

EXAMPLES -100-
C SPECIFY SUBSCRIPTING OF XDRA AND YDRA.
C
CALL AGSETI ('ROW.',2)
C
MAKE SURE GRID SHAPE IS SUCH THAT ONE UNIT IN X = ONE UNIT IN Y.
C
CALL AGSETF ('GRID/SHAPE.',2.)
C
TURN OFF BACKGROUND, THEN TURN LABELS BACK ON.
C
CALL AGSETF ('BACKGROUND.',4.)
CALL AGSETI ('LABEL/CONTROL.',2)
C
TURN OFF LEFT LABEL.
C
CALL AGSETF ('LABEL/NAME.','L')
CALL AGSETI ('LABEL/SUPPRESSION FLAG.',1)
C
CHANGE TEXT OF BOTTOM LABEL.
C
CALL AGSETF ('LABEL/NAME.','B')
CALL AGSETI ('LINE/NUMBER.',-100)
CALL AGSETP ('LINE/TEXT.','PURITY, BODY, AND FLAVOR$',1)
C
DRAW A BOUNDARY AROUND THE EDGE OF THE PLOTTER FRAME.
C
CALL BNDARY
C
DRAW THE GRAPH, USING EZMXY.
C
CALL EZMXY (XDRA,YDRA,201,6,201,'EXAMPLE 5 (EZMXY)$')
C
STOP
C
END

SUBROUTINE BNDARY
C
ROUTINE TO DRAW THE PLOTTER-FRAME EDGE.
C
CALL PLOTIT ( 0, 0,0)
CALL PLOTIT (32767, 0,1)
CALL PLOTIT (32767,32767,1)
CALL PLOTIT ( 0,32767,1)
CALL PLOTIT ( 0, 0,1)
RETURN
END
EXAMPLE 5 (EZXMY)

PURITY, BODY, AND FLAVOR
PROGRAM EXMPL6
C
C DEFINE THE DATA ARRAYS.
C
REAL XDRA(501),YDRA(501),GLAB(2)
C
C DEFINE THE GRAPH-WINDOW PARAMETER ARRAY.
C
REAL GWND (4,4)
C
DATA (GWND(I,1),I=1,4) / 0.0 , 0.5 , 0.5 , 1.0 /
DATA (GWND(I,2),I=1,4) / 0.5 , 1.0 , 0.5 , 1.0 /
DATA (GWND(I,3),I=1,4) / 0.0 , 0.5 , 0.0 , 0.5 /
DATA (GWND(I,4),I=1,4) / 0.5 , 1.0 , 0.0 , 0.5 /
C
C FILL THE DATA ARRAYS.
C
DO 101 I=1,501
THETA=.031415926535898*FLOAT (I-1)
XDRA(I)=500.+.9*FLOAT(I-1)*COS(THETA)
YDRA(I)=500.+.9*FLOAT(I-1)*SIN(THETA)
101 CONTINUE
C
C SUPPRESS THE FRAME ADVANCE.
C
CALL AGSETI ( 'FRAME. ',2)
C
C DO FOUR GRAPHS ON THE SAME FRAME, USING DIFFERENT BACKGROUNDS.
C
DO 102 IGRF = 1,4
C
C POSITION THE GRAPH WINDOW.
C
CALL AGSETP ( 'GRAPH WINDOW.',GWND(1,IGRF),4)
C
C DECLARE THE BACKGROUND TYPE.
C
CALL AGSETI ( 'BACKGROUND TYPE.',IGRF)
C
C TURN THE INFORMATIONAL LABELS BACK ON IF THEY WERE JUST TURNED OFF.
C
IF (IGRF.EQ.4) CALL AGSETI ( 'LABEL/CONTROL.',2)
C
C DECLARE THE LINEAR/LOGARITHMIC NATURE OF THE GRAPH.
C
CALL AGSETI ( 'X/LOGARITHMIC.',(IGRF-1)/2)
CALL AGSETI ( 'Y/LOGARITHMIC.',MOD(IGRF-1,2))
C
C ENCODE THE PROPER GRAPH LABEL.
C
ENCODE (12,1001,GLAB) IGRF

Examples -103-
C DRAW THE GRAPH, USING EZXY.
C
    CALL EZXY (XDRA,YDRA,501,GLAB)
C
102 CONTINUE
C
C DRAW A BOUNDARY AROUND THE EDGE OF THE PLOTTER FRAME.
C
    CALL BNDARY
C
C ADVANCE THE FRAME.
C
    CALL FRAME
C
    STOP
C
C FORMAT FOR ENCODE.
C
    1001 FORMAT ('EXAMPLE 6-',I1,'$')
END

SUBROUTINE BNDARY
C
C ROUTINE TO DRAW THE PLOTTER-FRAME EDGE.
C
    CALL PLOTIT ( 0, 0,0)
    CALL PLOTIT (32767, 0,1)
    CALL PLOTIT (32767,32767,1)
    CALL PLOTIT ( 0,32767,1)
    CALL PLOTIT ( 0, 0,1)
RETURN
END
EXAMPLES

---

EXAMPLE 6-1

EXAMPLE 6-2

EXAMPLE 6-3

EXAMPLE 6-4
PROGRAM EXMPL7

C DEFINE THE DATA ARRAYS AND THE DASH-PATTERN ARRAY.
C
REAL XDRA(101), YDRA(101,9), DSHP(4,9)
C
C DECLARE THE MAPPING ROUTINES EXTERNAL.
C
EXTERNAL EX7MFR, EX7MFT
C
C FILL THE DATA ARRAYS AND THE DASH PATTERN ARRAY.
C
DO 101 I=1,101
   XDRA(I)=-90.+1.8*FLOAT(I-1)
101 CONTINUE
C
DO 103 J=1,9
   ENCODE (32,1001,DSHP(1,J)) J
   FJ=J
   DO 102 I=1,101
      YDRA(I, J)=3.*FJ-(FJ/2700.)*XDRA(I)**2
102 CONTINUE
103 CONTINUE
C
C TURN ON WINDOWING (SOME CURVES RUN OUTSIDE THE CURVE WINDOW).
C
CALL 'AGSETI ('WINDOWING.', 1)
C
C MOVE THE EDGES OF THE CURVE WINDOW (GRID).
C
CALL AGSETF ('GRID/LEFT.', 3.10)
CALL AGSETF ('GRID/RIGHT.', 3.90)
CALL AGSETF ('GRID/BOTTOM.', 3.10)
CALL AGSETF ('GRID/TOP.', 3.85)
C
C SET THE X AND Y MINIMUM AND MAXIMUM.
C
CALL AGSETF ('X/MINIMUM.', -90.)
CALL AGSETF ('X/MAXIMUM.', +90.)
CALL AGSETF ('Y/MINIMUM.', 0.)
CALL AGSETF ('Y/MAXIMUM.', 18.)
C
C SET LEFT AXIS PARAMETERS.
C
CALL AGSETI ('LEFT/MAJOR/TYPE.', 1)
CALL AGSETF ('LEFT/MAJOR/BASE.', 3.)
CALL AGSETI ('LEFT/MINOR/SPACING.', 2)
C
C SET RIGHT AXIS PARAMETERS.
C
CALL AGSETI ('RIGHT/FUNCTION.', LOC(EX7MFR))
CALL AGSETF ('RIGHT/NUMERIC/TYPE.', 1.E36)
C
C SET BOTTOM AXIS PARAMETERS.
C
CALL AGSETI ('BOTTOM/MAJOR/TYPe.', 1)
CALL AGSETF ('BOTTOM/MAJOR/BASE.', 15.)
CALL AGSETI ('BOTTOM/MINOR/SPACING.', 2)
C
C SET TOP AXIS PARAMETERS.
C
CALL AGSETI ('TOP/FUNCTION.', LOC(EX7MFT))
CALL AGSETF ('TOP/NUMERIC/TYPe.', 1.E36)
C
C SET UP THE DASH PATTERNS TO BE USED.
C
CALL AGSETI ('DASH/ADDRESS.', LOC(DSHP))
CALL AGSETI ('DASH/NUMBER.', 9)
CALL AGSETI ('DASH/INDEX.', 4)
CALL AGSETI ('DASH/LENGTH.', 28)
C
C SET UP THE LEFT LABEL.
C
CALL AGSETF ('LABEL/NAME.', 'L')
CALL AGSETI ('LINE/NUMBER.', 100)
CALL AGSETP ('LINE/TEXT.', 'HEIGHT (KILOMETERS)$', 1)
C
C SET UP THE RIGHT LABEL.
C
CALL AGSETF ('LABEL/NAME.', 'R')
CALL AGSETI ('LINE/NUMBER.', -100)
CALL AGSETP ('LINE/TEXT.', 'PRESSURE (TONS/SQUARE FURLONG)$', 1)
C
C SET UP THE BOTTOM LABELS.
C
CALL AGSETF ('LABEL/NAME.', 'B')
CALL AGSETI ('LINE/NUMBER.', -100)
CALL AGSETP ('LINE/TEXT.', 'LATITUDE (DEGREES)$', 1)
C
CALL AGSETF ('LABEL/NAME.', 'SP')
CALL AGSETF ('LABEL/BASEPOINT/X.', .000001)
CALL AGSETF ('LABEL/BASEPOINT/Y.', 0.)
CALL AGSETF ('LABEL/OFFSET/Y.', -.015)
CALL AGSETI ('LINE/NUMBER.', -100)
CALL AGSETP ('LINE/TEXT.', 'SP$', 1)
C
CALL AGSETF ('LABEL/NAME.', 'NP')
CALL AGSETF ('LABEL/BASEPOINT/X.', .999999)
CALL AGSETF ('LABEL/BASEPOINT/Y.', 0.)
CALL AGSETF ('LABEL/OFFSET/Y.', -.015)
CALL AGSETI ('LINE/NUMBER.', -100)
CALL AGSETP ('LINE/TEXT.', 'NP$', 1)
C
C SET UP THE TOP LABEL.
C
EXAMPLES AUTOGRAPH-107-
CALL AGSETF ('LABEL/NAME.', 'T')
CALL AGSETI ('LINE/NUMBER.', 80)
CALL AGSETP ('LINE/TEXT.', 'DISTANCE FROM EQUATOR (MILES)', 1)
CALL AGSETP ('LINE/TEXT.', '$', 1)
CALL AGSETP ('LINE/NUMBER.', 100)
CALL AGSETP ('LINE/TEXT.', 'LINES OF CONSTANT INCRUDESCENCE', 1)
CALL AGSETP ('LINE/NUMBER.', 110)
CALL AGSETP ('LINE/TEXT.', 'EXAMPLE 7 (EZMXY)', 1)

C
C SET UP CENTERED (BOX 6) LABEL.
C
CALL AGSETF ('LABEL/NAME.', 'EQUATOR')
CALL AGSETI ('LABEL/ANGLE.', 90)
CALL AGSETI ('LABEL/NUMBER.', 0)
CALL AGSETP ('LINE/TEXT.', 'EQUATOR', 1)

C
C DRAW A BOUNDARY AROUND THE EDGE OF THE PLOTTER FRAME.
C
CALL BNDARY

C
C DRAW THE GRAPH, USING EZMXY.
C
CALL EZMXY (XDRA,YDRA,101,9,101,0)

C
STOP

C
C FORMAT FOR ENCODE ABOVE.
C
1001 FORMAT ('$$$$$$$$$$$$$$$$$$$$$$''J''=''I1'', ''')

C
END

SUBROUTINE EX7MFR (IDMA, VINP, VOTP)
IF (IDMA.GT.0) VOTP=ALOG10(20.-VINP)
IF (IDMA.LT.0) VOTP=20.-10.**VINP
RETURN
END

SUBROUTINE EX7MFT (IDMA, VINP, VOTP)
IF (IDMA.GT.0) VOTP=70.136*VINP
IF (IDMA.LT.0) VOTP=VINP/70.136
RETURN
END

EXAMPLES -108- AUTOGRAPH
SUBROUTINE BNDARY

C ROUTINE TO DRAW THE PLOTTER-FRAME EDGE.

CALL PLOTIT(0, 0, 0)
CALL PLOTIT(32767, 0, 1)
CALL PLOTIT(32767, 32767, 1)
CALL PLOTIT(0, 32767, 1)
CALL PLOTIT(0, 0, 1)
RETURN
END
PROGRAM EXMPL8
C
C DEFINE THE DATA ARRAYS.
C
REAL XDRA(101), YDRA(4,101)
C
C FILL THE DATA ARRAYS.
C
DO 101 I=1,101
   XDRA(I)=-3.14159265358979+.062831853071796*FLOAT(I-1)
101 CONTINUE
C
DO 103 I=1,4
   FLTI=I
   BASE=2.*FLTI-1.
   DO 102 J=1,101
      YDRA(I,J)=BASE+.75*SIN(-3.14159265358979+.062831853071796*
      + FLTI*FLOAT(J-1))
102 CONTINUE
103 CONTINUE
C
CHANGE THE LINE-END CHARACTER TO A PERIOD.
C
CALL AGSETF ('LINE/END.', ' .')
C
SPECIFY LABELS FOR X AND Y AXES.
C
CALL ANOTAT ('SINE FUNCTIONS OF T.', 'T.', 0, 0, 0, 0)
C
USE A HALF-AXIS BACKGROUND.
C
CALL AGSETI ('BACKGROUND.', 3)
C
MOVE X AXIS TO THE ZERO POINT ON THE Y AXIS.
C
CALL AGSETF ('BOTTOM/INTERSECTION/USER.', 0.)
C
SPECIFY BASE VALUE FOR SPACING OF MAJOR TICS ON X AXIS.
C
CALL AGSETF ('BOTTOM/MAJOR/BASE.', 1.)
C
RUN MAJOR TICS ON X AXIS TO EDGE OF CURVE WINDOW.
C
CALL AGSETF ('BOTTOM/MAJOR/INWARD.', 1.)
   CALL AGSETF ('BOTTOM/MAJOR/OUTWARD.', 1.)
C
POSITION X AXIS MINOR TICS.
C
   CALL AGSETI ('BOTTOM/MINOR/SPACING.', 9)
C
RUN THE Y AXIS BACKWARD.

EXAMPLES -111- AUTOGRAPH
CALL AGSETI ('Y/ORDER.', 1)
CALL AGSET ('Y/NICE.', 0)
CALL AGSETF ('GRID/SHAPE.', .01)
CALL AGSET ('DASH/SELECTOR.', -1)
CALL AGSET ('ROW', -1)
CALL AGSET ('INVERT.', 1)
CALL BNDARY
CALL EZMXY (XDRA, YDRA, 4, 4, 101, 'EXAMPLE 8. ')
STOP
END

SUBROUTINE BNDARY
C
C ROUTINE TO DRAW THE PLOTTER-FRAME EDGE.
C
CALL PLOTIT (0, 0, 0)
CALL PLOTIT (32767, 0, 1)
CALL PLOTIT (32767, 32767, 1)
CALL PLOTIT (0, 32767, 1)
RETURN
END
PROGRAM
PROGRAM EXMPLF
C
C DEFINE THE DATA ARRAY.
C
DIMENSION XYCD(226)
C
C FILL THE DATA ARRAY.
C
READ 1001 , XYCD
C
DO 101 I=1,226
   IF (XYCD(I).EQ.1.E36) GO TO 101
   XYCD(I)=2.**(XYCD(I)-15.)/2.5
101 CONTINUE
C
C SPECIFY LOG/LOG PLOT.
C
CALL DISPLA (0,0,4)
C
C BUMP THE LINE-MAXIMUM PARAMETER PAST 42.
C
CALL AGSETI ('LINE/MAXIMUM.',50)
C
C SPECIFY X- AND Y-AXIS LABELS, GRID BACKGROUND.
C
CALL ANOTAT ('LOGARITHMIC, BASE 2, EXPONENTIAL LABELING$\,\,$',
            +   'LOGARITHMIC, BASE 2, NO-EXPONENT LABELING$\,\,$',2,0,0)
C
C SPECIFY THE GRAPH LABEL.
C
CALL AGSETF ('LABEL/NAME.', 'T')
CALL AGSETI ('LINE/NUMBER.', 100)
CALL AGSETP ('LINE/TEXT.', 'FINAL EXAMPLE$',1)
C
C SPECIFY X-AXIS TICKS AND LABELS.
C
CALL AGSETI ('BOTTOM/MAJOR/TYPE.',3)
CALL AGSETF ('BOTTOM/MAJOR/BASE.',2.)
CALL AGSETI ('BOTTOM/NUMERIC/TYPE.',2)
CALL AGSETI ('BOTTOM/MINOR/SPACING.',4)
CALL AGSETI ('BOTTOM/MINOR/PATTERN.',125252B)
C
C SPECIFY Y-AXIS TICKS AND LABELS.
C
CALL AGSETI ('LEFT/MAJOR/TYPE.',3)
CALL AGSETF ('LEFT/MAJOR/BASE.', 2.)
CALL AGSETI ('LEFT/NUMERIC/TYPE.',3)
CALL AGSETI ('LEFT/MINOR/SPACING.',4)
CALL AGSETI ('LEFT/MINOR/PATTERN.',125252B)

EXAMPLES

-114-

AUTOGRAPH
C COMPUTE SECONDARY CONTROL PARAMETERS.
C
CALL AGSTUP (XYCD(1),1,0,113,2,XYCD(2),1,0,113,2)
C
C DRAW THE BACKGROUND.
C
CALL AGBACK
C
C DRAW THE CURVE TWICE TO MAKE IT DARKER.
C
CALL AGCURV (XYCD(1),2,XYCD(2),2,113,1)
CALL AGCURV (XYCD(1),2,XYCD(2),2,113,1)
C
C DRAW A BOUNDARY AROUND THE EDGE OF THE PLOTTER FRAME.
C
CALL BNDARY
C
C ADVANCE THE FRAME.
C
CALL FRAME
C
STOP
C
FORMAT.
C
1001 FORMAT (14E5.0)
C
END

SUBROUTINE BNDARY
C
ROUTINE TO DRAW THE PLOTTER-FRAME EDGE.
C
CALL PLOTIT ( 0, 0,0)
CALL PLOTIT (32767, 0,1)
CALL PLOTIT (32767,32767,1)
CALL PLOTIT ( 0,32767,1)
CALL PLOTIT ( 0, 0,1)
RETURN
END
LOGARITHMIC, BASE 2, EXPONENTIAL LABELING

LOGARITHMIC, BASE 2, NO-EXPONENT LABELING
TREE STRUCTURE OF AUTOGRAPH PARAMETER GROUPS

PRIMARY...................................................... Allows for setting and restoring all primary parameters.

DASH/SELECTOR............................................... Specifies which type of dashed lines the EZ routines will use.

FRAME.......................................................... Specifies when the EZ routines will do a frame advance.

SET.............................................................. "Set" parameter from the old AUTOGRAPH - not recommended.

ROW.............................................................. Specifies how x/y coordinates are stored for the EZ routines.

INVERT......................................................... Switch which, if set, reverses roles of x and y coordinates.

WINDOW........................................................ Switch which, if set, causes "windowing" of curves drawn.

BACKGROUND............................................... Background type selector - the action of setting it causes other parameters to be set.

NULL 1.............................................................. The AUTOGRAPH "null 1" - used as a special parameter value and data value.

2................................................................. The AUTOGRAPH "null 2" - used as a special parameter value.

GRAPH LEFT.................................................. Parameters specifying the edges of the graph window, in which the entire graph, including labels, is to be positioned.

RIGHT
BOTTOM
TOP

GRID LEFT..................................................... Parameters specifying the edges of the grid window, in which a "grid" or "curve window" of a specified shape is to be placed. Curves are drawn in this box.

RIGHT
BOTTOM
TOP
SHAPE

X MINIMUM.................................................. Parameters specifying how user "x" data is to be mapped onto the horizontal axis of the curve window ("y" data if the INVERT flag is set).

MAXIMUM
LOGARITHMIC
ORDER
NICE

Y MINIMUM.................................................. Parameters specifying how user "y" data is to be mapped onto the horizontal axis of the curve window ("x" data if the INVERT flag is set).

MAXIMUM
LOGARITHMIC
ORDER
NICE
CONTROL .................................. ............
Flag which specifies whether or not axis "s" is drawn and gives AUTOGRAPH more or less freedom to shift things around.

LINE........................................ Auxillary flag – can be used to turn off the axis line itself.

INTERSECTION GRID..................... Parameters which can be set to non-null values to move the axis around.

FUNCTION.................................. The address of a routine which specifies the label-system / coordinate-system mapping.

[TICKS] MAJOR [SPACING] TYPE........ Parameters which specify how major ticks are to be placed along the axis.

PATRON .................. Dash pattern for major ticks.

[LENGTH] OUTWARD... Lengths of outward- and inward- pointing parts of major ticks.

MINOR SPACING..... Positions minor ticks.

PATTERN .... Dash pattern for minor ticks.

[LENGTH] OUTWARD... Lengths of outward- and inward- pointing parts of minor ticks.

[NUMERIC] TYPE ...................... Specify whether and, if so, how numeric labels are to be placed along the axis.

EXPOSIITON
FRACTION

ANGLE 1ST ............... First and second choice for orientation of numeric labels.

2ND

OFFSET .................. Gap between axis and labels.

WIDTH MANTISSA .......... Sizes of characters used for the two portions of numeric labels.

EXPOSIITON

DASH ADDRESS ............................. Parameters defining the "user" set of dashed-line patterns – the array address, the number of patterns, the index increment between patterns, the length of each, character sizes to use.
Can be used to turn off all labels - except numeric labels.

Not to be set - value is label buffer length.

Used to save and restore entire label buffer.

Used to pick up just the names of the labels currently defined.

A sort of switch, set to point to the label whose definition one wants to access.

The parameters defining a label, including a suppression flag, x and y basepoint coordinates, x and y offset-vector components, the orientation angle, the centering flag, a count of the number of lines in the label, and a pointer to the first of those lines in the line list.

Two parameters specifying how input text lines are scanned.

Not to be set - value is line buffer length.

Used to save and restore entire line buffer.

A sort of switch, set to point to the particular line whose definition is to be accessed.

The parameters defining a line, including a suppression flag, the character size, the text, the length of the text, and a pointer to the next line.

Secondary parameters, intended only to be retrieved. To my knowledge, no one has had occasion to use any of these. I would be hard put to think of a situation in which one would do so, though such situations probably do exist.

Note: "s" stands for any one of "LEFT", "RIGHT", "BOTTOM", or "TOP" and "b" for any one of "LEFT", "RIGHT", "BOTTOM", "TOP", "CENTER", or "GRAPH".
APPENDIX B: TRANSFORMATIONS
APPENDIX B: TRANSFORMATIONS

INTRODUCTION
Where it is appropriate, the graphics routines contain the facility to transform the lines drawn from the default positions to other positions on the screen resulting in a new shape for the picture. For example, a contour map generated from data stored in a rectangular array in core can be transformed to produce a polar coordinate plot.

IMPLEMENTATION
In all packages with transformations, statement functions containing the default transformations are included in the appropriate routines. (These transformations are very short and result in little or no run time overhead.) The user wanting other than the default transformations can replace the statement functions with new ones, or, if the transformation is complicated, delete the statement functions and add external routines with the same name to do the transforming.

The range of the arguments to the statement functions is dependent on the routine in use. (Explanations of several of the more common transformations used in contouring and velocity vectors given below illustrate this point.) The range of the default transformation is the same as that of the arguments. If a user-supplied transformation has a different range, it must be considered. This is usually done by calling SET before calling the plotting routine and suppressing the SET call within the plotting routine via one of its arguments.

EXAMPLES
The following examples assume that the range of arguments to the statement functions is the range found in all the contouring and velocity vector routines for data stored in rectangular arrays. In both types of routines, M by N arrays are supplied by the user. The default transformations are:

\[ FX(X,Y) = X \]
\[ FY(X,Y) = Y \]

X corresponds to the first subscript direction. Y corresponds to the second subscript direction. \(1 < X < M\) and \(1 < Y < N\), but X and Y can take on noninteger values. When \(X = 7.0\) and \(Y = 1.0\), the algorithm has generated a coordinate corresponding to the array value with subscripts \((1,1)\).

† There is zero run time overhead at NCAR.
Users writing their own transformation functions FX and FY to be used with CONREC or VELVCT should be aware of the following information concerning the order in which FX and FY are evaluated. The calls to FX and FY in CONREC and VELVCT are of the form:

\[
\text{CALL SUB (FX(X,Y),FY(X,Y))}
\]

The 7600 will compile this statement to cause FX to be evaluated before FY. This is not the case on the CRAY-1, however. Example 3 below does in fact assume that FX will be computed before FY; this example will run at NCAR only on the 7600. The FORTRAN standards do not allow any assumptions to be made about the order in which FX and FY are to be evaluated in the above call, and as a general rule, such assumptions should not be made. Ultimately, the utilities CONREC and VELVCT will be changed to force the evaluation of FX before FY. As a temporary measure, users may put sufficient code in FX and FY to insure that they are independent. If this proves to be too costly in computation time, users may wish to put a parity switch in common with FX and FY (assuming that FX and FY are called in pairs) so that each function knows whether it is being called first or second and appropriate action can be taken based on this knowledge.

**Example 1: Polar Coordinates**
Default transformations

\[ \begin{align*}
FX(X,Y) & = X \\
FY(X,Y) & = Y
\end{align*} \]

Defaults are replaced by

\[
\begin{align*}
\text{COMMON /TRANS/A,B,C,D} \\
FX(X,Y) & = (A+B*(X-1.)/(\text{FLOAT}(M)-1.))*\cos(C+D*(Y-1.)/(\text{FLOAT}(N)-1.)) \\
FY(X,Y) & = (A+B*(X-1.)/(\text{FLOAT}(M)-1.))*\sin(C+D*(Y-1.)/(\text{FLOAT}(N)-1.))
\end{align*}
\]

Note:

- A, B, C and D are in common with the routine that called the plotting routine.
- C and D are the desired angles in radians.
- A and B are the desired distances.
- Here and in the subsequent examples, SET was called by the routine that called the plotting routine in such a way to accommodate the numbers generated by FX and FY. CALL SET(.1,.9,.1,.9,0,A+B,0.,A+B,1) would work in this particular case.

Example 2: Orthogonal unequally spaced

Z(I,J) is located at (XX(I),YY(J))

Default transformations are replaced by

\[
\begin{align*}
\text{COMMON /TRANS/XX(100),YY(100)} \\
FX(X,Y) & = XX(\text{IFIX}(X))+(XX(\text{IFIX}(X)+1)-XX(\text{IFIX}(X)))*(X-\text{AINT}(X)) \\
FY(X,Y) & = YY(\text{IFIX}(Y))+(YY(\text{IFIX}(Y)+1)-YY(\text{IFIX}(Y)))*(Y-\text{AINT}(Y))
\end{align*}
\]

Note:

- XX(I) has been set to the proper abscissa for I=1,...,M.
- YY(J) has been set to the proper ordinate for J=1,...,N.
- XX(M+1) has been set equal to XX(M).
- YY(N+1) has been set equal to YY(N).

Example 3: Overlaying on SUPMAP

The default transformations are deleted and the following routines are added to the program:
FUNCTION FX(X, Y)
COMMON /TRANS/M,N,XLONMI,XLONMA,YLATMI,YLATMA,YANS
XLON=XLONMI+(XLONMA-XLONMI)*(X-1.)/(FLOAT(M)-1.)
YLAT=YLATMI+(YLATMA-YLATMI)*(Y-1.)/(FLOAT(N)-1.)
CALL SUPCON(YLAT,XLON,XANS,YANS)
FX=XANS
RETURN
END

FUNCTION FY(X, Y)
COMMON /TRANS/M,N,XLONMI,XLONMA,YLATMI,YLATMA,YANS
FY=YANS
RETURN
END

Note:

* SUPMAP must be called before CONREC or VELVCT
* The longitude of the data are equally spaced between XLONMI and XLONMA.
* The latitudes of the data are equally spaced between YLATMI and YLONMA. Z(1,1) is at longitude XLONMI, latitude YLATMI.
* M,N,XLONMI,XLONMA,YLATMI, and YLATMA are initialized in the user's routine which calls the plotting routines.

Example 4: A parameterized distortion, 30 by 20 example

Z(I,J) is at (XX(I,J),YY(I,J))

The default transformations are deleted and the following routines are added to the program:
FUNCTION FX(X,Y)
  COMMON /TRANS/XX(30,20),YY(30,20),YS
  I=X
  J=Y
  IF(FLOAT(I).EQ.X) GO TO 1
  C Y=J
  FX=XX(I,J)+(XX(I+1,J)-XX(I,J))*(X-FLOAT(I))
  YS=YY(I,J)+(YY(I+1,J)-YY(I,J))*(Y-FLOAT(J))
  RETURN
  C TEST FOR CORNER
  1 IF(FLOAT(J).EQ.Y) GO TO 2
    FX=XX(I,J)+(XX(I,J+1)-XX(I,J))*(Y-FLOAT(J))
    YS=YY(I,J)+(YY(I,J+1)-YY(I,J))*(Y-FLOAT(J))
    RETURN
  2 FX=XX(I,J)
    YS=YY(I,J)
    RETURN
END
FUNCTION FY(X,Y)
  COMMON /TRANS/XX(30,20),YY(30,20),YS
  FY=YS
  RETURN
END
APPENDIX C: PROCESSING PARTS OF ARRAYS
APPENDIX C

APPENDIX C: PROCESSING OF PARTS OF ARRAYS

INTRODUCTION
The standards for argument lists of functions and subroutines in ULIB require a set of arguments (described below) which make it possible to process not only the entire array, but any part (sub-array) from the array.

REQUIRED ARGUMENTS
When an N dimensional array is being passed to a subroutine, the following arguments are required: The name (origin) of the array, its first N-1 dimensions in the dimension statement of the calling routine, and the number of array values to be processed in each of the N dimensions. (It is not necessary to pass all N dimensions from the dimension statement of the calling program because only the first N-1 are used in the addressing algorithm.)

Please refer to the examples on the following pages.
Example 1: Processing all of an array

PROGRAM MAIN
DIMENSION A(100), B(30,20), C(22,16,9)
C THESE CALLS PROCESS ALL OF EACH ARRAY
CALL LINEAR (A, 100)
CALL TWODIM (B, 30, 30, 20)
CALL THREED (C, 22, 22, 16, 9)
STOP
END
SUBROUTINE LINEAR (U, N)
DIMENSION U(N)
DO 1 I = 1, N
(Process U array)
1 CONTINUE
RETURN
END
SUBROUTINE TWODIM (V, LX, MX, MY)
DIMENSION V(LX, MY)
DO 1 I = 1, MX
DO 1 J = 1, MY
(Process V array)
1 CONTINUE
RETURN
END
SUBROUTINE THREED (W, LX, MX, LY, MY, MZ)
DIMENSION W(LX, LY, MZ)
DO 1 I = 1, MX
DO 1 J = 1, MY
DO 1 K = 1, MZ
(Process W array)
1 CONTINUE
RETURN
END
These same arguments allow a user to process any array contained within another array. When the user is doing this, the array argument can be used as the origin of the sub-array. The following examples demonstrate the processing of parts of arrays (and assume the same subroutines as listed in Example 1).

**Example 2:** Processing part of an array

```
PROGRAM MAIN
DIMENSION A(100), B(30,20), C(22,16,9)
C PROCESS FROM A(25) THROUGH A(80), See Figure 1.
C (80-25+1=56)
CALL LINEAR (A(25), 56)
C PROCESS PART OF B STARTING AT B(5,3) AND
C INCLUDING 11 B's IN THE FIRST SUBSCRIPT DIRECTION
C AND 10 B's IN THE SECOND SUBSCRIPT DIRECTION.
C (A TOTAL OF 11*10=120 B's.) See Figure 2.
CALL TWODIM (B(5,3), 30, 11, 10)
C PROCESS A CUBE IN C WITH ONE CORNER HAVING
C SUBSCRIPTS (7,4,2) AND THE OPPOSITE
C CORNER AT (18,14,8)
CALL THREED (C(7,4,2), 22, 12, 16, 11, 7)
STOP
END
```

This concept is consistent with standard FORTRAN and should be portable.

**Figure 1: ARRAY A**
Figure 2: ARRAY B

PART TO BE PROCESSED
Figure 3: ARRAY C
INDEX

Absolute value,
HAFTON-2,
PWRZI-2,
PWRZT-2
Accuracy, 2-2
DASHCHAR-4
DASHLINE-2
DASHSMTH-3
DASHSUPR-4
EZMAP-11
ISOSRFHR-4
SRFACE-4
SUPMAP-6
THREED-2
WINDOW-2
Activate or deactivate
options, CONRAQ-3
CONRAS-3
CONRAN-3
Address space,
plotter, DASHCHAR-3
AG.,
Routines, AUTOGRAPH-7
AGBACK, AUTOGRAPH-3
AGCURV, AUTOGRAPH-3
AGCURV,
Routine, AUTOGRAPH-8
AGGETF, AUTOGRAPH-2
AGGETI, AUTOGRAPH-2
AGGETP, AUTOGRAPH-2
AGGETP, AUTOGRAPH-2
AGGETP, AUTOGRAPH-2
AGSTUP, AUTOGRAPH-3
AGSTUP,
Routine, AUTOGRAPH-8
Algorithm, 2-2
CONRAN-2
CONRAQ-2
CONRAS-2
CONREC-4
CONRECQCK-1
DASHCHAR-3
DASHLINE-2
DASHSMTH-3
DASHSUPR-4
EZMAP-11
HAFTON-4
ISOSRFHR-3
PWRITY-2
SCROLL-5
SRFACE-3
SUPMAP-6
VELVCT-3
WINDOW-2
Angle in degrees, SRFACE-1
Angles,
negative, SRFACE-1
positive, SRFACE-1
Angular distance, SUPMAP-3
Angular movement, SUPMAP-2
Annotated axes, THREED-2
ANOTAT, AUTOGRAPH-2
ANOTAT,
Routine, AUTOGRAPH-4
Apex, SUPMAP-13
Approximation, ISOSRF-1
Arbitrary object, ISOSRFHR-1
Arc of a circle, SUPMAP-13
Arcsine mapping, HAFTON-3
Area,
triangulated, CONRAS-2
Area Specification,
Map, EZMAP-4
Area,
Lambert equal, SUPMAP-2
Areas,
Contiguous special value, HAFTON-3
Argument list, ISOSRFHR-3
Arguments, 2-1
Arguments,
Dimension of, 2-1
CONRAN-1
CONRAQ-1
CONRAS-1
CONREC-1
CONRECQCK-1
CONRECSMTH-1
CONRECSUPR-1
EZMAP-1
HAFTON-1
ISOSRF-1
ISOSRFHR-1
PWRITX-1
PWRITY-1
PWRZI-1
PWRZS-1
PWRZT-1
Arguments,
fixed or floating point,
DASHCHAR-1
DASHLINE-1
WINDOW-2

Arguments,
DASHLINE-1
DASHSUPR-1
THREED-1
WINDOW-1

Arguments to the Routines,
AUTOGRAPH-3

Array,
circular list, STRMLN-2
CONREC-1
integer work, CONRAN-1
CONRAS-2
large, ISOSRFHR-1
linear, SURFACE-2
model picture, DASHSUPR-1
of heights: SURFACE-1
real work, CONRAN-1
CONRAS-1
rectangular, CONRECQCK-1
CONRECSUPR-1
HAFTON-1
scratch, CONRAN-2
CONRAQ-2
CONRAS-2
ISOSRFHR-2
SURFACE-2
three-dimensional, ISOSRF-1
work, CONRAQ-1

Arrow,
length of the, VELVCT-1

Assumptions,
HAFTON-1
SCROLL-1
SURFACE-1
STRMLN-1
VELVCT-1

Author,
CONRAN-15
CONRAQ-12
CONRAS-14

AUTOGRAPH Control Parameter
Routines, AUTOGRAPH-2

AUTOGRAPH-1
Axes,
major and minor, SUPMAP-13
annotated, THREED-2
Azimuthal Default (AD), EZMAP-6
Azimuthal Equidistant, SUPMAP-2
Base,
vector, VELVCT-5
Bi-linear interpolation, HAFTON-4
Blank frames, SCROLL-1
Block USER,
Common, PWRITX-1

Blocks,
Common, 2-1
AUTOGRAPH-9
CONRAN-2
CONRAQ-2
CONRAS-2
CONREC-3
CONRECQCK-1
CONRECSUPR-1
DASHCHAR-3
DASHLINE-2
DASHSMTH-3
DASHSUPR-3
EZMAP-10
HAFTON-4
ISOSRFHR-3
PWRITX-6
PWRITY-2
SCROLL-4
SRFACE-3
STRMLN-3
SUPMAP-5
THREED-2
VELVCT-3
WINDOW-2

Border, STRMLN-2
grid and, SUPMAP-4
draw a, VELVCT-2

Buffer (IBUF),
Workspace, SCROLL-4
Workspace, SCROLL-2

C1 Surface Interpolation,
Lawson's, CONRAQ-2

Calling program, HAFTON-2

Capabilities, 1-1
smoothing, DASHSMTH-1
smoothing, DASHSUPR-1
character, CONRECSMTH-1
clipping, WINDOW-1

Card,
A header, SCROLL-2
continuation, SCROLL-3
Cards,
data, SCROLL-2
second dimension of, SCROLL-3
text, SCROLL-2
Cell, CONRECQCK-1
Cells in the x-direction, ISOSRFHR-2
Center, SCROLL-3
Centered horizontally., SCROLL-1
Centering option, SCROLL-3
Central meridian, SUPMAP-3
Change the data output format, CONRAN-3
Character capability, CONRECSMTH-1
Character height, SCROLL-3
Character labeling,
perspective, SRFACE-3
Character plotting routine,
PWRZI-1
PWRZS-1
PWRZT-1
Character problem,
hidden, PWRZT-1
PWRZS-1
Character width, PWRZI-1
Characters,
draw, DASHSUPR-1
Characters,
PWRZI-1
PWRZS-1
PWRZT-1
Characters,
relative size of, SCROLL-2
CHL, CONRAN-3
CONRAQ-3
CONRAS-3
GIL, CONRAN-1
CONRAQ-4
CONRAS-4
Circle,
arc of a, SUPMAP-13
Circular list arrays, STRMLN-2
Clipping capability, WINDOW-1
Codes,
Function, PWRITX-2
Common Block PUSER, PWRITX-1
Common Blocks, 2-1
AUTOGRAPH-9
CONRAN-2
CONRAQ-2
CONRAS-2
CONREC-3
CONRECQCK-1
CONRECSUPR-1
DASHCHAR-3
DASHLINE-2
DASHSMTH-3
DASHSUPR-3
EZMAP-10
HAFTON-4
ISOSRFHR-3
PWRITX-6
PWRITY-2
SCROLL-4
SRFACE-3
STRMLN-3
SUPMAP-5
THREED-2
VELVCT-3
WINDOW-2
Computes a scale factor, VELVCT-5
CON, CONRAN-4
CONRAQ-4
CONRAS-4
Conditions,
Special, AUTOGRAPH-9
SRFACE-3
Cone constant, SUPMAP-13
Configuration,
standard, CONREC-2
STRMLN-2
VELVCT-2
Conformal conic,
Lambert, SUPMAP-1
SUPMAP-2
SUPMAP-3
Conic,
Lambert conformal, SUPMAP-1
SUPMAP-2
Conical projection, SUPMAP-13
CONRAN Error Messages, CONRAN-15
CONRAQ Error Messages, CONRAQ-12
CONRAS Error Messages, CONRAS-15
CONREC,
On Input for, CONREC-2
On Output for, CONREC-3
Constant,
10-bit, CONRECOCK-1
cone, SUPMAP-13
Consuming,
time, ISOSRFHR-4
Contiguous special
value areas, HAFTON-3
Continental outlines, SUPMAP-1
Continental, SUPMAP-1
Continuation card, SCROLL-3
Continuous lines, SUPMAP-1
Contour level, highest, CONREC-2
lowest, CONREC-2
Contour Levels, CONREC-1
CONREC-2
Contour Lines, CONREC-1
CONRECQCK-1
CONRECSMTH-1
CONRECUSPR-1
label the, CONRAS-1
plots, CONRAQ-1
Contour Map, CONRAN-2
CONRAN-2
CONRAQ-2
CONREC-1
CONRECQCK-1
CONRECSMTH-1
CONRECUSPR-1
plots the, CONRAS-2
Contoured, CONREC-1
Contouring, CONRAQ-1
Contouring levels,
Non-uniform, CONREC-3
Contours,
different styled, CONRAN-14
Contours,
Options Which Effect the, CONRAN-14
CONRAQ-11
CONRAS-14
Control Parameter Routines,
AUTOGRAPH, AUTOGRAPH-2
Control Routines, EZMAP-3
Control scaling, STRMLN-2
Flag to, VELVCT-2
Coordinate,
the MX, SCROLL-3
WINDOW-2
plotter, ISOSRFHR-3
positioning, PWRZT-1
X, SRFACE-2
Y, SRFACE-2
Cost,
Picture Quality vs., 1-1
Counter-clock wise, SRFACE-1
Crowded lines, CONRECSUPR-1
removing, DASHSUPR-1
CRT units, SCROLL-3
VELVCT-2
Cumulative error,
DASHCHAR-4
DASHLINE-2
DASHSMTH-3
Current grid resolution,
PWRZS-1
PWRZT-1
Cylindrical Default (CD), EZMAP-6
Cylindrical equidistant, SUPMAP-2
Cylindrical projections, SUPMAP-3
DANDR (Draw AND Remember), ISOSRFHR-1
DAS, CONRAN-5
DAS, CONRAS-5
Dash line generators, DASHLINE-1
DASHCHAR, -1
DASHD,
On Input for, DASHCHAR-2
DASHSMTH-2
DASHSUPR-2
Dashed line generators, DASHCHAR-1
Dashed line package, DASHCHAR-1
Dashed line pattern, CONREC-1
CONREC-3
CONRECQCK-1
Data cards, SCROLL-2
Data,
irregularly distributed, CONRAN-1
triangulated, CONRAQ-1
sparse, CONRAQ-2
Data files, SUPMAP-7
Data output format,
change the, CONRAN-3
Data output format,
change the, CONRAN-3
Data point values, CONRECSQCK-2
Data points,
X-coordinates of the, CONRAN-1
Y-coordinates of the, CONRAN-1
Data points, CONRAQ-1
CONRAN-1
Data sets,
irregular, CONRAN-1
CONRAQ-1
CONRAS-1
random, CONRAN-1
CONRAQ-1
CONRAS-1
sparse, CONRAN-1
CONRAQ-1
CONRAS-1
Data values, HAFTON-1
Data values at the points,
CONRAN-1
Data values, CONRAQ-1
CONREC-2
HAFTON-2
DBP, CONRAN-5
DBP, CONRAS-5
dd80, HAFTON-4
Deactivate,
activate or, CONRAQ-3
Deactivate options,
activate or, CONRAS-3
activate or, CONRAN-3
DEF, CONRAN-5
CONRAQ-4
CONRAS-5
Default, EZMAP-6
Default Values,
Option, CONRAQ-13
Optional, CONRAN-10
CONRAS-12
Default Values for User
Specified Parameters,
CONRAN-13
CONRAQ-10
CONRAS-13
Defined,
object is, ISOSRFHR-1
earth, SUPMAP-1
Degrees,
Angle in, SRFACE-1
Dependent,
Machine word-length, CONRECSUPR-2
Derivatives,
estimated partial, CONRAN-3
Development methods,
film, HAFTON-2
Different styled contours, CONRAN-14
dimension,
First, ISOSRF-1
Second, ISOSRF-1
Dimension of Arguments, 2-1
CONRAN-1
CONRAQ-1
CONRAS-1
CONREC-1
CONRECKOCK-1
CONRECSMTH-1
CONRECSUPR-1
EZMAP-1
HAFTON-1
ISOSRF-1
ISOSRFHR-1
ISOSRFHR-1
PWRITX-1
PWRITY-1
PWRZI-1
PWRZS-1
PWRZT-1
SCROLL-1
SRFACE-1
STRMLN-1
SUPMAP-1
VELVCT-1
Dimension of CARDS,
Second, SCROLL-3
Dimension,
The first, STRMLN-2
Direct mapping, HAFTON-2
Direction,
the first subscript, CONREC-2
the second subscript, CONREC-2
the first subscript, HAFTON-2
the second subscript, HAFTON-2
the second subscript, ISOSRF-1
the third subscript, ISOSRF-2
first subscript, STRMLN-2
second subscript, STRMLN-2
Directness, HAFTON-2
Discontinuities,
extreme, SRFACE-3
Disjoint union, SUPMAP-7
DISPLA, AUTOGRAPH-2
DISPLA,
Routine, AUTOGRAPH-6
Distance,
angular, SUPMAP-3
Distance traversed, DASHLINE-2
Distortion, SUPMAP-12
WINDOW-1
Distributed data,
irregularly, CONRAN-1
DOT, EZMAP-6
Draw a border., VELVCT-2
Draw AND Remember,
DANDR, ISOSRFHR-1
Draw characters, DASHSUPR-1
Drawn,
perimeter, HAFTON-1
Draws the map, SUPMAP-5
DSHL, AUTOGRAPH-6
Dummy, SUPMAP-2
Earth defined, SUPMAP-1
Effect the Contours,
Options Which, CONRAN-14
CONRAQ-11
CONRAS-14
Ellipse, SUPMAP-13
Endpoints of the vector, VELVCT-3
Endpoints, WINDOW-2
Entry Points, 2-1
AUTOGRAPH-9
CONRAN-2
CONRAQ-2
CONRAS-2
CONREC-3
CONRECQCK-1
CONRECSMTH-2
CONRECSUPR-1
DASHCHAR-3
DASHLINE-2
DASHSMTH-3
DASHSUPR-3
EZMAP-6
HAFTON-4
ISOSRFHR-3
PWRITX-6
PWRITY-2
SCROLL-4
SRFACE-3
STRMLN-2
SUPMAP-5
THREED-2
VELVCT-3
WINDOW-2

Equal area,
Lambert, SUPMAP-2
Equally spaced, SRFACE-1
Equator, SUPMAP-2
Equidistant,
Azimuthal, SUPMAP-2
Cylindrical, SUPMAP-2

Error,
cumulative, DASHCHAR-4
DASHLINE-2
DASHSMTH-3

Error Messages,
CONRAN, CONRAN-15
CONRAQ, CONRAQ-12
CONRAS, CONRAS-15

Error routines,
port library, CONRAN-2
CONRAS-2

Errors,
visibility, SRFACE-3

Estimated partial derivatives, CONRAN-3
Example, SCROLL-8
Exponential,
mapping is, HAFTON-3
EXT, CONRAN-6
CONRAQ-5
CONRAS-6

Extra information, CONREC-3

Extreme discontinuities, SRFACE-3

EYE, ISOSRFHR-2
Eye position, ISOSRFHR-2
Eye,
The viewer's, SRFACE-2
EZ.,
Routines, AUTOGRAPH-3
EZCNTR,
On Input for, CONREC-1
On Output for, CONREC-1
EZCNTR, CONRECSMTH-1
EZCNTR, CONRECSUPR-1

EZHFTN, HAFTON-1
EZHFTN,
On Input for, HAFTON-1
On Output for, HAFTON-2

EZMAP,
Subroutine, EZMAP-1
EZMAP, SUPMAP-1
EZMAP,
On Input for, SUPMAP-2
On Output for, SUPMAP-2
EZMY, AUTOGRAPH-1
EZSRFC, SRFACE-1
EZSRFC,
On Input for, SRFACE-1
On Output for, SRFACE-2

EZSTRM, STRMLN-1
EZXY, AUTOGRAPH-1
EZY, AUTOGRAPH-1

Factor,
scale, VELVCT-1
Computes a scale, VELVCT-5

Factors,
scale, CONREC-1
VELVCT-3

Fastest member, CONRAQ-1

Feature,
special value, HAFTON-3
VELVCT-2

Features,
smoothing, DASHSMTH-1
nonstandard, SUPMAP-6

Fence lines, THREED-2

Field,
flow, STRMLN-1
strength of the, VELVCT-1
two-dimensional velocity, VELVCT-1
vector, VELVCT-3
velocity, STRMLN-1

File,
scratch, ISOSRFHR-2

Files,
data, SUPMAP-7
Film development methods, HAFTON-2
Fine-tune a plot,
CONRAN-14
CONRAQ-11
CONRAS-14
First dimension,
ISOSRF-1
STRMLN-2
First point of a line,
DASHSMTH-1
DASHSUPR-1
First subscript direction,
CONREC-2
HAFTON-2
ISOSRF-1
STRMLN-2
Fixed or floating point arguments, DASHCHAR-1
DASHLINE-1
WINDOW-2
Flag, HAFTON-2
Flag to control scaling, VELVCT-2
Flexibility, 1-2
FLO, HAFTON-2
Floating point arguments,
fixed or, DASHCHAR-1
DASHLINE-1
WINDOW-2
Floating point numbers, PWRZI-1
Floating point, WINDOW-1
Flow field, STRMLN-1
Flow speed, STRMLN-1
FMT, CONRAN-6
CONRAQ-5
CONRAS-6
For MTITLE,
On Input, SCROLL-1
On Output, SCROLL-2
For SCROLL,
On Input, SCROLL-2
On Output, SCROLL-4
Format,
change the data output, CONRAN-3
Formulas,
interpolation, STRMLN-2
Frame, DASHSUPR-1
Frames,
blank, SCROLL-1
title, SCROLL-1
Function Codes, PWRITX-2
Functions,
modify the interpolating, CONRAQ-11
statement, CONRAQ-11
modify the interpolating, CONRAS-14
Internal, VELVCT-5
FURA, AUTOGRAPH-7
FUSR, AUTOGRAPH-7
Generators,
dashed line, DASHCHAR-1
DASHLINE-1
Geo-political, SUPMAP-7
GLAB, AUTOGRAPH-4
Gnomonic, SUPMAP-2
Graphical output, SCROLL-4
Graphics Routines,
Interfacing into Other, CONRAN-14
CONRAQ-11
CONRAS-14
Graphics Utilities,
Overview Of the SCD, 1-1
Gray scale, HAFTON-7
Greenwich meridian, SUPMAP-2
GRI, CONRAN-6
CONRAQ-5
CONRAS-6
Grid and border, SUPMAP-4
Grid,
plot a, CONRAN-1
virtual, CONRAN-2
virtual, CONRAQ-2
plot a perimeter or, CONRAS-1
virtual, CONRAS-2
GRID, EZMAP-5
Grid is scanned, CONRAS-2
Grid resolution, PWRZI-1
Grid resolution,
current, PWRZS-1
current, PWRZT-1
Grid space,
The, CONRECQCK-1
Grids, SUPMAP-1
Group of lines, SCROLL-1
HAFTON,
On Input for, HAFTON-2
On Output for, HAFTON-4
Half-tone picture, HAFTON-1
Half-tone plot, HAFTON-1
Hardware, DASHCHAR-1
Head,
vector, VELVCT-5
Header card,
A, SCROLL-2
Height,
character, SCROLL-3
Heights,
array of, SRFACE-1
How To Use

IABS(JLTS) = 1, SUPMAP-3
IABS(JLTS) = 2, SUPMAP-3
IABS(JLTS) = 3, SUPMAP-3
IABS(JLTS) = 4, SUPMAP-3
IABS(JLTS) = 5, SUPMAP-3
ICNT, PWRITY-2
ID, PWRITY-1
IDIOT, AUTOGRAPH-2
IDOT, SUPMAP-4
IDPC, PWRTXT-2
IDXY, AUTOGRAPH-3
IER, SUPMAP-4
IIVX, AUTOGRAPH-8
Image plane model, ISOSRFHR-2
Importance,
lines of major, DASHSUPR-4
lines of minor, DASHSUPR-4
In degrees,
Angle, SRFACE-1
In MTITLE, SCROLL-6
In SCROLL, SCROLL-7
Increasing NCP, CONRAQ-11
Information,
extra, CONREC-3
Visibility, SRFACE-3
INIT3D, ISOSRFHR-1
Input,
On, CONRAN-1
On, CONQAQ-1
On, CONRAS-1
On, DASHLINE-1
On, EZMAP-3
On, ISOSRF-1
On, ISOSRFHR-2
On, PWRITY-1
On, PWRZI-1
On, PWRZS-1
On, PWRZT-1
On, STRMLN-1
On, THREED-1
On, VELVCT-1
On, WINDOW-1
Input for CONREC,
On, CONREC-2
Input for DASHD,
On, DASHCHAR-2
Input for DASHD,
On, DASHSMTH-2
Input for DASHD,
On, DASHSUPR-2
Input for DASHD,
On, DASHSUPR-2
Input for EZCNTR,
On, CONREC-1
Input for EZHFTN,
On, HAFTON-1
Input for EZMAP,
On, SUPMAP-2
Input for EZSRFC,
On, SRFACE-1
Input for HAFTON,
On, HAFTON-2
Input For MTITLE,
On, SCROLL-1
Input For SCROLL,
On, SCROLL-2
Input for SRFACE,
Input points, number of, CONRAN-3
Input points, number of, CONRAS-3
Input, PWRITX-1
Instructions, plotter, HAFTON-4
INT, CONRAN-6
CONRAQ-6
CONRAS-7
Integer work array, CONRAN-1
CONRAN-2
Intensity, HAFTON-1
highest, HAFTON-1
levels, HAFTON-2
Interfacing Into Other Graphics Routines, CONRAN-14
CONRAQ-11
CONRAS-14
Internal functions, VELVCT-5
Internal parameter list, SCROLL-4
Internal Parameters, 2-2
CONREC-3
CONRECKQ-2
CONRECUSR-2
DASHLNE-3
HAFTON-5
PWRITX-7
SCROLL-6
SRFACE-4
DASHSMTH-3
DASHSUPR-3
Isosurface, EZMAP-11
HAFTON-4
ISOSRFHR-3
PWRITX-6
PWRITY-2
SRFACE-3
SUPMAP-5
THREED-2
VELVCT-3
WINDOW-2
Irregular data sets,
CONRAN-1
CONRAQ-1
CONRAS-1
Irregularly distributed data, CONRAN-1
IS2, ISOSRFHR-2
ISIZE, PWRITX-1
Iso-valued surface, ISOSRF-1
ISPV, HAFTON-3
ITHETA, PWRITY-1
IU, ISOSRFHR-2
IUSOUT, SUPMAP-4
IUSR, AUTOGRAPH-7
IWHICH, EZMAP-7
JCR, DASHSUPR-3
JCTR, PWRITX-2
JGRID, SUPMAP-4
JLTS, EZMAP-1
JLTS, EZMAP-4
JPROJ, EZMAP-3
JPROJ, SUPMAP-2
JSIZE, DASHSUPR-3
JSIZE, PWRITX-2
KDSH, AUTOGRAPH-8
LAB, CONRAN-7
LAB, CONRAS-7
LABEL, EZMAP-6
Label the contour lines, CONRAS-1
Labeled and smoothed, CONRECSMTH-1
Labeled and smoothed, CONRECUSR-1
Labeling,
perspective character, SRFACE-3
Labels, CONRECQCK-1
Lambert conformal conic, SUPMAP-1
Lambert conformal conic, SUPMAP-2
Lambert conformal, SUPMAP-3
Lambert equal area, SUPMAP-2
Language, 2-1
  AUTOGRAPH-10
  CONRAN-2
  CONRAQ-2
  CONREC-4
  CONRECQCK-1
  DASHCHAR-3
  DASHLINE-2
  DASHSMTH-3
  DASHSUPR-4
  EZMAP-11
  HAFTON-4
  ISOSRFHR-3
  PWRITX-6
  PWRITY-2
  SCROLL-5
  SRFACE-3
  SUPMAP-6
  THREED-2
  VELVCT-3
  WINDOW-2
Large array, ISOSRFHR-1
Last point of a line, DASHSMTH-1
Latest Revision, 2-1
  AUTOGRAPH-1
  CONRAN-1
  CONRAQ-1
  CONRAS-1
  CONREC-1
  CONRECQCK-1
  CONRECQSMTH-1
  CONRECSUPR-1
  DASHCHAR-1
  DASHLINE-1
  DASHSMTH-1
  DASHSUPR-1
  EZMAP-1
  HAFTON-1
  ISOSRF-1
  ISOSRFHR-1
  PWRITX-1
  PWRITY-1
  PWRZI-1
  PWRZS-1
  PWRZT-1
  SCROLL-1
  SRFACE-1
  SUPMAP-1
  THREED-1
  VELVCT-1
  WINDOW-1
  Latitude and longitude, SUPMAP-1
  Lawson's C1 surface interpolation, CONRAQ-2
  LBAC, AUTOGRAPH-4
  Length of a vector, Computes the, VELVCT-5
  Length of the arrow, VELVCT-1
  Length, minimum vector, VELVCT-2
  Level, highest contour, CONREC-2
  lowest contour, CONREC-2
  Level Routines, Lower, AUTOGRAPH-3
  Levels, contour, CONREC-1
  CONREC-2
  non-uniform contouring, CONREC-3
  intensity, HAFTON-2
  LFRA, AUTOGRAPH-6
  Library error routines, port, CONRAS-2
  Library Routines, Required, 2-1
    CONRAN-2
    CONRAQ-2
    CONRAS-2
  Limit, practical, DASHCHAR-2
  DASHSUPR-2
  Line, first point of a, DASHSMTH-1
  last point of a, DASHSMTH-1
  remaining points of a, DASHSMTH-1
  first point of a, DASHSUPR-1
  remaining points of a, DASHSUPR-1
  Line generators,
dashed, DASHCHAR-1
dash, DASHLINE-1
  Line of sight, SRFACE-2
  Line package,
dashed, DASHCHAR-1
  Line pattern,
dashed, CONREC-1
  CONREC-3
  CONRECQCK-1
  Line segment, WINDOW-2
  Linear
array, SRFACE-2
mapping, HAFTON-2
Linearly spaced, HAFTON-1
Lines,
  contour, CONREC-1
  CONRECQCK-1
  CONRECSMTH-1
  CONRECSUPR-1
crowded, CONRECSUPR-1
label the contour, CONRAS-1
negative, CONREC-1
of major importance, DASHSUPR-4
of minor importance, DASHSUPR-4
plots contour, CONRAQ-1
removing crowded, DASHSUPR-1
smooth, DASHSUPR-4
solid, CONREC-3
  CONRECQCK-1
Lines removed,
  continuous, SUPMAP-1
fence, THREED-2
group of, SCROLL-1
hidden, ISOSRF-1
hidden, SRFACE-1
skips plotting, SCROLL-5
List arrays,
circular, STRMLN-2
List,
  argument, ISOSRFHR-3
Internal parameter, SCROLL-4
Logarithmic,
mapping, HAFTON-3
Longest vector, VELVCT-2
Longitude,
  latitude and, SUPMAP-1
LOT, CONRAN-1
  CONRAQ-6
  CONRAS-7
Lower Level Routines, AUTOGRAPH-3
Lowest contour level, CONREC-2
Lows, CONREC-1
  highs and, CONREC-3
LROW, AUTOGRAPH-6
LSET, AUTOGRAPH-4
LSZ, CONRAN-7
LSZ, CONRAS-7
LTYP, AUTOGRAPH-7
LURA, AUTOGRAPH-7
LX, ISOSRFHR-2
M, HAFTON-1
M, HAFTON-2
Machine word-length
  dependent, CONRECSUPR-2
Machine-dependent, ISOSRFHR-4
Major and minor axes, SUPMAP-13
Major importance,
  lines of, DASHSUPR-4
Making multiple runs, CONRAN-2
Manual,
  How to Use this, 2-1
  MANY, AUTOGRAPH-3
Map Area Specification, EZMAP-4
Map,
  contour, CONRAN-2
  CONRAQ-2
  CONREC-1
  CONRECQCK-1
  CONRECSMTH-1
  CONRECSUPR-1
draws the, SUPMAP-5
plots the contour, CONRAS-1
MAPDRW, EZMAP-8
MAPFST, EZMAP-9
MAPGRD, EZMAP-8
MAPI, EZMAP-9
MAPI, EZMAP-10
MAPIT, EZMAP-9
MAPL, EZMAP-8
MAPLOT, EZMAP-8
MAPLOT, SUPMAP-5
MAPOPT, EZMAP-7
mapping, CONRAQ-11
  direct, HAFTON-2
  inverse, HAFTON-2
Arcsine, HAFTON-3
  Sinusoidal, HAFTON-3
Mapping is exponential, HAFTON-3
Mapping is linear, HAFTON-2
Mapping is logarithmic, HAFTON-3
MAPROJ, EZMAP-7
MAPRS, EZMAP-7
Maps,
  publication quality, CONRAS-1
MAPSET, EZMAP-7
MAPTO, EZMAP-9
MAPTRN, EZMAP-9
MAPTRP, EZMAP-10
MAPVE, EZMAP-9
MAPVP, EZMAP-10
Mark the regions, DASHSUPR-1
Marks,
  tick, CONREC-2
Mathematical method, SUPMAP-7
Maximum values,
  the minimum and, SUPMAP-3
Member,
fastest, CONRAQ-1
Mercator, SUPMAP-2
Meridian,
Greenwich, SUPMAP-2
central, SUPMAP-3
Meridians,
the, SUPMAP-13;
MES, CONRAN-8
CONRAQ-6
CONRAS-8
Messages,
CONRAN Error, CONRAN-15
CONRAQ Error, CONRAQ-12
CONRAS Error, CONRAS-15
Method,
mathematical, SUPMAP-7
Methods,
film development, HAFTON-2
Mid-latitude regions, SUPMAP-14
Minimum and maximum values,
the, SUPMAP-3
Minimum vector length, VELVCT-2
Minor axes,
major and, SUPMAP-13
Minor importance,
lines of, DASHSUPR-4
Mode,
production, SCROLL-1
Model, ISOSRFHR-2
image plane, ISOSRFHR-2
picture array, DASHSUPR-1
picture, DASHSUPR-1
DASHSUPR-4
Modifications, CONREC-3
Modifier,
PWRITX, SCROLL-3
Mollweide type, SUPMAP-2
More history, SUPMAP-7
Movement,
angular, SUPMAP-2
Movie, SCROLL-1
Movie sequence, SCROLL-5
Movie titles, SCROLL-2
scrolled, SCROLL-1
MTITLE, SCROLL-1
On Input For, SCROLL-1
On Output For, SCROLL-2
In, SCROLL-6
Multiple runs,
making, CONRAN-2
MV, ISOSRFHR-3
MX coordinate,
the, SCROLL-3
MY coordinate,
the, SCROLL-3
N, HAFTON-2
N, PWRITY-1
Name,
Utility, 2-1
NC, DASHSUPR-3
NCHAR, PWRITX-2
NCP, CONRAN-8
CONRAQ-7
CONRAS-8
increasing, CONRAQ-11
NDSH, AUTOGRAPH-5
Negative lines, CONREC-1
Negative angles, SRFACE-1
Negative,
If NSET is, VELVCT-2
NEVX, AUTOGRAPH-8
NEXY, AUTOGRAPH-8
Nice values, CONREC-2
Nine projections, SUPMAP-7
NLEV, HAFTON-2
Non-standard features, SUPMAP-6
Non-uniform contouring levels, CONREC-3
NOPT, HAFTON-2
Note, AUTOGRAPH-9
CONRAN-14
CONRAQ-11
CONRAS-14
CONREC-3
CONRECSMTH-1
CONRECSUPR-1
DASHLINE-1
DASHSMTH-3
DASHSUPR-3
EZMAP-6
HAFTON-4
ISOSRFHR-4
PWRITX-5
SCROLL-4
SRFACE-3
THREED-1
VELVCT-3
WINDOW-3
NPRM, HAFTON-3
NPTS, AUTOGRAPH-4
NSET is negative,
if, VELVCT-2
NSET is positive,
if, VELVCT-2
NU, ISOSRFHR-2
Number of input points, CONRAN-3
CONRAS-3
Number of parameters, CONRAQ-3
Numbers,
  floating point, PWRZI-1
Numerical value, CONRAN-3
  CONRAS-3
  NV, ISOSRFHR-2
  NVIX, AUTOGRAPH-8
  NW, ISOSRFHR-2
  NX, ISOSRFHR-2
  NY, ISOSRFHR-2
Object is defined, ISOSRFHR-1
Object,
  arbitrary, ISOSRFHR-1
On Input,
  CONRAN-1
  CONRAQ-1
  CONRAS-1
  DASHLINE-1
  EZMAP-3
  ISOSRF-1
  ISOSRFHR-2
  PWRITY-1
  PWRZI-1
  PWRZS-1
  PWRZT-1
  STRMLN-1
  THREED-1
  VELVCT-1
  WINDOW-1
On Input for CONREC, CONREC-2
On Input for DASHD, DASHCHAR-2
On Input for DASHD, DASHSMTH-2
On Input for DASHD, DASHSUPR-2
On Input for EZCNTR, CONREC-1
On Input for EHZFTN, HAFTON-1
On Input for EZSRFC, SUPMAP-2
On Input for HAFTON, HAFTON-2
On Input For MTITLE, SCROLL-1
On Input For SCROLL, SCROLL-2
On Input for SRFACE, SRFACE-2
On Input for SUPMAP, SUPMAP-2
On Output, CONRAN-2
On Output for CONREC, CONREC-3
On Output for EZCNTR, CONREC-1
On Output for EHZFTN, HAFTON-2
On Output for EZSRFC, SRFACE-2
On Output for HAFTON, HAFTON-4
On Output For MTITLE, SCROLL-2
On Output For SCROLL, SCROLL-4
On Output for SRFACE, SRFACE-3
On Output for SUPMAP, SUPMAP-4
Operation, CONRAN-3
  CONRAQ-3
  CONRAS-3
Option,
  scaling, CONRAQ-11
  CONRAS-14
Option Default Values, CONRAN-13
  CONRAQ-10
  CONRAS-12
Option,
  centering, SCROLL-3
Option setting routine, CONRAN-1
Option setting routine, CONRAQ-1
Option setting routine, CONRAS-1
Option values, CONRAN-2
Options,
  CONRAN-3
  CONRAQ-3
  CONRAS-3
Options Which Effect the Contours,
  CONRAN-14
  CONRAQ-11
  CONRAS-14
OPVAL, EZMAP-7
Origins, VELVCT-1
Orthographic, SUPMAP-2
Other Graphics Routines,
  Interfacing Into, CONRAN-14
  CONRAQ-11
  CONRAS-14
Other Parameters, EZMAP-5
Outlines, EZMAP-5
  Continental, SUPMAP-1
  U.S. state, SUPMAP-1
Output,
  On, CONRAN-2
  On, CONRAQ-2
  On, CONRAS-2
Overview of the SCD Graphics Utilities, 1-1

Overwriting, WINDOW-1

Package,
dashed line, DASHCHAR-1

Package Routines Used,
System Plot, SCROLL-5

Package,
System PLOT, THREED-1

Package,
Windowing, WINDOW-1

Pairs,
STEREO, SRFACE-2

Parameter list,
internal, SCROLL-4

Parameter Routines,
AUTOGRAPH Control, AUTOGRAPH-2

Parameters,

Internal, 2-2

Parameters,

Default Values for User Specified,
CONRAN-13
CONRAQ-10
CONRAS-13

Parameters,

number of, CONRAQ-3

Internal,

CONREC-3
CONRECK-2
CONRECUPR-2
DASHLINE-3
HAFTON-5
PWRITX-7
SCROLL-6
SRFACE-4
STRMLN-1

Other, EZMAP-5

Partial derivatives,
estimated, CONRAN-3

Pattern,
dashed line,
CONREC-1
CONREC-3
CONRECK-1

PDV, CONRAN-8
CONRAQ-7
CONRAS-8

PER, CONRAN-8
CONRAQ-7
CONRAS-8

Perimeter,
plot a, CONRAN-1

CONRAN-3
CONRAS-3
drawn, HAFTON-1

Perimeter, EZMAP-6
VELVCT-1

Perimeter or grid,
plot a, CONRAS-1

Perspective character labeling, SRFACE-3

Perspective picture, ISOSRFHR-1

Philosophy, 1-1

Picture array,
model, DASHSUPR-1
DASHSUPR-4
half-tone, HAFTON-1
perspective, ISOSRFHR-1
Picture Quality vs. Cost, 1-1
Picture size, SRFACE-3
Picture,
perspective, SRFACE-1
shape of the, SRFACE-3
Picture,
Sample, THREED-3
Plane,
plotting, DASHSUPR-4
Plane model,
image, ISOSRFHR-2
Plot a grid, CONRAN-1
Plot a perimeter,
CONRAN-1
CONRAN-3
CONRAS-3
or grid, CONRAS-1
Plot,
fine-tune, CONRAN-14
CONRAQ-13
CONRAQ-14
or grid, CONRAS-1
Plot,
Sample, 2-2
AUTOGRAPH-13, 14
CONRAN-16
CONRAQ-13
CONRAS-16
CONREC-7, 8
CONRECQCK-3, 4
CONRECSTH-3, 4
CONRECUPR-3, 4
DASHCHAR-4
DASHLINE-4
DASHSMTH-6, 7
DASHSUPR-8
HAFTON-7, 8
ISOSRF-6, 7
ISOSRFHR-8
PWRITX-8, 9, 10, 11
PWRITY-3
SRFACE-6
SUPMAP-15, 16, 17, 18, 19, 20
THREED-4
VELVCT-6
WINDOW-4
Plots contour lines, CONRAQ-1
Plots produced, CONRAS-14
Plots the contour map, CONRAS-2
Plotter
address space, DASHCHAR-3
coordinates, ISOSRFHR-3
instructions, HAFTON-4
space, DASHLINE-2
units, DASHCHAR-2
DASHSUPR-3
Plotting lines,
skips, SCROLL-5
Plotting plane, DASHSUPR-4
Plotting routine,
character, PWRZI-1
character, PWRZS-1
character, PWRZT-1
Plotting Routines,
Required,
CONRAN-3
CONRAQ-2
CONRAS-3
Plotting routines,
regular, THREED-2
Plotting Routines Used, 2-2
CONREC-4
CONRECQCK-2
DASHCHAR-4
DASHLINE-2
DASHSMTH-4
DASHSUPR-4
HAFTON-4
ISOSRFHR-4
PWRITX-6
PWRITY-2
SRFACE-4
SUPMAP-7
THREED-3
VELVCT-3
WINDOW-3
PMM, CONRAN-9
CONRAS-9
Point arguments,
fixed or floating,
DASHCHAR-1
DASHLINE-1
WINDOW-2
Point locked at,
the, SRFACE-2
Point numbers,
floating, PWRZI-1
Point of a line,
first, DASHSMTH-1
last, DASHSMTH-1
first, DASHSUPR-1
Point of interpolation, CONRAN-14
Point,
floating, THREED-1
Point values, data, CONRECQCK-2
Points, Entry, 2-1
Points, Entry, AUTOGRAPH-9
CONRAN-2
CONRAQ-2
CONRAS-2
CONREC-3
CONRECQCK-1
CONRECSMTH-2
CONRECSUPR-1
DASHCHAR-3
DASHLINE-2
DASHSMTH-3
PWRITX-6
PWRITY-2
SCROLL-4
SRFACE-3
STRMLN-2
SUPMAP-5
THEEED-2
VELVCT-3
WINDOW-2
Points, data values at the, CONRAN-1
X-coordinates of the data, CONRAN-1
Y-coordinates of the data, CONRAN-1
number of input, CONRAN-3
data, CONRAQ-1
data, CONRAS-1
number of input, CONRAS-3
Points of a line, remaining, DASHSMTH-1
remaining, DASHSUPR-1
POLAT, POLONG, ROT, EZMAP-3
SUPMAP-2
Polar projections, SUPMAP-3
Port library error routines, CONRAN-2
CONRAS-2
Portability, 1-3, 2-2
AUTOGRAPH-10
CONRAN-3
CONRAQ-2
CONRAS-3
CONREC-4
CONRECQCK-2
CONRECSUPR-2
DASHCHAR-4
DASHLINE-2
DASHSMTH-4
DASHSUPR-4
HAFTON-4
ISOSRFHR-4
PWRITX-6
PWRITY-2
SCROLL-5
SRFACE-4
SUPMAP-6
THEEED-2
VELVCT-3
Position, DASHLINE-2
eye, ISOSRFHR-2
Positioning coordinates, PWRZS-1
PWRZT-1
Positive angles, SRFACE-1
Practical limit, DASHCHAR-2
Practice run, SCROLL-1
SCROLL-4
Precision, 2-1
CONRAN-2
CONRAQ-2
CONRAS-2
CONREC-3
CONRECQCK-1
DASHCHAR-3
DASHLINE-2
DASHSMTH-3
DASHSUPR-4
EZMAP-11
HAFTON-4
ISOSRFHR-3
PWRITX-6
PWRITY-2
SCROLL-5
SRFACE-3
SUPMAP-6
THEEED-2
VELVCT-3
WINDOW-2
Problem, hidden character, PWRZT-1
Production mode, SCROLL-1
Program, calling, HAFTON-2
Projection, conical, SUPMAP-13
cylindrical, SUPMAP-3
polar, SUPMAP-3
nine, SUPMAP-7
Projections Used, Description of, SUPMAP-8

xvi
Purpose, 2-1

AUTOGRAPH-1
CONRAN-1
CONRAQ-1
CONRAS-1
CONREC-1
CONRECQCK-1
CONRECSTMH-1
CONRECSUPR-1
DASHCHAR-1
DASHLINE-1
DASHSMTH-1
DASHSUPR-1
EZMAP-1
HAFTON-1
ISOSRF-1
ISOSRFHR-1
PWRITX-1
PWRITY-1
PWRZI-1
PWRZS-1
PWRZT-1
SCROLL-1
SFRFACE-1
STRMLN-1
SUPMAP-1
THREED-1
VELVCT-1
WINDOW-1

PUSER,

Common Block, PWRITX-1
PWRITX modifier, SCROLL-3

Random data sets,
CONRAN-1
CONRAQ-1
CONRAS-1

Reader/printer output, HAFTON-1

Real work array,
CONRAN-1
CONRAS-1

Rectangular array,
CONRECQCK-1
CONRECSUPR-1
HAFTON-1

Reference,
ISOSRFHR-3
SFRFACE-3

References,
CONRAN-15
CONRAQ-12
CONRAS-14
EZMAP-11
SUPMAP-6

Regions,
mark the, DASHSUPR-1
mid-latitude, SUPMAP-14

Regular plotting routines, THREED-2
Relative size of characters, SCROLL-2

Remaining points of a line,
DASHSMTH-1
DASHSUPR-1

Removed,
hidden lines, ISOSRF-1
hidden lines, SRFACE-1

Removing crowded lines, DASHSUPR-1

REP, CONRAN-9
CONRAQ-7
CONRAS-9

Representation,
streamline, STRMLN-1

Required,
Space, 2-2
AUTOGRAPH-10
CONRAN-2
CONRAQ-2
CONRAS-2
CONREC-4
CONRECQCK-1
CONRECSMTH-2
CONRECSUPR-2
DASHCHAR-3
DASHLINE-2
DASHSMTH-3
DASHSUPR-4
HAFTON-4
ISOSRFHR-4
PWRITY-2
SCROLL-5
SFRFACE-4
STRMLN-2
SUPMAP-6
THREED-2
VELVCT-3

Required Library Routines, 2-1

CONRAN-2
CONRAQ-2

Required Plotting Routines,
CONRAN-3
CONRAQ-2
CONRAS-3

Required Resident Routines, 2-2
AUTOGRAPH-11
CONRAN-3
CONRAN-3
Required ULIB Routines,
AUTOGRAPH-10
CONREC-4
CONRECQCK-1
CONRECSUPR-1
DASHCHAR-3
DASHLINE-2
DASHSMTH-3
DASHSUPR-4
EZMAP-11
HAFTON-4
ISOSRFHR-3
PWRITX-6
PWRITY-2
SCROLL-5
SUPMAP-6
THREED-2
VELVCT-3
WINDOW-2

Resident Routines,
Required, 2-2
AUTOGRAPH-11
CONRAN-3
CONRAQ-3
CONRAS-3
CONREC-4
CONRECQCK-2
DASHCHAR-4
DASHLINE-2
DASHSMTH-4
DASHSUPR-4
EZMAP-12
HAFTON-4
ISOSRFHR-4
PWRITX-6
PWRITY-2
SCROLL-5
SUPMAP-7
THREED-3
VELVCT-3
WINDOW-3

Resolution,
grid, PWRZI-1
current grid, PWRZS-1
PWRZT-1

Revision,
Latest, 2-1
AUTOGRAPH-1
CONRAN-1
CONRAQ-1
CONRAS-1
CONREC-1
CONRECQCK-1
CONRECSMTH-1
CONRECSUPR-1
DASHCHAR-1
DASHLINE-1
DASHSMTH-1
DASHSUPR-1
EZMAP-1
HAFTON-1
ISOSRF-1
ISOSRFHR-1
PWRITX-1
PWRITY-1
PWRZI-1
PWRZS-1
PWRZT-1
SCROLL-1
SRFACE-1
SUPMAP-1
THREED-1
VELVCT-1
WINDOW-1

Routine AGCURV, AUTOGRAPH-8
Routine AGSTUP, AUTOGRAPH-8
Routine ANOTAT, AUTOGRAPH-4

Routine,
option setting, CONRAN-1
CONRAQ-1
CONRAS-1
DISPLA, AUTOGRAPH-6
character plotting, PWRZI-1
PWRZS-1
PWRZT-1

Routines,
Required Library, 2-1

xviii
Routines,
Required Resident, 2-2
Routines AG., AUTOGRAPH-7
Plotting, 2-2
CONREC-4
CONRECQCK-2
DASHCHAR-4
DASHLINE-2
DASHSMTH-4
DASHSUPR-4
HAFTON-4
ISOSRFHR-4
PWRITX-6
PWRITY-2
SRFACE-4
SUPMAP-7
THREED-3
VELVCT-3
WINDOW-3
Run,
practice, SCROLL-1
SCROLL-4
Runs,
making multiple, CONRAN-2
S, ISOSRFHR-3
Sample Picture, THREED-3
Sample Plot, 2-2
AUTOGRAPH-13,14
CONRAN-16
CONRAQ-13
CONRAS-16
CONREC-7,8
CONRECQCK-3,4
CONRECQSMTH-3,4
CONRECQSUPR-3,4
DASHCHAR-4
DASHLINE-4
DASHSMTH-6,7
DASHSUPR-8
HAFTON-7,8
ISOSRF-6,7
ISOSRFHR-8
PWRITX-8,9,10,11
PWRITY-3
SRFACE-6
SUPMAP-15,16,17,18,19,20
THREED-4
VELVCT-6
WINDOW-4
SCA,
CONRAN-9
CONRAQ-8
CONRAS-9
Scale factor, VELVCT-1
Computes a, VELVCT-5
Scale factors, CONREC-1
VELVCT-3
Scale,
grey, HAFTON-7
Scaling, CONREC-2
SRFACE-1
Scaling option, CONRAQ-11
CONRAS-14
Scaling,
control, STRMLN-2
Flag to control, VELVCT-2
Scanned,
grid is, CONRAS-2
SCD Graphics Utilities,
Overview of the, 1-1
Scratch array,
ISOSRFHR-2
SRFACE-2
CONRAN-2
CONRAQ-2
CONRAS-2
file, ISOSRFHR-2
Screen width, PWRZI-1
PWRZS-1
PWRZT-1
window, SCROLL-5
SCROLL, -1
Scroll coordinate,
The, SCROLL-3
Scroll of text,
The, SCROLL-3
SCROLL,
On Input For, SCROLL-2
On Output For, SCROLL-4
Time to move the, SCROLL-4
In, SCROLL-7
Scrolled movie titles, SCROLL-1
SDC, CONRAN-9
SDC, CONRAS-9
Second dimension, ISOSRF-1
Second dimension of CARDS, SCROLL-3
Second subscript direction,
the, CONREC-2
Second subscript direction,
HAFTON-2
ISOSRF-1
STRMLN-2
Seconds,
Time in, SCROLL-4
Segment,
line, WINDOW-2
vector, DASHCHAR-1
Sequence,
title, SCROLL-2
movie, SCROLL-5
Sets,
irregular data, CONRAN-1
CONRAQ-1
CONRAS-1
random data, CONRAN-1
CONRAQ-1
CONRAS-1
sparse data, CONRAN-1
CONRAQ-1
CONRAS-1
Setting routine,
option, CONRAN-1
CONRAQ-1
CONRAS-1
Shape of the picture, SRFACE-3
Silhouette, ISOSRFHR-2
Sinusoidal mapping, HAFTON-3
Size and smoothness of Z, CONRECQCK-2
Size of characters,
relative, SCROLL-2
Size,
picture, SRFACE-3
Skips plotting lines, SCROLL-5
SML, CONRAN-10
CONRAS-10
Smooth lines, DASHSUPR-4
Smoothed,
labeled and,
CONRECSMTH-1
CONRECSUPR-1
Smoothing capabilities,
DASHSMTH-1
DASHSUPR-1
Smoothing features, DASHSMTH-1
Smoothness of Z,
size and, CONRECQCK-2
Solid lines, CONREC-3
CONRECQCK-1
Space,
The grid, CONRECQCK-1
plotter address, DASHCHAR-3
plotter, DASHLINE-2
Space Required, 2-2
AUTOGRAPH-10
CONRAN-2
CONRAQ-2
CONRAS-2
CONREC-4
CONRECQCK-1
CONRECSMTH-2
CONRECSUPR-2
DASHCHAR-3
DASHLINE-2
DASHSMTH-3
DASHSUPR-4
HAFTON-4
ISOSRFHR-4
PWRITY-2
SCROLL-5
SRFACE-4
STRMLN-2
SUPMAP-6
THREED-2
VELVCT-3
Spaced,
linearly, HAFTON-1
equally, SRFACE-1
Spacing,
Vertical, SCROLL-1
Sparse data, CONRAQ-2
Sparse data sets,
CONRAN-1
CONRAQ-1
CONRAS-1
SPD, CONRAN-10
CONRAQ-8
CONRAS-10
Special Conditions,
AUTOGRAPH-9
SRFACE-3
Special value areas,
Contiguous, HAFTON-3
Special value feature,
HAFTON-3
VELVCT-2
Specification,
Transformation, EZMAP-3
Map Area, EZMAP-4
Specified Parameters,
Default Values for User,
CONRAN-13
CONRAQ-10
CONRAS-13
Speed,
flow, STRMLN-1
Splines under tension,
CONRAN-1
CONRAS-1
SRFACE,
On Input for, SRFACE-2
On Output for, SRFACE-3
SSZ, CONRAN-10
Surface,
  iso-valued, ISOSRF-1
Surfaces,
  visible, ISOSRFHR-3
Switch, SCROLL-1
System Plot Package
  Routines Used, SCROLL-5
System Plot Package, THREED-1
Tangential, SUPMAP-8
TEN, CONRAN-11
Tension,
  splines under, CONRAN-1
  CONRAS-1
    text cards, SCROLL-2
TFR, CONRAN-11
    CONRAQ-9
    CONRAS-11
Three words long,
  EYE is an array, THREED-1
Three-dimensional array, ISOSRF-1
Three-space,
  ISOSRFHR-1
  PWRZI-1
  PWRZS-1
  PWRZT-1
  THREED-1
Tick marks, CONREC-2
Time consuming, ISOSRFHR-4
Time in seconds, SCROLL-4
Time, SCROLL-2
Time to move the scroll, SCROLL-4
Timing, 2-2
  CONREC-4
  CONRECQCK-2
  CONRECSMTH-2
  CONRECSUPR-1
  DASHCHAR-4
  DASHLINE-2
  DASHSMTH-3
  DASHSUPR-4
  EZMAP-12
  HAFTON-4
  ISOSRFHR-4
  PWRITY-2
  SCROLL-5
  SRFACE-4
  SUPMAP-6
  THREED-2
  WINDOW-3
Title frames, SCROLL-1
Title sequence, SCROLL-2
Titles,
  scrolled movie, SCROLL-1
movie, SCROLL-2
TLE, CONRAN-12
  CONRAQ-9
  CONRAS-12
TOP, CONRAN-12
  CONRAQ-9
  CONRAS-12
TPGN, AUTOGRAPH-7
Transformation Specification, EZMAP-3
Transformation,
  SUPMAP-5
  CONREC-3
Translate, SRFACE-3
Traversed,
  distance, DASHLINE-2
TRI, CONRAN-12
  CONRAQ-10
  CONRAS-12
Triangulated area, CONRAS-2
Triangulated data,
  interpolation of the, CONRAN-1
  CONRAQ-1
Two-dimensional
  velocity field, VELVCT-1
Two-dimensional, SRFACE-1
Type,
  Mollweide, SUPMAP-2
ULIB Routines,
  Required,
    AUTOGRAPH-10
    CONREC-4
    CONRECQCK-1
    CONRECSUPR-1
    DASHCHAR-3
    DASHLINE-2
    DASHSMTH-3
    DASHSUPR-4
    EZMAP-11
    HAFTON-4
    ISOSRFHR-3
    PWRITX-6
    PWRITY-2
    SCROLL-5
    SRFACE-3
    SUPMAP-6
    THREED-2
    VELVCT-3
    WINDOW-2
Under tension,
  splines, CONRAN-1
  CONRAS-1
Union,
  disjoint, SUPMAP-7

Units,
  plotter, DASHCHAR-2
  plotter, DASHSUPR-3
  CRT, SCROLL-3
  CRT, VELVCT-2
U.S. state outlines, SUPMAP-1
Usage, 2-1
  AUTOGRAPH-1
  CONRAN-1
  CONRAQ-1
  CONRAS-1
  CONREC-1
  CONRECQCK-1
  CONRECSMTH-1
  CONRECSUPR-1
  DASHCHAR-1
  DASHLINE-1
  DASHSMTH-1
  DASHSUPR-1
  EZMAP-1
  HAFTON-1
  ISOSRF-1
  ISOSRFHR-1
  PWRITX-1
  PWRITY-1
  PWRZI-1
  PWRZS-1
  PWRZT-1
  SCROLL-1
  SRFACE-1
  STRMLN-1
  SUPMAP-1
  THREED-1
  VELVCT-1
  WINDOW-1
Used,
  Plotting Routines, 2-2
    CONREC-4
    CONRECQCK-2
    DASHCHAR-4
    DASHSMTH-4
    DASHSUPR-4
    HAFTON-4
    ISOSRFHR-4
    PWRITX-6
    PWRITY-2
    SRFACE-4
    SUPMAP-7
    THREED-3
    VELVCT-3
    WINDOW-3

xxii
Description of Projections,
SUPMAP-8
SUPMAP-9
System Plot Package Routines,
SCROLL-5
User Specified Parameters,
Default Values For, CONRAN-13
CONRAQ-10
CONRAS-13
Utilities,
Overview of the SCD Graphics, 1-1
The Graphics, 2-2
Utility Name, 2-1
Value areas,
contiguous special, HAFTON-3
Value,
absolute, HAFTON-2
PWRZI-2
PWRZT-2
data, HAFTON-1
interpolated, CONRAS-2
numerical, CONRAN-3
CONRAS-3
Value feature,
special, HAFTON-3
VELVCT-2
Values at the points,
data, CONRAN-1
Values,
data, CONREC-2
CONRAQ-1
data point, CONRECQCK-2
nice, CONREC-2
Option Default, CONRAN-13
option, CONRAN-2
CONRAS-2
Optional Default, CONRAQ-10
Optional Default, CONRAS-12
Values for User Specified Parameters,
Default, CONRAN-13
CONRAQ-10
CONRAS-13
Values,
data, HAFTON-2
the minimum and maximum, SUPMAP-3
VECPPRT, SUPMAP-5
Vector base, VELVCT-5
Vector field, VELVCT-3
vector head, VELVCT-5
Vector length,
minimum, VELVCT-2
Vector segments, DASHCHAR-1
Vector,
longest, VELVCT-2
endpoints of the, VELVCT-3
computes the length of a, VELVCT-5
Velocity field,
two-dimensional, VELVCT-1
Velocity fields, STRMLN-1
VELVCT, -1
Vertical spacing, SCROLL-1
Viewer's eye,
the, SRFACE-2
Virtual grid, CONRAN-2
CONRAQ-2
CONRAS-2
Visibility errors, SRFACE-3
Visibility information, SRFACE-3
Visible surfaces, ISOSRFHR-3
Width,
character, PWRZI-1
screen, PWRZI-1
PWRZS-1
PWRZT-1
Window,
screen, SCROLL-5
WINDOW, SUPMAP-1
Windowing Package, WINDOW-1
Word-length dependent,
machine, CONRECSUPR-2
Work array,
integer, CONRAN-1
CONRAS-2
Real, CONRAN-1
Workspace buffer (IBUF), SCROLL-4
Workspace buffer, SCROLL-2
Written, STRMLN-1
X coordinates, SRFACE-2
X-coordinates, CONRAS-1
of the data points, CONRAN-1
x-direction,
cells in the, ISOSRFHR-2
XDRA, AUTOPRAG-3
AUTOPRAG-8
XLAB and YLAB, AUTOPRAG-4
XVEC, AUTOPRAG-8
X,Y, PWRITX-1
PWRITY-1
Y coordinates, SRFACE-2
Y-coordinates, CONRAQ-1
Y-coordinates, CONRAS-1
of the data points, CONRAN-1
y-direction,
cells in the, ISOSRFHR-2
YDRA, AUTOPRAG-3
YLAB,
XLAB and, AUTOGRAPH-4
YVEC, AUTOGRAPH-8
Z,
size and smoothness of, CONRECQCK-2
Z, HAFTON-1
Z, HAFTON-2