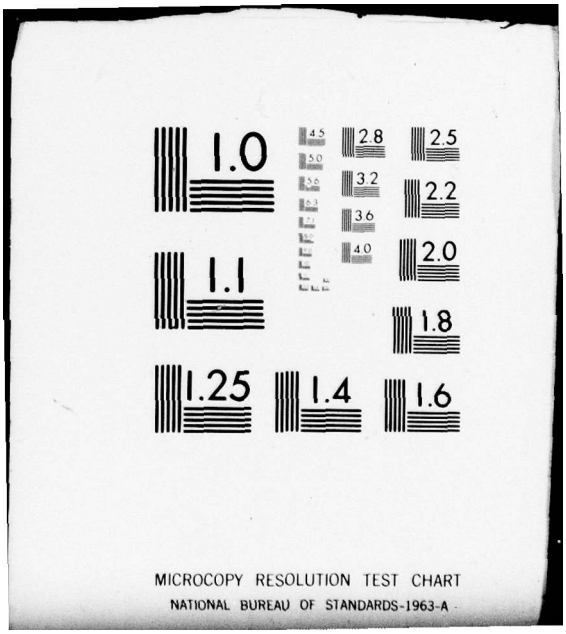
AD-A083 424 WESTINGHOUSE WORD PROCESSING CENTER PITTSBURGH PA F/G 9/2
GRAPHICS COMPATIBILITY SYSTEM (GCS) PROGRAMMER'S REFERENCE MANU--ETC(U)
OCT 79 DACW39-78-M-2676
UNCLASSIFIED WES-MP-0-79-5 NL

1 OF 3
AD A083424


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MISCELLANEOUS PAPER O-79-5

GRAPHICS COMPATIBILITY SYSTEM (GCS) PROGRAMMER'S REFERENCE MANUAL

by

Westinghouse Word Processing Center
Pittsburgh, Pa. 15230

October 1979

Final Report

Prepared for Office, Chief of Engineers, U. S. Army
Washington, D. C. 20314

Under Contract No. DACW39-78-M-2676

Monitored by Automatic Data Processing Center
U. S. Army Engineer Waterways Experiment Station
P. O. Box 631, Vicksburg, Miss. 39180

Destroy this report when no longer needed. Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Miscellaneous Paper 0-79-5	2. GOVT ACCESSION NO.: AD-A083 424
3. RECIPIENT'S CATALOG NUMBER:	
4. TITLE (and Subtitle): GRAPHICS COMPATIBILITY SYSTEM (GCS) PROGRAMMER's REFERENCE MANUAL,	5. TYPE OF REPORT & PERIOD COVERED: Final report.
	PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s):	8. CONTRACT OR GRANT NUMBER(s): DACW39-78-M-2676
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Westinghouse Word Processing Center, Pittsburgh, Pa. 411688	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 4A762725AT42
11. CONTROLLING OFFICE NAME AND ADDRESS: Office, Chief of Engineers, U. S. Army, Washington, D. C. 20314	12. REPORT DATE: October 1979 13. NUMBER OF PAGES: 282
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): U. S. Army Engineer Waterways Experiment Station, Automatic Data Processing Center, P. O. Box 631, Vicksburg, Miss. 39180	15. SECURITY CLASS. (of this report): Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18 WES 19 MP-0-79-5

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer graphics
Graphics Compatibility System
Manuals
Programmer's reference manual

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This reference manual explains the function of each subroutine in the Graphics Compatibility System (GCS). GCS is a Fortran-based computer graphics system designed for interactive/passive use on a wide variety of graphics hardware. GCS was originally developed by the U. S. Military Academy at West Point. In January 1975, the responsibility for maintenance, distribution, and future development was transferred to the U. S. Army Engineer Waterways Experiment Station.

→ to page 1

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

411688

PREFACE

This report is the result of work performed under Contract No. DACW39-78-M-2676, dated 3 April 1978, between the U. S. Army Engineer Waterways Experiment Station (WES), Vicksburg, Miss., and the Westinghouse Word Processing Center, Pittsburgh, Pa. The work concerned publishing a word processing version of the Programmer's Reference Manual for the Graphics Compatibility System (GCS). The original version of this manual was developed at the U. S. Military Academy.

Contributors to the document were Dr. Richard Puk, Sandia Laboratories; Dr. Steve Orbon and Mr. Robert Bruns, Westinghouse; and Mr. James M. Jones II, R&D Software Group, Automatic Data Processing (ADP) Center, WES. The work was administered by the ADP Center, WES, as part of Computer Technology-Engineering Software, Project No. 4A762725AT11, and the Civil Works Computation and Analysis Project sponsored by the Office, Chief of Engineers, U. S. Army (OCE). OCE points of contact were Mr. Robert McMurrer (DAEN-DSE) and Messrs. Richard Malm and Harry Hardin (DAEN-CWE-BA).

Mr. Jones monitored the contract under the general supervision of Dr. N. Radhakrishnan, Special Technical Assistant to the Chief of the ADP Center, WES, and Mr. D. L. Neumann, Chief of the ADP Center.

Directors of WES during the period of the contract and the preparation of this report were COL J. L. Cannon, CE, and COL N. P. Conover, CE. Technical Director was Mr. F. R. Brown.

Page 9 left blank intentionally

CONTENTS

CHAPTER I

INTRODUCTION

Contained within this section are the detailed descriptions and usage instructions of all of the user callable routines which make up the GCS graphics system, and a discussion of the various options available to the user.

GCS is available in two versions, a two-dimensional version, and a three-dimensional version.

The three dimensional version is a sophisticated, highly capable package having such features as graphical data structures, text processing, and of course, three-dimension graphics. The two-dimensional version is the older, less sophisticated version. While possessing fewer capabilities, it is still supported, since it offers the benefits of a smaller central memory requirement, and a higher execution speed, while still fulfilling most of the normal graphics requirements.

In general, the two-dimension version is a proper subset of the three dimension version meaning that all elements of two-dimension GCS are contained within the three dimensional version.

Those routines or options which are a part of the three-dimension version, and not a part of the two-dimension version are indicated by the notation '3D only.'

CHAPTER II

THE GRAPHICS STATUS AREA (GSA) AND USET/UPSET/UQUERY OPTIONS

The Graphics Status Area

Central to the structure of GCS is the concept of a distributed executive capability provided by the Graphics Status Area (GSA). This is a labeled COMMON area in which is stored information concerning the state of the graphics system, information about the characteristics of both the terminal and the computer system, and information about the current settings of GCS user-controlled variables. The GSA serves as a means of communication between the various subroutines which comprise the Graphics Compatibility System. Much of the power of GCS arises from the ability of every GCS routine to access any of the information about the status of the system conveniently and efficiently. One of the immediate results of having a status area is the great reduction in the number of arguments in the calling sequences of the routines, thus improving the efficiency of the system. It also makes it much easier for the user to remember the calling sequence of the different routines when writing application software.

The variables of the GSA can be set in any of three ways: they may be established during the execution of USTART with values taken from a predefined table; they may be set by the user through USET, UPSET, or other setting routines; or they may be set by GCS routines during execution. USTART will load the variables with their initial default setting. The user sets these variables under his control in USET and UPSET and through subroutine arguments for such routines as UWINDO, UMARGN, UDAREA, UCOSYS, and UROTAT. The user may also restore the GSA variables to the previous state through use of any of the 'UNSAVE' routines. GCS itself maintains many variables which keep track of such items as beam/pen position, and various status change flags. The parameters contained within the GSA can be obtained by the user by a call to UQUERY.

The ability to change GSA variables selectively in accordance with related subroutine options is one of the many attractive and powerful features of GCS. This chapter contains two tables which present descriptive grouping of USET and UPSET options. Table 1 gives a list of those USET options which are most frequently used. This list may be used as a quick reference guide to the most popular USET options. Table 2 gives the default conditions that are automatically preset as each call to GCS is initiated. This concept of default values is immensely valuable and convenient to both the beginning programmer and the experienced programmer who doesn't want to worry about the setting of the GSA variables. The complete list of options can be formed in the writeups associated with the USET/UPSET/UQUERY subroutines.

USET Options

TABLE 1

USET OPTIONS MOST FREQUENTLY USED:

TO REQUEST ABSOLUTE COORDINATE PLOTTING (DEFAULT):

USET ('ABSOLUTE')	USET ('ABSO')
USET ('ABSb')	

TO REQUEST RELATIVE OR INCREMENTAL COORDINATE PLOTTING:

USET ('RELATIVE')	USET ('RELA')

NOTE: b - is a blank or space

TO REQUEST RECTANGULAR COORDINATES (DEFAULT):

USET ('RECTANGULAR') USET ('RECT')

TO REQUEST POLAR COORDINATES:

USET ('POLAR') USET ('POLA')

TO REQUEST ANGULAR VALUES IN DEGREES (DEFAULT):

USET ('DEGREES') USET ('DEGR')

TO REQUEST ANGULAR VALUES IN RADIANS:

USET ('RADIANS') USET ('RADI')

TO REQUEST PLOTTING IN VIRTUAL SPACE (DEFAULT):

USET ('VIRTUAL') USET ('VIRT')

TO REQUEST PLOTTING IN DEVICE SPACE:

USET ('DEVICE') USET ('DEVI')

TO REQUEST DEVICE SPACE PLOTTING UNITS IN INCHES (DEFAULT):

USET ('INCHES') USET ('INCH')

TO REQUEST DEVICE SPACE PLOTTING IN RASTER UNITS:

USET ('RASTERUNITS') USET ('RAST')

TO REQUEST DEVICE SPACE PLOTTING IN FONT (CHARACTER SPACE) UNITS:

USET ('FONTUNITS') USET ('FONT')

TO REQUEST DEVICE SPACE PLOTTING IN PERCENT UNITS:

USET ('PERCENTUNITS') USET ('PERC')

TO REQUEST PLOTTING OF A VISIBLE LINE (DEFAULT):

USET ('LINE')

TO REQUEST PLOTTING OF INVISIBLE LINES:

USET ('MOVE')
USET ('NOLINE') USET ('NOLI')
USET ('NObLINE') USET ('NObL')

NOTE: b - is a blank or space

TO REQUEST PLOTTING OF LINES WITH ARROW TERMINATORS:

USET ('ARROW') USET ('ARRO')

TO REQUEST PLOTTING OF DASHED LINES:

USET ('DASH')

TO REQUEST PLOTTING OF TIC LINES:

USET ('TICLINE') USET ('TICL')

TO REQUEST PLOTTING OF LINES WITH ONLY ENDPOINTS VISIBLE:

USET ('POINT') USET ('POIN')

TO REQUEST PLOTTING OF LINES COMPOSED OF CHARACTERS:

USET ('ALPHANUMERICLINES') USET ('ALPH')

TO REQUEST PLOTTING OF INVISIBLE LINES WITH CHARACTERS AT THEIR ENDPOINTS:

USET ('CHARACTER') USET ('CHAR')

TO REQUEST CHARACTER OUTPUT IN THE FORM OF HARDWARE CHARACTERS (DEFAULT):

USET ('HARDWARE') USET ('HARD')

TO REQUEST CHARACTER OUTPUT IN THE FORM OF SOFTWARE CHARACTERS:

USET ('SOFTWARE') USET ('SOFT')

TO REQUEST UPRINT/UWRITE OUTPUT IN TEXT FORMAT (DEFAULT):

USET ('TEXT')

TO REQUEST UPRINT/UWRITE OUTPUT IN REAL FORMAT:

USET ('REALNUMBER') USET ('REAL')

TO REQUEST UPRINT/UWRITE OUTPUT IN INTEGER FORMAT:

USET ('INTEGER') USET ('INTE')

TO REQUEST PLOTTING WITH RESPECT TO THE SYSTEM AXIS (DEFAULT):

USET ('SYSTEM') USET ('SYST')

NOTE: b - is a blank or space

TO REQUEST PLOTTING WITH RESPECT TO THE USER AXIS:

USET ('USERAXIS') USET ('USER')

TO REQUEST A LINE TYPE/CHARACTER TERMINATOR COMBINATION:

USET ('X$bb') USET ('X$')

Where

$ IS THE DESIRED CHARACTER TERMINATOR

AND

X, THE LINE TYPE SPECIFICATION, IS AS FOLLOWS:

A	ALPHANUMERIC LINE
D	DASHED LINE
L	SOLID LINE
N	NULL OR INVISIBLE LINE
T	TIC LINE
b	denotes blank

TABLE 2

GCS DEFAULT CONDITIONS

This table discusses those options which are present in the Graphics Status Area as default options. After a call to Subroutine USTART, the Graphics Compatibility System is set to the default conditions as indicated below. The default options can be divided into two groups: Basic Plotting Options and High Level Plotting Options.

I. Default Basic Plotting Options

Plotting is done in 'RECTANGULAR' and 'ABSOLUTE' coordinates on the 'SYSTEM' coordinate axis. The 'USER' coordinate axes are identical to the system axis. Plotting is done in 'VIRTUAL' space with a virtual window whose limits are from 0.0 to 100.0 in the X direction, and from 0.0 to 100.0 in the Y direction. The virtual window is mapped into a display area which is the largest square area on the device display surface. The right hand edge of the square corresponds to the top edge of the display surface.

Character output in GCS will be, by default, in 'HARDWARE' character format, of type 9K and of 'MEDIUM' size. If 'SOFTWARE' characters are requested via USET then the default horizontal size is five (5.0) virtual units, and the default vertical size is seven (7.0) virtual units, with the font 'SGCS being the default font.

By default, angular units for angular specifications are in 'DEGREES'. If the user switches to 'DEVICE' space, then all length or distance units (i.e. from UPRINT or UREAD), the data will be assumed to be in 'TEXT' mode. The alphanumeric output defaults to the 'PLOT' device if possible, and any graphic output will go to all devices in a cluster. Graphic input is from the primary input device; the number of digits of precision for numeric output is four (4); and GCS detected errors are signalled as they occur.

A line which is drawn by a subroutine in GCS is considered to consist of two parts; a line type and a line terminator. The line type may be solid (visible), ticked, null (invisible), dashed, or alphanumeric. The line may be terminated by an arrow, a back arrow, double arrows, a character, a point, a symbol, a set of coordinate values, or nothing at all. The default line type in GCS is 'SOLID' with 'NULL' terminators ('LNULL'). If 'TICLINES' are requested via USET option, then the default tic interval is ten (10.0) virtual units. It is the user's responsibility to insure that the tic interval is appropriate if he switches to

NOTE: b - is a blank or space

'DEVICE' space or alters the default virtual window setting or the default display area setting. If 'DASHLINES' are requested, then the default dash specification (56.) will result in a dashed line which is alternately light and dark, in increments of approximately 0.075 inches. If a 'CHARACTER' line terminator is requested but no character is specified by way of Subroutine UPSET, then the character asterisk (*) is used. Similarly, the asterisk is used to compose the line type if 'ALPHANUMERIC' lines are requested.

For 3D operations, the viewing environment is set up to simulate a 2D environment. The view point is located at (0., 0., 150.), the view site is at (0., 0., 0.) and the view plane is located at the view site. The system coordinate system is considered to be 'RIGHT-HANDED' with the 'ZPOSITIVEAXIS' representing up. Note that since the view is down the z-axis, the up direction degenerates to 'YPOSITIVEAXIS'. The z-axis clipping planes are 150. (hither plane) and 1.0E+30 (yon plane) with 'NOZCLIPPING'.

II. Default High Level Plotting Options

For each call to UPLOT or UPLOT1, the data values which represent the curves are examined, and a 'NEWSCALE' is created. UAXIS will be invoked to create 'XYAXES'. The data values will be examined, and appropriate limits for the data established. Numeric labels only will be output for the X axis and the Y axis. The values which appear at the tic marks on the axes will be 'neat' numbers, and the axes will be positioned at the 'EDGE' of the plot. Both the X axis and the Y axis will be drawn in a linear coordinate space. The axis lines will be ticced. If a time series axis is plotted by invoking Subroutine UTAXIS, the default interval for the X axis will be 'DATE'. No curves will be fit to the data values, but if 'FITPOLYNOMIAL' is requested, then subroutine UPLOT will attempt to fit a fifth degree polynomial to the data.

III. Summary

COORDINATE SPACE:	'VIRTUAL'
COORDINATE TYPE:	'ABSOLUTE'
COORDINATE SYSTEM:	'RECTANGULAR'
COORDINATE AXIS:	'SYSTEM'
VIRTUAL WINDOW:	0.0 TO 100.0 X DIRECTION 0.0 TO 100.0 Y DIRECTION
DISPLAY AREA:	Largest square area which is right-justified on the display surface of the device.
DEVICE SPACE UNITS:	'INCHES'
ANGULAR UNITS:	'DEGREES'
LINE TYPE:	'LINE' or 'LNULL'
SYSTEM CHARACTER:	asterisk (*)
TIC INTERVAL:	10.0 virtual units
DASH SPECIFICATION:	56.
CHARACTER TYPE:	'HARDWARE'
CHARACTER FONT:	'GOTHIC'
CHARACTER SIZE:	'MEDIUM'
SOFTWARE CHARACTER SIZE:	5.0 virtual units horizontal 7.0 virtual units vertical
SOFTWARE CHARACTER FONT	'SGCS'
OUTPUT FORMAT:	'TEXT'
ALPHANUMERIC MARGINS:	Device display surface boundaries.

DIGITS OF PRECISION:	4
ERROR HANDLING:	'IMMEDIATEOUTPUT'
AXIS SCALING:	'AUTOSCALE'
AXIS LABELING:	'XNUMERICLABEL'
	'YNUMERICLABEL'
AXIS POSITIONING:	'EDGEAXIS'
AXIS TYPE:	'TICAXIS'
AXIS EXISTENCE:	'XYAXES'
AXIS COORDINATES:	'NOLOGARITHMS'
TIME SERIES AXIS SCALE:	'DATE'

CHAPTER III

GCS USER SUBROUTINES

This chapter provides the detailed description of each user callable subroutine. Each routine is presented in the following manner: The FUNCTION paragraph gives a brief statement of the main purpose of the subroutine. The CALLING SEQUENCE is presented with an explanation of each parameter required for the subroutine. Any USET or UPSET options that will affect the operation of the subroutine are listed under the OPTIONS paragraph. The COMMENTS paragraph provide the detailed description of all of those features considered to be both unique and important to each specific subroutine. On the following pages is listing of all of the user subroutines. The features that are only calculated in the three dimensional version are specified by italics.

ALPHABETICAL LISTING OF GCS SUBROUTINES

Subroutine UAIN *Interactive*

FUNCTION:

This routine obtains one character as entered to signify the completion of a graphics input operation. This is usually the result of a keyboard trike.

CALLING SEQUENCE:

CALL UAIN (CHAR)

Where

CHAR **contains the character obtained from the terminal in Hollerith format.**

OPTIONS which may apply:

'KEYBOARD', 'FUNCTIONKEY', 'CURSOR', 'JOYSTICK', 'BALL', 'MOUSE', 'LIGHTPEN', 'TABLET'

COMMENTS:

The character requested will be obtained from the specified graphics input device. If no explicit USET call has been used to specify an input device (see UGRIN), the character will be obtained from the principal input device for the terminal being used.

The USET option 'BALL' refers to Trackball input.

Programming Notes:

Subroutine UALPHA

FUNCTION:

This routine sets the terminal to alphanumeric mode so that character output will appear on the control device.

CALLING SEQUENCE:

CALL UALPHA

OPTIONS which may apply

No options apply.

COMMENTS:

This routine should be called whenever the user wants to insure that the terminal is in alphanumeric mode and to flush any buffered graphics I/O (e.g., before FORTRAN I/O).

Programming Notes:

Subroutine UAOUT

FUNCTION:

This routine displays one character at the current beam position subject to the margins or windows which are currently set. The beam is positioned ready to output an adjacent character.

CALLING SEQUENCE:

CALL UAOUT (CHAR)

Where

CHAR **is an alphanumeric character expressed in FORTRAN Hollerith (left-justified, blank-filled) format.**

OPTIONS which may apply:

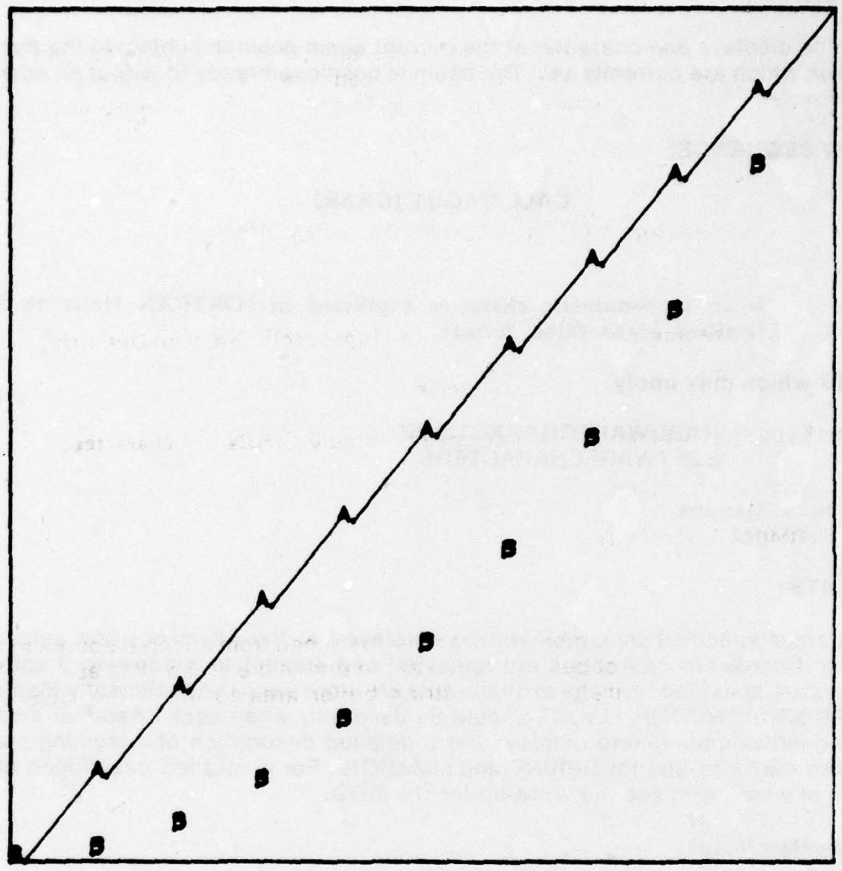
Character Type: **'HARDWARECHARACTERS'**
'SOFTWARECHARACTERS'

Alphanumeric Margins
Window Settings

COMMENTS:

The character specified by CHAR will be displayed at the beam position subject to margining if hardware characters are specified and subject to windowing if software characters are specified. Strings of characters are more easily and efficiently displayed using UPRINT or UWRITE. UAOUT should be used only when each character must be processed individually before display. For a detailed description of margining and its effects see the write-ups for UPRINT and UMARGN. For a detailed description of the concepts of windowing see the write-up for UWINDO.

Programming Notes:



```
      DIMENSION X(11),Y(11),Z(11)
      DATA X/0.,10.,20.,30.,40.,50.,60.,70.,80.,90.,100./
      DATA Y/0.,10.,20.,30.,40.,50.,60.,70.,80.,90.,100./
      DATA Z/0.,1.,4.,9.,16.,25.,36.,49.,64.,81.,100./
      CALL USTART
      CALL UOUTLN
      DO 1 I = 1, 11
      CALL UPEN (X(I),Y(I))
    1 CALL UAOUT ('A')
      CALL USET ('NOLINE')
      CALL UMOVE (X(1),Z(1))
      DO 2 I = 1, 11
      CALL UPEN (X(I),Z(I))
      CALL UAOUT ('B')
    2 CONTINUE
      CALL UEND
      STOP
      END
```

Subroutine UAPEND

FUNCTION:

This routine copies characters from the input Hollerith character string placing them in the output character buffer. When the designated number of characters have been moved, the GCS string terminator is appended to the end of the string in the output buffer.

CALLING SEQUENCE:

CALL UAPEND(COUNT,DATAIN,DATOUT)

Where

COUNT	is the number of characters in the input Hollerith character string.
DATAIN	is the input Hollerith character string.
DATOUT	is available or array large enough to hold COUNT+1 characters

OPTIONS which may apply:

UPSET('TERMINATOR',CHAR)

COMMENTS:

This routine is useful if a character string is being obtained from an input source and the GCS string terminator has not been included at the end of the string. It is assumed that enough room is available in the output string buffer area to insert the GCS string terminator character.

DATAIN, and DATOUT need not be distinct.

Programming Notes:

Subroutine UARC

FUNCTION:

This subroutine draws a part of a circle; i.e., an arc with a given center, radius, and angular span starting at the current bean or pen position. The arc will be either treated as a curvilinear line segment with the current line option (see UPEN for description of line options) or treated as a collection of straight line segments, each of which is affected by the current line option. Which method is used to draw the arc will depend on whether 'CONTINUOUS' or 'SEGMENTED' has been set.

CALLING SEQUENCE:

CALL UARC (X,Y,ANGLE)

Where

X	**is the X or RADIUS of the center of the circle in current user units.**
Y	**is the Y or THETA coordinate of the center of the circle in current user units.**
ANGLE	**is the number of angular units to be subtended by the arc.**

OPTIONS which may apply:

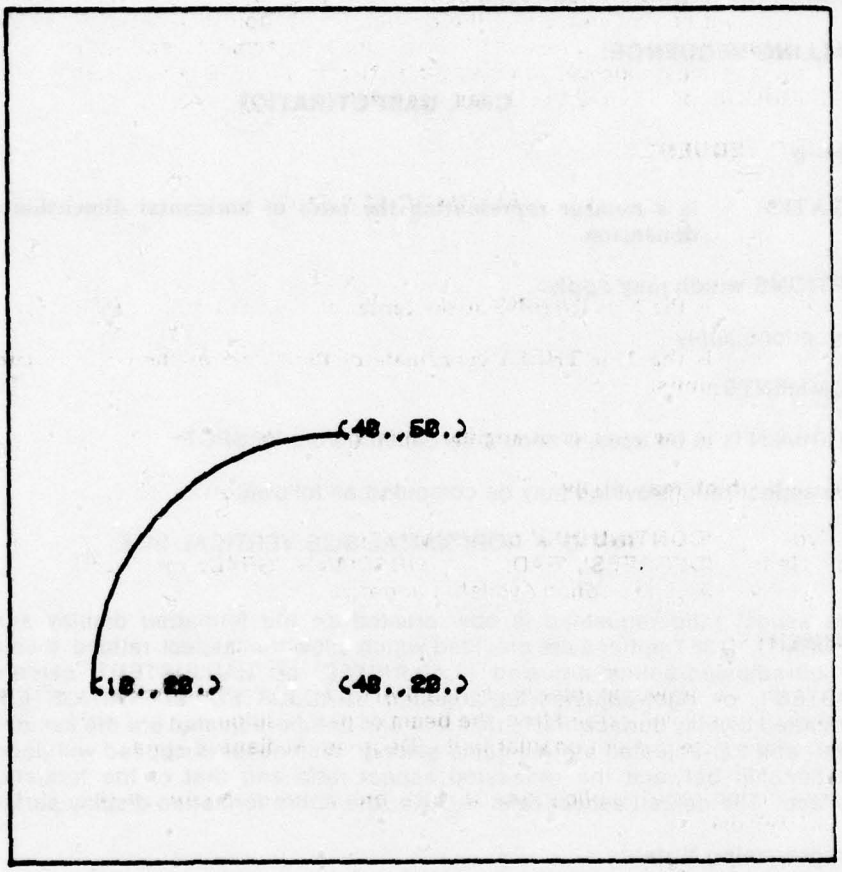
Arc Type:	**'CONTINUOUS'** or 'SEGMENTED'
Angle Units:	**'DEGREES'**, 'RADIANS', *'PIRADIANS'*, *'GRADS'* or *'MILS'*
Line Types:	Any line option available under UPEN

COMMENTS:

The arc will be drawn counterclockwise if ANGLE is positive or clockwise if ANGLE is negative. The arc will extend from the beam or pen position upon invoking UARC for the angle specified in current angular units (Degrees, Radians, or *Piradians).*

In three dimensional applications, the arc produced by a call to UARC will lie in the current X-Y plane.

Programming Notes:




```
CALL USTART
CALL UOUTLN
CALL UPEN! (40.0,50.0,'NCOORDINATES')
CALL UMOVE (40.0,50.0)
CALL UARC (40.0,20.0,90.0)
CALL UWHERE (X,Y)
CALL UPEN! (X,Y,'NCOORDINATES')
CALL UPEN! (40.0,20.0,'NCOORDINATES')
CALL UEND
STOP
END
```

FUNCTION:

This routine restricts the physical display surface boundaries to the largest area which satisfies the provided aspect ratio.

CALLING SEQUENCE:

CALL UASPCT(RATIO)

Where

RATIO **is a number representing the ratio of horizontal dimension to vertical dimension.**

OPTIONS which may apply:

No options apply

COMMENTS:

If UDIMEN is to be used, it should be called before UASPCT.

The aspect ratio provided may be computed as follows:

RATIO = HORIZONTAL SIZE/VERTICAL SIZE

The aspect ratio requested is now created on the formatted display surface (see UFORMT). USET options are provided which allow the 'aspect-ratioed' display surface to left-adjusted/bottom-adjusted ('LADJUSTED' or 'BADJUSTED'), centered ('CADJUSTED'), or right-adjusted/top-adjusted ('RADJUSTED' or 'TADJUSTED') on the formatted display surface. Note that left- and bottom-adjusted are the same setting and right- and top-adjusted are the same setting. Whichever is applied will depend on the relationship between the requested aspect ratio and that of the formatted display surface. The default aspect ratio is 0 (i.e., the entire formatted display surface).

Programming Notes:

Subroutine UAVERG

FUNCTION:

To compute a moving average forecast for time series data.

CALLING SEQUENCE:

CALL UAVERG (ARRAY,POINTS,FCST,TIME)

Where

ARRAY	**is a real array of size POINTS which is the input time series data.**
POINTS	**is the number of input points.**
FCST	**is a real array of size POINTS-TIME+1 which is the time series moving average forecast for periods TIME+1 to POINTS+1**
TIME	**is the number of previous periods to be used for each average.**

OPTIONS which may apply:

No options apply.

COMMENTS:

The number of input data points must be greater than one. The number of previous periods which are used to compute the moving average must be greater than zero and not greater than the number of input data points.

Note that this routine represents one of the basic tools for time series analysis. The user is urged to consult a qualified reference on time series analysis for a complete description of the methods involved in this routine. Other techniques such as base series correction, cyclic analysis, and exponential smoothing are necessary to synthesize a meaningful, composite forecast.

Programming Notes:

Subroutine UAXIS

FUNCTION:

This routine creates axes of the type specified by the user. Also provided are numeric scaling and labeling and alphanumeric labeling. The axes are created in the UDAREA which is set prior to entering UAXIS. The user window may also be modified to reflect the actual scaling used. Upon return from UAXIS, the user window will be set to the scaling used to draw the axes. The device area will be reduced if 'EDGEAXES' are specified to accommodate the labeling requested. If 'ZEROAXES' or 'PENAXES' are specified, the device area will not be changed since the labeling will appear in the device area setting.

CALLING SEQUENCE:

CALL UAXIS (XMIN,XMAX,YMIN,YMAX)

Where

XMIN	is the minimum X-value to be contained on the axis in virtual units.
XMAX	is the maximum X-value to be contained on the axis in virtual units.
YMIN	is the minimum Y-value to be contained on the axis in virtual units.
YMAX	is the maximum Y-value to be contained on the axis in virtual units.

OPTIONS which may apply:

Axis Existence:	'NOAXES', 'XAXIS', 'YAXIS', 'XYAXIS'
Axis Format:	'PLAINAXES', 'TICAXES', GRIDAXES'
X-Axis Labeling:	'XNUMERIC', 'XALPHALABEL', 'XBOTH', 'NOXLABEL'
Y-Axis Labeling:	'YNUMERIC','YALPHALBEL', 'YBOTH', 'NOYLABEL'
Axis Positioning:	'ZEROAXES', 'XZEROYEDGE', 'YEDGEXZERO', 'XEDGEYZERO', 'YZEROXEDGE', 'EDGEAXES', 'PENAXES'
Numeric Label Format:	'IFORMAT', 'GFORMAT', **'BESTFORMAT'**, 'EFORMAT'
Scaling Type:	**'AUTOSCALE'**, 'FULLSCALE', 'OWNSCALE'
X-Axis Type:	**'NOLOGARITHMS'**, 'LOGXAXIS'
Y-Axis Type:	**'NOLOGARITHMS'**, 'LOGAXIS'
Log Base:	UPSET('BASE',base) where base is either a number or the character string IHE
Coordinate Type:	'POLAR', **'RECTANGULAR'**

COMMENTS:

UAXIS is a general purpose axis drawing and/or plot labeling routine. The options indicated above are independent of each other type designations so that several different options might appropriately be chosen. The UAXIS subroutine is also used by the high-level graphing routines (see UPLOT) to create their axes.

Each of the USET options which effect the UAXIS routine is described below. These same options apply when subroutine UPLOT or subroutine UPLOT1 is called.

Programming Notes:

Axis Existence:

NOAXES	–	Neither the X-axis nor the Y-axis will be displayed.
XAXIS	–	Only the X-axis will be displayed
YAXIS	–	Only the Y-axis will be displayed
XYAXES	–	both the X- and Y-axis will be displayed

Axis Format:

PLAINAXES	–	The axis specified will appear as a solid line.
TICAXES	–	The axis specified will appear as a ticced line. The tic intervals will be determined from the current setting of the UPSET parameters TICX and TICY which should be specified in the same units to be used in plotting the functions. If a tic interval of zero is specified, a suitable tic interval will be chosen by UAXIS.
GRIDAXES	–	A grid will be formed in the display area. The interval between grid lines is determined in the same way as for TICAXES. In the case of log axes, the tic interval refers to the cycle interval. In the case of polar axes, the tic interval refers to the radial distance between the concentric circles of the grid. This radial distance is obtained from the value of TICX.

Axis Positioning:

The user is expected to establish the UDAREA he desires before calling UAXIS. For all the options indicated below, no part of the axes or the associated labels will appear outside the UDAREA which is set prior to entry to UAXIS.

EDGEAXES	–	The room needed for the 'axes' specified labeling is reserved from the UDAREA set on input and a new UDAREA is established which describes the remaining plotting area. The axes will be drawn along the edge of this new UDAREA.
ZEROAXES	–	If the zero point falls between the minimum and maximum input values for either X or Y, the axes are drawn along the zero value from border to border of the UDAREA. Numeric labels will appear directly below the X-axis and to the left of the Y-axis. Alphanumeric X-labels will appear along the bottom edge of the UDAREA and the alphanumeric Y-labels will appear along the left edge of the UDAREA. If the zero point for either axes is not between the minimum input values, the axis will default to EDGEAXIS format.
PENAXES	–	The intersection of the axes is defined to be at the beam position upon entry to UAXIS. If the beam position is outside the range of values furnished as input, the axis which is outside the range will default to EDGEAXIS format.
YEDGEXZERO XZEROYEDGE	–	The X-axis will be in ZEROAXIS format and the Y-axis will be in EDGEAXIS format.
XEDGEYZERO YZEROXEDGE	–	The X-axis will be EDGEAXIS format and Y-axis will be in ZEROAXIS format.

Axis Type:

NOLOGARITHMS	—	The X and Y axes are both linear
XLOGARITHM	—	The X axis is a logarithm based axis while the Y axis is linear
YLOGARITHM	—	The Y axis is a logarithm based axis while the X axis is linear
XYLOGARITHM LOGARITHMIC	--	Both the X and Y axes are logarithm axes

Note: The base of the logarithms used in the axes is set by a call to UPSET, i.e., CALL UPSET ('BASE', base) where base is either a numeric value or the character string IHE.

Coordinate Type:

RECTANGULAR	—	The input coordinates are rectangular. Logarithmic or linear axes can be specified.
POLAR	—	Input coordinates are polar. Only polar axes will be created by UAXIS.

X-Axis Labeling:

NOXLABEL	—	No labeling should appear along the X-axis.
XNUMERIC	—	Numeric labels should be created along the X-axis at the tic-intervals specified by TICX (see Axis Format above). The values for the numeric labels will be derived as specified in the scaling options described below.
XALPHANUMERIC	—	Alphanumeric labels will appear in the location specified in the Axis Position Option (see above). The contents of the label can be specified by calling UPSET with arguments 'XLABEL' and a GCS character string as the value (for a description of GCS character strings see UPRINT).
XBOTHLABELS	—	Both the alphanumeric and numeric labels will appear along the X-axis.

Y-Axis Labeling:

NOYLABEL	—	No labeling should appear along the Y-axis.
YNUMERIC	—	labels should be created along the Y-axis at the tic intervals specified by TICY (see Axis Format above). The values for the numeric labels will be derived as specified in the scaling options described below.
YALPHANUMERIC	—	Alphanumeric labels will appear in the location specified by the Axis Positioning Option (see above). The contents of the label can be specified by calling UPSET with arguments 'YLABEL' and a GCS character string as the value (for a description of GCS character strings see UPRINT).

YBOTHLABELS — Both the alphanumeric and numeric labels will appear along the Y-axis.

Scaling Type:

AUTOSCALE — The range of values to be plotted is examined and scaling occurs so that 'nice' numbers will appear at the tic marks. A slight expansion of the range may be necessitated to incorporate this feature. The UWINDO is set to the expanded range as identified.

FULLSCALE — The full range of values specified in the input arguments are examined. If zero is not within the range of X values, the appropriate limit is extended to zero. The UWINDO is set to this resulting range.

OWNSCALE — The UWINDO setting upon entry to UAXIS is used to determine the scale limits. No change is made to the UWINDO and a zero value is not forced to the X -scale. 'Nice' numbers may not necessarily appear as numeric labels.

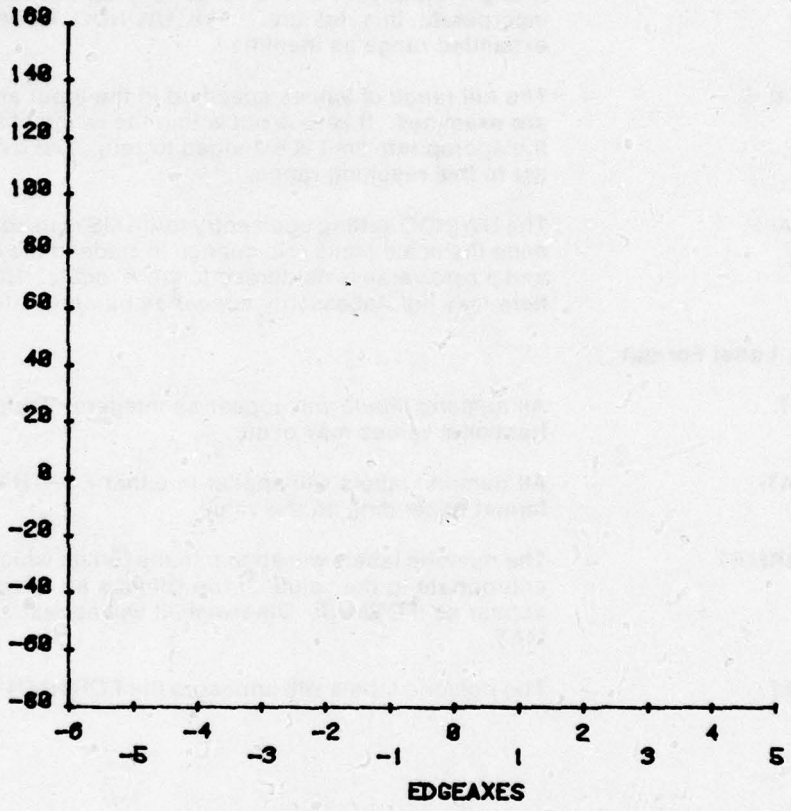
Numeric Label Format:

IFORMAT — All numeric labels will appear as integers. Truncation of fractional values may occur.

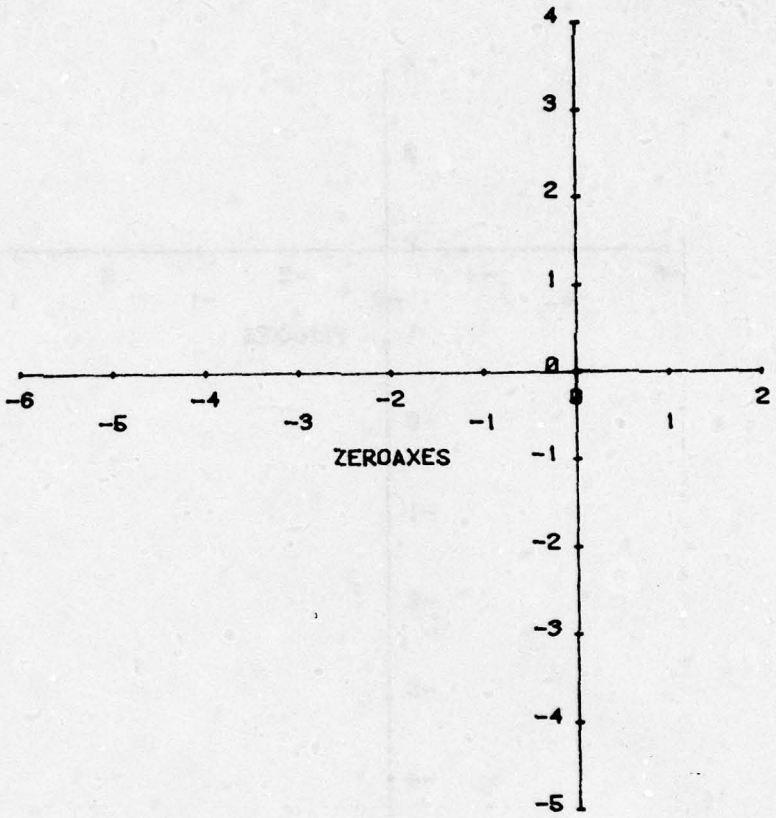
GFORMAT — All numeric labels will appear in either FORTRAN F or E format depending on the value.

BESTFORMAT — The numeric labels will appear in the format which is most appropriate to the value. If the value is an integer, it will appear as IFORMAT. Otherwise, it will appear as GFORMAT.

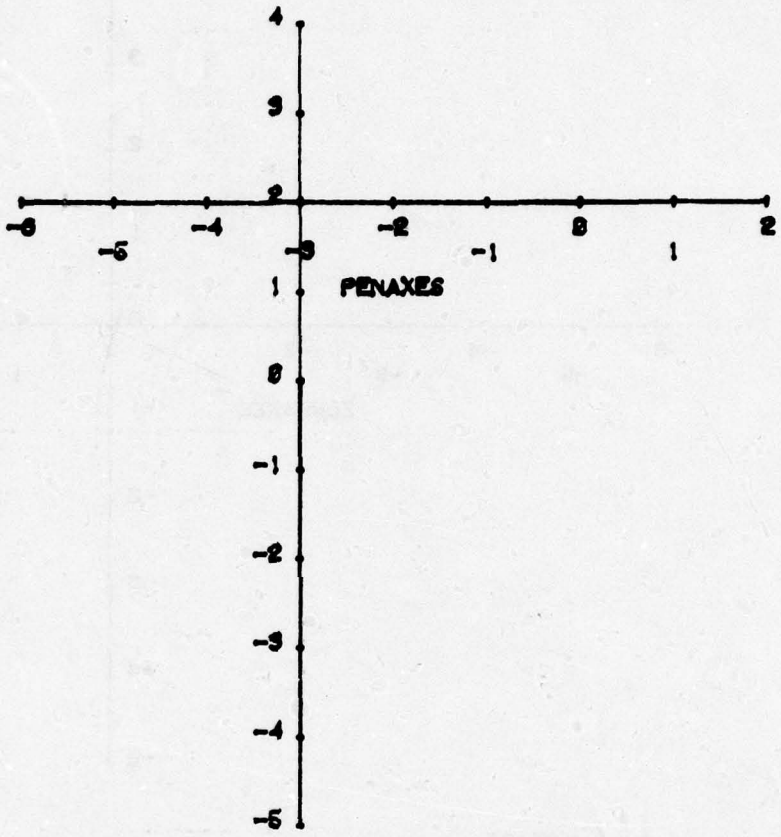
EFORMAT — The numeric labels will appear in the FORTRAN E format.



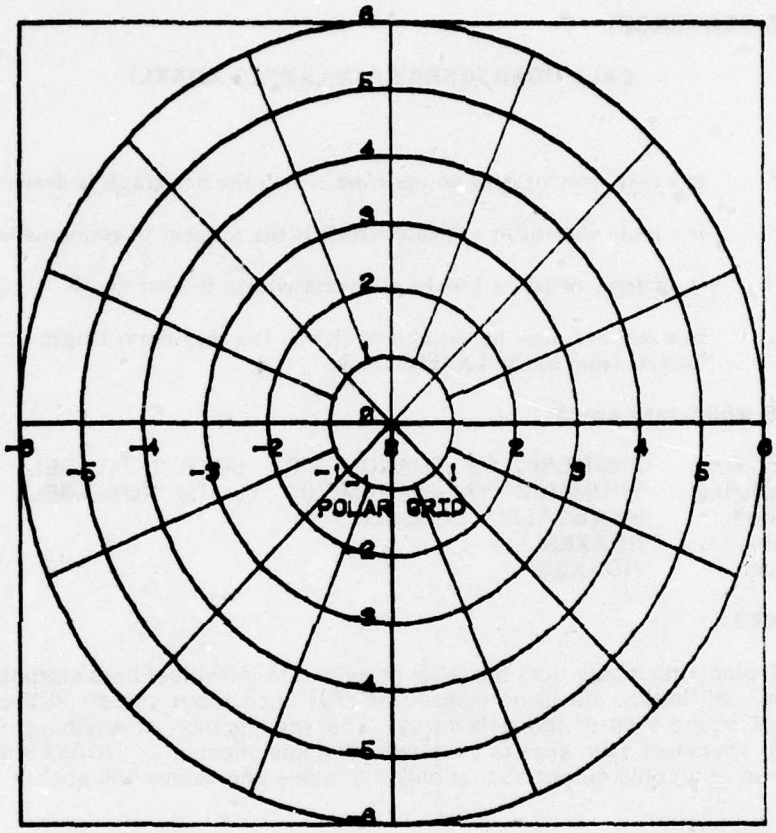

```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('XBOTHLABELS')
CALL UPSET ('XLABEL','      EDGEAXES;')
CALL UAXIS (-5.32,4.69,-83.4,156.459)
CALL UEND
STOP
END
```




```
      CALL USTART
      CALL UPSET ('TERMINATOR',';')
      CALL USET ('ZEROAXES')
      CALL USET ('XBOTHLABELS')
      CALL UPSET ('XLABEL','ZEROAXES;')
      CALL UAXIS (-6.,2.,-5.,4.)
      CALL UEND
      STOP
      END
```




```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('PENAXES')
CALL USET ('XBOTHLABELS')
CALL UPSET ('XLABEL','PENAXES;')
CALL UMOVE (-3.,2.)
CALL UAXIS (-6.,2.,-5.,4.)
CALL UEND
STOP
END
```




```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL UPSET ('XLABEL','POLAR GRID;')
CALL USET ('POLAR')
CALL USET ('GRID')
CALL USET ('XBOT')
CALL USET ('ZEROAXIS')
CALL UAXIS (-6.,6.,0.,360.)
CALL UEND
STOP
END
```

Subroutine UBAR

FUNCTION:

This routine draws a general purpose bar graph. The current user display area will be used and the bar graph will be drawn and scaled according to user specifications.

CALLING SEQUENCE:

CALL UBAR (ARRAY,PTS,LABELS,XMAXL)

Where

ARRAY	**is a real array of data points from which the bar graph is drawn.**
PTS	**is a real constant or variable which is the number of elements in ARRAY.**
LABELS	**is an array of labels for the elements within the bar graph.**
XMAXL	**is a real constant or variable which is the maximum length of a label or longest label in the LABELS array.**

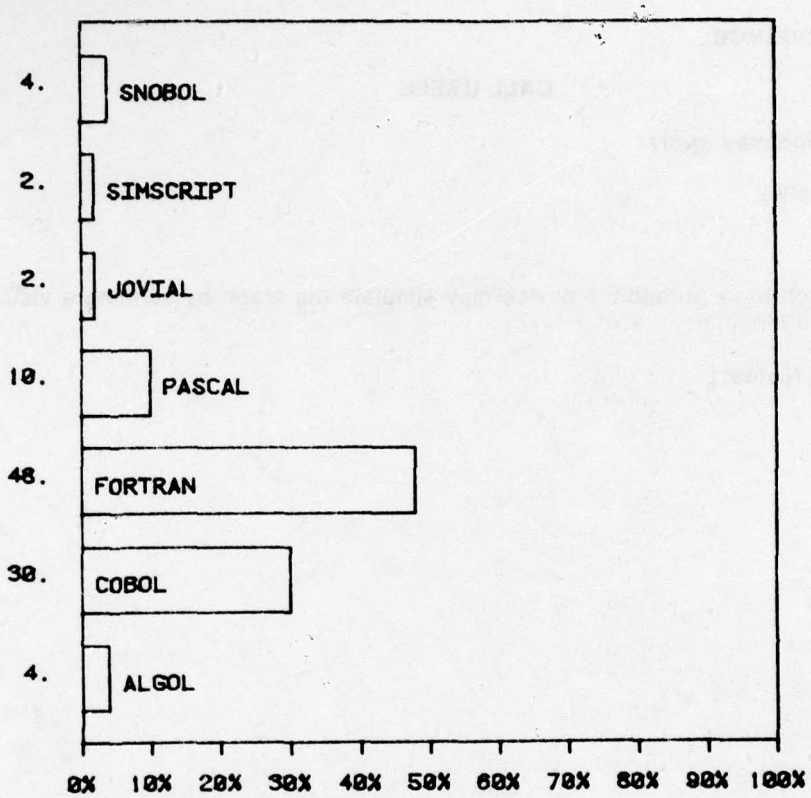
OPTIONS which may apply:

X-Axis Labeling:	**'XNUMERIC'** 'XALPHANUMERIC' 'XBOTH' 'NOXLABEL'
Y-Axis Labeling:	**'YNUMERIC'** 'YALPHANUMERIC' 'YBOTH' 'NOYLABEL'
Display Area:	**'NEWSCALE'** 'OLDSCALE'
Axis Existence:	**'NOAXES'**
Axis Format:	**'TICAXES'**

COMMENTS:

UBAR will place the labels from the label array on the individual bars starting from the bottom up. All input data is normalized so that each input value will appear as a percentage of the total of the data array. The specification of anything other than 'TICAXES' will cause plain axes to be drawn. The specification of 'NOAXES' will cause suppression of graphic output so that only the labels and values will appear.

Programming Notes:




```
DIMENSION DATA(7)
CHARACTER LABELS*10(7)
DATA DATA/4.,30.,48.,10.,2.,2.,4./
DATA LABELS/'ALGOL;','COBOL;','FORTRAN;','PASCAL;','JOVIAL;',
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('XBOTHLABELS')
CALL UPSET ('XLABEL','TYPICAL LANGUAGE UTILIZATION;')
CALL UBAR (DATA,7.,LABELS,10.)
CALL UEND
STOP
END
```

Subroutine UBELL *Interactive*

FUNCTION:

This routine causes an audible alarm (bell) to sound at the terminal.

CALLING SEQUENCE:

CALL UBELL

OPTIONS which may apply:

No options apply.

COMMENTS:

Terminals which have no audible device may simulate the alarm by sending a visual message to the terminal.

Programming Notes:

FUNCTION:

This routine invokes an already constructed data structure. The request coordinate system transformation is established and then each element of the structure is executed.

CALLING SEQUENCE:

CALL UCALL(X,Y,SX,SY,R,NAME)

Where

X,Y	**are the coordinates of the origin of the structure coordinate system in current coordinate system units.**
SX, SY	**are the scale factors of the structure coordinate system.**
R	**is the rotation of the structure coordinate system in relation to the current reference system.**
NAME	**is the eight character name of the structure being invoked. This structure must be defined in the current 'LIBRARY' file.**

COMMENTS:

The current coordinate system is saved prior to applying the structure coordinate transformation. Up to three levels of recursive calls to UCALL can be handled.

This subroutine can be used in a 3-D environment if the transformation so specified is to be applied only in the XY plane of the current coordinate system.

Programming Notes:

Subroutine UCHART

FUNCTION:

This routine draws a grouped bar chart for multi-valued data. The current user display area will be used and the chart will be drawn and scaled according to user specifications.

CALLING SEQUENCE:

CALL UCHART(ARRAY,GROUPS,BARS,LABELS,YMAXL)

Where

ARRAY	**is a real array of data points from which the chart is drawn**
GROUPS	**is a real constant or variable which is the number of groups**
BARS	**is a real constant or variable which is the number of bar graphs in each group**
LABELS	**is an array of labels for each group**
YMAXL	**is a real constant or variable which is the maximum length of a label or the length of the longest label in the array LABELS**

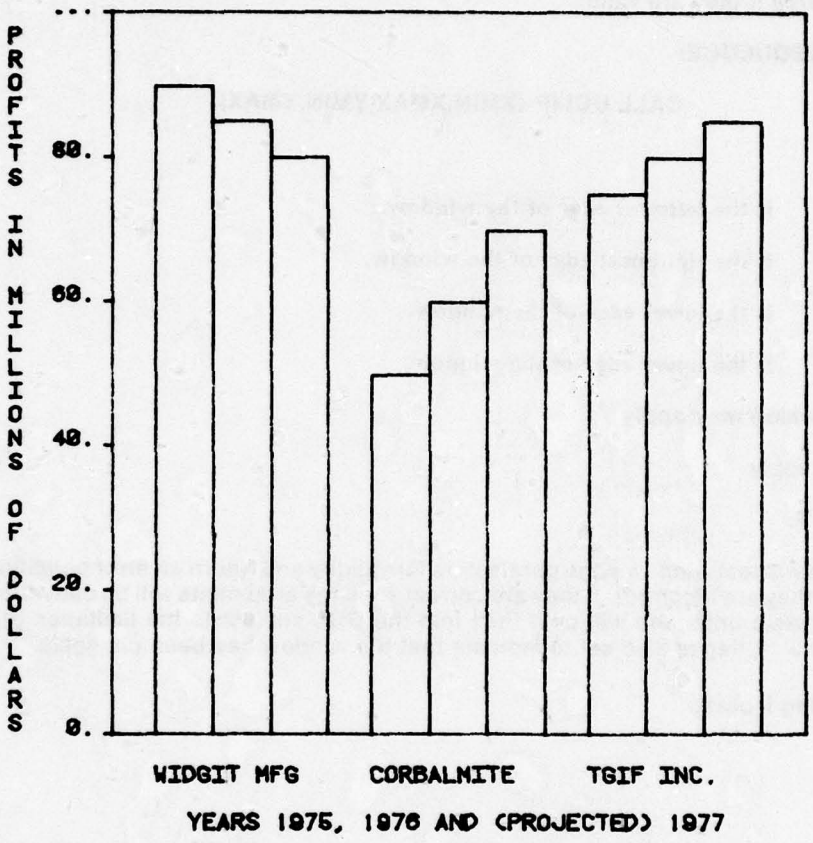
OPTIONS which may apply:

X-Axis Labeling:	**'XNUMERIC',** 'XALPHABETIC', 'XBOTH', 'NOXLABEL'
Y-Axis Labeling:	**'YNUMERIC',** 'YALPHABETIC', 'YBOTH', 'NOYLABEL'
Display Area:	**'NEWSCALE',** 'OLDSCALE'
Axis Existence:	**'NOAXES'**
Axis Format	**'TICAXES'**

COMMENTS:

The specification of anything other than 'TICAXES' will cause plain axes to be drawn. The specification of 'NOAXES' will cause suppression of graphic output so tht only the labels and values will appear. The X-axis and Y-axis labeling options will not affect the output of the group labels. If group labels are not desired, then YMAXL should be set to zero.

Programming Notes:




```
DIMENSION YEARS(9)
CHARACTER LABELS*10(3)
DATA YEARS/90.,85.,80.,50.,60.,70.,75.,80.,85./
DATA LABELS/'WIDGIT MFG,','CORBALMITE,','TGIF INC.,'/
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('XALPHABETIC')
CALL UPSET ('XLABEL','YEARS 1975, 1976 AND (PROJECTED) 1977;')
CALL USET ('YBOTHLABELS')
CALL UPSET ('YLABELS','PROFITS IN MILLIONS OF DOLLARS;')
CALL UCHART (YEARS,3.,3.,LABELS,10.)
CALL UEND
STOP
END
```

Subroutine UCLIP

FUNCTION:

This routine tests the validity of the clipping parameters and stores them into the GCS COMMON area if they are valid.

CALLING SEQUENCE:

CALL UCLIP (XMIN,XMAX,YMIN,YMAX)

Where

XMIN	is the leftmost edge of the window.
XMAX	is the rightmost edge of the window.
YMIN	is the lower edge of the window.
YMAX	is the upper edge of the window.

OPTIONS which may apply

No options apply

COMMENTS:

This routine will test window edge parameters for validity and return an error condition to the caller if they are incorrect. If they are correct, then the arguments will be converted to the device basic units and will post then into the GSA subject to the limitation of the device space. A flag is also set to indicate that the window has been changed.

Programming Notes:

FUNCTION:

This routine causes the currently open segment/frame to cease to be open.

CALLING SEQUENCE:

CALL UCLOSE(SEGID)

Where

SEGID	**is the segment identifier of the currently open segment (see UOPEN for description of SEGID).**

OPTIONS which may apply:

Segment/Frame Identifier Mode: **'FNAME'**, 'FNUMBER'

COMMENTS:

The currently open segment is closed irrelevant of the type of segment or the existence of image transformations. The closed frame will replace an existing segment with the same identifier thus insuring that a complete image is always visible.

An error will be generated if SEGID does not agree with that of the currently open segment. However, the segment will still be closed.

Programming Notes:

Subroutine UCOLOR

Function:

This routine defines entries in a program modifiable table of colors.

Calling sequence:

CALL UCOLOR(CLRIDX,CLRCNT,CLRNAM,CLRVAL)

Where

CLRIDX	is the index number of the color table entry to be redefined. Note that this is the same number used when setting the color using UPSET or USET
CLRCNT	specifies the number of components in the color definition vector. At least one component and a maximum of seven components may be specified
CLRNAM	is an array containing the names of the color components. Valid names are 'RED', 'GREEN', 'YELLOW', 'BLUE', 'MAGENTA' 'CYAN', AND 'WHITE'. Only the first four characters of each name are significant. The names must occupy no more than one word in the array.
CLRVAL	is an array containing the percentages of intensity for each component specified in the CLRNAM array. Percentage values may range from 0 to 100 and indicate the proportion of maximum application of the specified color component.

OPTIONS which may apply:

No options apply

COMMENTS:

This routine will verify correct color definitions for devices which have no modifiable color tables.

If duplicate color component names are provided, an error will be generated and the color will not be defined.

This routine uses the Red/Green/Blue (RGB) or Magenta/Yellow/Cyan method of color definition. It should be noted that the intensity attribute (UPSET,'BRIGHTNESS') serves as a master intensity control to modulate the colors.

Programming Notes:

Subroutine UCONIC

FUNCTION:

This routine allows the user to draw a generalized conic section having a given focus, directrix, eccentricity, and starting and ending sweep positions. The conic sections that may be drawn are the circular arc, ellipse, parabola, hyperbola, and, if the distance between the focus and the directrix is zero, point.

CALLING SEQUENCE:

CALL UCONIC (X,Y,P,E,TH1,TH2)

Where

X	is the X- or RADIUS coordinate of the focus of the conic section.
Y	is the Y- or THETA coordinate of the focus of the conic section.
P	is the distance from the focus to the directrix.
E	is the eccentricity.
TH1	is the initial sweep position in current angular units.
TH2	is the terminal sweep position in current angular units.

OPTIONS which may apply:

'CONTINUOUS', 'SEGMENTED' (see UARC write up)
Line options (see UPEN write up)
'RADIANS', **'DEGREES'**, *'PIRADIANS', 'GRADS' or 'MILS'* (see UARC write up)
Pen Coordinate Options (see UPEN write up)
UPSET Option: 'ORIENTATION'

COMMENTS:

The basis for UCONIC is the generalized conic equation:

$$R=(E*P)/(1-E*COS(THETA))$$

By suitably modifying the values of the parameters P and E, all types of conic sections can be created. The following describes the effects induced by different values of P of E.

P.GT.0	the focus is to the right (below the directrix).
P.LT.0	the focus is to the left (above the directrix).
P = 0	the conic will be a point at X,Y.
E = 0	the conic is a circular arc with center at X,Y and RADIUS P/2 subtending the angular range from the TH1 to TH2.
0.LT.ABS (E).LT.1	the conic is an ellipse.
ABS(E) = 1	the conic is a parabola.

ABS(E).GT.1	the conic is a hyperbola.
E.GT.0	the conic is oriented along the X-axis
E.LT.0	the conic is oriented along the Y-axis

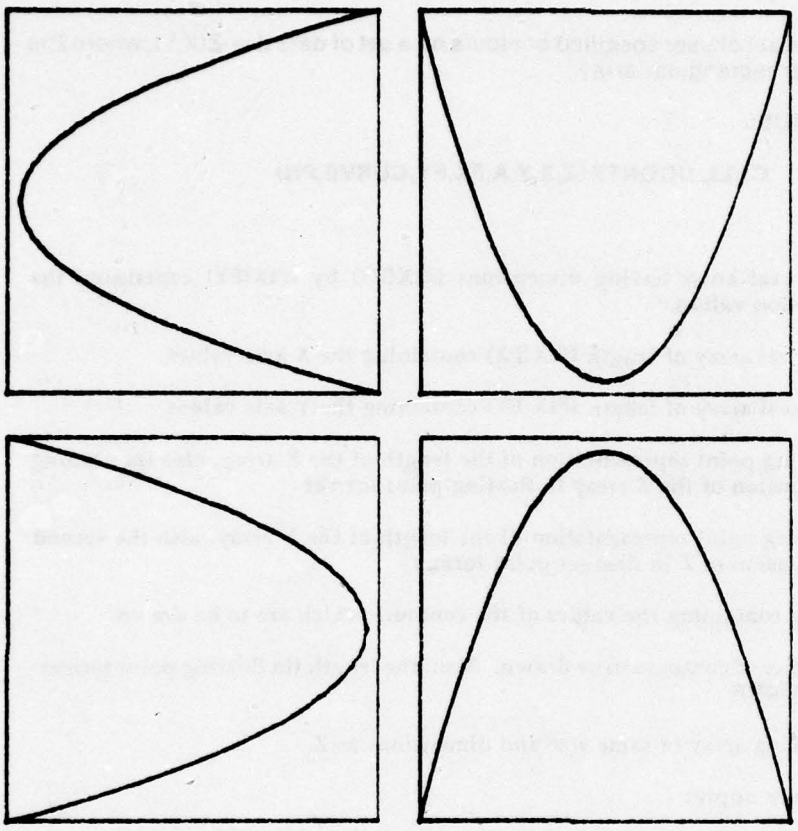
All of the line options and coordinate options which apply to UPEN will also apply to UCONIC. Thus, conic sections may be rotated to any angle by defining a suitable user coordinate system.

If a conic section of 360 degree sweep is specified, the line terminator will not be drawn when 'CONTINUOUS' mode has been specified. See the UARC write up for more information on the use of 'CONTINUOUS' and 'SEGMENTED'. The circle is a degenerate case not fully handled by the generalized conic equation. For completeness a circle with arbitrarily assigned RADIUS of P/2 will be generated when E has a value of zero.

If an angle of orientation of the conic is specified via UPSET, then the conic section will be oriented as specified around the point of intersection of the directrix and the semi-major axis.

In three dimensional applications, the conic section drawn by UCONIC will lie in the current XY plane.

Programming Notes:



```
      DIMENSION X(4),Y(4),P(4),E(4)
      DATA INDEX,Y0,X,Y,P,E/0,5.73,10.,50.,90.,50.,50.,10.,50.,90.,
     &                      13.,-13.,-13.,13.,1.,-1.,1.,-1./
      CALL USTART
      DO 1 I = 1, 2
      X0 = -1.82
      Y0 = Y0 - 2.86
      DO 1 J = 1, 2
      X0 = X0 + 2.86
      INDEX = INDEX + 1
      CALL UDAREA (X0,(X0+2.57),Y0,(Y0+2.57))
      CALL UOUTLN
      CALL UCONIC (X(INDEX),Y(INDEX),P(INDEX),E(INDEX),0.0,360.0)
    1 CONTINUE
      CALL UEND
      STOP
      END
```

Subroutine UCONTR

FUNCTION:

This routine draws a set of user specified contours on a set of data Z = Z(X,Y), where Z is defined on a regular rectangular array.

CALLING SEQUENCE:

CALL UCONTR(Z,X,Y,A,FX,FY,CURVE,FN)

Where

Z	**is a real array having dimensions IFIX(FX) by IFIX(FY) containing the function values.**
X	**is a real array of length IFIX(FX) containing the X axis values**
Y	**is a real array of length IFIX(FY) containing the Y axis values**
FX	**floating point representation of the length of the X array, also the leading dimension of the Z array in floating point format**
FY	**floating point representation of the length of the Y array, also the second dimension of Z in floating point format**
CURVE	**array containing the values of the contours which are to be drawn**
FN	**number of contours to be drawn. Also, the length (in floating point format) of CURVE**
A	**working array of same size and dimensions as Z.**

OPTIONS which may apply:

Coordinate Type Options:	**'RECTANGULAR'**, 'POLAR', 'LOGARITHMIC', UAXIS Options, (see UAXIS write up)
Scaling Type Options:	**'AUTOSCALE'**, 'FULLSCALE', 'OWNSCALE'
Scale Availability Options:	**'NEWSCALE'**, 'OLDSCALE'
Pen Options:	See UPEN write up

COMMENTS:

The algorithm used in UCONTR is based on linear interpolation between the grid points.

Programming Notes:

Subroutine UCOSYS

FUNCTION:

The specified user coordinate system is created using the position, scale, and rotation specified in the arguments, and plotting in the new coordinate system is automatically activated.

CALLING SEQUENCE:

CALL UCOSYS (X,Y,SCLX,SCLY,ANGLE)

Where

X	**is the X- or RADIUS coordinate (in the current coordinate system) of the origin of the coordinate system.**
Y	**is the Y- or THETA coordinate (in the current coordinate system) of the origin of the new coordinate system.**
SCLX	**is the ratio of the unit length in the new X direction to the unit length in the current X direction.**
SCLY	**is the ratio of a unit length in the new Y direction to the unit length in the current Y direction.**
ANGLE	**is the rotation factor which is applied to the new coordinate system in current angular units.**

OPTIONS which may apply:

'WORKINGAXIS'
'REFERENCEAXIS'
UPSET option 'ZVALUE'

COMMENTS:

The execution of this subroutine causes the creation of a new origin (0.,0.) at the specified position with respect to the current origin. The new coordinate system is rotated about the new origin by the specified angle and has unit scales in the X and Y direction as specified. If GCS is in 'WORKINGAXIS' mode, the coordinate system is created on a temporary basis. The coordinate system composition has no effect on subsequent coordinate system composition. If GCS is in 'REFERENCEAXIS' mode, the coordinate system composition is based upon the current coordinate system. All coordinate system or 'SYSTEM' coordinate system. At any time the user can ensure that his secondary coordinate system is composed with respect to the system coordinate system by the use of the following GCS subroutine call:

CALL USET('SYSTEM')

In addition, the above call may be used at any time to ignore the current user coordinate system. To restore the current user coordinate system, the following GCS subroutine call is used:

CALL USET ('USERAXIS')

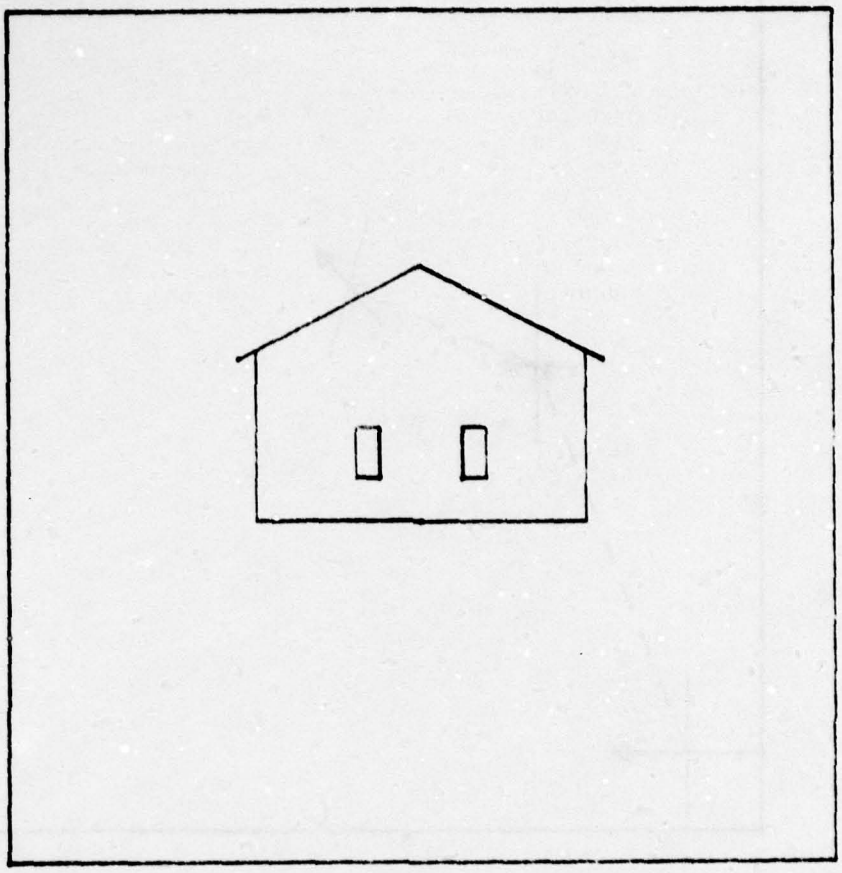
In a sense, a user always has two user coordinate systems, a permanent or reference system, and a temporary or working system. In 'WORKINGAXIS' mode, the working system is always 'ahead of' the reference system in a mathematical sense. In

'REFERENCEAXIS' mode, the working system is mathematically identical to the reference system. thus no new coordinate system is invoked in switching from reference system mode to working system mode. However, by switching in the opposite direction, the last reference system becomes the current system.

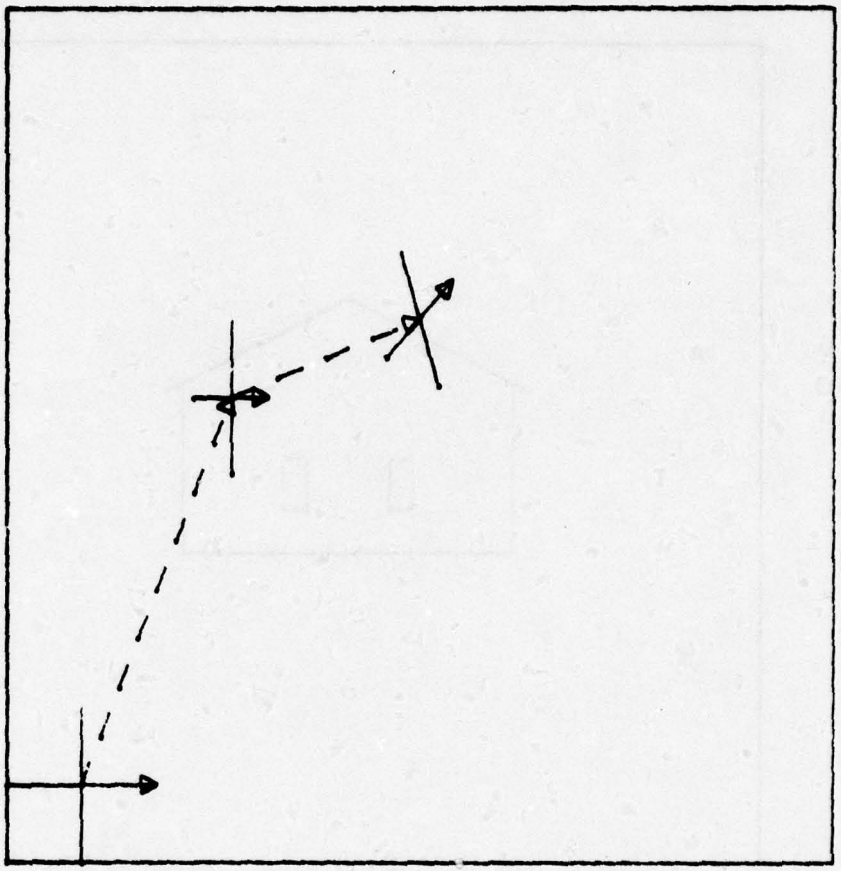
Considerable care should be exercised when utilizing the 'REFERENCEAXIS' of cumulative mode, especially in conjunction with non-uniform axis scaling. A coordinate system which is superimposed upon a nonuniform coordinate system and rotated by a non-zero angle will contain a skew factor resulting in axes which are not normal to each other. Graphic output drawn in such a system, although mathematically correct, may be difficult to understand and visualize.

In three dimension applications, the invocation of UCOSYS performs a three dimensional transformation in the current X-Y plane, specified by the UPSET parameter 'ZVALUE', with a scale factor of 1. in the Z direction, and rotation factors of 0. around the X and Y axis.

Programming Notes:



```
      CALL USTART
      CALL UOUTLN
      CALL UCOSYS (50.,50.,1.,1.,0.)
      I = 1
100   CALL UMOVE (0.,-10.)
      CALL UPEN (20.,-10.)
      CALL UPEN (20.,10.)
      CALL UMOVE (22.,9.)
      CALL UPEN (0.,20.)
      CALL UMOVE (5.,-5.)
      CALL URECT (8.,1.)
      IF (I .EQ. 2) GO TO 900
      CALL UCOSYS (50.,50.,-1.,1.,0.)
      I = 2
      GO TO 100
900   CONTINUE
      CALL UEND
      STOP
      END
```



```
CALL USTART
CALL UWINDO (-1.,10.,-1.,10.)
CALL UOUTLN
CALL AXIS
CALL UMOVE (0.,0.)
CALL USET ('REFERENCE')
CALL UCOSYS (2.,5.,.5,1.,0.)
CALL UPEN1 (0.,0.,'DARROW')
CALL AXIS
CALL UMOVE (0.,0.)
CALL UCOSYS (5.,1.,1.,1.,30.)
CALL UPEN1 (0.,0.,'DARROW')
CALL AXIS
CALL UEND
STOP
END
SUBROUTINE AXIS
CALL UMOVE (-1.,0.)
CALL UPEN1 (1.,0.,'LARROW')
CALL UMOVE (0.,-1.)
CALL UPEN (0.,1.)
RETURN
END
```

Subroutine UCOUNT

FUNCTION:

This routine returns the length of the input data as the number of characters required when displaying the data using UPRINT.

CALLING SEQUENCE:

CALL UCOUNT(DATA,COUNT)

Where

DATA	**contains input information ready for display by UPRINT.**
COUNT	**returns the number of characters which will be displayed when printing DATA on the display surface.**

OPTIONS which may apply:

Alphameric Output Option:	**'TEXT',** 'REALNUMBER', 'INTERGERNUMBER', 'XYCOORDINATES', *'XYZCOORDINATES'*
Numeric Precision Option:	UPSET option 'PRECISION'

COMMENTS:

More information about the option which apply may be found under UPRINT. In conjunction with UQUERY to determine current character size, this routine can be used to provide special character string positioning.

Programming Notes:

Subroutine UCRCLE

FUNCTION:

This routine draws a circle whose center is at the point specified. The current beam position is not affected.

CALLING SEQUENCE:

CALL UCRCLE (X,Y,RADIUS)

Where

X	**is the X- or RADIUS coordinate in current user units**
Y	**is the Y- or THETA coordinate in current user units.**
RADIUS	**is the radius of the circle in current user units.**

OPTIONS *which may apply*

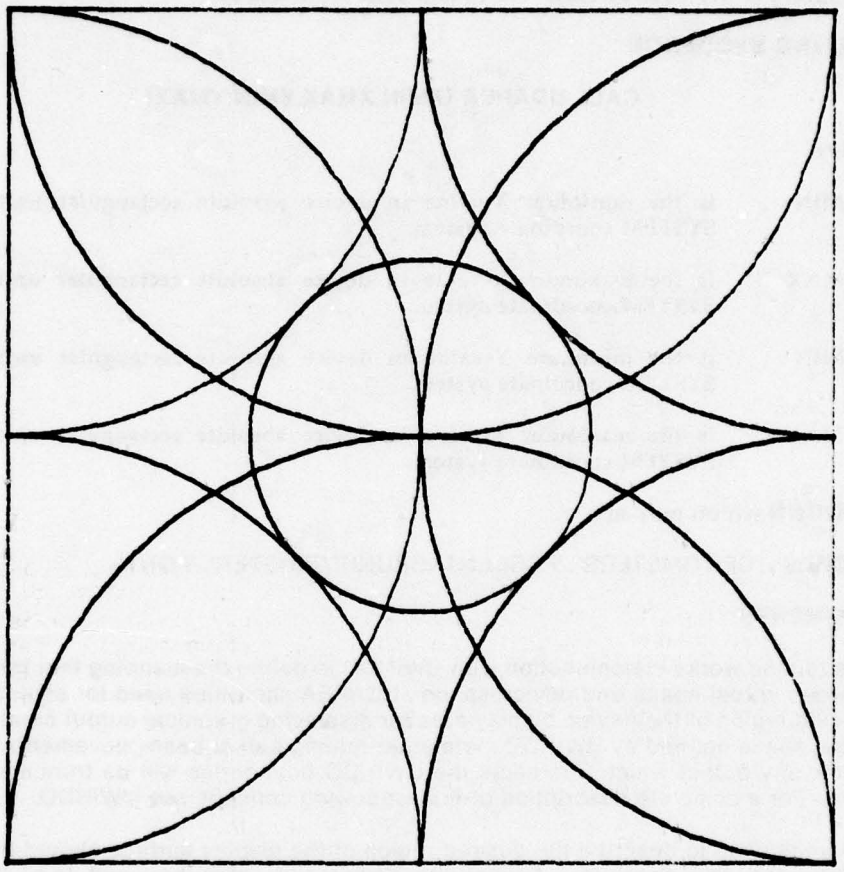
'CONTINUOUS' or 'SEGMENTED'
Pen Options (See UPEN write up)
UPSET 'ZVALUE' option

COMMENTS:

The circle created by UCRCLE will appear as either a continuous line or a collection of line segments depending on whether the user has specified 'CONTINUOUS' or 'SEGMENTED.' If 'CONTINUOUS' has been specified, the circle will appear as a single circular line segment of the line type of which is obtained from the current pen option. There will not be any type of line terminator generated. However, if 'SEGMENTED' is specified, then the circle will appear as an approximation with each line segment of the approximation taking on the characteristic defined by the current pen option. The terminator indicated will appear at the end of each line segment in the approximation.

A call to UCRCLE in a three dimensional graphics application program will result in a circle being drawn in the current X-Y plane, specified by the UPSET option 'ZVALUE'.

Programming Notes:



```
CALL USTART
CALL UOUTLN
CALL UCRCLE (0.,0.,50.)
CALL UCRCLE (50.,0.,50.)
CALL UCRCLE (100.,0.,50.)
CALL UCRCLE (100.,50.,50.)
CALL UCRCLE (100.,100.,50.)
CALL UCRCLE (50.,100.,50.)
CALL UCRCLE (0.,100.,50.)
CALL UCRCLE (0.,50.,50.)
CALL UCRCLE (50.,50.,20.7)
CALL UEND
STOP
END
```

Subroutine UDAREA

FUNCTION:

This routine defines the boundaries of the device display area into which the output appearing in the user window into virtual space will appear.

CALLING SEQUENCE:

CALL UDAREA (XMIN,XMAX,YMIN,YMAX)

Where

XMIN	**is the minimum X-value in device absolute rectangular units in the SYSTEM coordinate system.**
XMAX	**is the maximum X-value in device absolute rectangular units in the SYSTEM coordinate system.**
YMIN	**is the minimum Y-value in device absolute rectangular units in the SYSTEM coordinate system.**
YMAX	**is the maximum Y-value in device absolute rectangular units in the SYSTEM coordinate system.**

OPTIONS which may apply:

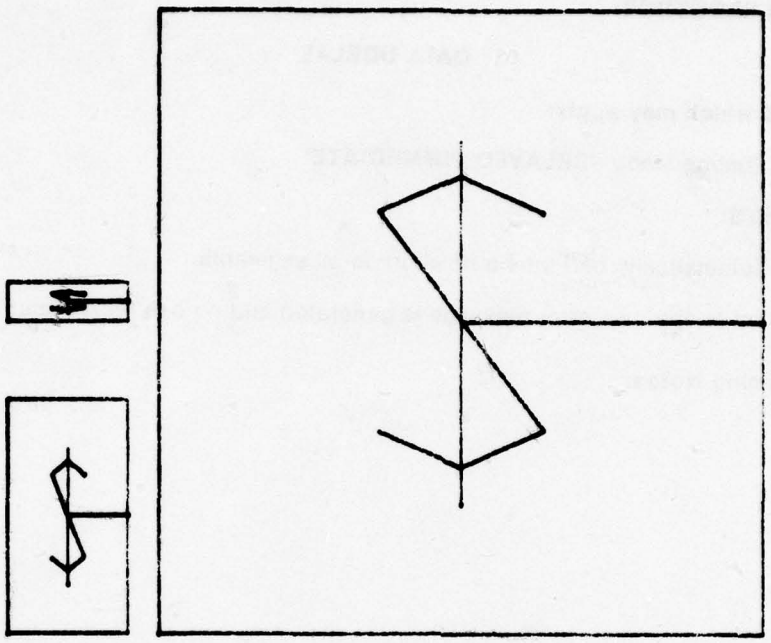
'INCHES', 'CENTIMETERS', 'PERCENTAGEUNIT', 'RASTER', 'FONT'

COMMENTS:

This routine works in conjunction with UWINDO to define the mapping that takes place between virtual space and device space. UDAREA should be used for setting aside a specific region of the device display area for displaying graphical output created in the virtual space defined by UWINDO. While performing pen or beam movements in virtual space, any output which intersects the UWINDO boundaries will be truncated at that point. For a complete description of the windowing concept, see UWINDO.

The units used to describe the desired region of the display surface should be device, rectangular, and absolute. Any of the device unts may be used (e.g., 'INCHES', 'CENTIMETERS,' etc.). If either the maximum X or maximum Y specification are less than the minimum X or minimum Y respectively, an error message will be generated (see Error Appendix for details) and the UDAREA specification in effect prior to entry to UDAREA will be retained.

Programming Notes:



```
CALL USTART
CALL UDAREA (2.,10.,0.,8.)
CALL USET ('SOFTWARE')
CALL UPSET ('HORIZONTAL SIZE',70.)
CALL UPSET ('VERTICAL SIZE',90.)
CALL USET ('L$')
CALL UOUTLN
CALL UMOVE (100.,50.)
CALL UPEN (50.,50.)
CALL UDAREA (0.,1.6,0.,3.)
CALL U3WHER (X,Y,Z)
CALL UOUTLN
CALL UMOVE (100.,50.)
CALL UPEN (50.,50.)
CALL UDAREA (0.,1.6,4.,4.5)
CALL UOUTLN
CALL UMOVE (100.,50.)
CALL UPEN (50.,50.)
CALL UEND
STOP
END
```

Subroutine UDELAL

FUNCTION:

This routine deletes all currently defined segments (or frames).

CALLING SEQUENCE:

CALL UDELAL

OPTIONS *which may apply*:

Segment Posting Mode: 'DELAYED', **'IMMEDIATE'**

COMMENTS:

UERASE automatically performs a deletion on all segments.

If a segment is open, an error message is generated and no deletions occur.

Programming Notes:

FUNCTION:

This routine deletes an existing retained segment and removes its image from display surface.

CALLING SEQUENCE:

CALL UDELET(SEGID)

Where

SEGID **is the segment identifier of the segment to be deleted (see UOPEN for a description of SEGID).**

OPTIONS which may apply:

Segment/Frame Identifier Mode: **'FNAME'**, 'FNUMBER'
Segment Posting Mode: **'DELAYED'**, 'IMMEDIATE'

COMMENTS:

The image of the deleted segment will be immediately removed, if visible, from the display surface if the segment posting made is 'IMMEDIATE'. Otherwise, the image will not be removed until the next required screen erase. This will occur when UPOST or UERASE is executed or if UDELET or UMODFY (to change a segment attribute to 'INVISIBLE') is called with the segment posting mode as 'IMMEDIATE'.

Programming Notes:

Subroutine UDIMEN

FUNCTION:

This routine adjusts the physical boundaries of the display surface. It may also be used to provide a display surface of a specific size.

CALLING SEQUENCE:

CALL UDIMEN (XSIZE,YSIZE)

Where

XSIZE	is the physical size in the X-direction in current device units.
YSIZE	is the physical size in the Y-direction in current device units.

OPTIONS which may apply:

Device Unit Type:	**'INCHES',** 'CENTIMETERS', 'FONTUNITS', 'PERCENTUNITS', 'RASTERUNITS', 'SPECIFICATION UNITS', 'NDCUNITS'
Specification Unit Size:	'XSPECIFICATION UNIT', 'YSPECIFICATION UNIT', 'SPECIFICATION UNITS'

COMMENTS:

This routine is an alternative method to UFORMT for specifying the size of the display surface. The actions of this routine will replace any settings performed by previous calls to UFORMT or UDIMEN.

For those devices which have a variable size display surface (e.g., drum plotters or dot-matrix printers), this routine provides that the appropriate size display surface is available. For devices with a fixed size display surface, this routine may be used to reduce the available size of the display surface. A display surface which exceeds the maximum available display surface cannot be specified. Attempts to do so will generate an error, and the display surface size will not be changed.

The currently set aspect ratio (provided by UASPCT) will apply to the newly formatted display surface. The UDAREA will be set to the entire resulting virtual display surface.

CAUTION:

This routine may change the size of PERCENTUNITS, SPECIFICATIONUNITS, or NDCUNITS.

To reduce the side effects of redefining the display surface, it is recommended that UDIMEN be called immediately following a call to UERASE, USTART, or URESET.

Programming Notes:

Subroutine UDOIT

Interactive

FUNCTION:

This subroutine enables the user to do various page layouts and control a cassette recorder such as the Tektronix 4923.

CALLING SEQUENCE:

CALL UDOIT (ACTION)

Where

ACTION **is a Hollerith string that indicates some function to perform**

ACTIONS

Must have called USET ('DEVICE')

'ALARM' — *Sounds the audible alarm*
'ALPHAMODE' — *Sets alphamode*
'AUDIBLE' — *Sounds the audible alarm*
'BACKSPACE' — *Move back one character space*
'BEEP' — *Sounds the audible alarm*
'BELL' — *Sounds the audible alarm*
'BLIP' — *Sounds the audible alarm*
'BOTTOM' — *Move to bottom margin*
'BSPb' — *Same as backspace*
'CARR' — *Do a carriage-return*
'CR' — *Do a carriage-return*
'CRLF' — *Do a carriage-return then a line-feed*
'DOWN' — *Do a line-feed*
'ERAS' — *Erase the screen*
'HDOWN' — *Home to the bottom*
'HOME' — *Home to the top*
'LEFT' — *Move to the left margin*
'LFbb' — *Do a line-feed*
'LINEFEED' — *Do a line-feed*
'NEWLINE' — *Do a carriage-return then a line-feed*
'NEWPAGE' — *Erase the screen*
'PAGE' — *Erase the screen*
'RIGHT' — *Move to right margin*
'SPbb' — *Move over one hardware character space*
'SPACE' — *Move over one hardware character space*
'TABb' — *Tab to next horizontal tab setting*
'TABH' — *Tab to next horizontal tab setting*
'TOPb' — *Move to top margin*
'UPbb' — *Move to top margin*
'TABV' — *Tab to next vertical tab setting*
'TABX'
'TBHX' — *Tab to designated setting (example: 'TBH8')*
'TBVX' *(see UPSET)*
WHERE X = 0-9
'UP' — *Move up one vertical hardware character position*

CASSETTE CONTROL: Not necessary to call USET ('DEVICE')

'RSTART' — *Start read*
'RSTOP' — *Stop read*
'WSTART' — *Start write*
'WSTOP' — *Stop write*

Programming Notes:

Subroutine UDRAW

FUNCTION:

This routine draws a solid line vector from the current position to that specified by the input arguments. All lines are drawn on the XY-plane specified by the current set Z-value.

CALLING SEQUENCE:

CALL UDRAW(X,Y)

Where

X,Y **are the coordinates of the end point of the line in curreny units**

OPTIONS:

Pen Coordinate Options:

Type: **'RECTANGULAR'**, 'POLAR'

Log: 'XLOG', 'YLOG', *'ZLOG'*, 'XYLOG', *'XZLOG'*, *'YZLOG'*, *'XYZLOG'*, 'LOGARITHMIC', **'NOLOG'**

MODE: **'ABSOLUTE'**, 'RELATIVE'

SPACE: **'VIRTUAL'**, 'DEVICE'

DEVICE SPACE UNITS: **'INCHES'**, 'CENTIMETERS', 'RASTERUNITS', 'FONTUNITS', 'PERCENTUNITS', *'SPECIFICATIONUITS'*

COORDINATE SYSTEM: **'SYSTEM'**, 'USER', 'REFERENCE', **'WORKING'**

COMMENTS:

This routine is equivalent to UPEN with line option 'LNULL'. The current setting of the line option has no effect on this routine. *The current Z-value will be used to draw the line in three-space.*

Programming Notes:

Subroutine UDRIN *Interactive*

FUNCTION:

This routine provides a flexible drafting operation which obtains the end of the next beam/pen movement and the type of movement via graphics input, and then performs the operation requested. Only one operation will be performed by a single call to UDRIN. If the operation requested is not a valid UDRIN operation, then the input values are simply passed back to the calling program.

CALLING SEQUENCE:

CALL UDRIN(X,Y,CHAR)

Where

X	**is the X- or RADIUS coordinate of the graphics input cursors in user units entered by the user during the graphics input operation.**
Y	**is the Y- or THETA coordinate of the graphics input cursors in user units entered by the user during the graphics input operation.**
CHAR	**is the character entered which identifies the type of operation performed. It will be returned in Hollerith (left justified, blank-fill) format.**

OPTIONS which may apply:

Any option which effects pen/beam.
Input Device Selection Options: 'LIGHTPEN', 'CURSORS', etc. (see UGRIN)

COMMENTS:

UDRIN supports the most basic line options for interactive drafting. When UDRIN is called, the current graphics input device (see UGRIN) is enabled. The user positions the cursor or tracking cross at the position he desires and strikes the particular keyboard key which indicates the option he desires. The following table indicates the UDRIN options available:

A – draws an arrow from current beam location to current cursor postion.

C – outputs the current system character at the current cursor location.

D – draws a double arrow from current beam location to cursor position.

E – erases the screen.

H – sets Graphic Status Area to hardware character type.

K – sets the system character to that character entered at next appearance of cursors.

L – draws a line from current beam position to cursor position.

M – moves the beam invisibly to the cursor position.

P – plots a point at cursor position.

S – sets Graphics Status Area to software character type.

T — draws a ticked line from current beam location to cursor position.

X — prints coordinates at cursor position.

Z — draws a dashed line according to the current dash specification from the current beam position to the cursor position.

All unused characters may be assigned special meanings by the user to support options not available in UDRIN. The characters indicated above are all upper case, thus all lower case characters are available to the user.

Programming Notes:

Subroutine UEND

FUNCTION:

The routine should be called to terminate graphics operations. Its effect is to flush any output buffers which may be used internally in GCS, take the terminal out of graphics mode, display any error messages whose output has been deferred, position the beam or pen at the home position, and return to the user's program.

CALLING SEQUENCE:

CALL UEND

OPTIONS which may apply:

No options apply.

COMMENTS:

Designed as the means of terminating graphics operations in a program, UEND is the antithesis of USTART. Depending on the type of graphics device being used, different operations will take place which flush any buffers which are partially filled (see UFLUSH), disenable the graphics operations at the terminal, position the pen or beam in the home position, and dump any GCS error messages (see Error Appendix) which have been deferred. UEND then returns control to the user's program. The user, if he wishes to perform additional graphics operations, should call USTART in order to reenter graphics mode (see USTART).

Programming Notes:

Subroutine UERASE

FUNCTION:

This routine causes the graphics display device plotting area to be cleared of any previous output. The pen or beam position is not affected.

CALLING SEQUENCE:

CALL UERASE

OPTIONS which may apply:

Security Banners: 'UNSECURED', 'UNCLASSIFIED', 'CONFIDENTIAL', 'SECRET', 'TOP SECRET'

COMMENTS:

The currently specified display surface(s) will be cleared. The pen or beam position at the time of the erasure will not be changed and the graphics status area will not be affected.

If a security banner other than 'UNSECURED' is selected, the banner will be displayed at the top and bottom of each frame in the largest hardware character size.

Any segments which are defined at the time of an invocation of UERASE will be deleted. If this is not desired, UPOST should be called instead.

Programming Notes:

Subroutine UERROR

FUNCTION:

This routine returns the identifier of the last GCS error which occurred and a count of the total number of GCS errors which have occurred.

CALLING SEQUENCE:

CALL UERROR(ERLAST,TOTAL)

Where

ERLAST is the GCS error number of the last GCS error which occurred. Zero is returned if no errors have occurred.

TOTAL is the number of GCS errors which have occurred since the last GSA initialization (USTART or URESET) or the last call to UERROR whichever occurred most recently.

OPTIONS which may apply:

No options apply

COMMENTS:

The error total is reset to zero whenever UERROR is called.

There are three modes of error processing in GCS: 'ERRORS', 'SUPPRESSED ERRORS', and 'DEFERRED ERRORS'. The default mode is 'ERRORS' which indicates that error messages are to be generated as they occur. 'SUPPRESSED ERRORS' will not be generated at all. The first 10 'DEFERRED ERRORS' will be generated when UEND is called. The GCS Error Mode does not effect the execution of UERROR. Regardless of error mode, errors will be counted and the last error to occur will be saved.

Programming Notes:

Subroutine UFLUSH

FUNCTION:

This routine terminates the frame.

CALLING SEQUENCE:

CALL UFLUSH

OPTIONS which may apply:

No options apply.

COMMENTS:

This routine must be called to terminate each frame. If not then plot data may be lost.

Programming Notes:

Subroutine UFONT

FUNCTION:

This routine changes the text font to that requested by the calling program.

CALLING SEQUENCE:

CALL UFONT (NAMFNT)

Where

NAMENT **is the Hollerith string containing the name of the font to be used by subsequent text operations.**

OPTIONS which may apply:

No options apply

COMMENTS:

The font name provided will be converted to standard GCS format (first four characters) and compared to the table of available fonts. The font will not be changed if an invalid font name if provided. The fonts are grouped by style and are listed below. Each font is shown in Table UFONT-1 with its corresponding ASCII reference symbol (the ASCII 'RUBOUT' character is represented by 'c'). With the exception of the Simplex GCS font, all fonts are derived from the characters digitized by Allan Hershey of the U.S. Naval Weapons Laboratory, Dahlgren, VA.

More information on these fonts may be found in National Bureau of Standards Publication 424 entitled "A Contribution to Computer Typesetting Techniques: Table of Coordinates for Hershey's Repetory of Occidental Type Fonts and Graphic Symbols."

Fonts may also be changed from within GCS text strings by including the font change symbol (default is the 'at' symbol) followed by the first four characters of the font name.

Fonts will be changed for both hardware and software characters, if possible. However, should a hardware character generator not support a particular font, a substitute (usually the basic font) will be provided.

AVAILABLE FONTS:

Simplex:

This style is a very simple style and uses the fewest strokes. The fonts available in this style are:

SGCS	Simplex GCS (default)
SGRE	Simplex Greek
SROM	Simplex Roman
SSCR	Simplex Script

Complex:

As the name implies, this style provides more complicated characters. The fonts available in the complex style are:

CCYR	Complex Cyrillic
CGRE	Complex Greek
CITA	Complex Italic
CSCR	Complex Script

Duplex:

This style is a less ornated representation of complicated characters. Only one font is available:

DROM	Duplex Roman

Triplex:

This style is similar to the complex style. By using three lines for each stroke, the characters appear more bold. Two fonts are available.

TITA	Triplex Italic
TROM	Triplex Roman

Gothic:

This is the most ornate style available. This ornateness comes at a severe penalty in generation time due to the large number of strokes in each character. The fonts available are:

GENG	Gothic English
GGER	Gothic German
GITA	Gothic Italian

Special:

The following fonts support applications areas where special symbols are frequently used. The SYMBOL font consists entirely of symbols. The other fonts incorporate all or portions of the roman alphabet and the ten numeric digits. The fonts available are:

ASTR — Astrology—This font incorporates the signs of the zodiac and of the planets. Included are a full triplex Italic alphabet and numerals.

LMAT — Lower Case Mathematics—This font includes a complete set of mathematical symbols. A lower case Simplex Roman alphabet is provided and it is referenced using upper case characters.

UMAT — Upper Case Mathematics—This font includes a complete set of mathematical symbols. Only an upper case Simplex Roman alphabet is provided and it is referenced using upper case characters.

METE — Meteorological—This font incorporates the standard set of meteorological symbols. Included are a full Simplex Roman alphabet and numerals.

MUSI — Music—This font provides symbols commonly used when writing music including symbols for cleffs, rests, and notes. A full Complex Italic alphabet and numberals are provided.

SYMB — Symbols—Only symbols will be found in this font. No alphabet or numerals are provided.

Programming Notes:

Subroutine UFORMT

FUNCTION:

This routine configures the display surface for generation of the requested standard format.

CALLING SEQUENCE:

CALL UFORMT (FORMAT)

Where

FORMAT **is the name of the desired format. A list of valid names is provided below.**

OPTIONS which may apply:

No options apply

COMMENTS:

The following table lists the supported standard formats grouped by media:

Publication:

'PAGE'	8-1/2 x 11 inch vertical page.
'HPAGE'	11 x 8-1/2 inch horizontal page.
'LEGAL'	8-1/2 x 14 inch vertical legal page.
'HLEGAL'	14 x 8-1/2 inch horizontal legal page.

16 mm Film:

'16CINE'	16 mm motion picture.
'16COMIC'	16 mm microfilm.

35 mm Film:

'35CINE'	35 mm motion picture.
'35COMIC'	35 mm slides.
'35APERTURE CARD'	35 mm non-sprocketed aperture cards.

70 mm Film:

'70CINE'	70 mm wide screen motion picture.
'70COMIC'	70 mm square comic orientation.

105 mm Film:

'105'	105 mm square full frame comic.

Microfiche:

'24XNARROW'	Vertically oriented narrow page 24X reduction.
'24XWIDE'	Horizontally oriented wide page 24X reduction.
'42XNARROW'	Vertically oriented narrow page 42X reduction.
'42XWIDE'	Horizontally oriented wide page 42X reduction.
'48XNARROW'	Vertically oriented narrow page 48X reduction.
'48XWIDE'	Horizontally oriented wide page 48X reduction.

Engineering Drawings:

'AVSIZE'	Vertical A-size	8-1/2 x 11 inches.
'AHSIZE'	Horizontal A-size	11 x 8-1/2 inches.
'BSIZE'	B-size	11 x 17 inches.
'CSIZE'	C-size	17 x 22 inches.
'DSIZE'	D-size	22 x 34 inches.
'ESIZE'	E-size	34 x 44 inches.

Miscellaneous:

'SHEETFILM'	4 x 5 inch sheet film.
'TELEVISION'	Television safe area.
'STRIPCHART'	Frame-abutted.
'SQUARE'	Square (10 inches).
'MAXIMUM'	Entire physical surface.

If the physical display surface cannot directly produce the requested format, a display surface will be created which simulates the requested format using the maximum available portion of the physical display surface which accepts the aspect ratio of the format requested. When the format is simulated, the formatted display surface will be centered on the physical display surface if practicable.

This routine is an alternative method to UDIMEN for specifying the size of the display surface. The actions of this routine will replace any settings performed by previous calls to UFORMT or UDIMEN.

The currently set aspect ratio (provided by UASPCT) will apply to the newly formatted display surface. The UDAREA will be set to occupy entirely the resulting virtual display surface.

CAUTION:

This routine may change the size of PERCENTUNITS, SPECIFICATIONUNITS, or NDCUNITS. If a simulated formatted display surface is provided, the size of INCHES and CENTIMETERS will be as if the format were actually being produced.

To reduce the side effects of redefining the display surface, it is recommended that UFORMT be called immediately following a call to UERASE, USTART, or URESET.

Programming Notes:

Subroutine UFRAME

FUNCTION:

This routine indicates to GCS that all graphic commands generated subsequent to the invocation of this subroutine and prior to the invocation of the matching subroutine UFREND are a named entity which replaces any previous occurrence of the same named entity.

CALLING SEQUENCE:

CALL UFRAME (NAME)

Where

NAME **is the eight character alphanumeric constant or variable.**

OPTIONS which may apply:

'INVISIBLE'
'VISIBLE'

COMMENTS:

The execution of this subroutine causes all subsequent graphic commands to be associated with the argument NAME. The association continues until the matching subroutine UFREND is executed. For refresh graphic terminals, the framed information replaces the previous occurrence of the frame on the display screen. If GCS is in INVISIBLE build mode (default) the previous occurrence is deleted only when the new occurrence is completely defined. That is, when the matching subroutine UFREND is executed. If GCS is in visible build mode, the previous frame is deleted when UFRAME is executed and the new frame is displayed as each graphic command is generated. Multiple frames can be defined, but only one frame may be open or active at a time. This means that frame definitions may not overlap or be nested.

Programming Notes:

Subroutine UFREND

FUNCTION:

This routine indicates to GCS that the named set of graphic commands currently being defined (the active frame) is terminated.

CALLING SEQUENCE:

CALL UFREND (NAME)

Where

NAME *is an eight character alphanumeric constant or variable.*

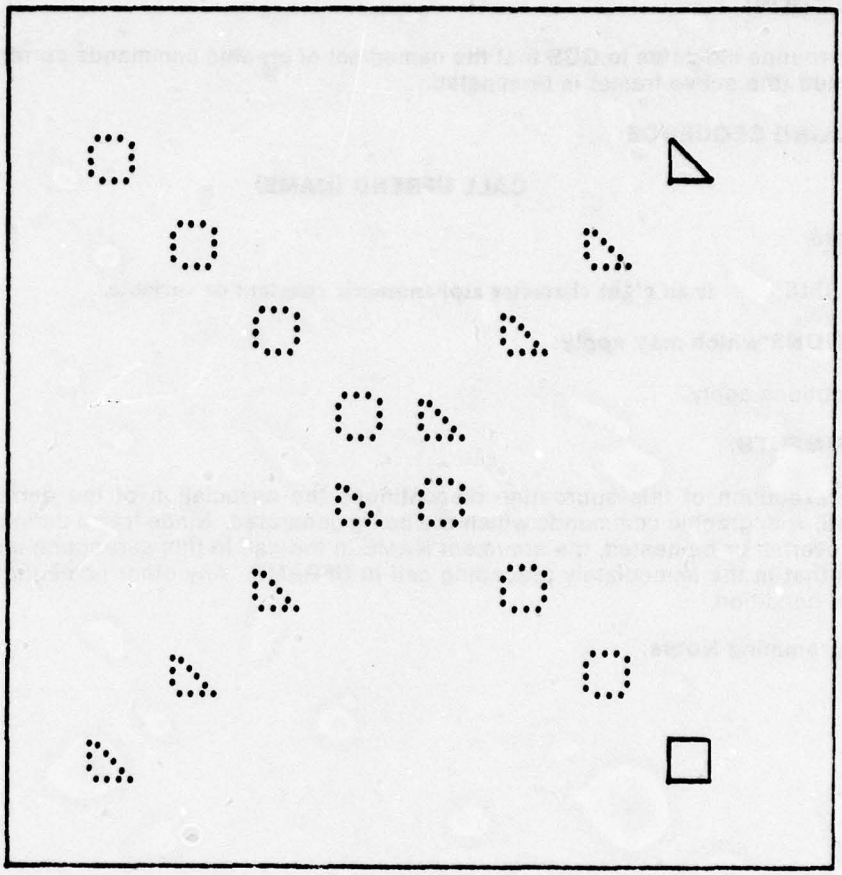
OPTIONS which may apply:

No options apply.

COMMENTS:

The execution of this subroutine discontinues the association of the current frame NAME with graphic commands which are being generated. Since frame definitions may not overlap or be nested, the argument NAME in the call to this subroutine must agree with that in the immediately preceding call to UFRAME. Any other name indicates an error condition.

Programming Notes:



```
      X = 10.0
      Y = 80.0
      CALL USTART
      CALL UPSET ('LIBRARY',1.)
      CALL UOUTLN
      DO 1 I = 1, 8
      CALL UFRAME ('TRIANGLE')
      CALL UMOVE (X,X)
      CALL UPEN (X,(X+5.0))
      CALL UPEN ((X+5.0),X)
      CALL UPEN (X,X)
      CALL UFREND ('TRIANGLE')
      CALL UFRAME ('SQUARE')
      CALL UMOVE (X,Y)
      X = X + 10.0
      Y = Y - 10.0
      CALL URECT ((X-5.0),(Y+15.0))
      CALL UFREND ('SQUARE')
    1 CONTINUE
      CALL UEND
      STOP
      END
```

Subroutine UGRIN *Interactive*

FUNCTION:

This routine activates a graphics input device at the terminal and returns to the user the coordinates specified and an alphanumeric character which may be specified by the user or derived from the actions of the input device.

CALLING SEQUENCE:

CALL UGRIN (X,Y,ICHAR)

Where

X	**is the X- or RADIUS coordinate of the point specified by the input device in user units.**
Y	**is the Y- or THETA coordinate of the point specified by the input device in user units.**
ICHAR	**is the alphanumeric entered by the user or provided by the device in Hollerith (left-justified, blank-filled) format.**

OPTIONS which may apply:

'CURSOR', 'JOYSTICK', 'LIGHTPEN', 'TABLET', 'KEYBOARD', 'FUNCTIONKEY', 'MOUSE', 'BALL'.

Graphic Input Device Options

COMMENTS:

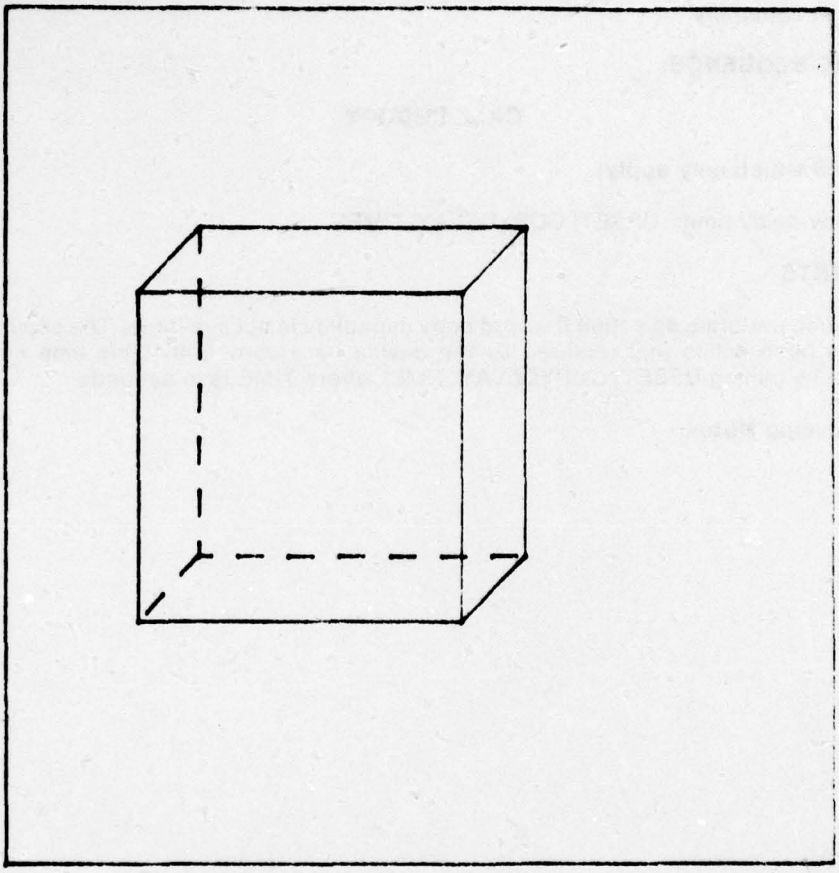
UGRIN is the user's general purpose graphics input routine. It is designed to support any of the input devices which may be available on any particular terminal. For terminals with more than one graphics input device, each type of device can be supported separately. Should the user request a device which is not available, the principal graphic input device will be enabled. Terminals with no graphics input device simulate graphics input by reading the coordinates and the character from the keyboard. In this case, the coordinates must be entered in *absolute, rectangular, raster units.*

The graphics input device which the user wishes enabled should be requested prior to calling UGRIN by setting the USET option for the device. A list of devices is indicated under 'Options which may apply' above. USTART will initialize the device selection to the primary graphics input device for the terminal.

When UGRIN is called, an indication that the graphics input device has been enabled will appear. This indication may be a tracking cross or cursors which appear on the screen, a ready light which turns on, or some other recognizable object. At this point, the user should position the cursor or tracking cross to the desired location and signal the end of the operation in the manner required by the device. This may consist of striking a keyboard character, removing the input device (pencil) from the input surface, applying pressure to the input device pencil, or some other means. For exact details for a particular device, see the System Manual. If graphics input is being simulated, enter the coordinates and the character from the keyboard upon request from the program. When the end of graphics input has been signaled, the graphics input enabled indicator will disappear.

The coordinates returned to the user's program in arguments X and Y will be in the units which are currently set in the Graphics Status Area. These values are suitable for direct entry as UPEN arguments. The character which has been entered will be in standard FORTRAN hollerith format (left-justified, blank-filled) and is suitable for use in FORTRAN LOGICAL IF statements for interrogation or as input to USET or UAOUT.

Programming Notes:



```
      CHARACTER CHAR*1
      CALL USTART
      CALL UOUTLN
    1 CALL UGRIN (X,Y,CHAR)
      IF (CHAR .EQ. 'S') CALL UPEN1 (X,Y,'LINE')
      IF (CHAR .EQ. 'I') CALL UMOVE (X,Y)
      IF (CHAR .EQ. 'D') CALL UPEN1 (X,Y,'DASH')
      IF (CHAR .EQ. 'E') GO TO 2
      GO TO 1
    2 CALL UEND
      STOP
      END
```

Subroutine UHDCPY

FUNCTION:

This routine causes a hard copy to be generated if the device has an on-line immediate hard copy capability.

CALLING SEQUENCE:

CALL UHDCPY

OPTIONS which may apply:

Hard copy delay time: UPSET('COPYDELAY',TIME)

COMMENTS

This routine performs no action if a hard copy capability is not available. The copy delay time has been set to that required by the device hard copy unit. This time may be adjusted by calling UPSET('COPYDELAY',TIME) where TIME is in seconds.

Programming Notes:

Subroutine UHISTO

FUNCTION:

To draw a histogram of the specified number of bars from the user data array. The histogram will be drawn in the current user display area and will be scaled and labeled as the user has specified. The display area will be reduced by the size of the labels, if any.

CALLING SEQUENCE:

CALL UHISTO (ARRAY,POINTS,BARS)

Where

ARRAY	**is a real array of data.**
POINTS	**is a real constant or variable which is the number of elements in ARRAY.**
BARS	**is a real constant or variable which is the number of bars in the histogram.**

OPTIONS which may apply:

AXIS EXISTENCE: 'NOAXES', 'XAXIS', YAXIS', **'XYAXIS'**
AXIS FORMAT: 'PLAINAXIS', **'TICAXIS'**
X-AXIS LABELING: **'XNUMERIC',** 'XALPHABETIC', 'XBOTH', 'NOXLABEL'
Y-AXIS LABELING: **'YNUMERIC',** 'XALPHABETIC', 'YBOTH', 'NOXLABEL'
SCALING TYPE: **'AUTOSCALE',** 'FULLSCALE', 'OWNSCALE'
DISPLAY AREA: **'NEWSCALE',** 'OLDSCALE'

COMMENTS:

UHISTO is a general purpose histogram drawing and labeling routine. The options indicated above are independent of each other so that several different options may appropriately be chosen. When the statistics for the histogram are developed, each interval is assumed to be closed on the lower end and open on the upper end. The last interval is closed on the upper end. That is, for example, the value of 2.0 will fall in the range of 2.0 to 3.0 rather than 1.0 to 2.0.

SUMMARY OF OPTIONS:

Axis Existence:

NOAXIS	– Neither the X-axis of the Y-axis nor the histogram will be displayed.
XAXIS	– Only the X-axis will be displayed.
YAXIS	***– Only the Y-axis will be displayed.***
XYAXIS	– Both the X- and the Y-axis will be displayed.

Axis Format:

PLAINAXIS	– The X-axis will appear as a solid line.

TICAXIS — The X-axis will appear as a ticked line. The tic interval will be one unit. Note that only the X-axis format is involved. The Y-axis will always remain plain. The GRIDAXIS option is not supported. If GRIDAXIS is selected, the option will revert to PLAINAXIS.

X-Axis Labeling:

NOXLABEL — No labeling should appear along the X-axis

XNUMERIC — Numeric labels should be created along the X-axis. The tic interval will be determined by UHISTO so that the number of labels which appear on the X-axis will be maximized.

XALPHANUMERIC — Alphanumeric labels will appear at the lower edge of the display area in which the histogram is drawn. The contents of the label can be specified by calling subroutine UPSET with arguments 'XLABEL' and a GCS character string as a value (for a description of GCS character strings, see UPRINT).

XBOTHLABELS — Both the alphanumeric and numeric labels will appear along the X-axis.

Y-Axis Labeling:

NOYLABEL — No labeling should appear along the Y-axis.

YNUMERIC — Numeric labels should be created along the Y-axis. One label will be created for each bar of the histogram. The numeric labels consist of the range of values represented by a histogram bar, and the count of the number of items in the range (the length of the bar).

YALPHAMERIC — Alphameric labels should be created along the Y-axis, at the left edge of the display area in which the histogram is drawn. The contents of the label can be specified by calling subroutine UPSET with arguments 'YLABEL' and a GCS character string as a value (for a description of GCS character strings, see UPRINT).

Scaling Type:

AUTOSCALE — For the X-axis, the range of values for the histogram will be from zero to a number larger than the size of the largest bar. For the Y-axis, the range of values to be histogrammed is examined so that 'nice' numbers will appear at the bar intervals. The range may be expanded to include these values. The UWINDO is set to the expanded range as identified.

FULLSCALE — For the X-axis, the range of values for the histogram will be from zero to a number larger than the size of the largest bar. For the Y-axis, the limits of the histogram will be the minimum and maximum values of the data. The UWINDO setting is modified to reflect the limits of the X-axis.

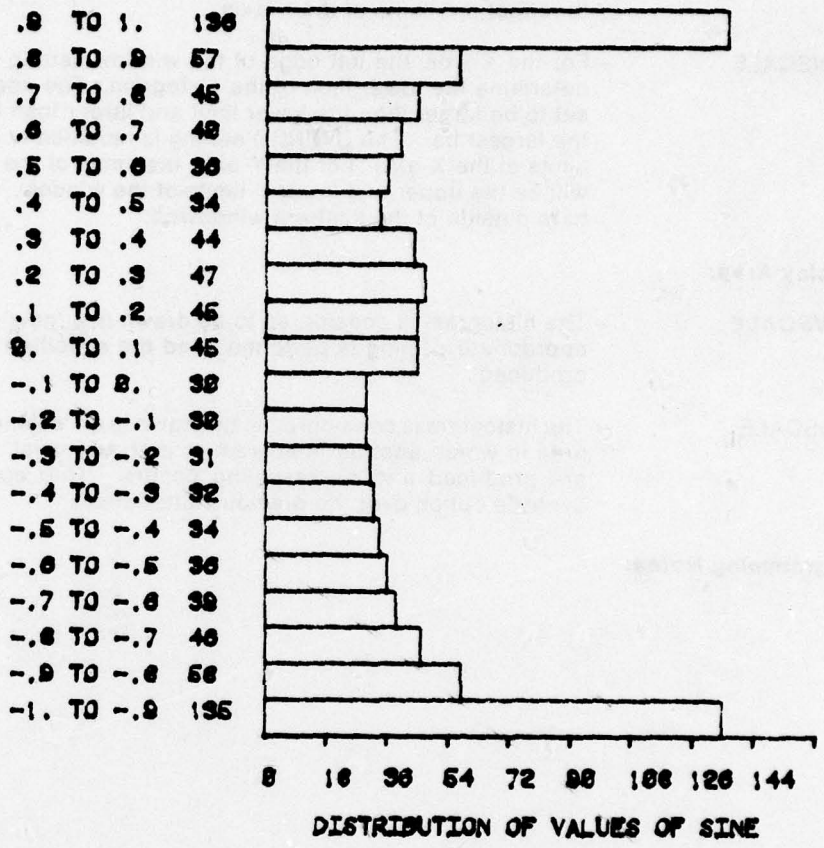
OWNSCALE — For the X-axis, the left edge of the window setting is used to determine the lower limit of the histogram. The upper limit is set to be larger than the lower limit and larger than the size of the largest bar. The UWINDO setting is modified to reflect the limits of the X-axis. For the Y-axis, the limits of the histogram will be the upper and lower Y limits of the window. Histogram bars outside of the limit are windowed.

Display Area:

NEWSCALE — The histogram is considered to be drawn in a 'new' area. The appropriate scaling is performed and the specified labels are produced.

OLDSCALE — The histogram is considered to be drawn in an 'existing' display area in which another histogram or plot may exist. No labels are produced and no rescaling occurs. This option is an override option over the previous alternatives.

Programming Notes:




```
      DIMENSION DATA(1000)
      DATA XN/1000./
      DO 2 I = 1, 1000
    2 DATA(I) = SIN(FLOAT(I)/150.)
      CALL USTART
      CALL UASPCT (1.)
      CALL USET ('LARGE')
      CALL USET ('PERCENTUNITS')
      CALL UDAREA (0.,100.,0.,100.)
      CALL USET ('FULLSCALE')
      CALL USET ('XBOTHLABELS')
      CALL UPSET ('XLABEL','DISTRIBUTION OF VALUES OF SINE\')
      CALL UHISTO (DATA,XN,20.)
      CALL UEND
      STOP
      END
```

Subroutine UHOME

FUNCTION:

This routine moves the beam to the home position.

CALLING SEQUENCE:

CALL UHOME

OPTIONS which may apply:

No options apply.

COMMENTS:

The home position is maintained in the Graphics Status Area. This position is normally the lower left corner of the display surface. The beam is moved to the home position with no visible output on the display surface.

Programming Notes:

FUNCTION:

This routine applies a general two-dimensional image transformation to the indicated segment/frame.

CALLING SEQUENCE:

CALL UIMAGE(X,Y,SX,SY,R,SEGID)

Where

X,Y	**is the new position of the segment in current 2D device units**
SX,SY	**is the scale factor to be applied along each axis of the display surface**
R	**is the rotation in current angular units to be applied around the Z-axis of the display surface**
SEGID	**is the identifier of a 'retained' segment/frame which was UOPENed for at least general 2D image transformations**

OPTIONS which may apply:

Device Units:	**'INCHES',** 'CENTIMETERS,' 'RASTER UNITS,' 'FONT UNITS,' 'SPECIFICATION UNITS,' 'PERCENT UNITS'
Specification Units (UPSET):	'SPECIFICATION UNITS,' 'XSPECIFICATION UNITS', 'YSPECIFICATION UNITS', 'ZSPECIFICATION UNITS'
Angular Units:	**'DEGREES',** 'RADIANS', 'PIRADIANS', 'GRADS', 'MILS'
Segment Identifier Mode:	**'FNAME',** 'FNUMBER', 'SNAME', 'SNUMBER'

COMMENTS:

The image transformation is applied in the order: rotation scaling, and translation. If the resulting image should exceed the display surface address space, the result is undetermined.

Image transformations are only applied if supported on the current display surface. Requests for image transformations will be ignored if the display device does not support this facility. 2D transformations will be applied if 3D transformations are supported.

Programming Notes:

Subroutine UINPUT *Interactive*

FUNCTION:

This routine enables the user to read character data entered from the alphanumeric key board of the terminal. The data will be formatted and returned to the user in one of five options. The options are: 'TEXT', 'REALNUMBER', 'INTEGER', OR 'XYCOORDINATE', 'XYZ COORDINATE'.

CALLING SEQUENCE:

CALL UINPUT (DATA,COUNT,FLAG,OPTION)

Where

DATA — is a single variable or array to contain either real number or a GCS text string. The size of DATA is variable: it is a single variable if 'REAL' or 'INTEGER' is specified; it is a two word array if 'XYCOORDINATES' is specified; and it is of variable length if 'TEXT' is specified.

COUNT — is a single variable or constant. If the user has specified 'TEXT' input, then COUNT is the maximum number of characters returned in DATA. The size of DATA must be large enough to contain COUNT characters. The characters are returned in Hollerith format; left justified and blank filled. If the terminal operator inputs fewer characters than requested in COUNT, then DATA will be blank filled to the required size. If the user has specified 'REAL', 'INTEGER', 'XYCOORDINATE' or 'XYZCOORDINATE' then the value of COUNT upon entry to UINPUT is the number of variables to be input. The size of DATA must be large enough to contain COUNT variables.

FLAG — is a single variable. The value of FLAG, upon exit from UINPUT, will be negative if the terminal operator has made an error in entering the numeric data. In this case, the value of DATA is undefined. If the required input was correct, then the value of FLAG is the number of elements returned in Data. The value of FLAG is always less than or equal to COUNT.

OPTION — is a character constant or variable which specifies the format into which the input string will be mapped. The format options are: 'TEXT', 'REAL', 'INTEGER', 'XYCOORDINATE' or 'XYZCOORDINATE'.

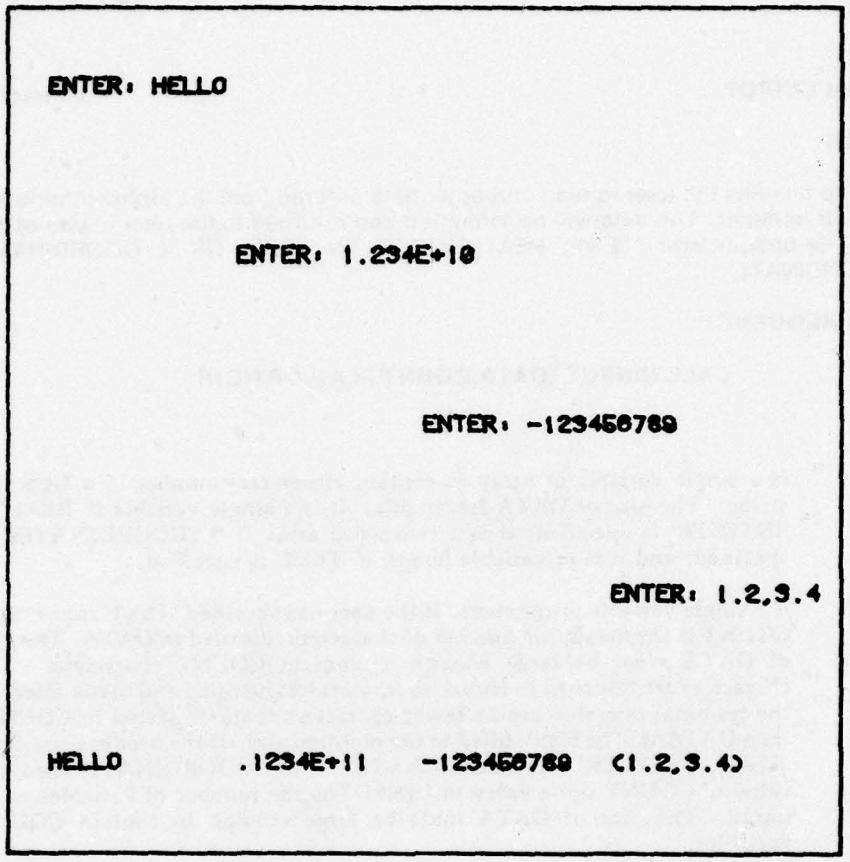
OPTIONS which may apply:

Device Switching Options: 'MESSAGEDEVICE', 'PLOTDEVICE'

COMMENTS:

The input begins at the current beam or alphanumeric cursor position. Upon termination of UINPUT, the beam position in effect prior to its invocation is restored. For a full explanation of the GCS alphanumeric input facility, see SUBROUTINE UREAD.

Programming Notes:




```
      CHARACTER OPTION*4(4)
      DIMENSION COUNT(4),DATA(8),INDEX(4)
      DATA COUNT,INDEX,X,Y/5.,1.,1.,1.,1,3,4,5,5.,99./
      DATA OPTION/'TEXT','REAL','INTE','XYCO'/
      CALL USTART
      CALL UPSET ('TERMINATOR',',')
      CALL UOUTLN
      DO 1 I = 1, 4
      IDX = INDEX(I)
      CALL UMOVE (X,Y)
      CALL UPRNTI ('ENTER: ','TEXT')
      CALL UINPUT (DATA(IDX),COUNT(I),FLAG,OPTION(I))
      IF(I.EQ.1)CALL UAPEND(COUNT(1),DATA(INDEX(1)),DATA(INDEX(1)))
      CALL USET (OPTION(I))
      CALL UPRINT (X,10.,DATA(IDX))
      X = X + 22.5
      Y = Y - 20.
    1 CONTINUE
      CALL UEND
      STOP
      END
```

Subroutine UINVOK

FUNCTION:

This invokes an already constructed GCS data structure. The current beam/pen position becomes the origin of the structure's coordinate system.

CALLING SEQUENCE:

CALL UINVOK (NAME)

Where

NAME **is an eight-character GCS structure name of an already defined structure.**

OPTIONS which may apply:

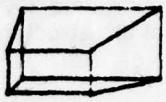
Invocation Scaling Options (UPSET): 'XSCALE', 'YSCALE', 'ZSCALE'

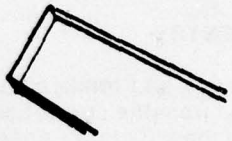
Invocation Rotation Options (UPSET): 'XROTATION', 'YROTATION', 'ZROTATION'

COMMENTS:

The scaling and rotation values for each axis of the structure's coordinate system are obtained from the current settings of the Invocation Scaling and Rotation Options. The origin of the structure's coordinate system is set to the current beam/pen position. A call to UINVOK is equivalent to a call using relative coordinates to U3CALL (0.,0.,-0.,SX,SY,SZ,RX,RY,RZ,NAME).

Programming Notes:





```
      CALL ATTACH (2,'/TEK3DEMO/SAVE;',3,0,IST,)
      CALL USTART
      CALL UASPCT (1.)
      CALL UWINDO (-100.,100.,-100.,100.)
      CALL UPSET ('LIBRARY FILE',1.)
      CALL UTILTY ('LOAD',2.)
      CALL U3MOVE (-90.,5.,0.)
      CALL UINVOK ('BOX   ')
      CALL UPSET ('XSCALE',2.)
      CALL U3MOVE (5.,5.,0.)
      CALL UINVOK ('BOX   ')
      CALL UPSET ('ZROTATION',45.)
      CALL U3MOVE (-50.,-90.,0.)
      CALL UINVOK ('BOX   ')
      CALL UPSET ('XROTATION',25.)
      CALL UPSET ('YROTATION',25.)
      CALL UPSET ('YSCALE',.5)
      CALL U3MOVE (50.,-90.,0.)
      CALL UINVOK ('BOX   ')
      CALL UEND
      STOP
      END
```

Subroutine ULINE

FUNCTION:

This routine creates a line string on the current XY-plane by connecting an array of coordinates with lines drawn with the current line option.

CALLING SEQUENCE:

CALL ULINE (X,Y,PTS)

Where

X	**is an array of length PTS containing the X-components of the coordinates of the points to be connected.**
Y	**is an array of length PTS containing the Y-components of the coordinates of the points to be connected.**
PTS	**is a real variable which specifies the number of points to be connected.**

OPTIONS which may apply:

All pen-related options (see UPEN)

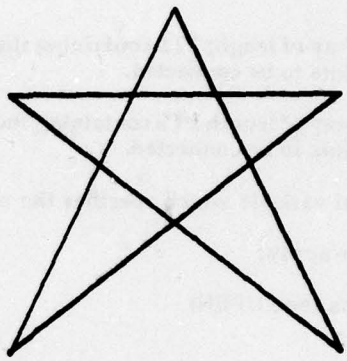
COMMENTS:

This routine will move to the first point specified and then draw a line to succeeding points until the last point is reached. The pen or beam will be left at the last point.

This routine is the complement of the UAXIS routine. UAXIS may be used to create an axis and a plotting environment. ULINE can then be used to plot within this environment.

In three-dimension applications, the line drawn by ULINE will lie in the current XY plane.

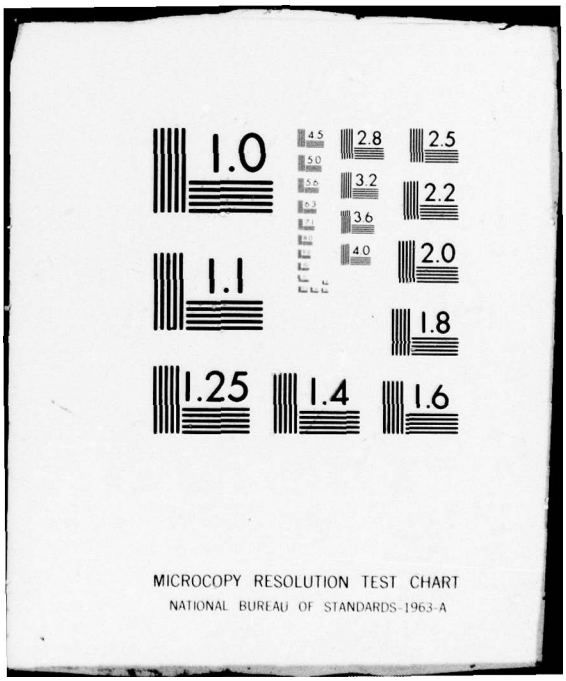
Programming Notes:



```
DIMENSION X(6),Y(6)
DATA X/30.,50.,70.,30.,70.,30./
DATA Y/30.,70.,38.,60.,60.,30./
CALL USTART
CALL ULINE (X,Y,6.)
CALL UEND
STOP
END
```

AD-A083 424 WESTINGHOUSE WORD PROCESSING CENTER PITTSBURGH PA F/G 9/2
GRAPHICS COMPATIBILITY SYSTEM (GCS) PROGRAMMER'S REFERENCE MANU--ETC(U)
OCT 79 DACW39-78-M-2676
UNCLASSIFIED WES-MP-0-79-5 NL

2 OF 3
AD A083424


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Subroutine ULINFT

FUNCTION:

This routine fits a linear equation to the specified input data in the least squares sense and returns the slope and intercept of the equation.

CALLING SEQUENCE:

CALL ULINFT (X,Y,XN,S,YI)

Where

X	**are an array of XN elements which is the X component of the data points to be fitted.**
Y	**are an array of XN elements which is the Y component of the data points to be fitted.**
XN	**is the number of points to be fitted**
S	**is the slope of the line which is fitted to the data points.**
YI	**is the intercept of the line which is fitted to the data points.**

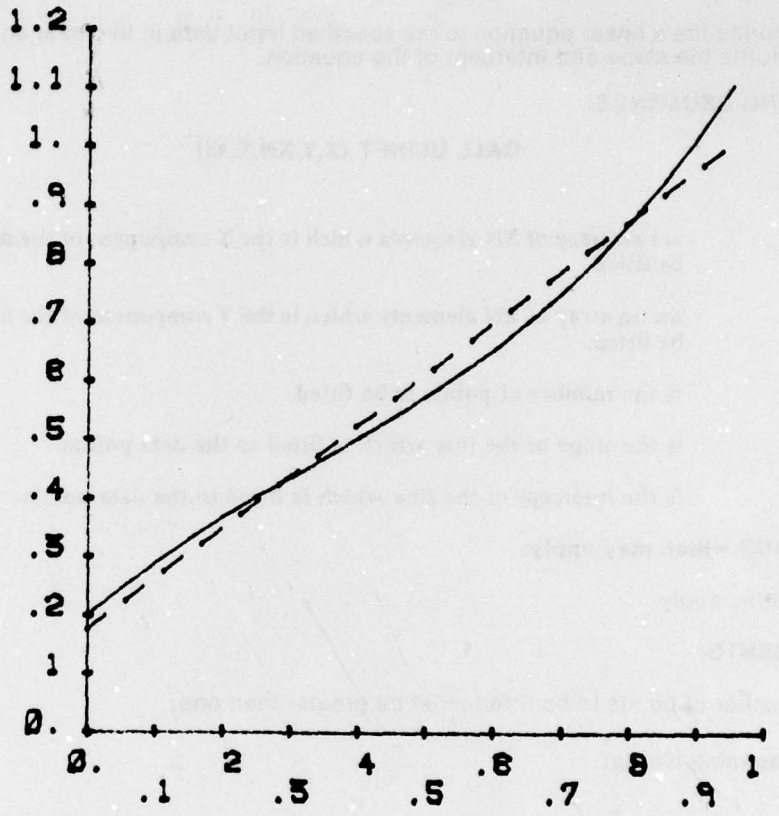
OPTIONS which may apply:

No options apply.

COMMENTS:

The number of points to be fitted must be greater than one.

Programming Notes:




```
      DIMENSION X(20),Y(20)
      DO 10 I = 1, 20
      X(I) = FLOAT(I-1) / 20.
   10 Y(I) = X(I)**3 - X(I)**2 + X(I) + .2
      CALL USTART
      CALL USET ('PERCENT UNITS')
      CALL USET ('SOFTWARE CHARACTERS')
      CALL UPSET ('HORIZONTAL',.03)
      CALL UPSET ('VERTICAL',.05)
      CALL UPLOT (X,Y,1,,20,,'LINE')
      CALL ULINFT (X,Y,20,,SLOPE,YI)
      CALL UMOVE (0.,YI)
      XI = 10. / 20.
      CALL USET ('DASHED LINES')
      CALL UPEN (XI,SLOPE*XI+YI)
      CALL UEND
      STOP
      END
```

FUNCTION:

This routine allows the user to specify a position on the display surface in which will be mapped the corresponding portion of the current virtual space.

CALLING SEQUENCE:

CALL ULOOK(XMIN,XMAX,YMIN,YMAX)

Where

XMIN,XMAX,YMIN,YMAX **are the boundaries of the desired display area in current device units.**

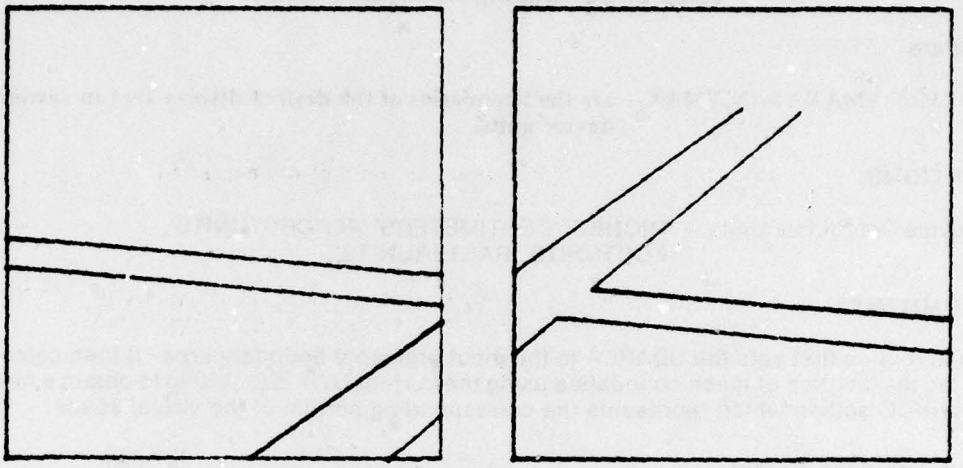
OPTIONS:

Device Coordinate Units — **'INCHES'**,'CENTIMETERS','PERCENTUNITS', 'FONTUNITS','RASTERUNITS'

COMMENTS:

This routine first sets the UDAREA to the input argument boundary area. It then calculates the location of these boundaries using the current UWINDO setting to obtain a new UWINDO setting which represents the corresponding portion of the virtual space.

Programming Notes:



```
CALL ATTACH (2,'/TEK3DEMO/SAVE;',3,0,IST,)
CALL USTART
CALL UPSET ('LIBRARY',1.)
CALL UTILTY ('LOAD',2.)
CALL UWINDO (-100.,100.,-100.,100.)
CALL UVIEW (100.,-250.,50.,0.,100.,0.)
CALL ULOOK (3.5,0.5,1.,4.)
CALL UOUTLN
CALL U3CALL (0.,0.,0.,1.,1.,1.,0.,0.,0.,'ROAD ')
CALL ULOOK (0.,3.,1.,4.)
CALL UOUTLN
CALL U3CALL (0.,0.,0.,1.,1.,1.,0.,0.,0.,'ROAD ')
CALL UEND
STOP
END
```

Subroutine ULSTSQ

FUNCTION:

This routine fits a polynomial equation to the input data by the method of least squares and returns the coefficients of the fitted equation.

CALLING SEQUENCE:

CALL ULSTSQ (X,Y,XN,COEFF)

Where

X	is an array of XN elements which are the X components of the points to be fitted.
Y	is an array of XN elements which are the Y components of the points to be fitted.
XN	is the number of data points.
COEFF	is the array of polynomial coefficients in ascending order, (i.e., COEFF(1)) is the coefficient for the polynomial factor raised to 'DEGREE'). The number of coefficients returned is equal to the degree of the polynomial fit plus one.

OPTIONS which may apply:

UPSET ('POLYNOMIAL', degree)

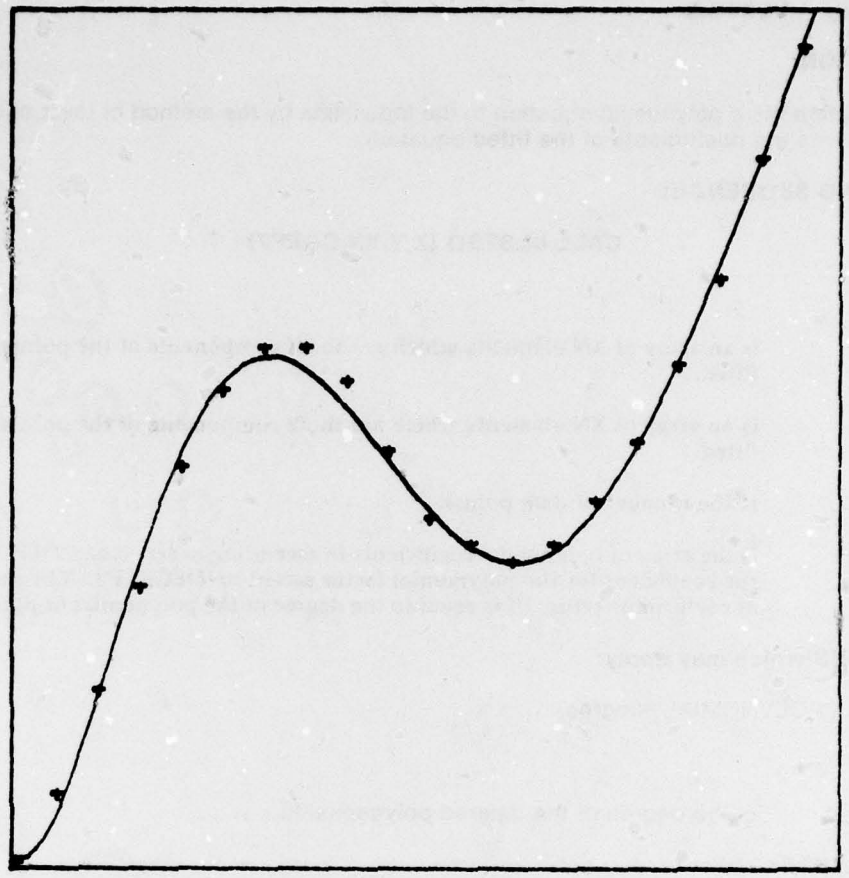
Where

degree	is the degree of the desired polynomial fit.

COMMENTS:

A request for a polynomial fit of degree less than or one degree, or greater than ten is an error. The points to be fitted must be in ascending order. Least squares computations within the subroutine are carried out in double precision arithmetic. Users are cautioned against attempting to perform a high degree fit with a small number of data points.

Programming Notes:



```
      PARAMETER IDEGRE=7
      DIMENSION A(IDEGRE),X(20),Y(20)
      DATA X/0.,5.,10.,15.,20.,25.,30.,35.,40.,45.,50.,55.,60.,
     &      65.,70.,75.,80.,85.,90.,95./
      DATA Y/0.,8.,20.,32.,40.,55.,60.,60.,50.,46.,40.,37.,35.,
     &      37.,42.,49.,58.,68.,82.,95./
      CALL USTART
      CALL UOUTLN
      CALL UPSET ('POLYNOMIAL',FLOAT(IDEGRE-1))
      DO 2 I =1, 20
      XN = FLOAT(I)
      CALL USET ('N+')
      CALL UPEN (X(I),Y(I))
      CALL USET ('LINE')
    2 CONTINUE
      CALL UMOVE (0.0,0.0)
      CALL ULSTSQ (X,Y,XN,A)
      DO 5 I =1, 100
      Y0 = A(1)
      X0 = FLOAT(I)
      XK = X0
      DO 4 J = 2, IDEGRE
      Y0 = A(J) * XK + Y0
      XK = XK * X0
    4 CONTINUE
      CALL UPEN (X0,Y0)
    5 CONTINUE
      CALL UEND
      STOP
      END
```

Subroutine UMARGN

FUNCTION:

This routine sets the left, right, top, and bottom alphanumeric margins for hardware characters to the device dimensions specified.

CALLING SEQUENCE:

CALL UMARGN (XLEFT,XRIGHT,YBOTTM,YTOP)

Where

XLEFT	**is the X-coordinate of the location of the left margin in device, rectangular, absolute user units.**
XRIGHT	**is th X-coordinate of the right margin in device, rectangular, absolute user units.**
YBOTTM	**is the Y-coordinate of the bottom margin device, rectangular, absolute user units**
YTOP	**is the Y-coordinate of the top margin in device, rectangular, absolute user units.**

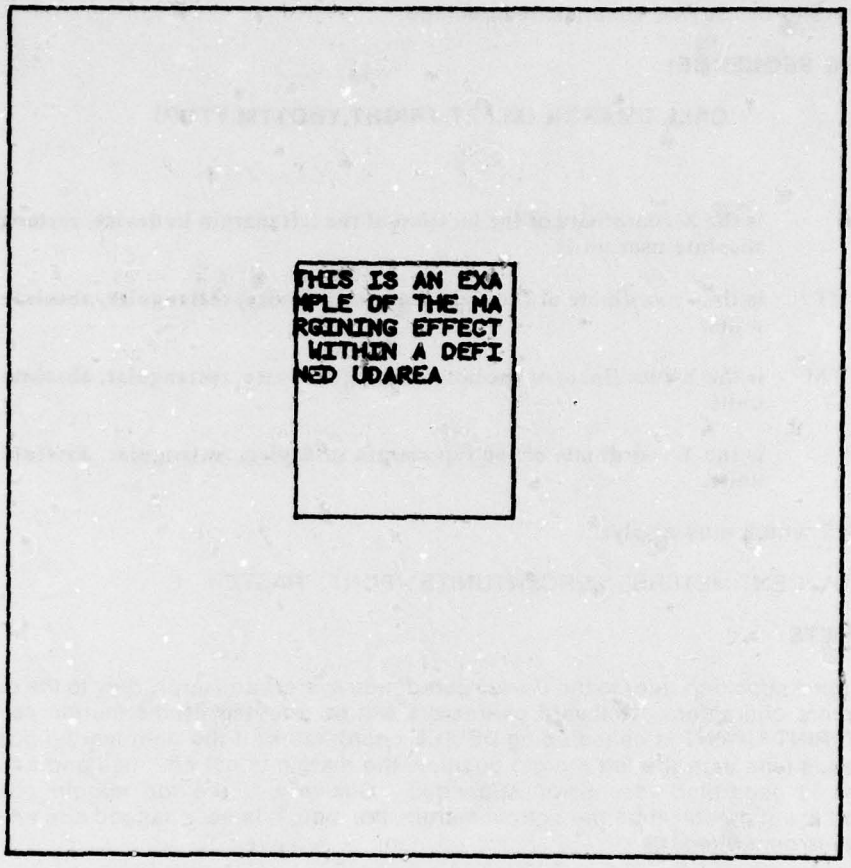
OPTIONS which may apply:

'INCHES', 'CENTIMETERS', 'PERCENTUNITS', 'FONT', 'RASTER'

COMMENTS:

The margins specified refer to the device coordinate system and apply only to the output of hardware characters. Software characters will be affected by the margin settings when UPRINT/UPRNT is called using DEVICE coordinates. If the right margin position specified is less than the left margin position, the margin is not changed and an error message is generated (see Error Appendix). Similarly, if the top margin position specified is not greater than the bottom margin, the margin is not changed and an error message is generated.

Programming Notes:




```
      CALL USTART
      CALL UPSET ('TERMINATOR',';')
      CALL UGET ('PERCENT UNITS')
      CALL UOUTLN
      CALL UDAREA (50.,70.,40.,70.)
      CALL UOUTLN
      CALL USET ('DEVICE')
      CALL UMARGN (50.,70.,40.,70.)
      CALL UPRINT (50.,70.,'THIS IS AN EXAMPLE OF THE MARGINING
     &EFFECTS WITHIN A DEFINED UDAREA;')
      CALL UEND
      STOP
      END
```

Subroutine UMENU

Interactive

FUNCTION:

This routine displays a menuboard containing up to ten options which are offered for selection. The user selects an option using the terminal's graphics input device (if it has one) or keyboard (if it doesn't). The number of the option selected is returned to the user.

CALLING SEQUENCE:

CALL UMENU (OPTNO,OPTNAM,SELECT)

Where

OPTNO	is a FORTRAN REAL variable whose absolute value indicates the number of options available and whose sign determines whether the menuboard already is being displayed. A positive value indicates that the menuboard should be created; a negative value indicates that the menuboard is already being displayed.
OPTNAM	is an array containing the eight-character names of the options for this menu. These names need not have GCS string terminators. There must be one name for each option.
SELECT	is the identifying number of the option chosen by the user which is returned to the calling program.

OPTIONS Which may apppy:

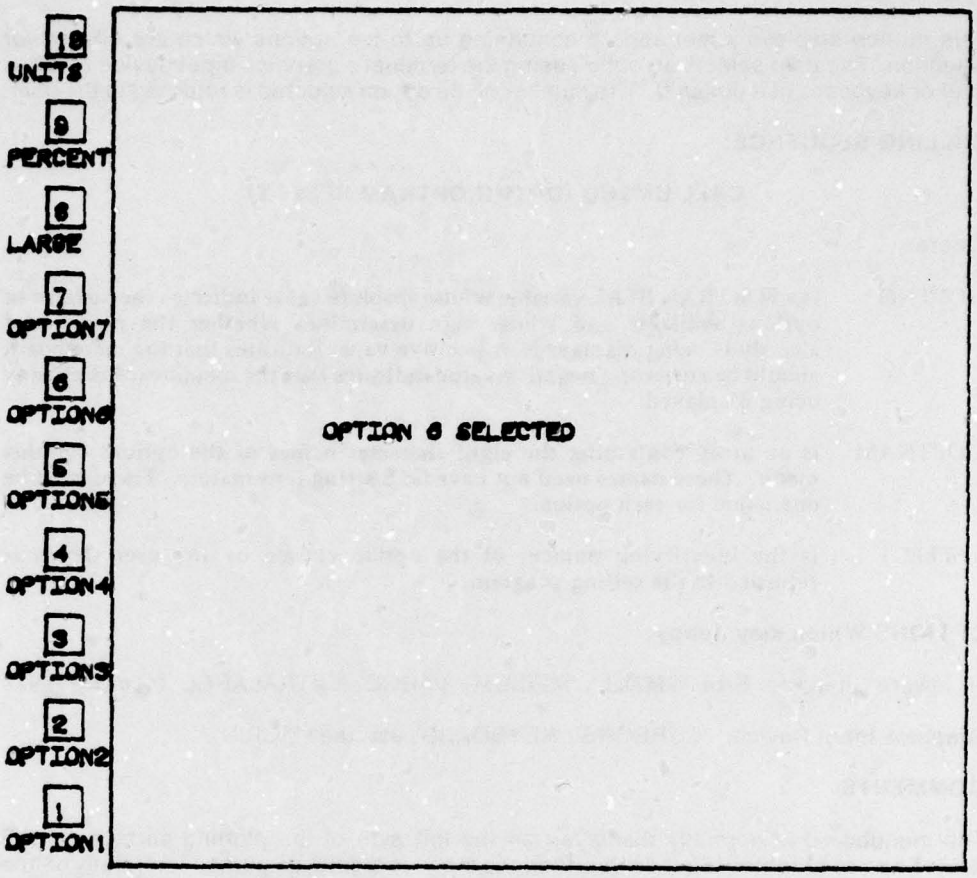
Hardware Character Size. **'SMALL'**, 'MEDIUM', 'LARGE', 'EXTRALARGE' (see U3PRNT).

Graphics Input Device: 'CURSORS', 'KEYBOARD', etc. (see UGRIN).

COMMENTS:

The menuboard is normally displayed on the left side of the plotting surface. If the device has no graphics input device, then the menu is displayed alphanumerically on the control device. The identification numbers have been designed so that the value returned to the user is a REAL number whose INTEGER value is suitable for use as a COMPUTED GO TO index. If the user selects outside of the displayed menu elements, the user is asked to select an option again.

Programming Notes:




```
      DIMENSION MENU(10)
      DATA MENU/'OPTION1','OPTION2','OPTION3','OPTION4','OPTION5',
      CALL USTART
      CALL UPSET ('TERMINATOR',',')
      CALL USET ('LARGE')
      CALL USET ('PERCENT UNITS')
      CALL UDAREA (20.,90.,0.,100.)
      CALL UOUTLN
      CALL UMENU (10.,MENU,SELECT)
      CALL UPRINT (25.,50.,'OPTION ,')
      CALL UPRNTI (SELECT,'INTEGER')
      CALL UPRNTI (' SELECTED,','TEXT')
      CALL UEND
      STOP
      END
```

FUNCTION:

This routine modifies the setting of segment attributes for existing retained segments.

CALLING SEQUENCE:

CALL UMODFY(SEGID,ATID,ATVAL)

Where

SEGID is the segment identifier of the segment whose attribute is to be changed (see UOPEN for a description of SEGID).

ATID is the attribute identifier of the segment attribute whose setting is to be modified. Only three segment attribute identifiers are allowed:

'VISIBILITY'
'HIGHLIGHTING'
'DETECTABILITY'

Only the first four characters are significant.

ATVAL is the value to which the indicated segment attribute is to be set. Valid values are 'ON' and 'OFF'. Any other value will generate an error and retain the existing setting.

OPTIONS which may apply:

Segment/Frame Identifier Mode: **'FNAME'**, 'FNUMBER'
Segment Posting Mode: **'IMMEDIATE'**, 'DELAYED'

COMMENTS:

If the new segment attribute value involves removing information from the display surface, then this will occur immeditely if in 'IMMEDIATE' segment posting mode. If in 'DELAYED' segment posting mode, the information will be removed at the next UPOST or UERASE, or UDELET or UMODFY in 'IMMEDIATE' segment posting mode, whichever occurs first.

Visibility specifies whether the segment is to be displayed or merely defined. Highlights will occur by blinking if possible on the selected display device. Otherwise, heightened intensity or wider lines may be substituted. Detectability controls whether the segment can be recognized during 'pick'-type input operations.

Programming Notes:

Subroutine UMOVE

FUNCTION:

The beam/pen is positioned at the location specified in the arguments. No visible output will appear on the termial.

CALLING SEQUENCE:

CALL UMOVE (X,Y)

Where

X	**is the X- or RADIUS coordinate in current user units.**
Y	**is the Y- or THETA coordinate in current user units.**

OPTIONS which may apply:

Pen Coordinate Options (see UPEN)
Coordinate System Options (see UCOSYS)

COMMENTS:

The execution of this subroutine has no effect on the pen status option (i.e., the line/terminator specification). The beam movement, although invisible is subject to windowing. The operation of UMOVE is identical to that of UPEN when the pen status specified in 'NNUL' (see UPEN).

Programming Notes:

Subroutine UNSAVE

FUNCTION:

This routine will restore the Graphics Status Area (GSA) to its condition at the time the array from which the status is obtained was loaded.

CALLING SEQUENCE:

CALL UNSAVE (SARRAY)

Where

SARRAY is an array in which a previous state of the GSA was saved by USAVE.

OPTIONS which may apply

No options apply.

COMMENTS:

The user should avoid modifying the contents of the save array before he calls UNSAVE as this may cause indeterminate effects during subsequent graphics operations. The current pen position is restored to the position which is specified in the saved array.

Programming Notes:

Subroutine UNSHOW

FUNCTION:

To request that the named frame be placed in omit status.

CALLING SEQUENCE:

CALL UNSHOW (NAME)

Where

NAME	is an eight character alphanumeric constant or variable of a currently defined frame.

OPTIONS which may apply:

No options apply.

COMMENTS:

The execution of this subroutine causes the frame NAME to be deactivated on the face of the display screen. The frame must have been previously defined by a subroutine UFRAME and subroutine UFRAME and subroutine UFREND pair.

Programming Notes:

Subroutine UNSVPN

FUNCTION:

This routine restores the state of all pen-related variables in the Graphics Status Area (GSA) to their condition at the time the array from which the status is obtained was loaded.

CALLING SEQUENCE:

CALL UNSVPN (SARRAY)

Where:

SARRAY is an array in which the state of all pen-related variables of the GSA was saved by USVPN.

OPTIONS which may apply:

No options apply.

COMMENTS:

For a description of when UNSVPN should be used, see USVPN. The user should avoid altering the contents of the save array before he calls UNSVPN as this may cause indeterminate effects during subsequent graphics operations. The current pen position is restored to the position which is specified in the saved array.

Programming Notes:

Subroutine UNSVTR

FUNCTION:

This routine will restore the user coordinate system transformation status of the Graphics Status Area (GSA) to its condition at the time the array from which the status is obtained was loaded.

CALLING SEQUENCE:

CALL UNSVTR (SARRAY)

Where

SARRAY is an array in which a previous state of the coordinate system transform was saved by USVTR.

OPTIONS which may apply:

No options apply.

COMMENTS:

The invocation of this subroutine will cause the automatic reactivation of the user coordinate system which was in effect at the time that the array was loaded. The user should avoid modifying the contents of the save area before he calls UNSVTR as this may cause indeterminate effects during subsequent graphics operations.

Programming Notes:

Subroutine UOPEN **3D**

FUNCTION:

This routine creates a segment/frame with the specified name and characteristics. Subsequent primitives will be inserted into the segment.

CALLING SEQUENCE:

CALL UOPEN(SEGID)

Where

SEGID **is a real number whose integer value is a positive integer (if in 'FNUMBER' mode) or is a Hollerith string of eight characters (if in 'FNAME' mode) which will be used as the segment identifier.**

OPTIONS which may apply:

Retained Segment Type:	**'NOTRANSFORMATION'**	Segment cannot be transformed
	'2DTRANSLATION'	Segment can be translated in 2D
	'2DGENERAL'	Segment can be translated, scaled and rotated in 2D
	'3DTRANSLATION'	Segment can be translated in 3D
	'3DGENERAL'	Segment can be translated , scaled and rotated in 3D

Segment/Frame Identifier Mode:	**'FNAME'**, 'FNUMBER'
Segment Retention Mode:	**'RETAINED'**, 'NONRETAINED', 'STORAGE', 'REFRESHED'
Segment Visibility:	**'VISIBLE'**, 'INVISIBLE'
Segment Highlighting:	**'NOHIGHLIGHTING'**, 'HIGHLIGHTING'
Segment Detectability:	**'UNDECTABLE'**, 'DETECTABLE', 'SENSITIZED', 'DESENSITIZED'

COMMENTS:

Segments may be created in either of two modes, 'RETAINED' ('REFRESHED') and 'NONRETAINED' ('STORAGE'). Segments which are 'RETAINED' may be transformed (if specified) and their segment attributes may be modified (see description of segment attributes below). Segments which are 'NONRETAINED' are displayed but their descriptions are not saved. Such segments will disappear when the display surface is erased or posted (see UPOST). The following discussion pertains only to 'RETAINED' segments.

Each 'RETAINED' segment has three segment attributes associated with it: visibility, highlighting, and detectability. The visibility attribute determines whether a segment is displayed or not. The highlighting attribute indicates whether the segment should be displayed in some enhanced mode. The detectability attribute governs the sensitivity of the segment to 'pick' input. When a segment is UOPENed, the segment attributes are set to current settings contained in the GSA. Segment attributes may be modified at any time by the UMODFY (q.v.) function.

After the segment has been opened, the current value of the following GCS modes will be placed in the segment:

line width
intensity/brightness
line style
text angle
hardware character size
text font

Image transformations may be applied to 'RETAINED' segments if the display device is capable of supporting these transformations. Attempts to perform image transformations on devices which do not have the capability will be ignored and the segment will remain untransformed. It should be noted that image transformations involve moving images around on the display surface and are not the same as moving objects in virtual space. Four routines are provided for applying image transformations as indicated:

Transformation	Applied By
'2DTRANSLATION'	UPLACE(X,Y,NAME)
'2DGENERAL'	UIMAGE(X,Y,SX,SY,R,NAME)
'3DTRANSLATION'	U3PLAC(X,Y,Z,NAME)
'3DGENERAL'	U3IMAG(X,Y,Z,SX,SY,SZ,RX,RY,RZ,NAME)

Image transformation may only be applied to a retained segment which has been 'UOPEN'ed with the retained segment type mode set to a type at least as general as the image transformations to be performed on the segment. The default type is 'NOTRANSFORMATIONS'.

The highlighting attribute will be implemented by the use of blinking whenever possible. The rate of blink may be controlled by the USET options 'SLOWBLINK' and 'FASTBLINK'. The defult rate is 'SLOWBLINK'.

Programming Notes:

Subroutine UORIGN

FUNCTION:

A user coordinate system is composed at the current beam position. No rotation or scale factors are applied to the new coordinate system.

CALLING SEQUENCE:

CALL UORIGN

OPTIONS which may apply:

'WORKINGAXIS'
'REFERENCEAXIS'

COMMENTS:

The invocation of this subroutine causes the creation of a new origin (0.,0.) at the current beam position with respect to the current origin. The rotation factor is assumed to be zero and the scale factors are assumed to be one. For a detailed discussion of the GCS coordinate system faculty, refer to subroutine UCOSYS.

Programming Notes:

Subroutine UOUTLN

FUNCTION:

This routine draws a line along each edge of the current device display area into which the user's virtual window will be mapped, or around each edge of the entire device plotting surface.

CALLING SEQUENCE:

CALL UOUTLN

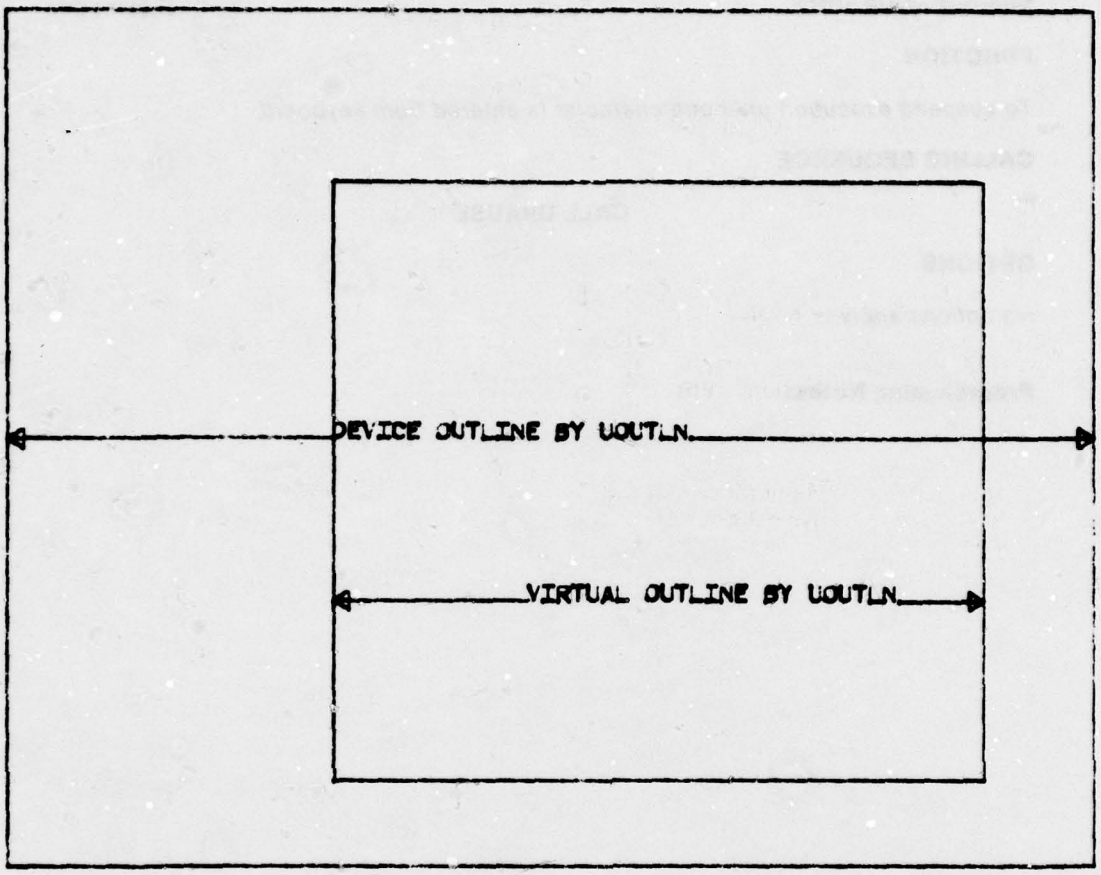
OPTIONS which may apply:

UDAREA setting
Coordinate space selection: **'VIRTUAL'**, 'DEVICE'

COMMENTS:

This routine is convenient for outlining the virtual window plotting area if GCS is set to 'VIRTUAL' and the physical boundaries of the device if GCS is set to 'DEVICE'.

Programming Notes:




```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('EXTRALARGE')
CALL USET ('PERCENT UNITS')
CALL UDAREA (30.,90.,10.,60.)
CALL UOUTLN
CALL USET ('BACKARROW')
CALL UMOVE (0.,30.)
CALL UPEN (30.,30.)
CALL UPRINT (30.,30.,'VIRTUAL OUTLINE BY UOUTLN;')
CALL USET ('ARROW')
CALL UPEN (100.,30.)
CALL USET ('DEVICE')
CALL UOUTLN
CALL USET ('BACKARROW')
CALL UMOVE (0.,50.)
CALL UPEN (30.,50.)
CALL UPRINT (30.,50.,'DEVICE OUTLINE BY UOUTLN;')
CALL USET ('ARROW')
CALL UPEN (100.,50.)
CALL UEND
STOP
END
```

Subroutine UPAUSE **Interactive**

FUNCTION

To suspend execution until one character is entered from keyboard.

CALLING SEQUENCE

CALL UPAUSE

OPTIONS

No options apply.

Programming Notes:

Subroutine UPEN

FUNCTION:

This routine draws a vector of the type indicated by the current line option from the current beam position (vector tail) to the location specified by the arguments (vector head). The current beam position is updated to point to the head of the vector.

CALLING SEQUENCE:

CALL UPEN (X,Y)

Where

X	**is the X- or RADIUS coordinate of the head of the vector in current user units.**
Y	**is the Y- or THETA coordinate of the head of the vector in current user units.**

OPTIONS which may apply:

See U3PEN
Default Z value - UPSET('ZVALUE',value)

COMMENTS:

All pen movements performed by GCS are made in a three dimensional space. For UPEN, the third dimension is specified by the current X-Y plane, which is zero by default.

In the three dimensional version of GCS, this plane can be changed by a call to UPSET, i.e.

CALL UPSET ('ZVALUE',VALUE)

If the current coordinate system is in SPHERICAL coordinates, then VALUE is the PHI component.

Programming Notes:

Subroutine UPEN1

FUNCTION:

To draw a vector from the current beam/pen position to the specified endpoint and to set an option which is in effect only during the execution of the subroutine.

CALLING SEQUENCE:

CALL UPEN1 (X,Y,OPT)

Where

X	is the X- or RADIUS coordinate of the head of the pen/beam movement desired.
Y	is the Y- or THETA coordinate of the head of the pen/beam movement desired.
OPT	is the USET option to be set only during the execution of this subroutine.

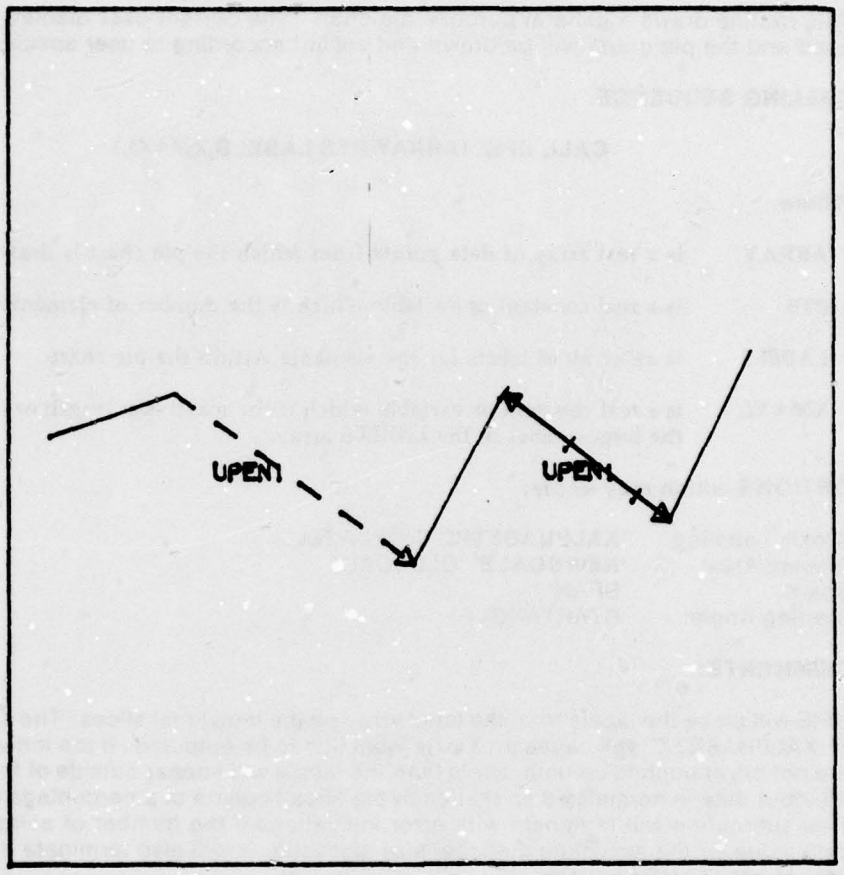
OPTIONS which may apply:

Any USET options
See UPEN for a description of options effecting pen/beam movements.

COMMENTS:

UPEN1 is convenient for setting options effective for only one pen/beam movement; although any USET option may be specified, only those which may apply to the execution of UPEN (see UPEN writeup) will be effective. The Graphics Status Area variable which is modified will be restored to its previous condition after execution of the beam/pen movement.

Programming Notes:




```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('EXTRALARGE')
CALL UOUTLN
CALL UMOVE (5.,50.)
CALL UPEN (20.,55.)
CALL UPEN1 (40.,35.,'DARROW')
CALL UPEN (60.,55.)
CALL UPEN1 (80.,40.,'TDOUBLEARROW')
CALL UPEN (90.,60.)
CALL UPRINT (25.,45.,'UPEN1;')
CALL UPRINT (65.,45.,'UPEN1;')
CALL UEND
STOP
END
```


141

Subroutine UPIE

FUNCTION:

This routine draws a general purpose pie chart. The current user display area will be used and the pie graph will be drawn and scaled according to user specifications.

CALLING SEQUENCE:

CALL UPIE (ARRAY,PTS,LABELS,XMAXL)

Where

ARRAY	**is a real array of data points from which the pie chart is drawn**
PTS	**is a real constant or variable which is the number of elements in ARRAY**
LABELS	**is an array of labels for the elements within the pie chart.**
XMAXL	**is a real constant or variable which is the maximum length or the length of the largest label in the LABELS array.**

OPTIONS which may apply:

X Axis Labeling:	**'XALPHABETIC'** 'NOXLABEL'
Display Area:	**'NEWSCALE'** 'OLDSCALE'
Span:	'SPAN'
Starting Angle:	'STARTANGLE'

COMMENTS:

UPIE will place the labels from the label array on the individual slices. The specification of 'XALPHABETIC' will cause an X axis label title to be outputed. If the individual slices are not big enough to contain labels then the labels will appear outside of the pie chart. All input data is normalized so that each pie slice appears of a percentage of the total. This subroutine will terminate with error indications if the number of points any input data value, or the maximum pie label size is invalid. It will also terminate if the display area is not of sufficient size.

The default piechart drawn by a call to UPIE encompasses a full circle with the initial wedge starting at 0. A portion of the circle can be drawn instead by specifying the angle subtended by the graph by calling

UPSET ('SPAN',ANGLE).

The piechart can be started at an angle other than zero by calling

UPSET ('STARTANGLE',ANGLE)

Programming Notes:

ALGOL - 4%

COBOL - 30%

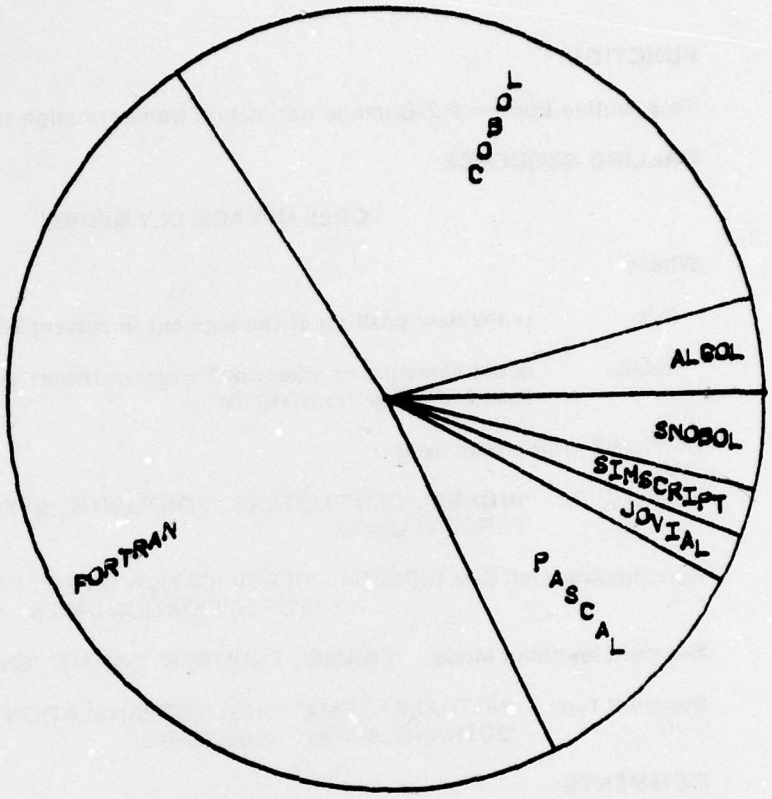
FORTRAN - 46%

PASCAL - 18%

JOVIAL - 2%

SIMSCRIPT - 2%

SNOBOL - 4%



TYPICAL LANGUAGE UTILIZATION

```
      DIMENSION DATA(7)
      CHARACTER LABELS*10(7)
      DATA DATA,Y/4.,30.,46.,10.,2.,2.,4.,100./
      DATA LABELS/'ALGOL;','COBOL;','FORTRAN;','PASCAL;','JOVIAL;',
     &            'SIMSCRIPT;','SNOBOL;'/
      CALL USTART
      CALL UPSET ('TERMINATOR',';')
      CALL USET ('XALPHABETIC')
      CALL UPSET ('XLABEL','TYPICAL LANGUAGE UTILIZATION;')
      CALL UPIE (DATA,7.,LABELS,10.)
      CALL USET ('DEVICE')
      CALL USET ('PERCENTUNITS')
      CALL UDAREA (0.,100.,0.,100.)
      DO 1 I = 1, 7
      Y = Y - (100. / FLOAT(7+1))
      CALL UMOVE (0.,Y)
      CALL UPRNT1 (LABELS(I),'TEXT')
      CALL UPRNT1 (' - ','TEXT')
      CALL UPRNT1 (DATA(I),'INTEGER')
      CALL UPRNT1 ('%;','TEXT')
    1 CONTINUE
      CALL UEND
      STOP
      END
```

FUNCTION:

This routine applies a 2-D image translation transformation to the indicated segment.

CALLING SEQUENCE:

CALL UPLACE (X,Y,SEGID)

Where

X,Y	**is the new position of the segment in current 2-D device units.**
SEGID	**is the identifier of a 'retained' segment/frame which was UOPENed for at least 2-D image translations.**

OPTIONS which may apply:

Device Units: **'INCHES'**, 'CENTIMETERS', 'FONTUNITS', 'SPECIFICATION UNITS', 'PERCENT UNITS'

Specification Unit Size (UPSET): 'SPECIFICATION UNITS', 'XSPECIFICATION UNITS', 'YSPECIFICATION UNITS', 'ZSPECIFICATION UNITS'

Segment Identifier Mode: **'FNAME'**, 'FNUMBER', 'SNAME', 'SNUMBER'

Segment Type: 'NOTRANSFORMATIONS', '2DTRANSLATION', '2DGENERAL', '3DTRANSLATION', '3DGENERAL'

COMMENTS:

The image transormation is applied to the specified segment. If the resulting image exceeds the display dimensions, the result is undetermined.

Image transformations are only applied if supported on the current display surface. Requests for image transformations will be ignored if the display device does not support this facility.

Programming Notes:

Subroutine UPLOT

FUNCTION:

This routine provides a general purpose numeric plotting capability. Given two arrays of corresponding coordinates of one or more curves, it will scale and plot these points along with suitable axes and labels as specified by the user within the current UDAREA. The virtual window will be modified to reflect the resultant scaling.

CALLING SEQUENCE:

CALL UPLOT (X,Y,CURVES,PTS,OPTS)

Where

X	**is the array of X or RADIUS coordinates for the points for all the curves in current user units.**
Y	**is an array of Y or THETA coordinates for the points for all the curves in user units.**
CURVES	**is a single variable which indicates the number of curves to be plotted.**
PTS	**is the array which indicates how many points are in each curve.**
OPTS	**is the array which specifies which USET option will apply to each curve as it is being plotted. One option must be specified for each curve and only the first four characters of the option name should be specified.**

OPTIONS which may apply:

Coordinate Type Options: **'RECTANGULAR'**, 'POLAR', 'LOGARITHMIC' UAXIS Options, (see UAXIS)

Scaling Type Options: **'AUTOSCALE'**, 'FULLSCALE', 'OWNSCALE'

Scale Availability Options: **'NEWSCALE'**, 'OLDSCALE'

Pen Options: see UPEN

Color Options: see UPEN

Curve Fitting Options: **'NOFITTING'**, 'FITLINEAR', 'FITPOLYNOMIAL', 'FITSPLINE'

COMMENTS:

UPLOT is a powerful routine which provides a flexible, yet easy to use means of plotting tabular data. Plotting will take place in 'VIRTUAL' space and the 'SYSTEM' coordinate system (see UCOSYS). The input arrays, X and Y, contain the points for each curve. The number of points in any curve is independent of the number of points in any other curve. This flexibility is possible because the input array POINTS specifies how many points are in each curve. The OPTS array will contain one element for each curve. Each element will contain the first four characters of the name of the USET option which is to apply to the corresponding curve. Normally, these options will be line options (see UPEN). However, they can be any legal USET option. So special effects may be possible by using such options as 'RELATIVE', a cclor option (see below), or a coordinate type option in conjunction with the use of 'OWNSCALE'. The choice of option is left to the

ingenuity of the user. However, the user is warned that some options may produce ambiguous or indeterminate results.

THE AXES AND LABELS:

Although there is no restriction as to the number of points in any curve, all points should be of the same coordinate type since this specification will dictate the kind of axes which will appear, and the mode in which the pints will be plotted. The axes and labels will be created by a call to the subroutine UAXIS. UAXIS options are available when UAXIS is called from UPLOT. For a description of these options see UAXIS.

THE SCALING:

Scaling is performed by UAXIS based on its input parameters. If 'NEWSCALE' has been specified, the input parameters will either be calculated by UPLOT from the input arrays, X and Y, ('AUTOSCALE' or 'FULLSCALE') or will be obtained from the current window specification ('OWNSCALE'). If 'OLDSCALE' has been specified the previous scale calculations will be considered active and the scaling factors will not be recalculated.

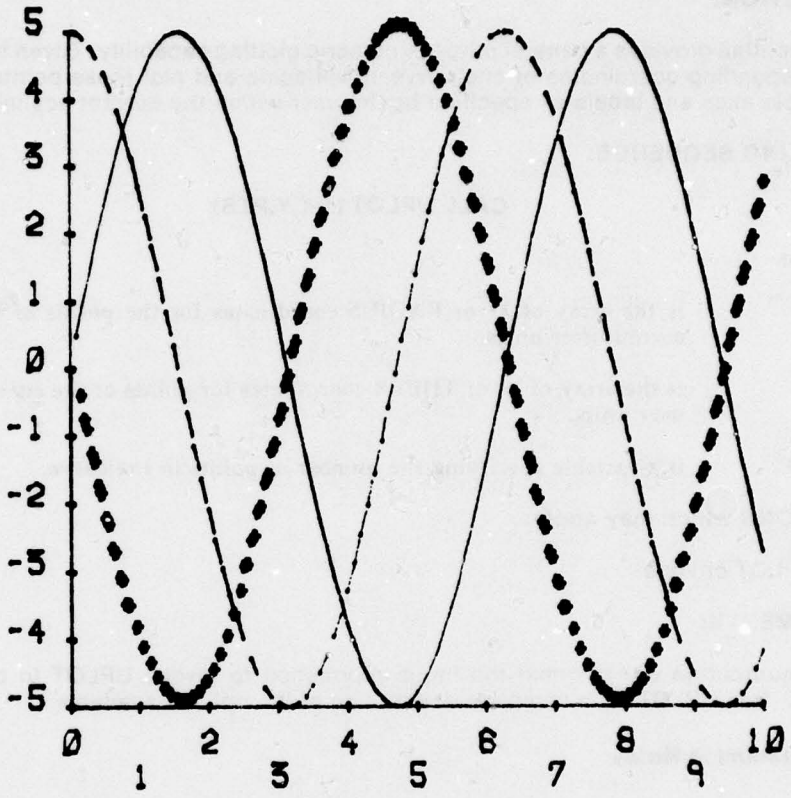
CURVE FITTING:

The default condition for automatic curve fitting is 'NOFITTING'. However, the user can also specify a 'FITLINEAR', a 'FITPOLYNOMIAL', a or a 'FITSPLINE'. For a detailed description of the different methods or to calculate the fits yourself, see ULINFT for 'FITLINEAR', ULSTSQ for 'FITPOLYNOMIAL', and 'UPSLIN' for 'FITSPLINE'. The OPTS array specification used for the original points will also be used for the fitted curve.

COORDINATE REPEATABILITY

The X and/or Y components of the coordinate provided may be held constant or repeated for all curves. Thus, if several curves are to be drawn with identical X values, but different Y values, then the X values need specified only once, and the USET option 'XREPEAT' specified. If one component is to be held constant, then the USET options 'XCONSTANT' or 'YCONSTANT' may be specified, and only a single X or Y value provided.

Programming Notes:




```
      DIMENSION X(300), Y(300), POINTS(3)
      INTEGER OPTS(3)
      DATA OPTS/'LINE','DASH','ANUL'/
      DATA POINTS/100.,100.,100./
      DO 10 I = 1, 100
         X(I) = FLOAT(I) / 10.
         Y(I) = SIN(X(I)) * 5.0
         Y(I+100) = COS(X(I)) * 5.0
         Y(I+200) = SIN(X(I)+3.14159) * 5.0
   10 CONTINUE
      CALL USTART
      CALL UPSET ('SETDASH',1.)
      CALL USET ('PERCENT UNITS')
      CALL USET ('SOFT')
      CALL UPSET ('HORIZONTAL',.3)
      CALL UPSET ('VERTICAL',.5)
      CALL USET ('XREPEAT')
      CALL UPLOT (X,Y,3.,POINTS,OPTS)
      CALL UEND
      STOP
      END
```

Subroutine UPLOT1

FUNCTION:

This routine provides a general purpose numeric plotting capability. Given two arrays or corresponding coordinates of one curve, it will scale and plot these points along with suitable axes and labels as specified by the user within the current scaling.

CALLING SEQUENCE:

CALL UPLOT1 (X,Y,PTS)

Where

X	**is the array of X- or RADIUS coordinates for the points of the curve in current user units.**
Y	**is the array of Y- or THETA coordinates for points of the curve in current user units.**
PTS	**is a variable specifying the number of points in the curve.**

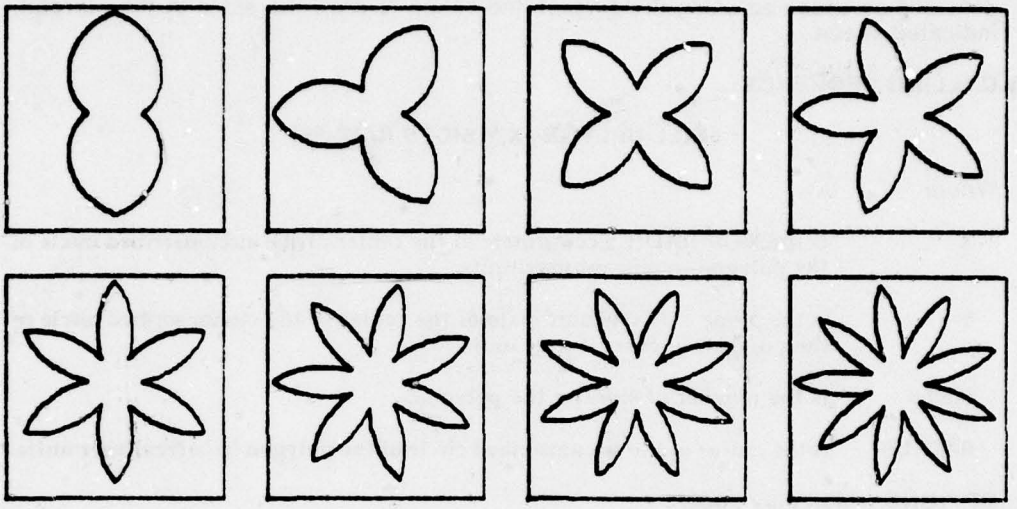
OPTIONS which may apply:

All UPLOT options

COMMENTS:

This subroutine will reformat the input information to invoke UPLOT to produce the curve. See UPLOT for a complete description of the options available.

Programming Notes:



```
      DIMENSION R(361),THETA(361)
      DATA PI,W,Y0/3.14159265,0.5,4.71/
      DO 1 I = 1, 361
    1 THETA(I) = FLOAT(I) * PI / 180.0
      CALL USTART
      CALL USET ('POLAR')
      CALL USET ('RADIANS')
      CALL USET ('NOAXES')
      CALL USET ('NOXLABEL')
      CALL USET ('NOYLABEL')
      DO 3 I = 1, 2
      X0 = -1.5
      Y0 = Y0 - 1.6
      DO 3 J = 1, 4
      W = W + 0.5
      X0 = X0 + 1.6
      CALL UDAREA (X0,(X0+1.5),Y0,(Y0+1.5))
      CALL UOUTLN
      DO 2 K = 1, 361
    2 R(K) = 1.2-0.7*(ABS(COS(W*THETA(K)))-ABS(SIN(W*THETA(K))))
      CALL UPLOT1 (R,THETA,361.0)
    3 CONTINUE
      CALL UEND
      STOP
      END
```

Subroutine UPLYGN

FUNCTION:

This routine creates a regular ploygon of the indicated number of sides and radius centered as specified using the current line option to draw the sides in the currently indicated space.

CALLING SEQUENCE:

CALL UPLYGN (X,Y,SIDES,RADIUS)

Where

X	is the X- or RADIUS coordinate of the center of the circumscribed circle of the polygon in current user units.
Y	is the Y- or THETA coordinate of the center of the circumscribed circle of the polygon in current user units.
SIDES	is the number of sides in the polygon.
RADIUS	is the radius of the circumscribed circle of the polygon in current user units.

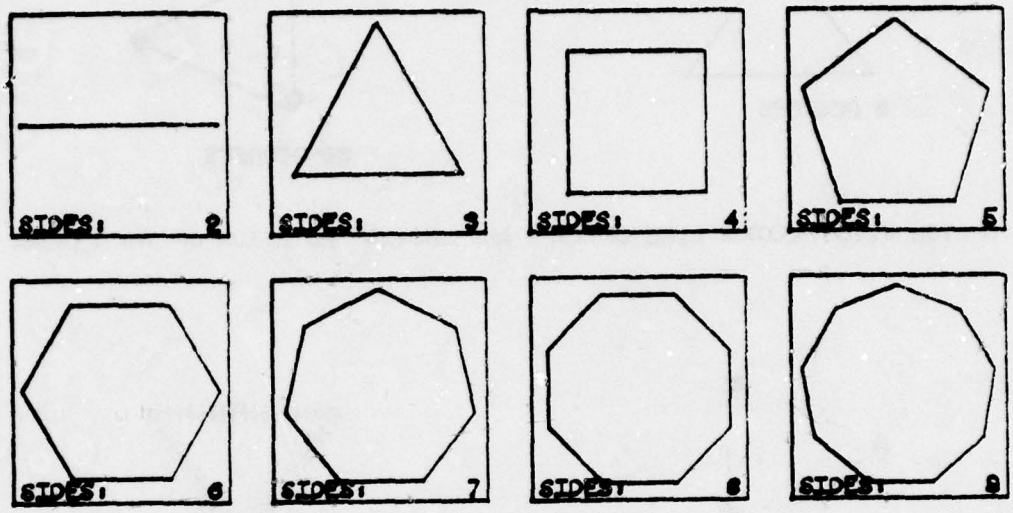
OPTIONS which may apply:

Line Options
Upset Option 'Orientation'

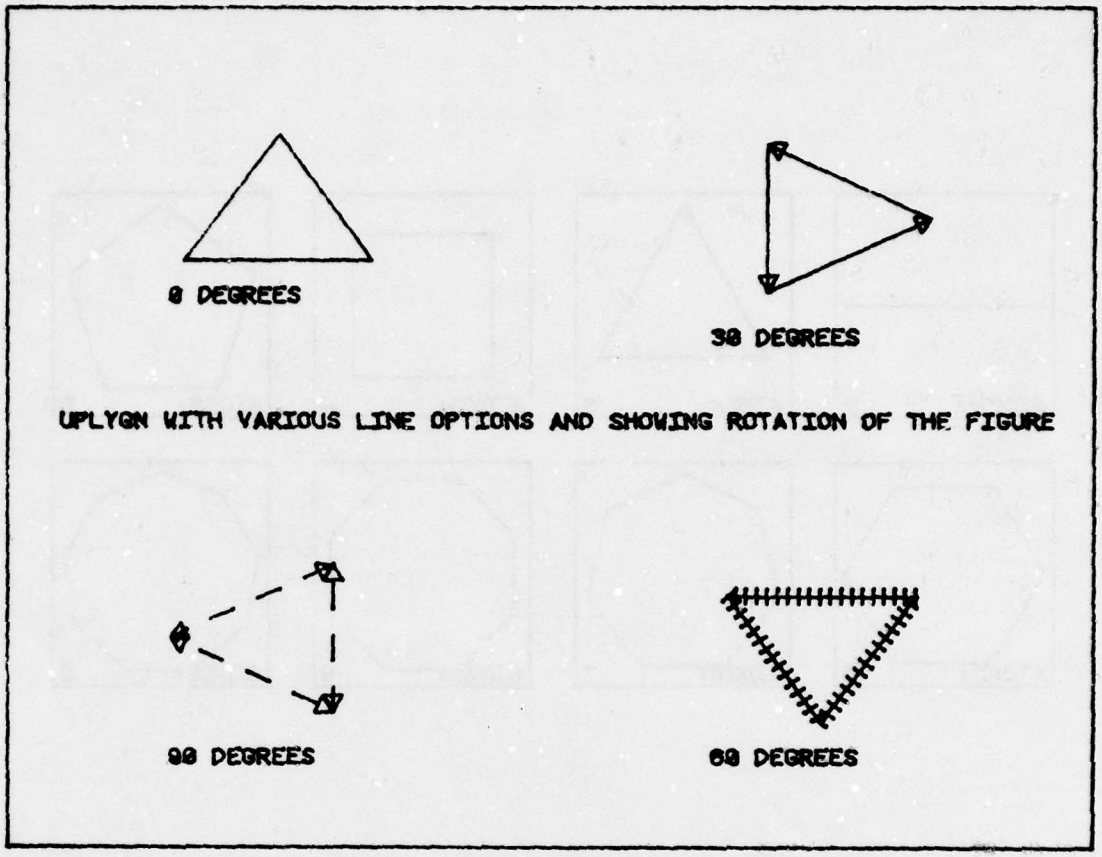
COMMENTS:

The polygon will be drawn centered at the location specified by the input variables X and Y. It will have a radius (i.e., distance from the center to any point) as indicted by RADIUS, each side of the polygon will be a line of the option currently set in the GSA. The default ORIENTATION of zero degrees will result in the polygon having one side parallel to the current X axis. By specifying an orientation in current angular units, the entire polygon will be displayed rotated appropriately around its center.

Programming Notes:




```
      DATA SIDES,Y0/1.0,4.7/
      CALL USTART
      CALL UPSET ('TERMINATOR',',')
      CALL UWINDO (-1.1,1.1,-1.1,1.1)
      DO 1 I = 1, 2
      X0 = -1.5
      Y0 = Y0 - 1.6
      DO 1 J = 1, 4
      X0 = X0 + 1.6
      SIDES = SIDES + 1.0
      CALL UDAREA (X0,(X0+1.5),Y0,(Y0+1.5))
      CALL UOUTLN
      CALL UPLYGN (0.0,0.0,SIDES,1.0)
      CALL USET ('TEXT')
      CALL UPRINT (-1.0,-1.05,'SIDES:',')
      CALL USET ('INTEGER')
      CALL UPRINT (0.0,-1.05,SIDES)
    1 CONTINUE
      CALL UEND
      STOP
      END
```




```
      CALL USTART
      CALL UPSET ('TERMINATOR',',')
      CALL USET ('PERCENT UNITS')
      CALL USET ('DEVICE')
      CALL UDAREA (10.,40.,60.,100.)
      CALL UOUTLN
      CALL UPLYGN (25.,75.,3.,10.)
      CALL UPRINT (15.,65.,'0 DEGREES;')
      CALL UDAREA (60.,100.,60.,100.)
      CALL USET ('ARROW')
      CALL UPSET ('ORIENTATION',30.)
      CALL UPLYGN (75.,75.,3.,10.)
      CALL UPRINT (65.,60.,'30 DEGREES;')
      CALL UDAREA (60.,100.,10.,40.)
      CALL USET ('TPOINT')
      CALL UPSET ('ORIENTATION',60.)
      CALL UPLYGN (75.,25.,3.,10.)
      CALL UPRINT (65.,10.,'60 DEGREES;')
      CALL UDAREA (10.,40.,10.,40.)
      CALL USET ('DDOUBLEARROW')
      CALL UPSET ('ORIENTATION',90.)
      CALL UPLYGN (25.,25.,3.,10.)
      CALL UPRINT (15.,10.,'90 DEGREES;')
      CALL UPRINT (5.,50.,'UPLYGN WITH VARIOUS LINE OPTIONS AND
     & SHOWING ROTATION OF THE FIGURE,')
      CALL UEND
      STOP
      END
```

FUNCTION:

This routine defines the point which will be used along with the two end points of the line, to define the plane for line terminators and tic marks.

CALLING SEQUENCE:

CALL UPOINT(X,Y,Z)

Where

X,Y,Z **are the coordinates of some point in current units.**

OPTIONS:

Pen Coordinate Options: see U3PEN

COMMENTS:

The point specified is stored for later use when drawing vectors. The user should be careful to define a point which will not be colinear with any line being drawn.

Programming Notes:

Subroutine UPOST

FUNCTION: *3D*

This routine updates the display surface to contain only the defined visible retained segments/frames, deleting all non-retained information.

CALLING SEQUENCE:

CALL UPOST

OPTIONS which may apply:

Security Classfication Mode: 'UNSECURED', 'UNCLASSIFIED', 'CONFIDENTIAL', 'SECRET', 'TOPSECRET'

Segment Visibility Attribute: **('VISIBILITY', 'ON'),** ('VISIBILITY', 'OFF')

COMMENTS:

All non-retained information will be cleansed from the display surface. This includes remaining segments made invisible while in 'DELAYED' segment posting mode. Non-retained information is considered to be all information generated while not within a 'retained' segment or 'retained' segments where visibility attribute has been set to 'OFF'.

Programming Notes:

Subroutine UPRINT

FUNCTION:

This subroutine enables the user to print information at the position specified. The five (5) options available to the user are: 'TEXT', 'REALNUMBER', 'INTEGERNUMBER', *'XYCOORDINATES'*, and 'XYZCOORDINATES'. The output characters will be either hardware or software depending on the current setting in the Graphics Status Area. Margining will occur with hardware characters and windowing will occur with software characters. The beam will remain positioned at the next character position following the last character sent.

CALLING SEQUENCE:

CALL UPRINT (X,Y,DATA)

Where

X	is the X- or RADIUS coordinate of the lower left corner of the first character of the output.
Y	is the Y- or THETA coordinate of the lower left corner of the first character of the output.
DATA	is a single variable or array containing either real numbers or a GCS text string. The size of data is variable; it is a single variable if 'REALNUMBER' or 'INTEGERNUMBER' is specified; it is a two word array if 'XYCOORDINATES' is specified; it is a three word array if **'XYZCOORDINATES'** are specified; and it may be any length if 'TEXT' is specified.

OPTIONS which may apply:

All U3PRNT options

Default Z value: UPSET ('ZVALUE',VALUE)

COMMENTS:

All text functions performed by GCS are performed a three dimensional space. For UPRINT, the third dimension is defined as the current X-Y plane, which is at Z=0, by default.

In the three dimensional version of GCS, this plane can be changed by a call to UPSET, i.e.

CALL UPSET ('ZVALUE',VALUE)

If the two dimensional version of GCS is used, then the XY plane corresponds to the screen surface.

Programming Notes:

Subroutine UPRNT1

FUNCTION:

This routine displays the data provided at the current beam position in the format currently specified in either hardware or software characters subject to the one time setting of the USET option furnished, which only applies during the executing of this subroutine. Upon return, the beam is positioned at the next character position following the last character produced.

CALLING SEQUENCE:

CALL UPRNT1 (DATA,OPTION)

Where

DATA	is a single variable or array containing either real numbers or a GCS text string. The size of DATA is variable: it is a single variable if 'REALNUMBER' or 'INTEGERNUMBER' is specified; it is a two word array if 'XYCOORDINATES' is specified; it is a three word array if 'XYZCOORDINATES' is specified; and it may be any length if 'TEXT' is specified.
OPTION	is a Hollerith-string variable or literal defining the USET option to apply during the execution of this subroutine.

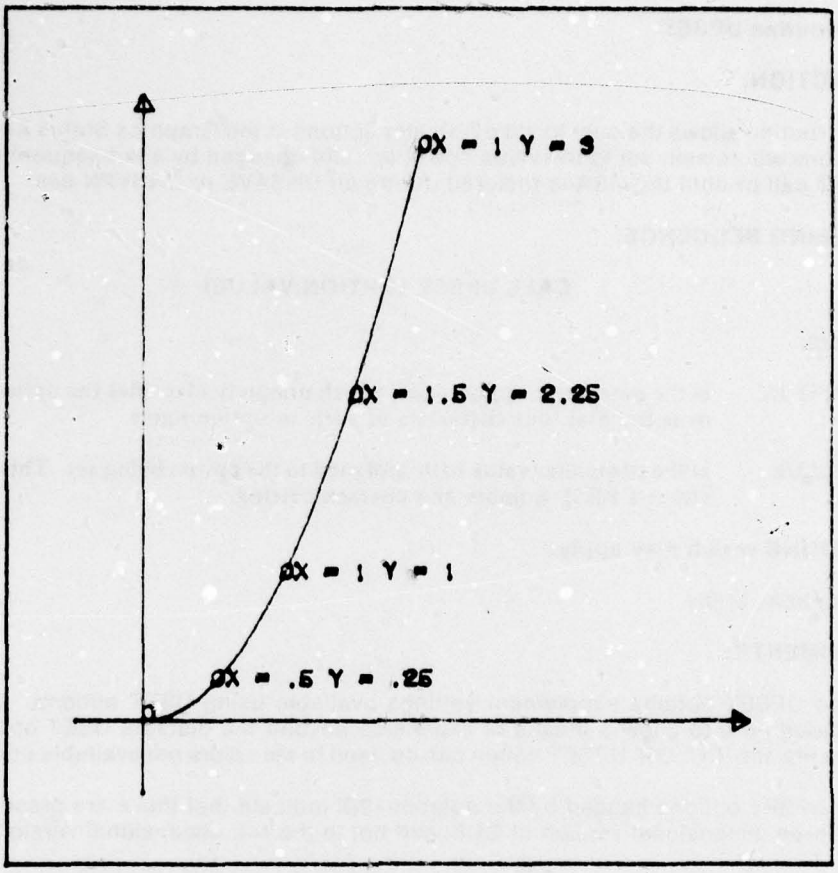
OPTIONS which may apply:

Same as those for UPRINT except that Coordinate System Options do not apply.

COMMENTS:

Calling UPRINT1 is equivalent to the sequence of calling USET with OPTIONS, calling UPRINT at the current beam position with DATA, and then calling USET with the appropriate option which would restore the status cnanged by OPTION. UPRNT1 is convenient for producing strings of output which contain differing data types.

Programming Notes:




```
      CHARACTER OPTION*10(4)
      DATA OPTION/'REALNUMBER','INTEGER',
     &            'REALNUMBER','INTEGER'/
      CHARACTER TITLE*8(2)/' X = ',' Y = '/
      DATA X,Y/0.,0./
      CALL USTART
      CALL UPSET ('TERMINATOR',';')
      CALL UOUTLN
      CALL UWINDO (-1.,5.,-1.,5.)
      CALL UMOVE (0.,-.5)
      CALL UPEN1 (0.,4.4,'LARROW')
      CALL UMOVE (-.5,0.)
      CALL UPEN1 (4.4,0.,'LARROW')
      CALL UPEN1 (X,Y,'NO')
      DO 1 I = 1, 4
      DO 2 J = 1, 5
      X = X + .10
      Y = X**2
      CALL UPEN (X,Y)
    2 CONTINUE
      CALL UPEN1 (X,Y,'NO')
      CALL UPRNT1 (TITLE(1),'TEXT')
      CALL UPRNT1 (X,OPTION(I))
      CALL UPRNT1 (TITLE(2),'TEXT')
      CALL UPRNT1 (Y,OPTION(I))
      CALL UMOVE (X,Y)
    1 CONTINUE
      CALL UEND
      STOP
      END
```

Subroutine UPSET

FUNCTION:

This routine allows the user to set parameter options in the Graphics Status Area (GSA). Options will remain set to the value specified until changed by a subsequent UPSET or USET call or until the GSA is restored during an UNSAVE or UNSVPN call.

CALLING SEQUENCE:

CALL UPSET (OPTION,VALUE)

Where

OPTION **is the mnemonic option name which uniquely identifies the option to be set or is the first four characters of such an option name.**

VALUE **is the parameter value to be assigned to the option being set. This is always either a REAL number or a character string.**

OPTIONS which may apply:

No options apply.

COMMENTS:

Some UPSET options supplement settings available using USET options. These are included here to allow a means of expansion beyond the discrete USET options. For example, the 'COLOR' UPSET option can be used to set colors not available under USET.

The UPSET options flagged by the notation '3D' indicate that these are present only in the three dimensional version of GCS, and not in the two dimensional version.

Programming Notes:

UPSET OPTIONS

Option Name	Value
'ANGLE OF TEXT'	Is an angular value which specifies the angle of the text string in relation to the current X axis. Default value is 0.
'ASPECT RATIO'	Specify aspect ratio of display surface.
'ATTITUDE'	Is an angular value which specifies the orientation of the up direction axis with the sides of the window. Default value is 0. meaning that the up direction is parallel with the left and right window boundaries and pointing towards the top of the window.
'BASE OF LOGARITHMS'	Is a positive value which specifies the base of the logarithms used to perform logarithmic scaling. Default base is 10. This option sets the specified base along each coordinate component. THe base of Naperian logarithm is specified by the character string IHE.
'BRIGHTNESS'	Is a value between 0 and 100. percent indicating the position in the range of possible line intensity settings from dimmest to brightest. Default brightness is 60%.
'CHARACTER'	Is a Hollerith character which becomes the current system character. The default system character is '*'.
'COLOR'	Is an integer which specifies the color index within the color table. Values 0-7 are predefined to represent black, white, red, green, yellow, blue, magenta, and cyan, respectively. Default color is device-dependent.
COPY DELAY'	Is a value which indicates the number of seconds of delay required during generation of a hard copy. Default value is device-dependent and is preset to the value required by the selected device.
'DISTANCE'	Is a value measured from the current view plane distance base specifying the position of the view (projection) plane. The default is 0. measured from the view site.
'FNTFILE'	Is a Fortran file number representing the file containing the font descriptors. Default is zero indicating no font file has been specified.
'FONT NAME'	Is a Hollerith string indicating the desired character font. Default is 'SGCS' which is the most efficient font.
'GREYSCALE'	Is a value indicating a particular grey level for terminals which support multiple grey scales rather than colors.
'GRID SPECIFICATION'	Is a value containing a dash specification to be used when generating grid axes. Default is 0. which indicates a solid line should be used.

'HORIZONTAL SIZE'	Indicates the width of a software character position in current user units. Default is 5 virtual units.
'INPUT FILE'	Is a Fortran file number indicating which file will be used to obtain graphics input. The default is set to the appropriate computer-system dependent file.
'LABELbROTATION'	Is the number of angular units the axes labels are to be rotated around the axes.
'LIBRARY FILE'	Is a Fortran file number indicating which file should be used by the GCS structure and segmentation facilities as a random work file. Default is 0. indicating no file has been provided.
'LOWER'	Is a Hollerith character which will be used by GCS as the indication to shift to lower case. Default character is the greater than sign on the keyboard.
'MARKER INDEX'	Is a value selecting a marker/symbol. Default marker symbol is 0. indicating a point.
'ORIENTATION'	Is an angular value indicating the display orientation of GCS created software symbols and figures such as software characters, polygons, and rectangles. Default orientation is 0.
'OUTPUT FILE'	Is a Fortran file number indicating which file will be used for sending graphics output to the display device. The default is set to the appropriate computer system dependent file. This is a file to which the pseudo display file is written.
'POLYNOMIAL DEGREE'	Specifies the degree of the polynomial to be created in calculating a least squares fit through a collection of points. Default value is 5.
'PRECISION'	Specifies the number of significant digits to appear when displaying real numbers. Default value is 4.
'ROTATION'	Same as 'ORIENTATION'.
'SCALEFACTOR'	Specifies a scale to be applied to GCS-created geometric figures such as polygons and rectangles.
'SCRIPTLEVEL'	Is a value indicating the scripting level to be set for textual output. Default value is 1.
'SETDASH'	Specifies the characteristics of the dashed lines to be plotted by UPEN. Default value is 56.
'SIZE'	Is a positive whole number which sets hardware character sizes. Values of 1. through 4. correspond to USET options 'SMALL', 'MEDIUM', 'LARGE', and 'EXTRA LARGE', respectively. Default value is 1. for 'SMALL' characters.

'SLANTANGLE'	Is an angular value indicating the amount of slant from the vertical for italicized software characters. Default value is approximately 18 degrees.
'SPAN ANGLE'	Is an angular value indicating the portion of a circle to be occupied by the pie chart. Default value is 360. degrees.
'SPECIFICATION UNITS'	Is a positive value indicating the number of specification units contained in device space for all directions. Default value is 1000.
'SPEED'	Specifies the speed of the communication line in characters per second. Default value is system dependent.
'START ANGLE'	Is an angular value indicating the starting position of the first wedge of the pie chart. Default value is 0.
'STRUCTURE TABLE SIZE'	Is a whole number indicating the number of words in the user provided structure table. Default value is 100.
'SUBSCRIPT CHARACTER'	Is a Hollerith character which will be used to decrease the scripting level by 1. Default subscript character is the underline (0/8/6).
'SUPERSCRIPT CHARACTER'	Is a Hollerith character which will be used to increase the scripting level by 1. Default superscript character is the up arrow (0/8/5).
'SZMARKER'	Is a value in current device units which specifies the size of software generated markers. Default value is device-dependent.
'TABHORIZONTAL'	Is an array of 10 elements containing 10 tab positions in current device units. Default value has all tab stops set to zero.
'TABVERTICAL'	Is an array of 10 elements containing 10 vertical tab positions in current device units. Default value has all tab stops set to zero.
'TERMINATOR'	Is a Hollerith character which will be used as the GCS string terminator character. Default value is the backslash
'TICINTERVAL' TICLENGTH'	Specifies the distance in current user units between tic marks of a UPEN created tic line. Default value is 10.
'TICMINUS'	Specifies in current units the size of that portion of a tic mark which lies on the clockwise side of the tic line. Default value is 0.5 inches.
'TICPLUS'	Specifies in current units the size of that portion of a tic mark which lies on the clockwise side of the tic line. Default value is 0.5 inches.

TICX'	Specifies the distance between tic marks or grid lines along X axes. Default value is 0. indicating that a 'nice' number should be chosen.
'TICY'	Is the same as 'TICX' for Y axes.
'TICZ'	Is the same as 'TICX' for Z axes
'UPPER'	Is a Hollerith character which will be used by GCS as the indication to shift to upper case. Default character is the less than symbol on the keyboard.
'VERTICAL SIZE'	Indicate the height of a software character position in current user units. Default value is seven virtual units.
'WIDTH'	Is a value in current units of the width of a line. Default value is 0. indicating a thin line.
'WRITE FILE'	Is a Fortran file number indicating which file will be used to generate non-graphic output. The default is set to the appropriate computer system dependent file.
'XBASE OF LOGS'	Is a positive value or the character string IHE which specifies the base for logarithmic scaling along the x-component. Default value is 10.
'XLABEL'	Specifies the alphanumeric label to be displayed along X axis. Default value is 'X'.
'XPERCENT'	Is a value specifying the portion of the width of a software character position to be occupies by a character. Default value is .65 indicating 65% of the width.
'XROTATION'	Is an angular value indicating the amount of rotation around the X axis for UINVOK structure invocations. Default value is 0.
'XSCALE'	Is the scale factor to be supplied along the X axis for UINVOK structure invocations. Default value is 1.
'XSIZE'	Is the size of hardware or simulated hardware character positions in current device units. Default value is device-dependent and corresponds to 'SMALL' hardware character size.
'XSPECIFICATION UNITS'	Is a positive value indicating the number of x-specification units in device space. Default value is 1000.
'YBASE OF LOGS'	Is a positive value or the character string IHE which specifies the base for logarithmic scaling along the y-component. Default value is 10.
'YBASE OF LOGS'	Is a postivie value which specifies the base for logarithmic scaling along the Y component. Default value is 10
'YLABEL'	Specifies the alphanumeric label to be displayed along the Y axis. Default value is 'Y'.

'YPERCENT'	Is a value specifying the portion of the height of a software character position to be occupies by a character. Default value is .65 indicating 65% of the height.
'YROTATION'	Is an angular value indicating the amount of rotation around the Y axis for UINVOK structure invocations. Default value is 0.
'YSCALE'	Is the scale factor to be applied along the Y axis for UINVOK structure invocations. Default value is 1.
'YSIZE'	Is the size of hardware or simulated hardware character positions in current device units. Default value is device-dependent and corresponds to 'SMALL' hardware character size.
'YSPECIFICATION UNITS'	Is a positive value indicating the number of Y specification units in device space. Default value is 1000.
'ZBASE OF LOGS'	Is a positive value or the character string IHE which specifies the base for logarithmic scaling along the z-component. Default value is 10.
'ZLABEL'	Specifies the alphanumeric label to be displayed along the Z axis. Default value is 'Z'.
'ZROTATION'	Is an angular value indicating the amount of rotation aroung the Z axis for UINVOK structure invocations. Default value is 0.
'ZSCALE'	Is the scale factor to be applied along the Z axis for UINVOK structure invocations. Default value is 1.

*means not implemented

Subroutine UQUERY

FUNCTION:

This routine obtains the current setting of USET or UPSET options and returns them to the user.

CALLING SEQUENCE:

CALL UQUERY(OPTION,VALUE)

Where

OPTION	**is a character string indicating the USET or UPSET option value desired. Valid options are listed below.**
VALUE	**is either an integer array dimensioned to (at least) 10, an integer scalar, or a real scalar, depending on the option queried.**

OPTIONS which may apply:

All USET and UPSET options

COMMENTS:

VALUE is always a scalar except for axis titles in which case it is an array ten elements in length. VALUE is a HOLLERITH string for all USET UQUERY options. For UPSET options with numeric settings, the value returned is converted to the current corresponding user units.

Those options flagged by the notation '3D' are available only in the three dimensional version of GCS, and not in the two dimensional version.

Programming Notes:

UQUERY OPTIONS

Type	Query Name		Returns (Default in Parentheses)
USET	'ABUTTING'		Page abutting mode ('NONABUTTING')
USET	'ACENTERING'		Alphanumeric centering ('NOCENTERING')
USET	'ADJUSTMENT'	3D	Axis view adjustment option (plane axis plotted on view port or viewed from current view point) ('XYZVIEW')
USET	'ANGULARUNITS'		Current user angular units ('DEGREES')
USET	'ANGLE OF TEXT'		Current angle of text output (0.)
UPSET	'ASPECTRATIO'	3D	Display Surface aspect ratio
UPSET	'ATTENTION QUEUE SIZE'		Attention queue size (100.)
UPSET	'BACKGROUND COLOR'		Background color index (Black = 0.)
USET	'BLENDMODE'*		Color Blending mode ('SUBTRACTIVE')
USET	'BLINKRATE'		Blink rate ('NOBLINK')
UPSET	'BRIGHTNESS'		Display intensity (60%)
USET	'BUILD'	3D	Structure build mode flag ('NOBUILD')
UPSET	'CHARACTER'		Current system character as Hollerith string (*)
USET	'CLIPPING'		Device space clipping flag ('NOCLIP', 'CLIP' or 'INVERTED')
USET	'COLOR'		Current color index (device dependent)
UPSET	'COPY DELAY'		Copy delay time (device dependent)
UPSET	'CSPACING'		Character spacing mode ('HORIZONTAL')
USET	'CURVE'		Current curve approximation mode ('CONTINUOUS')
UPSET	'DASH'		Numeric value corresponding to current dashline specification (56.)
USET	'DESCRIPTION'		Current axis option ('TICAXES')
USET	'DETECTABILITY'		Detectability of new segments ('DESENSITIZED')
USET	'DIMENSION'	3D	2D or 3D coordinate terminator switch ('2DCOORDINATES')
UPSET	'DISTANCE'	3D	Distance from viewport to screen plane (150.)
USET	'EDIT'	3D	Data structure edit mode ('COMPRESS' or 'EXPAND')
USET	'ERRORMODE'		Error presentation (ERROR)
USET	'EXECUTE'	3D	Structure building visability ('EXECUTE')
USET	'EXISTENCE'		Current axis existence option ('XYZAXES')
UPSET	'FACTOR'		Scale factor for GCS created software symbols (1.)
USET	'FITMODE'		Curve fitting mode for autoplotting ('NOFITTING')
USET	'FNAMING MODE'		Frame naming mode ('FNAME')
UPSET	'FNTFILE NUMBER'*		Font file number (0.)

UPSET	'FONT NAME'*		Font name ('GCS')
USET	'FORMAT'		Text number format for numeric labelling ('BESTFORMAT')
USET	'GAPMODE'		Gapped line mode ('UINTERRUPTED')
UPSET	'GREYSCALE'		Numeric valve indicating current grey level (device dependent)
UPSET	'GRID'	3D	Numeric value indicating grid axis type option (0.)
USET	'HANDEDNESS'	3D	Left or right handed coordinate system ('RIGHTHANDED')
UPSET	'HARDWARE'		Current hardware character size ('SMALL', device dependent)
UPSET	'HORIZONTAL'		Current horizontal software character position size (5.)
UPSET	'INFILE'		Graphics input file designation (computer system dependent)
USET	'INPUT'		Graphics input device medium
USET	'ITALICIZATION'		Italicization mode ('NOITALICS')
UPSET	'LABELANGLE'		Label angle around perpendicular (0.)
USET	'LETTERTYPE'		Character type ('HARDWARE')
UPSET	'LEVEL'	3D	Subscript/superscript spacing level (1.)
UPSET	'LIBRARY'	3D	Data structure library file mode (0.)
UPSET	'LIMIT'	3D	Number of errors before automatic stop. 0 means no limit (0.)
USET	'LINEOPTION'		Current line option setting ('LNULL')
UPSET	'LOWERCASE'		Lowercase shift character as Hollerith string ('F')
USET	'LOGARITHMIC'	3D	Axes selected for logarithmic transform (logarithmic transform switch) ('NOLOGSCALING')
USET	'LOGTIME OF APPLICATION'		Time of application of logarithmic scaling ('LOGSYSTEM COORDINATES')
USET	'LOGTYPE'	3D	Flag indicating when logarithmic scaling performed. ('LOGSYSTEM', 'LOGUSER', 'LOGOBJECT')
USET	'MAPPINGTYPE'	3D	3D to 2D mapping type ('PERSPECTIVE')
USET	'MENUTYPE'	3D	Menu board type ('STANDARD')
USET	'MERGE'	3D	Structure merge mode switch ('IGNORE')
USET	'MESSAGEDEVICE'		Destination of alphanumeric I/O ('PLOTDEVICE')
USET	'MODE'		Current coordinate mode ('ABSOLUTE')
USET	'NUMERIC'	3D	Numeric labels parallel or perpendicular to axis ('PARALLEL')
USET	'ORDER'	3D	3D coordinate system rotation application order ('ZYX')
UPSET	'ORIENTATION'		Angular orientation of display of GCS created symbols (0.)

USET	'ORIGIN'		Forced origin switch for axis scaling ('ORIGIN')
UPSET	'OUTFILE'		File number of graphics output file (computer system dependent)
USET	'OUTPUTDEVICE'		Graphical output destination device ('ALLDEVICES')
USET	'PLANE'	3D	Axis label plane ('XYPLANE')
USET	'PLOTSCALE'		Scale option ('NEWSCALE')
UPSET	'POLYNOMIALDEGREE'		Current degree of polynominal fit for curve fitting (5.)
USET	'POSITION'	3D	Current axes positioning ('EDGEAXES')
UPSET	'PRECISION'		Number of digits of precision to be displayed for real numbers. (4.)
USET	'PROJECTION TYPE'		Type of projection ('PERSEPCTIVE')
UPSET	'READ FILE'		File designator for non-graphic input (computer system dependent)
USET	'REPEAT'	3D	Data structure invocation option ('SINGLE' or 'REPEAT')
USET	'REWIND'	3D	Structure file rewind mode ('REWIND')
USET	'SCALE'		Axis scale option ('AUTOSCALE')
USET	'SCRIPT'	3D	Subscript/superscript control ('NOSCRIPT')
USET	'SECURITY LEVEL'		Control of security banners ('UNSECURED')
USET	'SENSITIVITY'	3D	Light pen sensitivity switch ('DESENSITIZE')
USET	'SIDE'	3D	Side of axes on which labels will appear ('NEGATIVE')
USET	'SIZE'		Hardware character size ('SMALL')
UPSET	'SLANT'	3D	Software character italic slant angle (18. degrees)
USET	'SPACE'		Coordinate space ('VIRTUAL')
UPSET	'SPAN'	3D	Angular span for pie charts (360. degrees)
UPSET	'START'	3D	Pie chart starting angle (0.)
USET	'STOP'		Error stopping control ('NOABORT')
USET	'STORAGEMODE'		Segment/Frame retention mode ('RETAINED')
UPSET	'STRUCTURE LIMIT'		Maximum number of structures which can be defined (100.)
UPSET	'SUBSCRIPTCHARACTER'	3D	Current subscript shift character ( )
UPSET	'SUPERSCRIPTCHARACTER'	3D	Current superscript shift character ( )
UPSET	'SYMBOL INDEX'		Choice of marker (0.)
USET	'SYSTEM'		Current coordinate system ('SYSTEM')
UPSET	'TABHORIZONTAL'		Location of current horizontal tab stop (10.)
UPSET	'TABVERTICAL'		Location of current vertical tab stop (10.)
UPSET	'TERMINATOR'		GCS string termination character
USET	'TEXT'		Textual I/O mode ('TEXT')

UPSET	'TICINTERVAL'		Length of UPEN ticintervals (10.)
UPSET	'TICLENGTH'		Length of UPEN ticintervals (10.)
UPSET	'TICMINUS'		Clockwise tic mark size (0.05)
UPSET	'TICPLUS'		Counter-clockwise tick mark size (0.05)
UPSET	'TICX'		X axis tic interval (0.)
UPSET	'TICY'		Y axis tic interval (0.)
UPSET	'TICZ'	3D	Z axis tic interval (0.)
USET	'TIME'		Time series plotting period ('DATES')
USET	'TYPE'		Type of coordinates ('RECTANGULAR)
USET	'UNIFORM'	3D	High level plotting option ('NONUNIFORM')
USET	'UNITS'		Device space units ('INCHES')
USET	'UPAXIS'	3D	Coordinate system viewport vertical axis ('ZPOSITIVE')
UPSET	'UPPERCASE'		Uppercase shift character as Hollirith string ('A')
USET	'USER'		User coordinate system switch (7.)
UPSET	'VERTICAL'	3D	Vertical software character position switch
USET	'VIEWPORT'		Viewport distance base switch ('SITE')
USET	'VISIBILITY'		Framed/segment creation visibility mode ('VISIBLE')
UPSET	'WIDTH OF LINES'	3D	Width of lines (0.)
USET	'WINDOW'		Window type ('NOWINDOW')
UPSET	'WRITE FILE'		Non-graphic alphanumeric output file (Computer system dependent)
UPSET	'XBASE'		Base of log scaling along x-component (real number or string 'E') (10.)
USET	'XLABEL'		X axis labelling option ('XNUMERICLABEL')
USET	'XLOGARITHMIC'		X axis linearity option
UPSET	'XPERCENTAGE OF CHARACTER SPACE'		Portion of horizontal character space occupied by character (0.65)
USET	'XREPETITION MODE'		X component repetition mode for plotting ('NOXREPEAT')
UPSET	'XROTATION'		Rotation factor around X axis for structure invocation (0.)
UPSET	'XSCALING'		Scaling factor along X axis for structure invocation (1.)
UPSET	'XSPECIFICATION UNIT'		Number of XSPECIFICATION unts in device space (1000.)
USET	'XSIZE'		Horizontal hardware character position size (device dependent)
UPSET	'XTITLE'		X axis alphanumeric label ('X')
UPSET	'YBASE'	3D	Base of log scaling along Y-component (real number or string 'E') (10.)

USET	'YLABEL'		Y axis labelling option ('YNUMERICLABELS')
USET	'YLOGARITHMIC'		Y axis linearity option
UPSET	'YPERCENTAGE OF CHARACTER SPACE'		Portion of vertical character space occupied by character (0.65)
USET	'YREPETITION'		Y component repetition made for plotting ('NOYREPEAT')
UPSET	'YROTATION'		Rotation factor around Y axis for structure invocation (0.)
UPSET	'YSCALING'		Scaling factor along Y axis for structure invocation (1.)
UPSET	'YSIZE'		Vertical size for hardware characters (device dependent)
UPSET	'YSPECIFICATIONUNITS'		Number of Y specification units in device space (1000.)
UPSET	'YTITLE'		Y axis alphanumeric label
UPSET	'ZBASE'	3D	Base of log scaling along Z axis (10.)
USET	'ZCLIP'		Hither/yon clipping mode ('NOZCLIPPING')
USET	'ZLABEL'	3D	Z axis label option ('ZNUMERICLABELS')
USET	'ZREPETITION MODE'		Z component repetition mode for plotting ('NOZREPEAT')
UPSET	'ZROTATION'		Rotation factor around Z axis for structure invocation (0.)
UPSET	'ZSCALING'		Scaling factor along Z axis for structure invocation (1.)
UPSET	'ZSPECIFICATIONUNITS'		Number of Z specification units in device space (1000.)
UPSET	'ZTITLE'	3D	Z axis title
UPSET	'ZVALUE'	3D	Numeric default Z-value for 2D coordinates (0.)

Subroutine URAXIS

FUNCTION:

This routine draws a scaled and labeled set of axis. The position of the beam/pen when URAXIS is called will be the intersection point of the two axes or their extensions. The axis will be drawn from the minimum to maximum values furnished be the user.

CALLING SEQUENCE:

CALL URAXIS (XMIN,XMAX,YMIN,YMAX)

Where

XMIN is the minimum value to be displayed for the X axis in current user units. It also indicates the minimum RADIUS component component if polar coordinates are being used.

XMAX is the maximum value to be displayed for the X axis in current user units. It also indicates the maximum RADIUS component if polar coordinates are being used.

YMIN is the minimum value to be displayed for the Y axis in current user units. It also indicates the minimum THETA component if polar coordinates are being used.

YMAX is the maximum value to be displayed for the Y axis in current user units. It also indicates the maximum THETA component if polar coordinates are being used.

OPTIONS which may apply:

Coordinate System Options: 'RECTANGULAR', 'POLAR', or 'LOGARITHMIC', or 'LOGARITHMIC'

Coordinate Type Options: 'ABSOLUTE' or 'RELATIVE'

Axis Existence Options: 'XYAXES', 'XAXIS', 'YAXIS', or 'NOAXES'

Axis Drawing Options: 'TICAXES', 'GRIDAXES', or 'PLAINAXES'

X Axis Type Options (Rectangluar): 'LINXAXIS', 'LOGXAXIS', or 'LNXAXIS'

Y Axis Type Options (Rectangular): 'LINYAXIS', 'LOGYAXIS', or 'LNYAXIS'

X Axis Labeling Options: 'XNUMERICLABEL', 'XALPHANUMERICLABEL', 'XBOTHLABEL', 'NOXLABEL'

Y Axis Labeling Options: 'YNUMERICLABEL', 'YALPHANUMERICLABEL', 'YBOTHLABEL', or 'NOYLABEL'

Numeric Scaling Format Options: 'BESTFORMAT', 'IFORMAT', or 'GFORMAT'

PARAMETERS which may be set:

'TICX' — The interval between tic marks or grid lines on the X axis in current user units (Rectangular, Linear).
The interval between radial tic marks or grid lines in current user units (Polar).

'TICY' — The interval between tic marks or grid lines on the Y axis in current user units (Rectangular, Linear).
The interval between axial grid lines in current user units (Polar).

'XLABEL' — The X axis alphahbetic iabel

'YLABEL' — The Y axis alphabetic label

'PRECISION' — The number of digits of precision for the numeric labels

COMMENTS:

The default options for URAXIS are the first options listed in each option type. Unless specified by the user, the tic or grid intervals will be determined automatically by URAXIS. The default for PRECISION is four digirs of precision.

This routine operates entirely in the current user coordinate system using the user's units. The vertical position of the X axis will be determined by the vertical position of the beam/pen when URAXIS is called. Similarly, the horizontal position of the Y axis will be determined by the horizontal position of the beam/pen. Normally, the numeric and alphabetic labels will be below the X axis and to the left of the Y axis. The X axis labels will move above the X axis if the beam/pen is positioned at or above YMAX, and the Y axis labels will move to the right of the Y axis if the beam/pen is positioned at or to the right of XMAX. The user is cautioned not to position the beam/pen on the boundary of the virtual window, but to leave sufficient room for the labels by positioning the beam/pen well inside the range of the URAXIS arguments, or by making the virtual window boundaries large enough so that room will exist outside the range of the URAXIS arguments.

If POLAR is specified, a check will be made on the arguments to see if a first quadrant or a full four quadrant set of axes is needed. The user should keep this in mind and make sure there is sufficient room for a four quadrant set of axes if needed. If the user is in an ABSOLUTE coordinate system, the axes will be centered at (0.,0.) regardless of the beam/pen position. If the user is in a RELATIVE coordinate system, the axes will be centered on the beam/pen position. The labels will be to the extreme left and bottom of the axes in both the one quadrant and four quadrant case.

Programming Notes:

1. This obsolete subroutine has been replaced by USAXIS. The subroutine description is being included in order to aid in the upgrading of a program already written which used an earlier version of GCS.

Subroutine UREAD *Interactive*

FUNCTION:

This routine enables the user to read character data entered from the alphanumeric keyboard of the terminal. The data will be formatted and returned to the user according to one of five options. The options are: 'TEXT', 'REALNUMBER', 'INTEGER', 'XYCOORDINATES', or 'XYZCOORDINATES' (3D only).

CALLING SEQUENCE:

CALL UREAD (X,Y,DATA,COUNT,FLAG)

Where

X — **is the X or RADIUS coordinate of the lower left corner of the point at which the input will begin, if possible.**

Y — **is the Y or THETA coordinate of lower left corner of the point at which the input will begin, if possible.**

DATA — **is a single variable or array to contain either real numbers or a GCS text string. The size of DATA is variable.**

COUNT — **is a single variable. If the user has specified 'TEXT' input, then COUNT in the maximum number of characters returned in DATA. The size of DATA must be large enough to contain COUNT characters. The characters are returned in Hollerith format, left justified and blank filled. If the terminal operator inputs fewer characters than requested in COUNT, then DATA will be blank filled to the required size. If the user has specified 'REAL' or 'INTEGER', then the value of COUNT is the number of variables to be input. For 'XYCOORDINATES', the value of COUNT is the number of coordinate pairs to be read. For 'XYZCOORDINATES', the value of COUNT is the number of coordinate triplets to be read.**

FLAG — **is a single variable. The value of FLAG will be zero if the terminal operator has made an error in entering the numeric data. In this case, the value of DATA is undefined. If the required input was correct, then the value of FLAG is the number of elements in DATA, or the number of characters that the terminal operator entered.**

OPTIONS which may apply:

Coordinate System Options: See UPEN and UCOSYS
ALPHANUMERIC Options: '**TEXT**', 'INTEGER', 'REALNUMBER', 'XYCOORDINATE'
Device Switching Options: 'MESSAGEDEVICE', 'PLOTDEVICE'

A. **TEXT** — Under this default option, UREAD will accept and store COUNT characters into DATA; hence, the user must insure that DATA has been suitably dimensioned to hold the number of characters which have been requested. Should fewer than COUNT characters be entered as input, UREAD will store the actual number of characters entered into FLAG, and blank-fill the remaining (COUNT - FLAG) characters of DATA. It should be noted that UREAD does not append the termination character to the end of the input. Therefore, the user is responsible for inserting this character via UAPEND if the string is to be passed as a parameter to UPRINT or UWRITE.

B. **REALNUMBER** — This option directs UREAD to edit the alphanumeric input as a REAL number, and to store the resulting floating-point number into the single-valued REAL parameter DATA. Should UREAD encounter any illegal characters during the edit, FLAG will be returned with a negative value, and DATA will be undefined.

C. **INTEGER** — As in the case of REALNUMBER, DATA is assumed to be single-valued and of type REAL. UREAD will edit the alphanumeric input as an INTEGER, perform an INTEGER to REAL conversion, and store the result into DATA. The user may check if the operation was successfully performed by examining FLAG upon return from UREAD.

D. **XYCOORDINATES** — This option directs UREAD to accept two REAL numbers (separated by a comma) as input and to store them into the two element REAL array, DATA. FLAG is set to reflect the status of the input and editing operation.

E. **XYZCOORDINATES** — This option directs UREAD to accept three REAL numbers (separated by commas) as input, and store them into the three element REAL array DATA. FLAG is set to reflect the status of the input and editing system.

COMMENTS:

The input begins at the current alphanumeric cursor position. For those devices which have alphanumeric capability on the PLOT device, then the alphanumeric cursor position is the current beam position. Otherwise, the input will be from the MESSAGE device. The maximum number of characters retrieved by UREAD is unlimited, however, the number of characters which may be input by the terminal operator may be limited. The user should consult the proper system manual to determine what limits exist, if any, for a particular terminal and system.

Programming Notes:

Subroutine URECT

FUNCTION:

This routine draws a rectangle which spans from the current beam position to the diagonally opposite corner.

CALLING SEQUENCE:

CALL URECT (X,Y)

Where

X	**is the X- or RADIUS component of the specified endpoint in current user units.**
Y	**is the Y- or THETA component of the specified endpoint in current user units.**

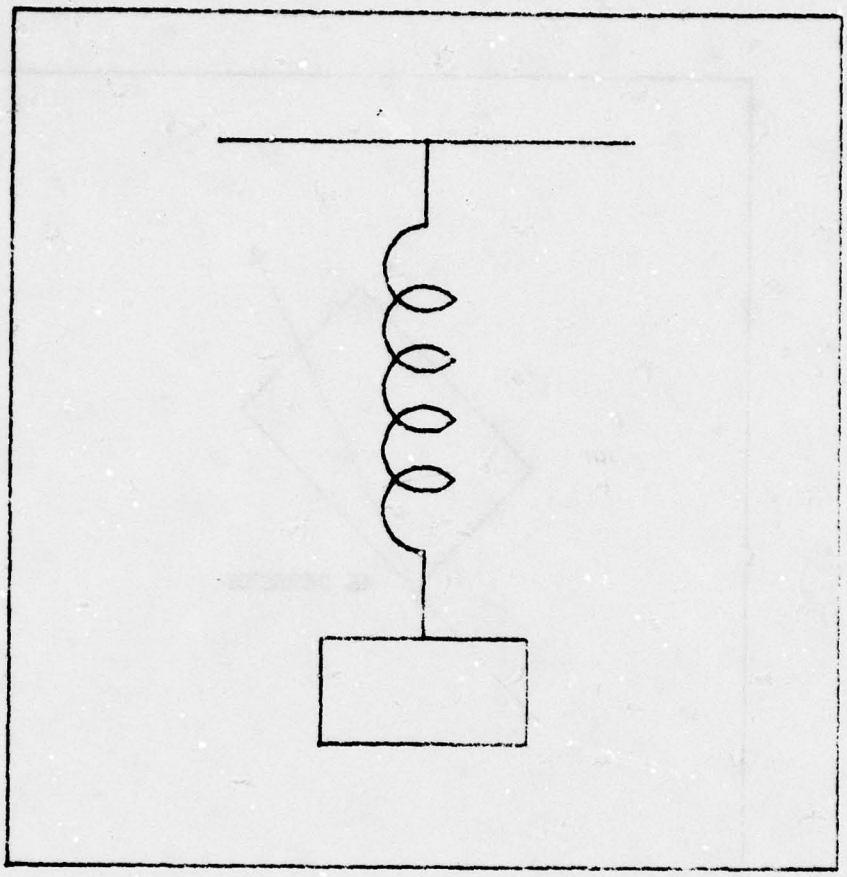
OPTIONS which may apply:

All Line Options
All Pen Coordinate Options
UPSET ('ORIENT', ANGLE)

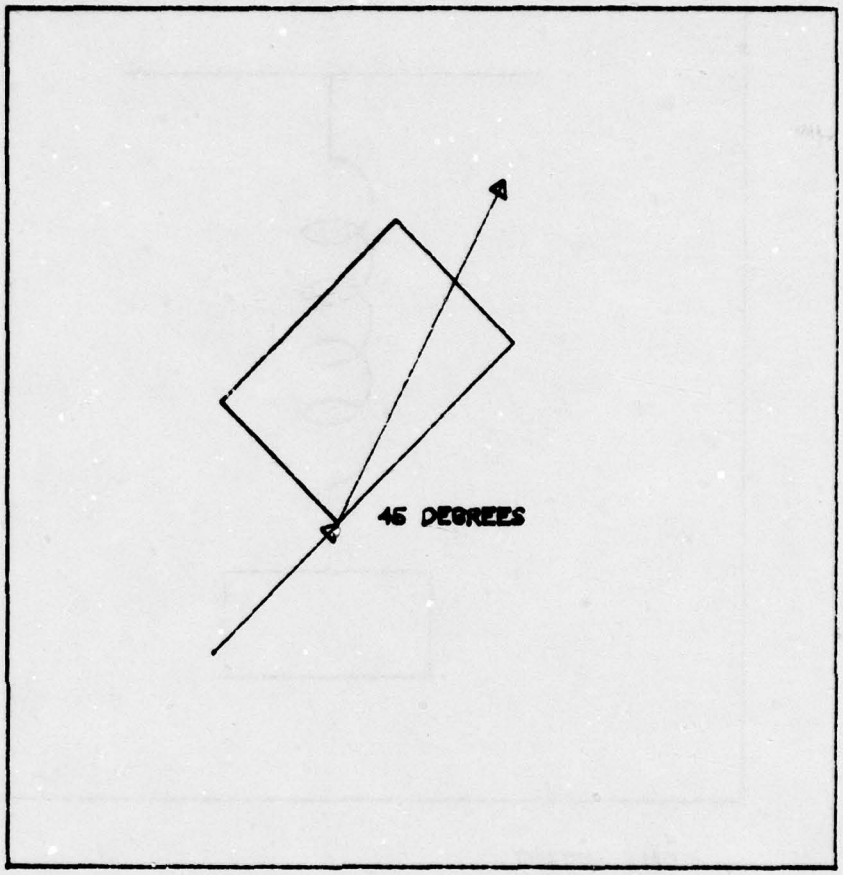
COMMENTS:

The orientation factor specifies the amount of rotation to be applied to the rectangle. Note that the orientation factor applies to the entire rectangle, and not the endpoint which determines the rectangle.

Programming Notes:



```
      CALL USTART
      CALL UOUTLN
      CALL SPRMAS
      CALL UEND
      STOP
      END
      SUBROUTINE SPRMAS
      CALL USET ('RELATIVE')
      CALL UMOVE (25.0,65.0)
      CALL UPEN (50.0,0.0)
      CALL UMOVE (-25.0,0.0)
      CALL UPEN (0.0,-10.0)
      CALL UARC (0.0,-5.0,225.0)
      DO 1 I = 1, 3
      CALL UARC (-3.535,-3.535,270.0)
    1 CONTINUE
      CALL UARC (-3.535,-3.535,225.0)
      CALL UPEN (0.0,-10.0)
      CALL UMOVE (-12.5,-12.5)
      CALL URECT (25.0,12.5)
      RETURN
      END
```




```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('EXTRALARGE')
CALL UOUTLN
CALL UMOVE (25.,25.)
CALL USET ('ARROW')
CALL UPEN (40.,40.)
CALL UPSET ('ORIENTATION',45.)
CALL USET ('LNULL')
CALL URECT (70.,60.)
CALL USET ('ARROW')
CALL UPEN (60.,60.)
CALL UWRITE (45.,40.,'45 DEGREES,')
CALL UEND
STOP
END
```

Subroutine UREPRO

FUNCTION:

This routine reproduces the contents of the provided pseudo-device file on the current device configuration.

CALLING SEQUENCE:

CALL UREPRO (FILENR,STATUS)

where

FILENR — is the Fortran file number of a file generated in the pseudo-device file format.

STATUS — is a value returned indicating the type of completion from UREPRO. Valid codes are:

1 -- End of Page
2 -- End of File
3 -- End of all Files
4 -- Error

OPTIONS which may apply:

Page Advancement: **'ADVANCE'**, 'NOADVANCE'
Reproduction Extent: 'PAGE', **'FILE'**, 'ALLFILES'.

COMMENTS:

This routine provides the mechanisms whereby graphics output generated in previous runs on to the pseudo-device may be post-processed to an actual device. The reproduction extent mode allows a whole queue to be processed ('ALLFILES'), a single file ('FILE'), or a single page ('PAGE'). Whenever control is returned from UREPRO, the current beam/pen position will be at the last location specified on the file just processed. If 'PAGE' was specified, the frame advance will occur whenever UREPRO is called again. This allows additional graphics operations on the page just processed.

The calling program can retain control of page advances by selecting 'NOADVANCE'. However, not that this will result in overlapping frames if more than one page is defined in a file and 'FILE' is specified. Similarly, frame overlap will occur if 'ALLFILE' is specified and more than one file is to be processed. A non-specified page advance will occur at the beginning of the second and following files if 'ALLFILES' is specified. This prevents overlapping the last frame of one file with the first frame of the next.

UREPRO can reproduce any file created using the pseudo-device file format even though the file was generated from some other graphics systems output from GCS can thus be merged with other graphics packages.

Programming Notes:

Subroutine URESET

FUNCTION:

To reinitialize the Graphics Status Area (GSA) to the default condition.

CALLING SEQUENCE:

```
CALL URESET
```

OPTIONS which may apply:

No options apply.

Programming Notes:

Subroutine UROTAT

FUNCTION:

A user coordinate system is composed at the current beam position and rotated according to the rotation factor specified in the argument.

CALLING SEQUENCE:

CALL UROTAT (ANGLE)

Where

ANGLE **is the rotation factor applied to the new coordinate system which is composed at the current beam position. Angle is in current angular units.**

OPTIONS which may apply:

WORKINGAXIS
REFERENCEAXIS

COMMENTS:

The invocation of this subroutine causes the creation of a new origin (0.,0.) at the current beam position. The unit scale factors are assumed to be equal to one. For a detailed discussion of the GCS coordinate system facility, refer to UCOSYS.

Programming Notes:

Subroutine USAREA

FUNCTION:

This subroutine avoids distortion by changing the device boundaries to maintain a 1 to 1 ratio with the current window boundaries.

CALLING SEQUENCE:

CALL USAREA

OPTIONS which may apply:

No options apply

COMMENTS:

The resulting device boundaries can be found by CALL USTUD(ARRAY)

Programming Notes:

Subroutine USAVE

FUNCTION:

This routine saves the entire contents of the Graphics Status Area (GSA) in an array furnished by the user. This array will be suitable for later use as input to UNSAVE to restore the status area to the current setting.

CALLING SEQUENCE:

CALL USAVE (SARRAY)

Where

SARRAY	**is an array large enough to contain the variables to be saved from the status area.**

OPTIONS which may apply:

No options apply.

COMMENTS:

The size of the array necessary to save the GSA is dependent upon the particular version of GCS being used. For the two dimensional version, SARRAY should be 300 words (decimal) long; *for the three dimensional version SARRAY should be 2100 (decimal) words long.*

Programming Notes:

161

FUNCTION:

This routine is the basic axis creation routine. It draws a single axis from the designated point for the distance specified parallel to the axis specified.

CALLING SEQUENCE:

CALL USAXIS(AXIS,XSTART,YSTART,ZSTART,DIST)

Where

AXIS specifies the coordinate system axis which is parallel to the desired axis. Its values are as follows:

'XAXIS': axis will be parallel to coordinate system X axis
'YAXIS': axis will be parallel to coordinate system Y axis
'ZAXIS': axis will be parallel to coordinate system Z axis

XSTART, YSTART, ZSTART are the coordinates of the starting point of the axis to be drawn

DIST specifies the length of the axis. A positive value will cause the axis to extend in the positive direction. A negative value will cause the axis to extend in the negative direction.

OPTIONS which may apply:

X Axis Labeling Options:	'NOXLABELS', **'XNUMERICLABELS'**, 'XALPHABETICALABEL', 'XBOTHLABLES'
Y Axis Labeling Options:	'NOYLABELS', **'YNUMERICLABELS'**, 'YALPHABETICLABELS', 'YBOTHLABELS'
Z Axis Labeling Options:	'NOZLABELS', **'ZNUMERICLABELS'**, 'ZALPHABETICLABELS', 'ZBOTHLABELS'
Alphabetic Label Specification:	UPSET('XLABEL', GCS character string) UPSET('YLABEL', GCS character string) UPSET('ZLABEL', GCS character string)
Numeric Label Format Options:	**'BESTFORMAT,'** IFORMAT', 'GFORMAT'
Axis Description Options:	'PLAINAXES', **'TICAXES'**
Tic Interval Specifications:	UPSET('TICX',value) UPSET('TICY',value) UPSET('TICZ',value)
Tic Size Specifications:	UPSET('TICPLUSSIZE', value) UPSET('TICMINUSSIZE', value)
Log Scaling Options:	**'NOLOG'**, 'XLOG', 'YLOG', 'ZLOG', 'XYLOG', 'XZLOG', 'YZLOG', 'XYZLOG', 'LOGARITHMS'

Log Base Specifications:	UPSET('BASE', value) UPSET('XBASE', value) UPSET('YBASE', value) UPSET('ZBASE' value)
Axis Label Orientation Options:	'PARALLELABELS', **'PERPENDICULARABELS'**
Axis Label Position Options:	'POSITIVESIDE', **'NEGATIVESIDE'**
Axis Label Plane Options:	**'XYPLANE',** 'XZPLANE', 'YZPLANE'
Label Rotation Option:	UPSET('LABELROTATION', value)

COMMENTS:

The axis label options provide for producing numbers which correspond to the tic marks (numeric labels) and for producing a description of the axis (alphabeticlabels). These labels are individually selectable for each axis.

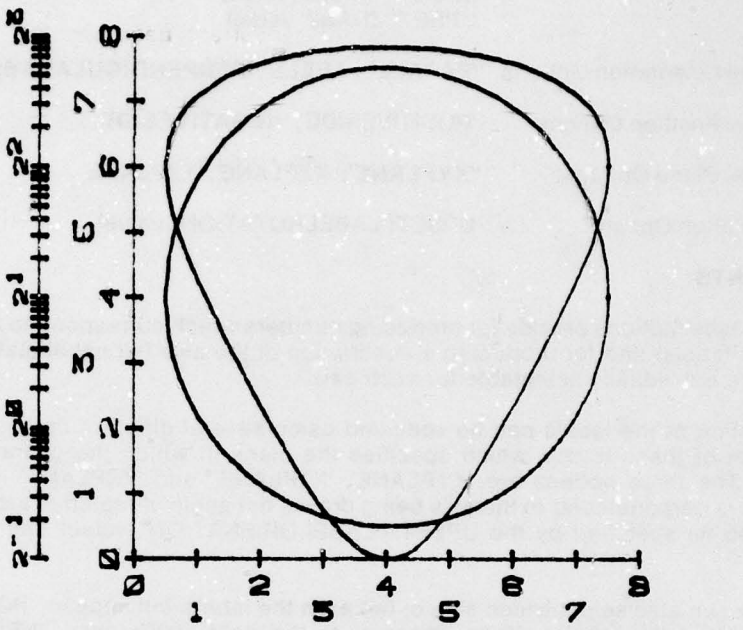
The location of the labels can be specified using several different options. The most signficant of these is that which specifies the plane in which the characters will be drawn. The three options are 'XYPLANE', 'XZPLANE', and 'YZPLANE'. If the plane selected is perpendicular to the axis being drawn, the angle of roatation around the axis must also be specified by the UPSET ('LABELORIENATION', value) call (default is 0 degrees).

The user can also select which side of the axes the labels will appear. 'POSITIVESIDE' specifies that the labels will be above or to the right of the axis. 'NEGATIVESIDE' specifies that the labels are to the left or below the axis. Default is 'NEGATIVESIDE'.

The orientation of the labels can be specified by either the 'PARALLEL' or 'PERPENDICULAR' USET options. PARALLEL specifies that the major axis of the numeric labels (their lengths) will be along the axis. 'PERPENDICULAR' specifies that the major axis of the numeric labels will be perpendicular to the axis. Alpha labels always have their major axis parallel to the axis being drawn. It should be noted that if hardware characters are to be used, vertical spacing may be necessary to maintain the correct label orientation. In some instances, it may even be necessary to produce labels at some oblique angles. Software character labels will, however, always be produced in the proper orientation.

USAXIS will always draw the requested axis in the current coordinate system and space. No attempt is made to insure that the axis is visible. Also no scaling is performed although a tic interval may be calculated if not specified.

Programming Notes:




```
      CALL USTART
      CALL UPSET ('TERMINATOR',';')
      CALL USET ('SOFTWARE CHARACTERS')
      CALL UPSET ('VERTICAL',.4)
      CALL UPSET ('HORIZONTAL',.4)
      CALL UWINDO (-3.,10.,-3.,10.)
      CALL UPSET ('TICX',1.)
      CALL UPSET ('TICY',1.)
      CALL UPSET ('XLABEL','LOGGED AND NONLOGGED CIRCLES;')
      CALL USET ('XBOTHLABELS')
      CALL USAXIS ('XAXIS',0.,0.,0.,8.)
      CALL USAXIS ('YAXIS',0.,0.,0.,8.)
      CALL UCRCLE (4.,4.,3.5)
      CALL USCSYS (0.,2.,0.,1.,2.,1.,0.,0.,0.)
      CALL UPSET ('HORIZONTAL',.15)
      CALL UPSET ('VERTICAL',.3)
      CALL UPSET ('BASE',2.)
      CALL USET ('YLOG')
      CALL USET ('LOGUSER')
      CALL USAXIS ('YAXIS',-1.5,.5,0.,7.5)
      CALL UCRCLE (4.,4.,3.5)
      CALL UEND
      STOP
      END
```

Subroutine USCALE

FUNCTION:

A user coordinate system is composed at the current beam position and scaled according to the scale factors specified in the argument.

CALLING SEQUENCE:

CALL USCALE(SCLX,SCLY)

Where

SCLX	**is the ratio of a unit length in the current X direction to the length of a unit length in the new X direction.**
SCLY	**is the ratio of a unit length in the current Y direction to the length of a unit length in the new Y direction.**

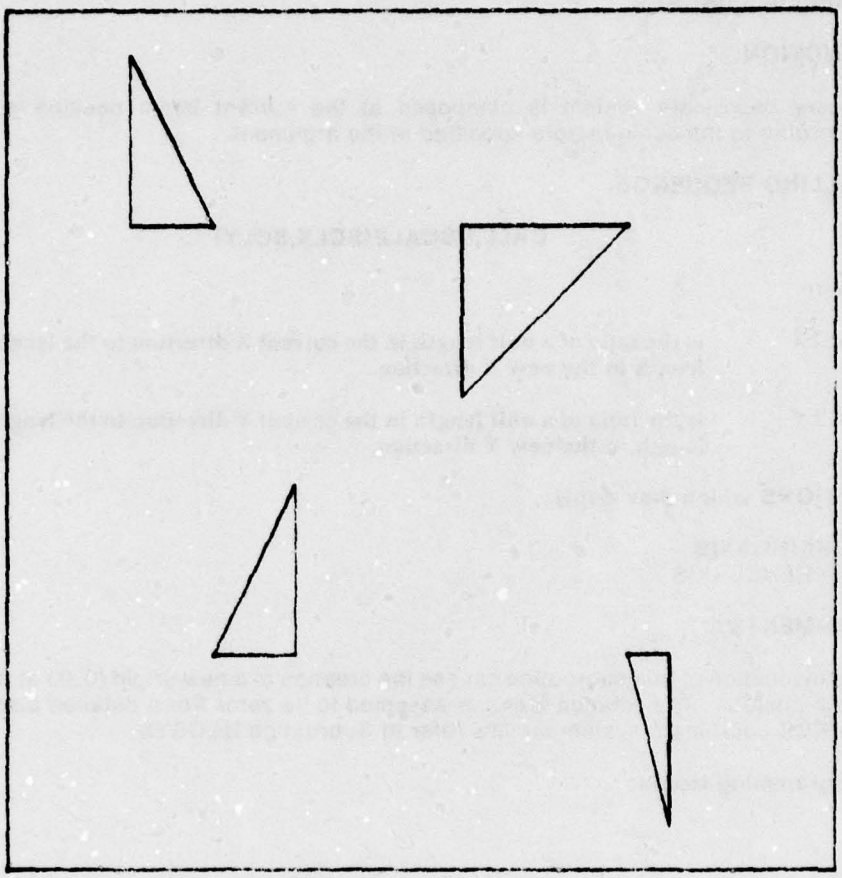
OPTIONS which may apply:

WORKINGAXIS
REFERENCEAXIS

COMMENTS:

The invocation of this subroutine causes the creation of a new origin (0.,0.) at the current beam position. The rotation factor is assumed to be zero. For a detailed discussion of the GCS coordinate system facility, refer to Subroutine UCOSYS.

Programming Notes:



```
CALL USTART
CALL UOUTLN
CALL UMOVE (25.,25.)
CALL USCALE (1.,1.)
CALL FIGURE
CALL USET ('SYSTEMAXIS')
CALL UMOVE (25.,75.)
CALL USCALE (-1.,1.)
CALL FIGURE
CALL USET ('SYSTEMAXIS')
CALL UMOVE (75.,25.)
CALL USCALE (.5,-1.)
CALL FIGURE
CALL USET ('SYSTEMAXIS')
CALL UMOVE (75.,75.)
CALL USCALE (-2.,-1.)
CALL FIGURE
CALL UEND
STOP
END
SUBROUTINE FIGURE
CALL UPEN (10.,0.)
CALL UPEN (10.,20.)
CALL UPEN (0.,0.)
RETURN
END
```

Subroutine USCATR

FUNCTION:

This routine draws a scatter plot from the user input data, drawing axes as specified.

CALLING SEQUENCE:

CALL USCATR(X,Y,PTS)

Where

X	is a real array of X or RADIUS coordinates for the points in the diagram
Y	is a real array of Y or THETA coordinates for the points in the diagram
PTS	is a real constant or variable specifying the number of points in the diagram

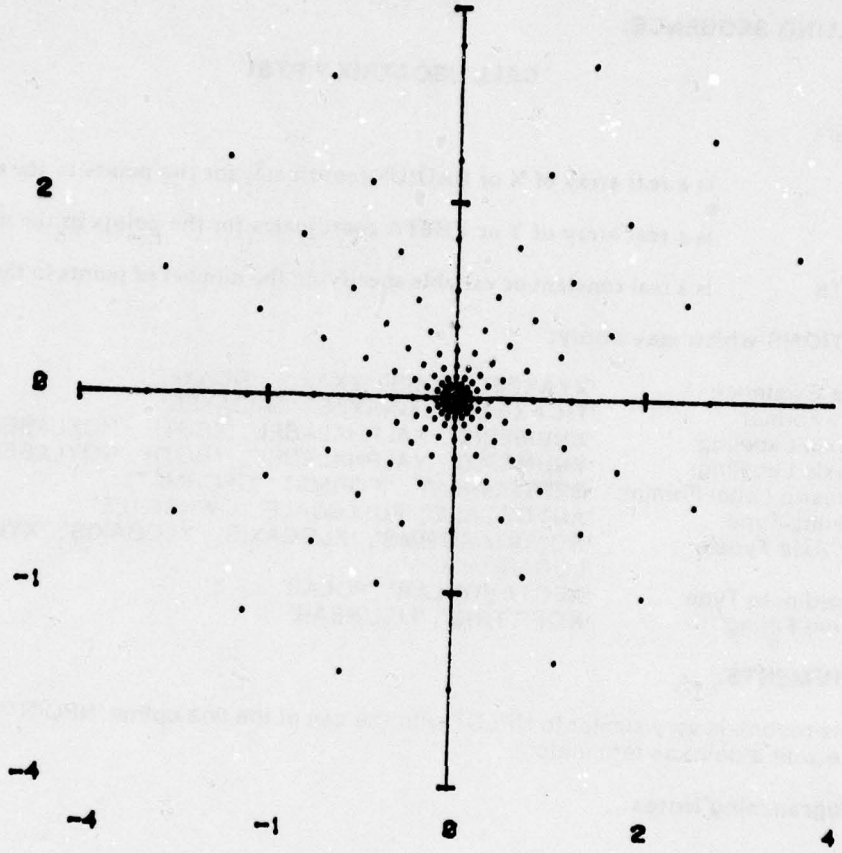
OPTIONS which may apply:

Axis Existence:	**'XYAXES',** 'XAXIS', 'YXAXIS', 'NOAXES'
Axis Format:	**'TICAXES',** 'PLAINAXES', 'GRIDAXES'
X Axis Labeling:	**'XNUMERIC',** 'XALPHALABEL', 'XBOTH', 'NOXLABEL'
Y Axis Labeling:	**'YNUMERIC',** 'YALPHALABEL', 'YBOTH', 'NOYLABEL'
Numeric Label Format:	**'BESTFORMAT',** 'IFORMAT', 'GFORMAT'
Scaling Type:	**'AUTOSCALE',** 'FULLSCALE', 'OWNSCALE'
X,Y Axis Types:	**'NOLOGARITHMS',** 'XLOGAXIS', 'YLOGAXIS', 'XYLOGAXIS', 'LOGARITHM'
Coordinate Type:	**'RECTANGULAR',** 'POLAR'
Curve Fitting'	**'NOFITTING',** 'FITLINEAR'

COMMENTS:

This routine is very similar to UPLOT with the use of the line option 'NPOINT', i.e., a null line with a point as terminator.

Programming Notes:




```
      DIMENSION X(182), Y(182)
      DATA PIBY8/0.3926990822/
      CALL USTART
      CALL USET ('POLAR')
      CALL USET ('RADIANS')
      DO 10 I = 1, 182
      THETA = PIBY8 * FLOAT(I)
      X(I) = 4. * EXP(-THETA/16.)
   10 Y(I) = THETA
      CALL USCATR (X,Y,182.)
      CALL UEND
      STOP
      END
```

Subroutine USET

FUNCTION:

This routine allows the user to set options in the Graphics Status Area (GSA). Options will remain set until changed by a subsequent USET or UPSET call or until the GSA is restored during an UNSAVE or UNSVPN call.

CALLING SEQUENCE:

CALL USET (OPTION)

Where

OPTION **is the mnemonic option name which uniquely identifies the option to be set, or is the first four characters of such an option name.**

OPTIONS which may apply:

No options apply.

COMMENTS:

In the following table, those options flagged by the notation '3D' are contained within only the three dimensional version of GCS.

The notation 'I' indicates that these options pertain to a system having an interactive capability. On passive graphics systems, the options have no effect.

Programming Notes:

USET OPTIONS BY ALPHABETICAL ORDER:
(Default Options are in Bold Type)

Option Name		To Request
'AARROW'		Alphanumeric lines with arrow terminators
'ABACKARROW'		Alphanumeric lines with back arrow terminators
'ABEND'	3D,*	To halt execution when error count exceeds certain limit
'ABORT'	3D	To halt execution when error count exceeds certain limit
'ABSb'		Plotting in an absolute coordinate system
'ABSOLUTE'		Plotting in an absolute coordinate system
'ABUTTING'		Abutting of display surface pages
'ACENTER'		To center character output about given location
'ACHARACTER'		Alphanumeric lines with character terminators
'ACOORDINATE'		Alphanumeric lines with ending coordinates indicated
'ADDITIVE'	*	Additive color blending mode
'ADOUBLEARROW'		Alphanumeric lines with arrowhead terminators
'ALLDISPLAYS'		Routing of graphical output to all devices
'ALPHANUMERIC'		Alphanumeric lines with no terminators
'ALTERNATEDISPLAY'		Routing of graphical output to first alternate device
'ANNUAL'		Time series axis scale in yearly intervals
'ANULL'		Alphanumeric lines with no terminators
'APOINT'		Alphanumeric lines with point terminators
'ARROWLINE'		Solid lines with arrow terminators
'ASYMBOL'		Alphanumeric lines with character terminators
'AUTOSCALE'		Automatic scaling for higher level graphing
'BACKARROWLINE'		Solid lines with back arrow terminators
'BALL'		Track ball graphical input
'BLACK'		Background color to be black
'BBLUE'		Background color to be blue
'BCYAN'		Background color to be cyan
'BESTFORMAT'		Numeric label output in best possible format
'BIHOURLY'		Time series axis scale in two hour intervals
'BLACK'		Switch to pen color black
'BLUE'		Switch to pen color blue
'BMAGENTA'		Background color to be magenta
'BRED'		Background color to be red
'BRIGHT'		Highest possible intensity for output
'BUILD'	3D	Structure building
'BWHITE'		Background color to be white
'BYELLOW'		Background color to be yellow
'CENTIMETERS'		Device space coordinates are in centimeters
'CHARACTER'		Null or invisible lines with character terminators
'CJUSTIFICATION'		Alphanumeric center justification
'COMPRESSED'*	3D	Data structure editing option
'CONTINUOUS'		Curved lines be interpreted as one pen operation
'COORDINATES'		UPRINT/UWRITE output in (X,Y) coordinate format
'CURSOR'		Graphic cursor as graphic input device
'CWINDOWING'	3D	Circular windowing
'CYAN'		Switch to pen color cyan
'CYLINDRICAL'		Coordinates are to be of the form (R,THETA,Z), where R is the number of units of radius in the X,Y plane, THETA is the number of angular units

		around the Y axis, and Z is the number of units along the Z axis
'DAILY'		Time series axis scale in daily intervals
'DARROW'		Dashed lines with arrow terminators
'DASH'		Dashed lines with null terminations
'DATE'		Time series axis scale in date series interval
'DBACKARROW'		Dashed lines with arrow terminators
'DCHARACTER'		Dashed lines with character terminators
'DCOORDINATES'		Dashed lines with endpoint coordinates indicated
'DDOUBLEARROW'		Dashed lines with double arrow terminators
'DEFERRED'	*	Deferred error output until UEND is called
'DEGREES'		Angular information be interpreted in degrees
'DESENSITIZE'	I	Disabling of pick sensitivity
'DETECTABLE'		Enabling of pick sensitivity
'DEVICE'		Plotting in device space
'DIGITIZER'		Digitizer is graphics input device
'DIMb'		Lowest possible intensity for output
'DIMENSIONLINE'		Solid lines with arrow terminators
'DISPLAY'		Plotting in device space
'DNULL'		Dashed lines with no terminators
'DOUBLEARROW'		Solid lines with double arrow terminators
'DPOINT'		Dashed lines with point terminators
'DSYMBOL'		Dashed lines with character terminators
'DUMP'		To select dump option
'ECHO'*	I	Echo alphanumeric input option
'EDGEAXIS'		X and Y axis labels at edge of graph
'ERROROUTPUT'		Immediate error output
'EXECUTE'	3D	Execution of data structure commands as they are built
'EXPANDED'*	3D	Data structure editing option
'EXTENDEDMENU'*	I,3D	Extended menuing option
'EXTRALARGE'		Extra large character size
'FAST'		Fast blink rate
'FITLINEAR'		Fit linear function to plotted lines
'FITPOLYNOMIAL'		Fit least squares polynomial to plotted points
'FITSPLINE'		Fit cubic spline curve to plotted points
'FNUMBERMODE'		Frame identifiers provided as numbers
'FONTUNITS'		To indicate device spage plotting in font units
'FULLSCALE'		Full scaling for higher level graphing
'FUNCTIONKEYS'		Function key is graphics input device
'GAPPED'		Alternate light and dark line output
'GFORMAT'		Numeric label output in FORTRAN real (E or F) format
'GOTHIC'		Gothic character font
'GRADS'	3D	Angular units to be measured in grads
'GREEN'		Switch to pen color green
'GRIDAXIS'		Grid axes for higher level graphing
'HARDWAREFONT'		Output of hardware generated characters
'HIGHLIGHTED'		Highlighted segments
'HITHER/YONbCLIPPING'		Z axis clipping
'HORIZONTAL'	3D	Alphanumeric output to be printed horizontally
'HOURLY'		Time series axis scale in twenty-four hourly intervals
'IFORMAT'		Numeric label output in integer format
'IGNORE'	3D	Ignore duplicate studies on merge file
'INCHES'		Device space coordinates in inches
'INCREMENTAL'		Plotting in an incremental coordinate system

'INTEGER'		UPRINT/UWRITE output in integer format
'INVISIBLE'		Invisible construction of frames and segments
'ITALICS'		Italic character format
'JOYSTICK'	I	Joystick as graphic input device
'KEYBOARD'	I	Keyboard as (pseudo) graphic input device
'LARGE'		Large character size
'LARROW'		Solid lines with arrow terminators
'LBACKARROW'		Solid lines with back arrow terminators
'LCHARACTER'		Solid lines with character terminators
'LCOORDINATES'		Solid lines with endpoint coordinates indicated
'LDOUBLEARROW'		Solid lines with double arrow terminators
'LEFT'	3D	Left handed coordinate system
'LETTER'		UPRINT/UWRITE output in text format
'LIGHTPEN'	I	Light pen as graphic input device
'LINE'		Solid lines with null terminators
'LJUSTIFICATION'		Alphanumeric left justification
'LNULL'		Solid lines with null terminators
'LOGARITHMIC'		Applies logarithmic transforms to all components
'LOGOBJECT'	3D	Logarithmic transforms are to be applied before any other transformations. (In this mode, log transforms may be applied to angle or radius components of 'CYLINDRICAL', 'POLAR', or 'SPHERICAL' coordinates)
'LOGORIGINALUNITS'		Application of log scaling before conversion to rectangular
'LOGSYSTEM'	3D	Logarithmic to be applied after conversion to 'SYSTEM' coordinates (in this mode, the logarithmic coordinate system axes.)
'LOGUSER'	3D	Logarithmic transforms are to be applied after conversion to 'ABSOLUTE', 'RECTANGULAR', 'USER' coordinates but before conversion to 'SYSTEM' coordinates. (In this mode, the logarithmic scaling will be applied along the current 'USER' coordinate system axes.)
'LOWERCASE'		Lower case to be 'TEXT' case
'LPOINT'		Solid lines with point terminators
'LSYMBOL'		Solid lines with character terminators
'MAGENTA'		Switch to pen color magenta
'MEDIUM'		Medium character size
'MESSAGEDEVICE'		Alphanumeric I/O routed to a message device
'MILS'	3D	Angular units to be measured in mils
'MINUTELY'		Time series axis scale in minute intervals
'MONTHLY'		Time series axis scale in monthly intervals
'MOUSE'	I	Analog mouse as graphic input device
'MOVE'		Invisible lines with null terminators
'MULTIPLE'	3D	Multiple data structure invocation
'NARROW'		Invisible lines with arrow terminators
'NBACKARROW'		Invisible lines with back arrow terminators
'NCHARACTER'		Invisible lines with character terminators
'NCOORDINATES'		Invisible lines with double arrow terminators
'NDOUBLEARROW'		Invisible lines with double arrow terminators
'NEGATIVESIDE'	3D	Labels will be to the left or below the axes
'NEWSCALE'		New scale for higher level graphing
'NNULL'		Invisible lines with null terminators
'NObLINE'		Invisible lines with null terminators
'NOABORT'	3D	Do not terminate if error count exceeds specified limit

'NOAXES'		No axes be drawn for higher level graphing
'NOBLINK'		No blinking to occur
'NOBUILD'	3D	No structure building
'NOCENTER'		Text output starts at given point
'NODUMP'	3D	No dump performed
'NOECHO'	I,3D	No echoing of alphanumeric input
'NOEXECUTE'	3D	No execution of data structure commands as they are built
'NOFIT'		No curve fitting for higher level graphics
'NOHIGHLIGHTING'		No segment highlighting
'NOITALICS'		No slanting of software characters
'NOLINE'		Invisible lines with null terminators
'NOLOGARITHMS'	3D	To remove logorithmic transform application
'NONUNIFORM'	3D	High level plotting option
'NOMARK'		Invisible marks with null terminators
'NONABUTTING'		No abutting of display surface pages
'NONRETAINEDSEGMENTS'		Create segments in non-retained form
'NONUNIFORM'		Nonuniform scaling of higher level grading
'NOORIGIN'	3D	No origin to be forced for 'AUTOSCALE' or 'FULLSCALE' scaling options
'NOREPEAT'	3D	No coordinate repeating for high level plotting
'NOREWIND'		No rewind of structure save files
'NORMALINTENSITY'		Normal intensity for output
'NOSCRIPT'	3D	To disable any superscripting or subscripting of text output
'NOSIGNIFICANTZEROES'		Suppression of display of significant zeros
'NOSUPERSCRIPT'	3D	Same as 'NOSCRIPTING'
'NOTRAIL'	3D	Record of which GCS routines are invoked is not listed
'NOWINDOWING'	3D	Disable GCS windowing routine
'NOXLABEL'		No labels are to be drawn for graphing
'NOXREPEAT'	3D	To indicate that a component is provided for every X value in every curve in higher level graphing
'NOYLABEL'		No Y labeling for graphing
'NOYREPEAT'	3D	To indicate that a component is provided for every Y value in every curve in higher level graphing
'NOZCLIPPING'		No hither/yon clipping
'NOZLABELS'	3D	To indicate that no Z labels are to be drawn for higher level graphing
'NOZREPEAT'	3D	To indicate that a component is provided for every Z value in every curve in higher level graphing
'NObLINE'		Invisible line with null terminator
'NObMARK'		Line type
'NPOINT'		Invisible lines with point terminators
'NSYMBOL'		Invisible lines with character terminators
'OLDSCALE'		Old scale to be used for higher level graphing
'ORIGIN'	3D	An origin to be forced for 'AUTOSCALE' and 'FULLSCALE' scaling options
'ORMODE'	I,3D	Asynchronous event processing option
'ORTHOGRAPHIC'	3D	To specify orthographic projection in which the projection is parallel from all points
'OWNSCALE'		Own scale option for higher level graphing
'PARALLELLABELS'	3D	To specify that the main axis of the numeric labels will be parallel to the axis
'PENAXIS'		Axis intersection at current position for graphing
'PENDOWN'		Solid lines with null terminators
'PENORIGIN'	3D	To force the current pen position to be included in the axis range.

'PENUP'		Invisible lines with null terminators
'PERIODIC'		Time series axis scale in accounting period intervals
'PERCENTUNITS'		Device space coordinates specified in percent units
'PERPENDICULARLABELS'	3D	To specify that the major axis the numeric labels will be perpendicular to the axis
'PERSPECTIVE'	3D	To specify perspective projection in which line length diminishes as the distances from the viewing position become greater
'PIRADIANS'	3D	Angular information be interpreted in PI radians
'PLAINAXIS'		Plain axes to be drawn for high level graphing
'PLOTDEVICE'		Alphanumeric I/O to the plotting device
'POINT'		Invisible lines with point terminators
'POLAR'		Plotting in polar (RHO, THETA) units
'POSITIVESIDE'	3D	Labels will be above or to the right of the axis
'PRIMARYDEVICE'		Routing of graphical output to primary device
'QUARTERLY'		Time series axis scale in quarter year intervals
'RADIANS'		Angular information be interpreted in radians
'RASTERUNITS'		To indicate device space coordinates are specified as is in raster units
'REAL'		UPRINT/UWRITE output in real number format
'RECTANGULAR'		Plotting on the user's reference axis
'REDb'		Switch to pen color red
'REFERENCE'		Plotting on the user's reference axis
'REFRESHEDSEGMENT'		Segments to be retained
'RELb'		Plotting in a relative coordinate system
'RELATIVE'		Plotting in a relative coordinate system
'REPLACE'	3D	Data structure building option
'RETAINEDSEGMENTS'		Segments to be retained structures from merge file
'REWIND'	3D	Data structure file handling command
'RIGHTHAND'	3D	Right handed coordinate system
'RJUSTIFICATION'		Alphanumeric right justification
'RWINDOWING'	3D	Rectangular windowing
'SECONDLY'		Time series axis scale in second intervals
'SECRET'		Security classification secret
'SEGMENTED'		Curved lines be interpreted as multiple pen operations
'SEMIANNUAL'		Time series axis scale in semi annual intervals
'SENSITIZE'	3D	Make graphic segments visible
'SIGNIFICANTZEROES'		Display of significant zero
'SIMULATED HARDWARE CHARACTERS'		Output of simulated hardware characters
'SINGLE'	3D	Data structure invocation option
'SITEPOINT'	3D	Viewpoint distance to be measured from the view site
'SLOWBLINK'		Slow blink rate
'SMALL'		Use smallest hardware character size
'SOFTWAREFONT'		Output of software generated characters
'SONICPEN'	I	Sonic pen is graphics input device
'SPECIFIC'		To specify particular device units instead of percent units
'SPHERICAL'		Coordinates are of the form (R,THETA,PHI) where R is the number of units of radius, THETA is the number of angular units around the Z axis, and PHI is the number of angular units around the X axis

'STANDARDMENU'	I,3D	Menuing option
'SUBSCRIPT'	3D	To specify that the output will be lowered from the specified line of text
'SUPERSCRIPT'	3D	To specify that the output will be raised from the specified line of text.
'SUPPRESSERRORS'		Error output be suppressed
'SYMBOL'		Invisible lines with character terminators
'SYSTEMAXIS'		Plotting on the system axis
'TABLET'	I	Analog tablet as graphic input device
'TARROW'		Tic lines with arrow terminators
'TBACKARROW'		Tic lines with back arrow terminators
'TCHARACTER'		Tic lines with character terminators
'TCOORDINATES'		Tic lines with endpoint coordinates indicated
'TDOUBLEARROW'		Tic lines with double arrow terminators
'TEXT'		UPRINT/UWRITE output in text format
'SUBTRACTIVE'*		Subtractive color blending mode
'TICAXES'		Tic axes to be drawn for higher level graphing
'TICLINE'		Tic lines with null terminators
'TNULL'		Tic lines with null terminators
'THINLINES'		Line width to be thin
'TOPSECRET'		Security Classification Top Secret
'TPOINT'		Tic lines with point terminators
'TRAIL'	3D	To indicate by an identification number which GCS routine is involved.
'TSYMBOL'		Tic lines with character terminators
'TWELVEHOUR'		Time series axis scale in twelve hour intervals
'TWENTYFOURHOUR'		Time series axis scale in twenty four-hour intervals
'UNCLASSIFIED'		Security classification unclassified
'UNDETECTABLE'		Disabling pick sensitivity
'UNIFORM'*	3D	High level graphing system
'UNINTERRUPTED'		Non-gapped line output
'UPPERCASE'		Upper case to be 'TEXT' case
'USER'		Plotting on a user defined axis system
'VERTICAL'	3D	Alphanumeric output to be spaced vertically
'VIEWPOINT'	3D	View port distance to be measured from the view point
'VIRTUAL'		Plotting in virtual space
'VISIBLE'		Visible framed output
'WEEKLY'		Time series axis scale in weekly intervals
'WHITE'		Switch to pen color white
'WIDELINES'		Line width to be wide
'WORKINGAXIS'		Plotting on the user's working (temporary axis)
'WORLDCOORDINATESYSTEM'		Coordinate system to be the default axis system
'XABSOLUTE'	3D	To specify the X coordinates with respect to the origin of the current coordinate system for indicated components. Y and Z components are to be specified with respect to the current beam/pen position
'XALPHANUMERIC'		An X axis alphanumeric label
'XAXIS'		The X axis be drawn for high level graphing
'XBOTHLABELS'		X axis alphanumeric and numeric labels
'XCONSTANT'	3D	To indicate that the X component does not vary during the drawing of any curve in higher level graphics
'XEDGEYZEROAXIS'		The X axis at edge of graph
'XLOGARITHMIC'		Logarithmic X and linear Y plotting

'XNEGATIVE'	3D	Negative X axis represents up in 3D graphics
'XNUMERIC'		An X axis numeric label
'XPOSITIVE'	3D	Positive X axis represents up in 3D graphics
'XRELATIVE'	3D	To specify the X coordinates with respect to the current beam/pen position. Y and Z components are to be specified with respect to the origin of the current coordinate system
'XREPEAT'	3D	To indicate that one set of X valves is provided which will be reused for every curve in higher level graphing
'XYAXES'		The X and Y axes be drawn for high level graphing
'XYABSOLUTE'	3D	To specify the X and Y coordinates with respect to the origin of the current coordinate system for the indicated components. Z components are to be specified with respect to the current beam/pen position
'XYCOORDINATES'		To specify that the data printed by UPRINT/ UWRITE is in the form of an (X,Y) coordinate pair.
'XYLOGARITHMIC'	3D	Logarthmic X and Y plotting, linear Z plotting
'XYPLANE'	3D	Labels are to be drawn in the plane formed by the X and Y axes
'XYRELATIVE'	3D	To specify the X and Y components with respect to the current beam/pen position. Z components are to be specified with respect to the origin of the current coordinate system
'XYVIEW'	3D	To view plane formed by X and Y axes
'XYZAXES'	3D	All three axes are to be drawn for higher level graphing
'XYZCOORDINATES'	3D	Text printed by U3PRNT/U3WRIT to be in form of (X,Y,Z) triplet
'XYZLOG'	3D	Applies logamithmic transforms to all three components
'XYZVIEW'	3D	To view all three axes
'XYZb'	3D	To set the rotation application order as indicated
'XYCOORDINATES'		UPRINT/UWRITE output in (X,Y) coordinate format
'XZABSOLUTE'	3D	To specify the X and Z coordinates with respect to the origin of the current coordinate system for the indicated components. Y components are to be specified with respect to the current beam posi-tion
'XZAXES'	3D	The X and Z axes are to be drawn for higher level graphing
'XZEROYEDGEAXIS'		The X axis adjacent to boundary of display area
'XZLOGARITHMIC'	3D	Applies Logarithmic transformation to X and Z components
'XZPLANE'	3D	Label plane to be plane formed by X and Z axis
'XZRELATIVE'	3D	To specify the X and Z components with respect to the current beam/pen position. Z components are to be specified with respect to the origin of the current coordinate system
'XZVIEW'	3D	To view the plane formed by the X and Z axes
'XZYb'	3D	To set the rotation application order as indicated
'YABSOLUTE'	3D	Y coordinates are specified with respect to the origin of the current coordinate system for the indicated units. X and Z components are specified with respect to the current beam/pen position
'YALPHANUMERIC'		Y axis alphabetic label
'YAXIS'		The Y axis to be drawn for high level graphing
'YBOTHLABELS'		Y axis having alphabetic and numeric labels

'YCONSTANT'	3D	To indicate that the Y component does not vary during the drawing of any curve
'YEARLY'		Time series axis scale in yearly intervals
'YEDGEXZEROAXIS'		The Y axis at edge of graph
'YELLOW'		Switch to pen color yellow
'YLOGARITHMIC'		Logarithmic Y plotting
'YNEGATIVE'	3D	Negative Y direction represents up in 3D graphics
'YNUMERIC'		Y axis numeric label
'YPOSITIVE'	3D	Positive Y direction represents up in 3D graphics
'YRELATIVE'	3D	Y coordinates are specified with respect to the current beam/pen position for the indicated components X and Z components are to be specified with respect to the origin of the current coordinate system
'YREPEAT'	3D	To indicate that one set of Y valves is provided which will be reused for every curve in higher level graphing
'YXZb'	3D	To set the rotation application order as indicated
'YZABSOLUTE'	3D	Y and Z coordinates are specified with respect to the origin of the current coordinate system for the indicated under X components are specified with respect to the current beam/pen position
'YZAXES'	3D	The Y and Z axes to be drawn for higher level graphing
'YZPLANE'	3D	Label plane to be plane formed by Y and Z axes
'YZEROXEDGEAXIS'		The Y axis adjacent to boundary of display area
'YZLOGARITHMIC'	3D	Applies logarithmic transformations to Y and Z components
'YZRELATIVE'	3D	Y and Z coordinates to be specified with respect to the current beam/pen position for the Y and Z components, and the X component to be specified with respect to the current beam/pen position for the Y and Z components, and the X component to be specified with respect to the origin of the current coordinate system
'YZVIEW'	3D	To view the plane formed by the Y and Z axes
'YZXb'	3D	To set the rotation application order as specified
'ZABSOLUTE'	3D	Z coordinates to be specified with respect to the origin of the current coordinate system for the indicated component. X and Y components are to be specified with respect to the current beam/pen position
'ZALPHANUMERIC'	3D	Alphanumeric labels to be drawn for higher level drawing
'ZAXIS'	3D	Z axis is to be drawn for higher level graphing
'ZBOTHLABELS'	3D	Both numeric and alphanumeric labels to be drawn for higher level drawing
'ZCLIP'	3D	Clip in Z direction
'ZCONSTANT'	3D	To indicate that the Z component does not vary during the drawing of any curve
'ZEROAXES'		X and Y axes adjacent to boundary of display area
'ZLOGARITHMIC'	3D	Applies logarithmic transform to Z component
'ZNEGATIVE'	3D	Negative Z axis represents up direction in 3D graphics
'ZNUMERIC'	3D	Numeric Z labels to be drawn for high level graphing

'ZPOSITIVE'	3D	Positive Z axis represents up direction in 3D graphics
'ZRELATIVE'	3D	Z coordinates are to be specified with respect to the current beam/pen position. X and Y components are to be specified with respect to the origin of the current coordinate system
'ZREPEAT'	3D	To indicate that one set of Z values is provided which will be reused for every curve in higher level graphing
'ZXYb'	3D	To set the rotation application order as indicated
'ZYXb'	3D	To set the rotation application order as indicated
'12HOUR'		Twelve hour time axis
'13WEEK'		Thirteen week time axis
'2DCOORDINATES'	3D	To specify A coordinate terminator in which two components are listed. (This option is independent of the text coordinate options of 'XYCOORDINATES' and 'XYZCOORDINATES')
'24HOUR'		Twenty four time axis
'3DCOORDINATE'	3D	To specify a coordinate terminator in which all three components are listed. (This option is independent of the text coordinate option of 'XYCOORDINATES' and 'XYZCOORDINATES')

NOTE: b - is a blank or space

GCS DEFAULT CONDITIONS

This section addresses those options which are present in the Graphics Status Area as default options. After a call to Subroutine USTART, the Graphics Compatibility System is set to the default conditions, as indicated. The default options can be divided into two groups: Basic Plotting Options and High Level Plotting Options.

I. Default Basic Plotting Options

Plotting is done in 'RECTANGULAR' and 'ABSOLUTE' coordinates on the 'SYSTEM' coordinate axis. The 'USER' coordinate axes are identical to the system axis. Plotting is done in 'VIRTUAL' space with a virtual window whose limits are from 0.0 to 100.0 in the X direction, and from 0.0 to 100.0 in the Y direction. The virtual window is mapped into a display area which is the largest square area on the device display surface. The right hand edge of the square corresponds to the right edge of the display surface.

Character output in GCS will be, by default, in 'HARDWARE' character format, of type 'GOTHIC' and of 'MEDIUM' size. If 'SOFTWARE' characters are requested via USET then the default horizontal size is five (5.0) virtual units, and the default vertical size is seven (7.0) virtual units.

By default, angular units for angular specifications are in 'DEGREES'. If the user switches to 'DEVICE' space, then all length or distance units (i.e. from UPRINT or UREAD), the data will be assumed to be in 'TEXT' mode. The alphanumeric output defaults to the 'PLOT' device if possible, and any graphic output will go to all devices in a cluster. Graphic input is from the primary input device; the number of digits of precision for numeric output is four (4); and GCS detected errors are signalled as they occur.

A line which is drawn by a subroutine in GCS is considered to consist of two parts; a line type and a line terminator. The line type may be solid (visible), ticced, null (invisible), dashed, or alphanumeric. The line may be terminated by an arrow, a back arrow, double arrows, a character, a point, a symbol, a set of coordinate values, or nothing at all. The default line type in GCS is 'SOLID' with 'NULL' termintors ('LNULL'). If 'TICLINES' are requested via USET option, then the default tic interval is ten (10.0) virtual units. It is the user's responsibility to insure that the tic interval is appropriate if he switches to 'DEVICE' space or alters the default virtual window setting or the default display area setting. If 'DASHLINES' are requested, then the default dash specification (56.) will result in a dashed line which is alternately light and dark, in increments of approximately 0.075 inches. If a 'CHARACTER' line terminator is requested but no character is specified by way of Subroutine UPSET, then the character asterisk (*) is used. Similarly, the asterisk is used to compose the line type if 'ALPHANUMERIC' lines are requested.

II. Default High Level Plotting Options

For each call to UPLOT or UPLOT1, the data values which represent the curves are examined, and a 'NEWSCALE' is created. UAXIS will be invoked to create 'XYAXES'. The data values will be examined, and zero will always appear somewhere on the axis. Numeric labels only will be output for the X axis and the Y axis. The values which appear at the tic marks on the axes will be 'neat' numbers, and the axes will be positioned at the 'EDGE' of the plot. Both the X axis and the Y axis will be drawn in a linear coordinate space. The axis lines will be ticked. If a time series axis is plotted by invoking Subroutine UTAXIS, the default interval for the X axis will be 'DATE'. No curves will be fit to the data values, but if 'FITPOLYNOMIAL' is requested, then subroutine UPLOT will attempt to fit a fifth degree polynomial to the data.

III. Summary

COORDINATE SPACE:	'VIRTUAL'
COORDINATE TYPE:	'ABSOLUTE'
COORDINATE SYSTEM:	'RECTANGULAR'
COORDINATE AXIS:	'SYSTEM'
VIRTUAL WINDOW:	0.0 TO 100.0 X DIRECTION 0.0 TO 100.0 Y DIRECTION
DISPLAY AREA:	Largest square area which is right justified on the display surface of the device.
DEVICE SPACE UNITS:	'INCHES'
ANGULAR UNITS:	'DEGREES'
LINE TYPE:	'LINE' or 'LNULL'
SYSTEM CHARACTER:	asterisk (*)
TIC INTERVAL:	10.0 virtual units
DASH SPECIFICATION:	56.
CHARACTER TYPE:	'HARDWARE'
CHARACTER FONT:	'GOTHIC'
CHARACTER SIZE:	'MEDIUM'
SOFTWARE CHARACTER SIZE:	5.0 virtual units horizontal 7.0 virtual units vertical
INPUT/OUTPUT FORMAT:	'TEXT'
ALPHANUMERIC MARGINS:	Device display surface boundaries.
OUTPUT ROUTE:	'PLOTDEVICE'
OUTPUT DISTRIBUTION:	'ALLDEVICES'
GRAPHIC INPUT:	Primary input device
DIGITS OF PRECISION:	4
ERROR HANDLING:	'IMMEDIATE OUTPUT'
AXIS SCALING:	'AUTOSCALE'
AXIS LABELING:	'XNUMERICLABEL' 'YNUMERICLABEL'
AXIS POSITIONING:	'EDGEAXIS'
AXIS TYPE:	'TICAXIS'
AXIS EXISTENCE:	'XYAXES'
AXIS COORDINATES:	'NOLOGARITHMS'
TIME SERIES AXIS SCALE:	'DATE'

USET OPTIONS BY CLASS

Coordinate Mode

'ABSOLUTE'	'INCREMENTAL'	'RIGHTHANDED'	'ZABSOLUTE'
'RELATIVE'	'XYRELATIVE'	'YABSOLUTE'	'ZRELATIVE'
'XABSOLUTE'	'XZABSOLUTE'	'YRELATIVE'	
'XRELATIVE'	'XZRELATIVE'	'YZABSOLUTE'	
'XYABSOLUTE'	'LEFTHANDED'	'YZRELATIVE'	

Coordinate Type

'RECTANGULAR'	'CYLINDRICAL'	'LOGOBJECT'	'XYZLOGARITHMIC'
'POLAR'	'SPHERICAL'	'LOGSYSTEM'	'XZLOGARITHMIC'
'LOGARITHIMC'	'XLOGARITHMIC'	'LOGUSER'	'YZLOGARITHMIC'
'YLOGARITHMIC'	'XYLOGARITHMIC'	'NOLOGARITHMS'	'ZLOGARITHMIC'

Coordinate Space

'VIRTUAL'	'SPECIFIC'
'DEVICE'	'DISPLAY'

Device Space Units

'INCHES'
'CENTIMETERS'
'FONTUNITS'
'PERCENTUNITS'
'RASTERUNITS'

Angular Units

'DEGREES'	'GRADS'
'RADIANS'	'MILS'
'PIRADIANS'	

Frame Composition

'INVISIBLE'
'VISIBLE'

Line Type

'LINE'	'LNULL'	'NOLINE'	'LBACKARROW'
'DASH'	'DNULL'	'NOMARK'	'LCOORDINATE'
'POINT'	'NPOI'	'NNULL'	'NO LINE'
'TICLINE'	'TNULL'	'LARROW'	'NO MARK'
'CHARACTER'	'BACKARROWLINE'	'LDOUBLEARROW'	'DIMENSIONLINE'
'SYMBOL'	'COORDINATELINE'	'MOVE'	'NSYMBOL'
'ALPHANUMERIC'	'NCHA'	'ARROWLINE'	'ANULL'
'DOUBLEARROWLINE'			

Lines that are drawn by GCS are composed on a line type and a line terminator. A large number of line types and terminators are possible. The specification is composed by combining the first letter of the line type with the name of the terminator. The following tables illustrate the possible linetypes and terminators:

LINE TYPE	**FIRST LETTER**
LINE	L
DASH	D
ALPHANUMERIC	A
NULL (INVISIBLE)	N
TICLINE	T

Line Terminators

NULL (NO TERMINATORS)
CHARACTER
SYMBOL
COORDINATE
POINT
ARROW
BACKARROW
DOUBLEARROW

Line Repeatability

'UNINTERRUPTED'
'GAPPED'

Curve Approximation

'CONTINUOUS'
'SEGMENTED'

Blink Rate

'NOBLINK'
'FASTBLINK'
'SLOWBLINK'

Color

'WHITE'	'BLUE'
'BLACK'	'RED'
'GREEN'	'MAGENTA'
'CYAN'	'YELLOW'

Intensity

'DIM'
'BRIGHT'

Coordinate Axis

'SYSTEMAXIS'
'USERAXIS'

Coordinate Axis Composition

'WORKINGAXIS'
'REFERENCEAXIS'

Character Format

'GOTHIC'
'UPPERCASE'
'ITALIC'
'LOWERCASE'

Character Size

'SMALL'
'MEDIUM'
'LARGE'
'EXTRALARGE'

Device Routing

'PLOTDEVICE'
'MESSAGEDEVICE'

Device Selection

'ALLDEVICES'
'PRIMARYDEVICE'
'ALTERNATEDEVICE'

Graphic Input

'CURSORS'	'NOECHO'
'KEYBOARD'	'ECHO'
'LIGHTPEN'	'ORMODE'
'JOYSTICK'	'ANDMODE'
'BALL'	'PROCEED'
'MOUSE'	'WAIT'

Character Type

'HARDWARE'
'SOFTWARE'

Error Conditions

'ERROR OUTPUT'	'DUMP'	'NOABORT'	'TRAIL'
'SUPPRESSERRORS'	'ABEND'	'NODUMP'	
'DEFERERRORS'	'ABORT'	'NOTRAIL'	

Structure Definition

'BUILD'
'NOBUILD'

Structure Building

'EXECUTE'
'NOEXECUTE'

Structure File Manipulation

'APPEND'	'REPLACE'
'IGNORE'	'REWIND'
'MULTIPLE'	'SINGLE'

Structure Editing

'COMPRESSED'
'EXPANDED'

Axis Scaling

'AUTOSCALE'
'FULLSCALE'
'OWNSCALE'

Axis Scale Existence

'NEWSCALE'
'OLDSCALE'

Segment Pickability

'DESENSITIZE'
'SENSITIZE'

Axis Existence

'XYAXES'	'NOAXES'
'XAXIS'	'XYZAXES'
'YAXIS'	'XZAXES'
'YZAXES'	'ZAXIS'

Axis Positioning

'EDGEAXIS'
'ZEROAXIS'
'PENAXIS'

'XEDGEYZEROAXIS'	'YEDGEXZEROAXIS'
'XZEROYEDGEAXIS'	'YZEROXEDGEAXIS'

Numeric Labels

'BESTFORMAT'
'FORMAT'
'GFORMAT'

Label Positioning

'NEGATIVESIDE'	'PERPENDICULAR-LABELS'		
'PARALLELLABELS'	'POSITIVESIDE'		
'XYPLANE'	'XZPLANE'	'YPLANE'	

Axis Type

'NOLOGARITHMS'	'XYLOGARITHM'		
'XLOGARITHM'	'XZLOGARITHM'		
'YLOGARITHM'	'YZLOGARITHM'	'ZLOGARITHM'	'XYZLOGARITHM'
'LOGARITHMS'			

X Axis Labels

'XNUMERIC'
'XALPHANUMERIC'
'XBOTHLABELS'
'NOXLABEL'

Y Axis Labels

'YNUMERIC'
'YALPHANUMERIC'
'NOYLABEL'
'YBOTHLABELS'

Z Axis Label

'ZNUMERIC'
'ZALPHANUMERIC'
'NOZLABEL'
'ZBOTHLABELS'

Axis Type

'TICAXIS'
'PLAINAXIS'
'GRIDAXIS'

Three Dimensional Windowing

'SITEPOINT'
'VIEWPOINT'

Windowing

'CWINDOWING'
'NOWINDOWING'
'RWINDOWINE'

Text Output

'NOCENTER'	'SUBSCRIPT'	'HORIZONTAL'
'ACENTER'	'NOSUBSCRIPTING'	'UPPERCASE'
'NOSCRIPT'	'SUPERSCRIPT'	'VERTICAL'
'XYZCOORDINATES'	'INTEGER'	'REAL'
'XYCOORDINATES'	**'TEXT'**	
'2DCOORDINATES'	'3DCOORDINATES'	

Menuing Option

'STANDARDMENU'
'EXTENDEDMENU'

Origin Inclusion

'NOORIGIN'
'ORIGIN
'PENORIGIN

Device Space Clipping

'NOZCLIPPING'
'ZCLIP'

High-Level Plotting Option

'NONUNIFORM'
'UNIFORM'

Coordinate Repeat Option

'NOREPEAT'	'NOZREPEAT'	'YCONSTANT'	'ZREPEAT'
'NOXREPEAT'	'XCONSTANT'	'YREPEAT'	
'NOYREPEAT'	'XREPEAT'	'ZCONSTANT'	

Transformation Type

'PERSPECTIVE'
'ORTHOGRAPHIC'

Axis Orientation

'XNEGATIVE'	'YNEGATIVE'	'ZNEGATIVE'
'XPOSITIVE	'YPOSITIVE'	'ZPOSITIVE'

Three Dimension Viewing

'XYVIEW'	'XZVIEW'
'XYZVIEW'	'YZVIEW'

Rotation Application Order

'XYZ'	'YZX'
'XZY'	**'ZXY'**
'YXZ'	'ZYX'

Time Axis Scaling

12Hour	'YEARS'	**'DATE'**	'SECONDS'
24Hour	'DAYS'	'HOURS'	'PERIODIC'

FUNCTION:

To request that the named frame be placed in show status.

CALLING SEQUENCE:

CALL USHOW (NAME)

Where:

NAME	**is the eight character alphanumeric constant or variable of a currently defined frame.**

OPTIONS which may apply:

No options apply.

COMMENTS:

The execution of the subroutine causes the frame NAME to be activated on the face of the display screen. The frame must have been previously defined by a subroutine UFRAME and subroutine FREND pair. The initial status of a frame is show status.

Programming Notes:

Subroutine USPLIN

FUNCTION:

This routine fits a cubic spline curve to the specified input data and returns a set of X and Y coordinates of data points which lie on the spline curve.

CALLING SEQUENCE:

CALL USPLIN (X,Y,XN,RX,RY,RN)

Where

X	**is an array of XN elements which is the X component of the data points to be fitted**
Y	**is an array of XN elements which is the Y component of the data points to be fitted.**
XN	**is the number of points to be fitted**
RX	**is an array of RN elements which is the X component of the points which are returned.**
RY	**is an array of RM elements which is the Y component of the points which are returned.**
RN	**is the number of points to be returned (.LE.200).**

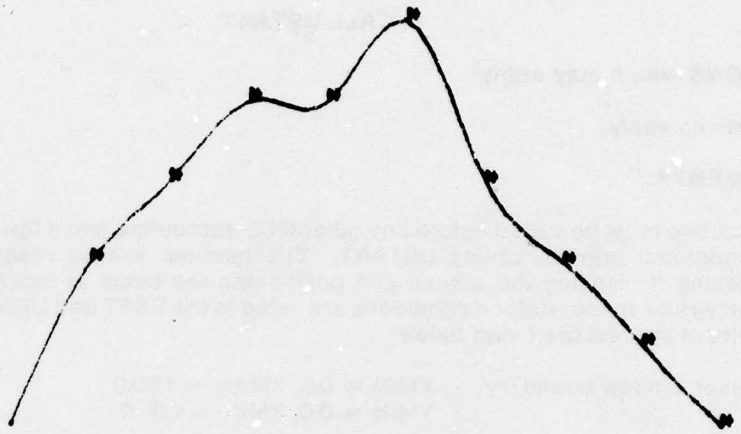
OPTIONS which may apply:

No options apply.

COMMENTS:

The number of points to be fitted must be greater than two. A maximum of one hundred data points may be fitted. The number of data points to be returned must be at least twice the number of input data points. As the ratio of output points to input points is increased, the resultant spline curve fit appears smoother. All data points are assumed to be in rectangular units.

Programming Notes:



```
DIMENSION X(10),Y(10),RX(50),RY(50)
DATA X/1.,2.,3.,4.,5.,6.,7.,8.,9.,10./
DATA Y/2.,4.,5.,6.,6.,7.,5.,4.,3.,2./
CALL USTART
CALL UWINDO (0.,10.5,0.,10.5)
CALL USET ('NCHARACTER')
CALL ULINE (X, Y, 10.)
CALL USPLIN (X, Y, 10.,RX,RY,50.)
CALL USET ('LINE')
CALL ULINE (RX,RY,50.)
CALL UEND
STOP
END
```

Subroutine USTART

FUNCTION:

This routine initializes the Graphics Status Area (GSA) to the default condition and insures that the terminal is ready for graphic output.

CALLING SEQUENCE:

CALL USTART

OPTIONS which may apply:

No options apply.

COMMENTS:

This routine must be called before any other GCS subroutine since the contents of GSA are undefined prior to calling USTART. The terminal will be readied for graphics processing by erasing the screen and positioning the beam at location (0.,0.) in the default virtual space. Default conditions are listed in the USET and UPSET tables. Other defaults of interest are listed below:

user window boundary	-	XMIN = 0.0, XMAX = 100.0 YMIN = 0.0, YMIN = 100.0
user display area	-	largest right-justified square that can be defined on the plotting surface.
margin boundaries	-	edges of entire display surface.

Programming Notes:

FUNCTION:

This routine starts construction of a new graphics data structure. It creates the structure header and places GCS in BUILD mode.

CALLING SEQUENCE:

CALL USTRCT(NAME)

Where

NAME	**is an eight character data structure name not currently used for this purpose in the current structure file.**

OPTIONS:

UPSET('LIBRARY',– –) must be specified for the structure random file to be defined.

Structure Construction Visibility Options: **'EXECUTE'** or 'NOEXECUTE'

COMMENTS:

The user again must specify a random file for the data structure library. The structure name must be eight characters, thus blanks must be put in by the user if the name is too short. If the user wishes to display data not to be saved in the structure, he may call USET('NOBUILD') to halt data structure creation. Later on, he may resume construction by calling USET('BUILD'). If the user does not wish to view the construction of his structure, he should specify 'NOEXECUTE'. Otherwise 'EXECUTE' is the default and he will watch his structure being build.

Programming Notes:

Subroutine USTUD

FUNCTION:

This routine returns to the user the limits of his display area.

CALLING SEQUENCE:

CALL USTUD (ARRAY)

Where

ARRAY **is an array of at least eight words defined as follows:**

ARRAY(1) = minimum X value in virtual space
ARRAY(2) = maximum X value in virtual space
ARRAY(3) = minimum Y value in virtual space
ARRAY(4) = maximum Y value in virtual space
ARRAY(5) = minimum X value in device space
ARRAY(6) = maximum X value in device space
ARRAY(7) = minimun Y value in device space
ARRAY(8) = maximum Y value in device space

OPTIONS:

Device Units: **'INCHES'**, 'CENTIMETERS', 'PERCENT', 'FONT', 'RASTER', 'SPECIFICATION'

COMMENTS:

A call to this routine will return to the user the limits of his virtual window and display area (virtual) or the limits of the device surface and clip area (device). The device units are in current units.

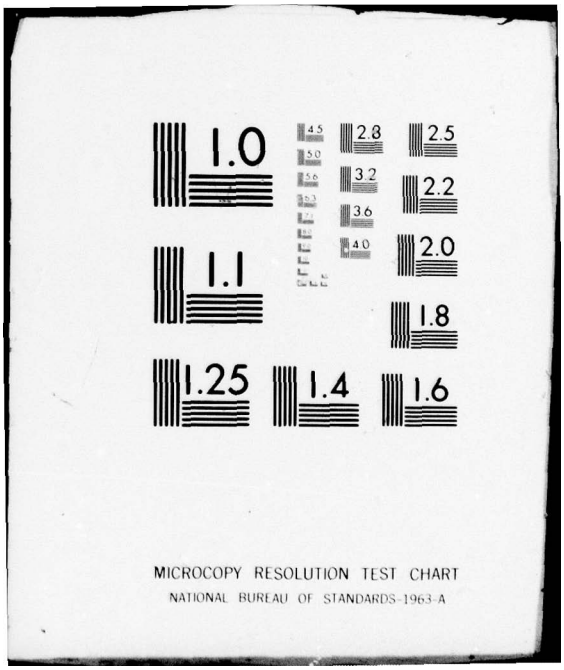
Programming Notes:

AD-A083 424 WESTINGHOUSE WORD PROCESSING CENTER PITTSBURGH PA F/G 9/2
GRAPHICS COMPATIBILITY SYSTEM (GCS) PROGRAMMER'S REFERENCE MANU--ETC(U)
OCT 79 DACW39-78-M-2676

UNCLASSIFIED WES-MP-0-79-5 NL

3 OF 3
AD A083424

END
DATE FILMED
5-80
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Subroutine USVPN

FUNCTION:

This routine saves the pen-related variables in the Graphics Status Area (GSA) in an array furnished by the user. This array will be suitable for later use as input to UNSVPN to restore the pen-related variables of the status area.

CALLING SEQUENCE:

CALL USVPN (ARRAY)

Where:

ARRAY **is an array large enough to contain the pen-related variables to be saved from the status area.**

OPTIONS which may apply:

No options apply.

COMMENTS:

The size of the array necessary to save the pen-related variables of the GSA is dependent upon the particular version of GCS being used.

Variables to be saved are only those which relate to the functions of the pen. Not saved are those variables which relate to such items as alphanumeric output, high-level graphics, dynamic graphics, and data structure controls. Also not saved are coordinate system specifications which have their own save and restore routines (see USVTR and UNSVTR).

The user is cautioned that any modification to the save array prior to its restoration into the GSA may yield indeterminate results.

Programming Notes:

Subroutine USVTR

FUNCTION:

This routine will save user coordinate system transformation status specification of the Graphics Status Area (GSA).

CALLING SEQUENCE:

CALL USVTR (SARRAY)

Where

SARRAY **is an array of 29 words into which the current coordinate system transform is stored.**

OPTIONS which may apply:

No options apply.

COMMENTS:

The invocation of this subroutine will save the current user coordinate system in effect, but will not alter the GCS status.

Programming Notes:

Subroutine UTAXIS

FUNCTION:

This routine creates a set of axes which displays time increments along the horizontal axis and numeric values along the vertical axis. It will support scaling and labeling options as provided for UAXIS. All output from UTAXIS will be contained within the currently defined UDAREA.

CALLING SEQUENCE:

CALL UTAXIS (BEGPER,PERIOD,YMIN,YMAX)

Where

BEGPER	**is the number which represents the beginning time period (1=first period, 2=second period etc.).**
PERIOD	**indicates the number of periods to be displayed.**
YMIN	**is the minimum Y value to be displayed.**
YMAX	**is the maximum Y value to be displayed.**

OPTIONS which may apply:

Axis Labeling Options:	'NOXLABELS', **'XNUMERIC',** 'XALPHANUMERIC', 'XBOTHLABELS', 'NOYLABELS', **'YNUMERIC',** 'YALPHANUMERIC', 'YBOTHLABELS'
Axis Titling Options:	UPSET option 'TICX', UPSET option 'YLABEL'
Tic-Interval Specification:	UPSET option 'XLABEL', UPSET option 'TICY'
Scaling Options:	**'AUTOSCALE',** 'FULLSCALE', 'OWNSCALE'
Period Options:	'MONTHS', 'YEARS', 'WEEKDAYS', 'DAYS', **'DATE',** 'MINUTES', 'HOURS', 'SECONDS', '12 HOUR' '24 HOUR', 'QUARTERS', 'PERIODIC'
Numeric Precision Option:	UPSET option 'PRECISION'

COMMENTS:

This routine will provide a set of axes similar to the UAXIS format. However, at those locations along the X axis where numeric labels would appear, UTAXIS will provide suitable labels to specify those options set in the Graphics Status Area. For a general description of all of the options indicated above except the period options, see UAXIS. Specific comments will be provide below on deviations from those general descriptions.

Programming Notes:

Period Options:

The period options indicate time divisions which will be used to label the horizontal axis. The time periods are cyclical and will be repeated as necessary until the number of periods specified has been indicated.

'SECONDS'	Continuous from that specified in BEGPER.
'MINUTES'	Continuous from that specified in BEGPER.
'HOURS'	Continuous from that specified in BEGPER.
'DAYS'	Continuous from that specified in BEGPER.
'WEEKDAYS'	Starts with BEGPER; recycles after 'SATURDAY'.
'MONTHS'	Starts with BEGPER; recycles after 'DECEMBER'.
'QUARTERS'	Starts with BEGPER; recycles after '4THQTR'.
'YEARS'	Continuous from that specified in BEGPER.
'DATE'	Starts with BEGPER; recycles on month boundaries. This value should have the format YYYMMDD or YYMMDD.

Axis Scaling Options:

The axis scaling option will not be effective for the time period axis. If the axis labels are non-numeric ('WEEKDAYS', 'MONTHS', or 'QUARTERS'), then the first period is assigned value zero and the last period has the value PERIOD-1.

FUNCTION:

This routine terminates construction of the current graphics data structure.

CALLING SEQUENCE:

CALL UTERM(NAME)

Where

NAME **is the eight character name of the structure currently being built.**

COMMENTS:

This routine puts a zero word after the last structure element and places GCS in NOBUILD mode.

Programming Notes:

FUNCTION:

This routine performs standard utility functions for data structure files. Some options will manipulate files, others manipulate structures themselves and their names.

CALLING SEQUENCE:

CALL UTILTY (ACTION,VALUE)

Where

ACTION is a Hollerith character string which specifies one of the following options:

'LOAD'	loads library file from user sequential file
'SAVE'	creates user sequential file from library file
'PURGE'	empties current structure library file
'MERGE'	merges requested file into library file
'DELETE'	deletes structure from library file
'RENAME'	renames old structure to new structure name.

VALUE is a Fortran file code number for LOAD,SAVE, and

MERGE is a structure name for DELETE, and is an array of two rows containing two structure names for RENAME.

COMMENTS:

This routine provides utility manipulations on structures and structure files.

The following comments apply to each separate action.

SAVE writes card image records to a user sequential file of all structures in the library.

LOAD reads the card image user file and puts the information into data structure format in the random file. This action has a compacting effect.

PURGE empties the current library file.

MERGE merges the user card image file into the current structure library. Duplicate names will be REPLACEd or IGNOREd according to these USET options.

DELETE removes a structure from the library file.

RENAME will rename the structure of the first name given with that of the second name.

Programming Notes:

Subroutine UTSFIT

FUNCTION:

To compute an expoentially smoothed forecast of the input time series data. If desired, the forecast will be trend adjusted.

CALLING SEQUENCE:

CALL UTSFIT(ARRAY,POINTS,FCST,ALPHA)

Where

ARRAY	**is a real array of size POINTS which is the input time series data.**
POINTS	**is the number of input points.**
FCST	**is a real array of size POINTS which is the time series forecast for periods 2 to POINTS +1.**
ALPHA	**is the weighting factor which determines the amount of reliance placed upon older data. The selection of a large ALPHA causes the system to respond rapidly to differences between the actual and forecast data.**

OPTIONS which may apply:

No options apply

COMMENTS:

The absolute value of ALPHA must be less than or equal to one. If the value of ALPHA is negative, the forecast will be trend adjusted.

Note that this routine represents one of the basic tools for time series analysis. The user is urged to consult a qualified reference on time series analysis for a complete description of the methods involved in this routine. Other techniques, such as base series correction, cyclic analysis, and higher order smoothing are necessary to synthesize a meaningful, composit forecast. The expoential smoothing of the input data is returned for periods 2...N+1 where N is the number of input time series periods. The first forecast period which is to be considered accurate must be determined by the user.

Programming Notes:

FUNCTION:

This routine defines the position of the viewer in relation to the environment and specifies his direction of view.

CALLING SEQUENCE:

CALL UVIEW(XVIEW,YVIEW,ZVIEW,XSITE,YSITE,ZSITE)

Where

XVIEW,YVIEW,ZVIEW	**is the location of the viewing position in current units.**
XSITE,YSITE,ZSITE	**is the location of the site at which the viewer is looking in current units.**

OPTIONS:

Pen Coordinate Options:	see U3PEN
Viewing Attitude:	UPSET('ATTITUDE',angle)
Up Direction:	'ZPOSITIVE', 'YPOSITIVE', 'XPOSITIVE', 'ZNEGATIVE', 'YNEGATIVE', 'XNEGATIVE'

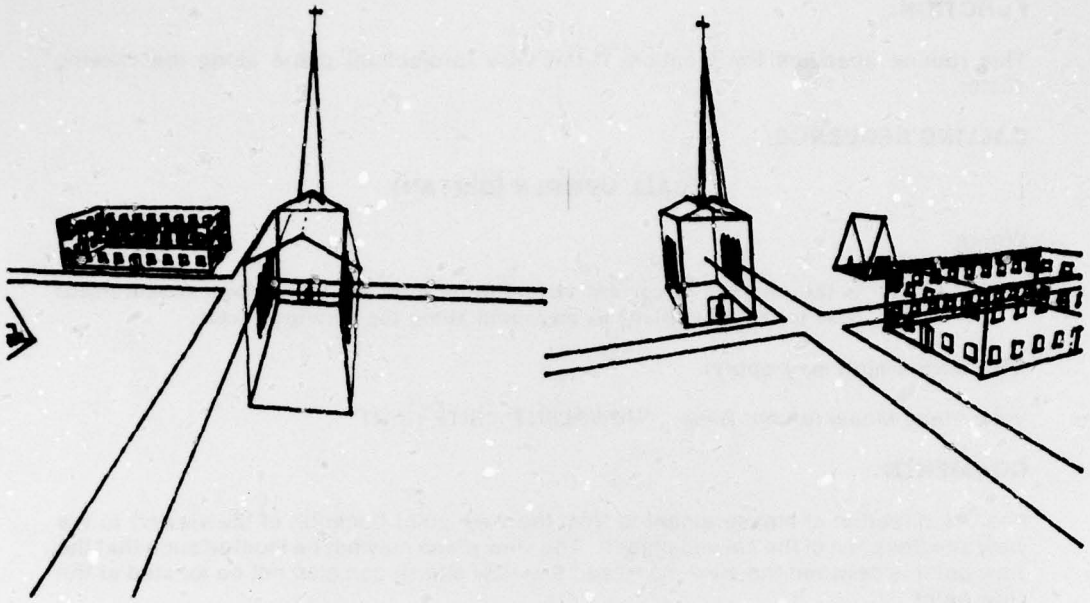
COMMENTS:

UVIEW sets the viewing position and direction in current units. Changing the current unit will not effect the viewing position and direction. The attitude of the view is a rotation around the viewing axis (a line drawn from the view point to the view site). It is normally set so that the view will be horizontal. This is the default value (UPSET('ATTITUDE',0.)).

Up Direction:

In creating 3-D systems, it has become standard to consider the positive Z axis as representing the direction 'up'. This is the default chosen by GCS. Should the user desire, however, he may specify that any other axis represents the 'up' direction. The 'up' direction refers to the attitude of the viewer in relationship to the coordinate system of the environment.

Programming Notes:



```
CALL ATTACH (2,'/TEK3DEMO/SAVE;',3,0,IST,)
CALL USTART
CALL UPSET ('TERMINATOR','<')
CALL USET ('PERCENT UNITS')
CALL USET ('REFERENCE COORDINATE SYSTEM')
CALL UPSET ('LIBRARY',1.)
CALL UTILTY ('LOAD',2.)
CALL USET ('VIEWSITE')
CALL UVWPRT (150.)
CALL UWINDO (-100.,100.,-100.,100.)
CALL UDAREA (0.,50.,0.,70.)
CALL UVIEW (-40.,200.,70.,-20.,20.,0.)
CALL VILLAG
CALL UVIEW (-70.,-150.,50.,0.,0.,10.)
CALL UDAREA (50.,100.,0.,70.)
CALL VILLAG
CALL UEND
STOP
END
SUBROUTINE VILLAG
CALL USET ('XYZ ')
CALL USET ('SYSTEMAXIS')
CALL USET ('REFERENCEAXIS')
CALL U3CALL (-50.,20.,0.,1.,1.,1.,90.,0.,0.,'CHURCH')
CALL U3CALL (24.,-10.,0.,1.,1.,1.,90.,-90.,0.,'SCHOOL')
CALL U3CALL (70.,70.,0.,0.7,0.7,0.7,0.,0.,0.,'PIZZA')
CALL U3CALL (0.,0.,0.,1.,1.,1.,0.,0.,0.,'ROAD ')
RETURN
END
```

FUNCTION:

This routine specifies the location of the view (projection) plane along the viewing vector.

CALLING SEQUENCE:

CALL UVWPLN (DISTAN)

Where

DISTAN **is the distance in current virtual units from the view plane measurement base to the view plane as measured along the viewing vector.**

OPTIONS which may apply:

View Plane Measurement Base. **'VIEWPOINT',** 'SITEPOINT'

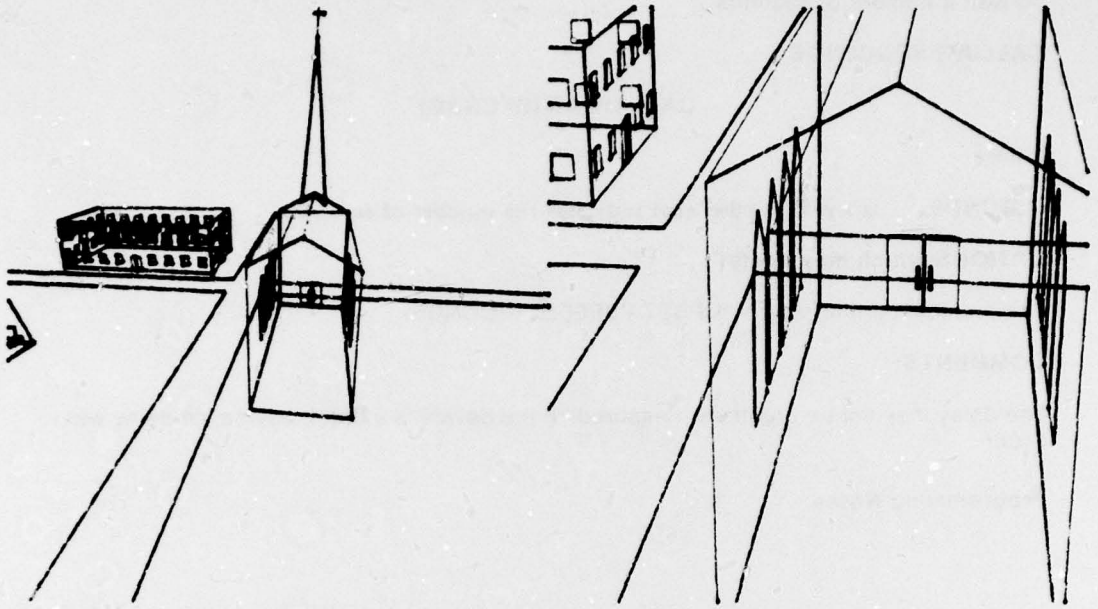
COMMENTS:

Positive direction of measurement is from the view point (location of the viewer) to the view site (location of the viewed object). The view plane may not be located such that the view point is between the view plane and the view site. It can also not be located at the view point.

Specification of the view plane will effect the amount of perspective which will result when generating the image. The closer the view (projection) plane to the view point and the wider (higher) the window on this plane, the more pronounced will be the perspective. This routine specifies the location of the view plane relative to either the view point or the view site. If 'VIEWPOINT' mode is specified, the view plane distance is measured from the site point. All measurements are along the viewing vector. 'SITEPOINT' is the default mode.

The view plane specification indicates the plane on which the window width and height are measured (see U3WNDO).

Programming Notes:



```
CALL ATTACH (2,'/TEK3DEMO/SAVE;',3,0,IST.)
CALL USTART
CALL UPSET ('TERMINATOR','<')
CALL USET ('PERCENT UNITS')
CALL USET ('REFERENCE COORDINATE SYSTEM')
CALL UPSET ('LIBRARY',1.)
CALL UTILTY ('LOAD',2.)
CALL USET ('VIEWSITE')
CALL UVWPLN (150.)
CALL UWINDO (-100.,100.,-100.,100.)
CALL UDAREA (0.,50.,0.,70.)
CALL UVIEW (-40.,200.,70.,-20.,20.,0.)
CALL VILLAG
CALL UVWPLN (500.)
CALL UDAREA (50.,100.,0.,70.)
CALL VILLAG
CALL UEND
STOP
END
SUBROUTINE VILLAG
CALL USET ('XYZ ')
CALL USET ('SYSTEMAXIS')
CALL USET ('REFERENCEAXIS')
CALL U3CALL (-50.,20.,0.,1.,1.,1.,90.,0.,0.,'CHURCH')
CALL U3CALL (24.,-19.,0.,1.,1.,1.,90.,-90.,0.,'SCHOOL')
CALL U3CALL (70.,70.,0.,0.7,0.7,0.7,0.,0.,0.,'PIZZA')
CALL U3CALL (0.,0.,0.,1.,1.,1.,0.,0.,0.,'ROAD ')
RETURN
END
```

Subroutine UWAIT *Interactive*

FUNCTION:

To wait a number of seconds.

CALLING SEQUENCE:

CALL UWAIT(SECNDS)

Where

SECNDS **is a real number that indicates the number of seconds.**

OPTIONS which may apply:

Communication line speed: UPSET ('SPEED',SECONDS)

COMMENTS:

The delay may not be precisely measured. If the device is a batch device, no delay will occur.

Programming Notes:

Subroutine UWHERE

FUNCTION:

This routine places in the arguments the coordinates of the current beam/pen position in current absolute user units.

CALLING SEQUENCE:

CALL UWHERE (X,Y)

Where

X	**will be on return from this routine, the X- or RADIUS coordinate of the current beam/pen position in current absolute user units.**
Y	**will be on return from this routine, the Y- or THETA coordinate of the current beam/pen position in current absolute user units.**

OPTIONS which may apply:

Pen Coordinate options - see UPEN

COMMENTS:

This routine always returns the coordinates of the current beam position in ABSOLUTE units since RELATIVE units are always (0.,0.). A handy use for this routine is to obtain the coordinates of the beam/pen in different units from those used to move the beam/pen to where it is located.

Programming Notes:

Subroutine UWINDO

FUNCTION:

This routine defines the boundaries of the user window into virtual space.

CALLING SEQUENCE:

CALL UWINDO (XMIN,XMAX,YMIN,YMAX)

Where

XMIN	**is the left-most window boundary in current VIRTUAL RECTANGULAR units.**
XMAX	**is the right-most window boundary in current VIRTUAL RECTANGULAR units.**
YMIN	**is the bottom-most boundary in current VIRTUAL RECTANGULAR units.**
YMAX	**is the top-most window boundary in current VIRTUAL RECTANGULAR units.**

OPTIONS which may apply:

No options apply.

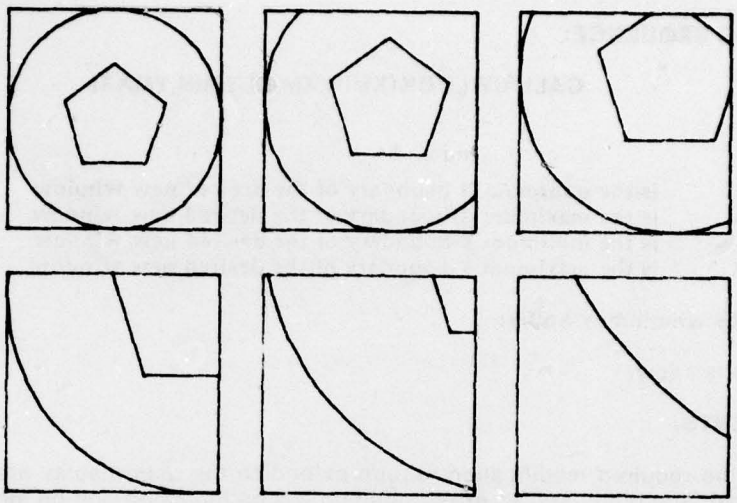
COMMENTS:

The concept of windowing into a virtual space expands greatly the capabilities of the user. It is this ability which allows him to work in his own units, plotting in a space which theoretically exists from minus infinity (largest negative number of the computer) to plus infinity (largest positive number of the computer) in both the X and Y directions, but observing only that subset of the vast space that is of interest to him. The user defines the rectangular area of interest by calling this routine (UWINDO). The user can also specify a corresponding rectangle on the physical plotting surface where any output visible in the virtual user window will appear. This area is called the user display area and may be specified by calling UDAREA. The output in virtual space is mapped on-to-one onto the user display area with resulting distortion should the ratio of the window to the display area in the X direction not be equal to that of the Y direction. Multiple user windows may be superimposed on one user display area by changing the user window without changing the user display area. Conversely multiple displays may be created from one area of virtual space by changing the user display area without changing the user window.

Subroutine UWINDO is used to set the user window boundaries at the viewplane. An error condition will occur (see Error Appendix) and the previous setting will not be modified if the maximum boundary is specified less than the minimum boundary.

In a two dimension environment, the viewplane is set to be the default Z value.

Programming Notes:



```
      CALL USTART
      DO 1 I = 1, 6
      CALL UERASE
      BOUNDS = 100. - (15. * FLOAT(I-1))
      CALL UWINDO (0.,BOUNDS,0.,BOUNDS)
      CALL UOUTLN
      CALL DRWFIG
    1 CONTINUE
      CALL UEND
      STOP
      END
      SUBROUTINE DRWFIG
      CALL UCRCLE (50.,50.,50.)
      CALL UPLYGN (50.,50.,5.,25.)
      CALL UBELL
      CALL UPAUSE
      RETURN
      END
```

FUNCTION:

This routine adjusts both the virtual window and the user display area to cover the portion of virtual space specified as input arguments.

CALLING SEQUENCE:

CALL UWLOOK(XMIN,XMAX,YMIN,YMAX)

Where

XMIN	**is the minimum X-boundary of the desired new window**
XMAX	**is the maximum X-boundary of the desired new window**
YMIN	**is the minimum Y-boundary of the desired new window**
YMAX	**is the maximum Y-boundary of the desired new window**

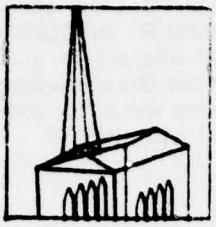
OPTIONS which may apply:

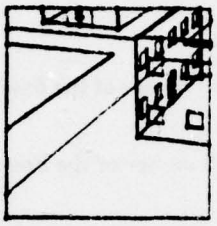
No options apply.

COMMENTS:

Should the required modification require extending the user display area beyond the display surface boundary, an error condition will be indicated and no modification will take place.

Programming Notes:





```
CALL ATTACH (2,'/TEK3DEMO/SAVE,',3,0,IST,)
CALL USTART
CALL UPSET ('TERMINATOR','<')
CALL UPSET ('LIBRARY',1.)
CALL UTILTY ('LOAD',2.)
CALL UWINDO (-100.,0.,0.,100.)
CALL UDAREA (3.5,6.5,4.5,7.5)
CALL UVIEW (100.,-250.,80.,0.,100.,0.)
CALL UWLOOK (-100.,-10.,10.,100.)
CALL UOUTLN
CALL VILLAG
CALL UWLOOK (-100.,-10.,-100.,-10.)
CALL UOUTLN
CALL VILLAG
CALL UEND
STOP
END
SUBROUTINE VILLAG
CALL USET ('XYZ ')
CALL USET ('SYSTEMAXIS')
CALL USET ('REFERENCEAXIS')
CALL U3CALL (-50.,20.,0.,1.,1.,1.,90.,0.,0.,'CHURCH')
CALL U3CALL (24.,-10.,0.,1.,1.,1.,90.,-90.,0.,'SCHOOL')
CALL U3CALL (70.,70.,0.,0.7,0.7,0.7,0.,0.,0.,'PIZZA')
CALL U3CALL (0.,0.,0.,1.,1.,1.,0.,0.,0.,'ROAD ')
RETURN
END
```

Subroutine UWRITE

FUNCTION:

This subroutine enables the user to print information at the pen position specified. The five (5) options available to the user are: 'TEXT', 'REALNUMBER', 'INTEGERNUMBER', *'XYZCOORDINATES'*, and 'XYCOORDINATES'. The output characters will be either hardware or software depending on the current setting in the Graphics Status Area. Margining will occur with hardware characters and windowing will occur with software characters. The beam will be restored to its position on entry to UWRITE.

CALLING SEQUENCE:

CALL UWRITE (X,Y,DATA)

Where

X	**is the X- or RADIUS coordinate of the lower left corner of the first character of the output.**
Y	**is the Y- or THETA coordinate of the lower left corner of the first character of the output.**
DATA	**is a single variable or array containing either real numbers or a GCS text string. The size of DATA is variable; it is a single variable if 'REALNUMBER' or 'INTEGERNUMBER' is specified; it is a two word array if 'XYCOORDINATES' is specified; and it may be any length if 'TEXT' is specified.**

OPTIONS which may apply:

Alphanumeric Output Options: 'REALNUMBER', 'INTEGERNUMBER', 'XYCOORDINATES', **'TEXT', 'XYZCOORDINATES'**

Numeric Precision Option: UPSET option 'PRECISION'

Character Type Options: **'HARDWARE',** 'SOFTWARE'

Character Size Options:

for 'HARDWARE' - **'SMALL',** 'MEDIUM', 'LARGE', 'EXTRALARGE'
for 'SOFTWARE' - UPSET options 'HORIZONTAL', 'VERTICAL'

Alternate Character Format: **'GOTHIC',** 'ITALIC'

Margin Boundaries: (see UMARGN)

Window Boundaries: (see UWINDO)

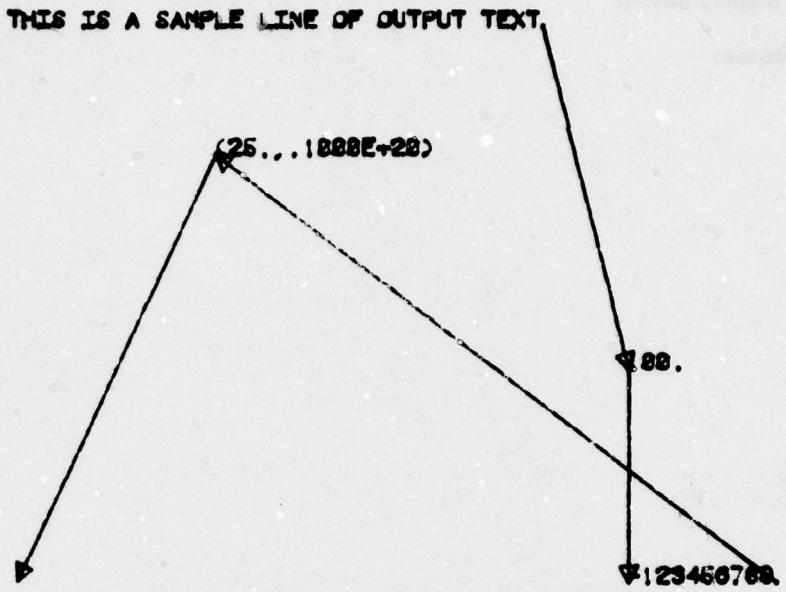
Coordinate System Options (see UPEN and UCOSYS)

COMMENTS:

UWRITE is functionally equivalent to UPRINT except that the beam is restored to the position it occupied upon entry to UWRITE. For a complete description of all the options available using either UWRITE or UPRINT, see UPRINT.

In three dimensional qpplications, the text is positioned at the point specified by X,Y,ZVALUE, where ZVALUE is established by a call to UPSET with the 'ZVALUE' option. For two dimensional applications, this corresponds to output being placed on the plane of the graphics display device.

Programming Notes:




```
DIMENSION COORD(2)
DATA COORD/25.,0.99999999E 19/
CALL USTART
CALL UPSET ('TERMINATOR',',')
CALL UPSET ('SPEED',120.)
CALL UERASE
CALL USET ('EXTRALARGE')
CALL UPRINT (0.,90.,'THIS IS A SAMPLE LINE OF OUTPUT TEXT,')
CALL USET ('REALNUMBER')
CALL USET ('ARROW')
CALL UPEN (75.,50.)
CALL UWRITE (75.,50.,100.)
CALL USET ('INTEGER')
CALL UPEN (75.,25.)
CALL UPRINT (75.,25.,-123456789.)
CALL USET ('XYCOORDINATE')
CALL UPEN (25.,75.)
CALL UWRITE (25.,75.,COORD)
CALL UPEN (1.,25.)
CALL UEND
STOP
END
```

Subroutine UWRIT1

FUNCTION:

This routine displays the data provided at the current beam position in the format currently specified in either hardware or software characters subject to the one-time setting of the USET option furnished which only applies during the execution of this subroutine. Upon return the beam is positioned at the location it held upon entering ths subroutine.

CALLING SEQUENCE:

CALL UWRIT1 (DATA,OPTION)

Where

DATA	is a single variable or array containing either real numbers or a GCS text string. The size of DATA is variable: it is a single variable if 'REALNUMBER' or 'INTEGERNUMBER' is specified; it is a two-word array if 'XYCOORDINATES' is specified it is a three word array if 'XYZCOORDINATES' is specified; and it may be any length if 'TEXT' is specified.
OPTION	is a Hollerith string variable or literal defining the USET option to apply during the execution of this subroutine.

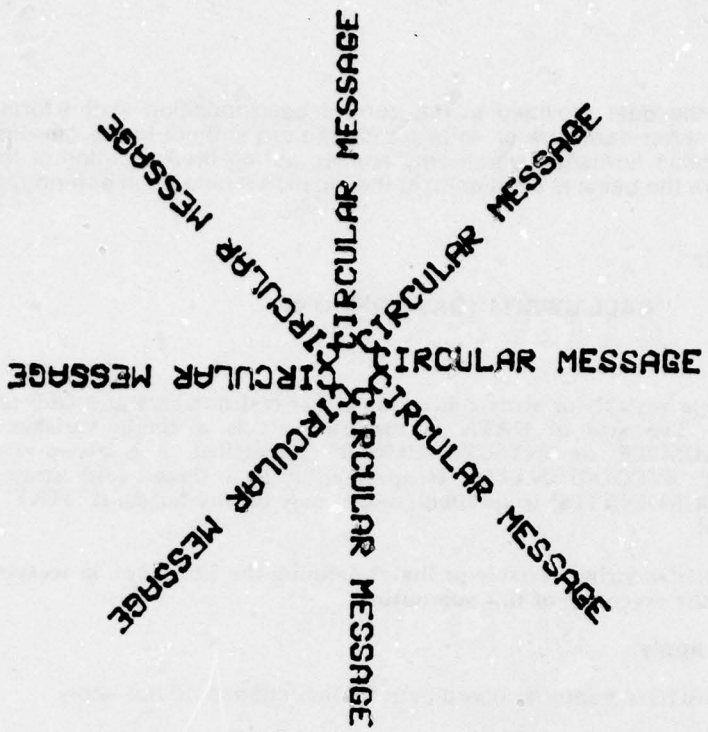
OPTIONS which may apply:

Same as those for UWRITE except that coordinate system options do not apply.

COMMENTS:

Calling UWRIT1 is equivalent to the sequence of calling USET with OPTION, calling UWRITE at the current beam position with DATA, and then calling USET with the appropriate options which would restore the status changed by OPTION. UWRIT1 is convenient for producing alphanumeric information at points being plotted.

Programming Notes:




```
      CALL USTART
      CALL UPSET ('TERMINATOR',',')
      CALL UASPCT (1.)
      CALL USET ('SOFTWARE CHARACTERS')
      CALL UPSET ('HORIZONTAL SIZE',2.5)
      CALL UPSET ('VERTICAL SIZE',3.5)
      CALL UMOVE (50.,50.)
      DO 10 I = 1, 6
      CALL UROTAT (45.*(FLOAT(I-1)))
   10 CALL UWRITE (' CIRCULAR MESSAGE,','TEXT')
      CALL UEND
      STOP
      END
```

Subroutine UZWNDO

FUNCTION:

This routine specifies the hither and yon boundary planes of the user window without altering the currently specified X and Y window boundaries.

CALLING SEQUENCE:

CALL UZWNDO(ZMIN,ZMAX)

Where

ZMIN	is the hither or minimum Z-boundary of the new window.
ZMAX	is the yon or maximum Z-boundary of the new window.

OPTIONS which may apply:

Z axis clipping: **'NOZCLIPPING'**, 'ZCLIPPING'

COMMENTS:

A new window boundary specification is set in which the new X and Y boundaries are the same as the old X and Y boundaries and the new Z boundaries are obtained from the input parameters. If ZMAX .LT. ZMIN, an error is generated and no change is made in the window. Clipping on the Z boundaries will only occur if 'ZCLIPPING' is specified (see U3WNDO).

The arguments of UZWNDO are given in eye coordinates.

Programming Notes:

FUNCTION:

This routine specifies the screen location within which the contents of the user window (U3WNDO) will be displayed.

CALLING SEQUENCE:

CALL U3AREA(XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX)

Where

XMIN	is the minimum X boundary in current device units of the new user display area
XMAX	is the maximum X boundary in current device units of the new user display area
YMIN	is the minimum Y boundary in current device units of the new user display area
YMAX	is the maximum Y boundary in current device units of the new user display area
ZMIN	is the minimum Z boundary in current device units of the new user display area
ZMAX	is the minimum Z boundary in current device units of the new user display area

OPTIONS which may apply:

Device Units: **'INCHES'**, 'CENTIMETERS', 'PERCENT UNITS', 'RASTER', 'FONT', 'SPECIFICATIONS'

COMMENTS:

This routine works in close conjunction with U3WNDO to define the mapping to take place between virtual space and device space. UDAREA defines a rectangular parallelopiped which is a subset of the total device space. The six-sided viewing solid defined by U3WNDO and the current projection mode (perspective or orthogonal) is mapped into this parallelopiped.

The units used to describe the desired region of the display area are device, rectangular, absolute units. Any of the device units may be used (e.g., 'INCHES', 'CENTIMETERS', etc.). If any of the maximum boundary parameters specifies a boundary less than or equal to the minimum boundary parameters, an error is generated and the old U3AREA specification is retained.

If a display-device with only a 2-D address space is being utilized, the Z values are checked for validity but are not used. The projection of the U3WNDO onto the view plane will then be mapped into the U3AREA X and Y space of the screen plane. The U3AREA is frequently referred to as the 'viewport' in the literature.

FUNCTION:

This routine creates a set of coordinate axes in 3-D space.

CALLING SEQUENCE:

CALL U3AXIS(XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX)

Where

XMIN	**is the minimum value of the X axis**
XMAX	**is the maximum value of the X axis**
YMIN	**is the minimum value of the Y axis**
YMAX	**is the maximum value of the Y axis**
ZMIN	**is the minimum value of the Z axis**
ZMAX	**is the maximum value of the Z axis**

OPTIONS which may apply:

Origin Inclusion Options: **'ORIGIN'**, 'NOORIGIN', 'PENORIGIN'

View Adjustment Options: 'XYVIEW', 'XZVIEW', 'YZVIEW', **'XYZVIEW'**

Grid Type Specifications: UPSET('GRID',value)

Axis Existence Options: 'NOAXES', 'XAXIS', 'YAXIS', 'ZAXIS', 'XYAXES', 'XZAXES', 'YAXES', **'XYZAXES'**

Axis Scaling Options: **'AUTOSCALE'**, 'FULLSCALE', 'OWNSCALE'

Coordinate Type Options: **'RECTANGULAR'**, 'POLAR', 'CYLINDRICAL', 'SPHERICAL'

Logarithmic Transforms: **'NOLOG'**, 'XLOG', 'YLOG', 'ZLOG', 'XYLOG', 'XZLOG', 'YZLOG', 'LOGARITHMIC'

Logarithmic Application Time: 'LOGSYSTEM', 'LOGUSER', 'LOGORIGINAL'

Logarithm Base: UPSET 'BASE', 'XBASE', 'YBASE', 'ZBASE'

COMMENTS:

This routine is designed to supercede the 2-D URAXIS routine.

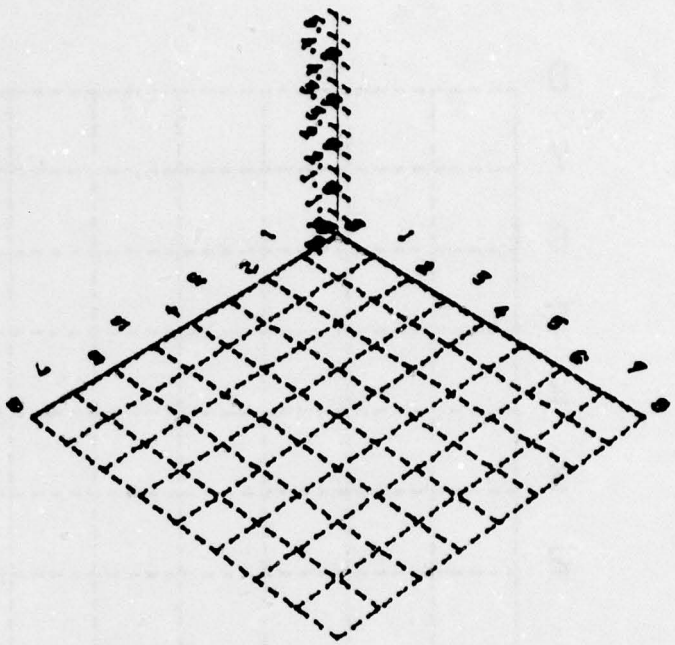
U3AXIS performs all operations in virtual space. This means that all axes and labels will be clipped if they are outside of the viewing pyramid. Which viewing pyramid will be used will depend upon the View Adjustment Options. If 'XYZVIEW' is specified, the current view will be used. This is the default. If 'XYVIEW', 'XZVIEW', or 'YZVIEW' is specified, a viewing vector is defined which is at positive 150. along the axis perpendicular to the designated plane. The view site is defined to be at (0.,0.,0.) in current units. Note that UWINDO/UDAREA modifications will only occur if 'XYZVIEW' is not specified.

Scaling options allow the user to select the intervals or allow U3AXIS to choose them. The Origin Inclusion Option allows the user to deselect the forced origin of the 'AUTOSCALE' and 'FULLSCALE' options. The default, 'ORIGIN', forces an origin for these options. 'NOORIGIN' will now allow the natural range of numbers of the U3AXIS

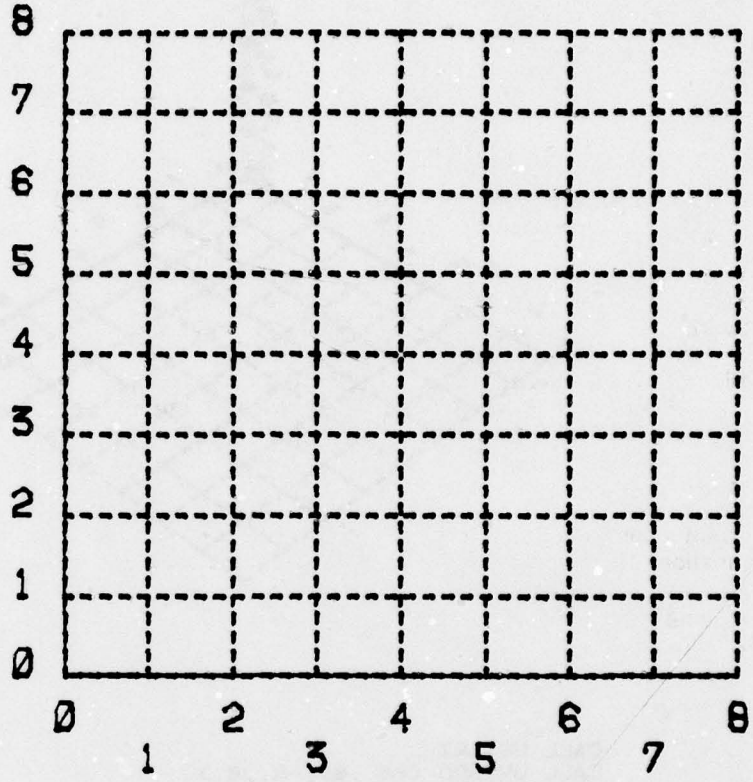
input arguments to determine the axis range. Note that under the 'OWNSCALE' option, origins are never forced. Use of the 'PENORIGIN' option allows the user to force the current pen position to be included in the axis range.

Grids will only be generated in the view adjustment plane (if specified) or the X-Y plane if 'XYZVIEW' is specified. The type of line to be used can be user selected by calling UPSET('GRID',value) where value is either 0. or a valid dash specification. If the default value of 0. is specified, a solid line will be used to create the grids.

Programming Notes:



```
CALL USTART
CALL UWINDO (-8.,8.,-8.,8.)
CALL UVIEW (30.,30.,30.,2.,2.,0.)
CALL UPSET ('GRID SPECIFICATION',34.)
CALL USET ('GRIDAXES')
CALL USET ('SOFTWARE CHARACTERS')
CALL UPSET ('HORIZONTAL',.3)
CALL UPSET ('VERTICAL',.5)
CALL UPSET ('LABEL',90.)
CALL UPSET ('TICPLUS',.3)
CALL UPSET ('TICMINUS',.3)
CALL U3AXIS (0.,8.,0.,8.,0.,5.)
CALL UEND
STOP
END
```




```
CALL USTART
CALL UWINDO (-8.,8.,-8.,8.)
CALL UVIEW (30.,30.,30.,2.,2.,0.)
CALL UPSET ('GRID SPECIFICATION',34.)
CALL USET ('GRIDAXES')
CALL USET ('SOFTWARE CHARACTERS')
CALL UPSET ('HORIZONTAL',.3)
CALL UPSET ('VERTICAL',.5)
CALL UPSET ('LABEL',90.)
CALL USET ('XYVIEW')
CALL U3AXIS (0.,8.,0.,8.,0.,5.)
CALL UEND
STOP
END
```

FUNCTION:

This routine invokes an already constructed graphics data structure. The requested coordinate system transformation is established,then each element of the structure is executed.

CALLING SEQUENCE:

CALL U3CALL(X,Y,Z,SX,SY,SZ,RX,RY,RZ,NAME)

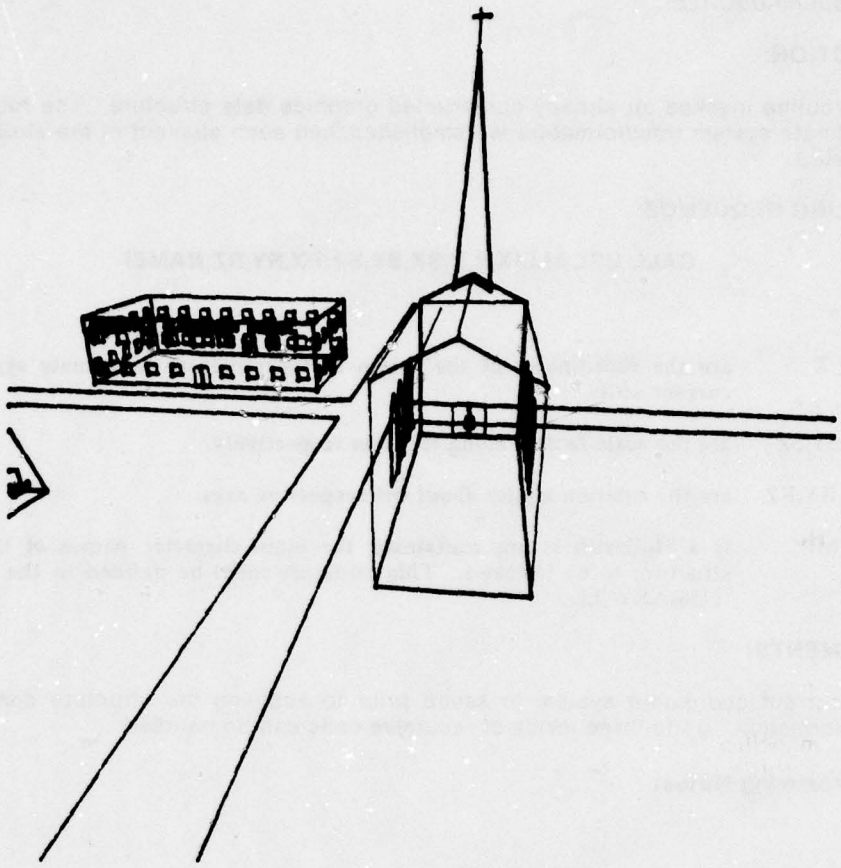
Where

X,Y,Z	**are the coordinates of the origin of the structure coordinate system in current units.**
SX,SY,SZ	**are the scale factors along the axes respectively.**
RX,RY,RZ	**are the rotation angles about the respective axes.**
NAME	**is a Hollerith string containing the eight-character names of the data structure to be invoked. This structure must be defined in the current 'LIBRARY' file.**

COMMENTS:

The current coordinate system is saved prior to applying the structure coordinate transformation. Up to three levels of recursive calls can be handled.

Programming Notes:



```
CALL ATTACH (8,'/TEK3DEMO/SAVE;',3,0,IST,)
CALL USTART
CALL UPSET ('LIBRARY',1.)
CALL UTILTY ('LOAD',8.)
CALL UPSET ('TERMINATOR','<')
CALL USET ('VIEWDISTANCE')
CALL UVWPRT (150.)
CALL UWINDO (-100.,100.,-100.,100.)
CALL UVIEW (-40.,200.,70.,-20.,20.,0.)
CALL VILLAG
CALL UEND
STOP
END
SUBROUTINE VILLAG
CALL USET ('XYZ ')
CALL USET ('SYSTEMAXIS')
CALL USET ('REFERENCEAXIS')
CALL USET ('BLACK')
CALL U3CALL (-50.,20.,0.,1.,1.,1.,90.,0.,0.,'CHURCH')
CALL USET ('RED ')
CALL U3CALL (24.,-18.,0.,1.,1.,1.,90.,-90.,0.,'SCHOOL')
CALL USET ('BLUE')
CALL U3CALL (70.,70.,0.,0.7,0.7,0.7,0.,0.,0.,'PIZZA')
CALL USET ('BLACK')
CALL U3CALL (0.,0.,0.,1.,1.,1.,0.,0.,0.,'ROAD ')
RETURN
END
```

FUNCTION:

This routine composes a new user coordinate system based on the current reference coordinate system using the input arguments for translation, scaling, and rotation operations.

CALLING SEQUENCE:

CALL U3CSYS(X,Y,Z,SX,SY,SZ,RX,RY,RZ)

Where

X,Y,Z	**the coordinates of the new origin in units of the old coordinate system.**
SX,SY,SZ	**are the multiplicative scale factors along each of the axes.**
RX,RY,RZ	**indicate the amount of rotation around each of the axes in current angular units.**

OPTIONS:

(Only those which differ from SUBROUTINE UCOSYS are listed)

Rotation Application Order: 'XYZ', 'XZY', 'ZXY', **'ZYX'**, 'YXZ', 'YZX'.

COMMENTS:

The composition of user coordinate systems is a powerful tool for use in describing problem pictures in their own environment. A full description of the use of this facility is available under UCOSYS and a tutorial is available in the primer. In using this facility in three-dimensions, only one additional requirement is imposed on the user: specification of the order in which the rotations will occur. This is a result of the non-commutativeness of the rotation operation. To set the order, the user merely specifies, as a USET option, the three characters which designate the axes in the order in which he wishes the rotations to be applied. The default setting is CALL USET('ZYX').

The positive direction of rotation is counterclockwise around the axis of rotation when viewed from a position on the axis of rotation looking in the negative direction.

Programming Notes:

FUNCTION:

This routine draws a solid line vector in 3-D space from the current pen position to the position specified by the input arguments.

CALLING SEQUENCE:

CALL U3DRAW(X,Y,Z)

Where

X,Y,Z **are the coordinates of the end point of the line in current units.**

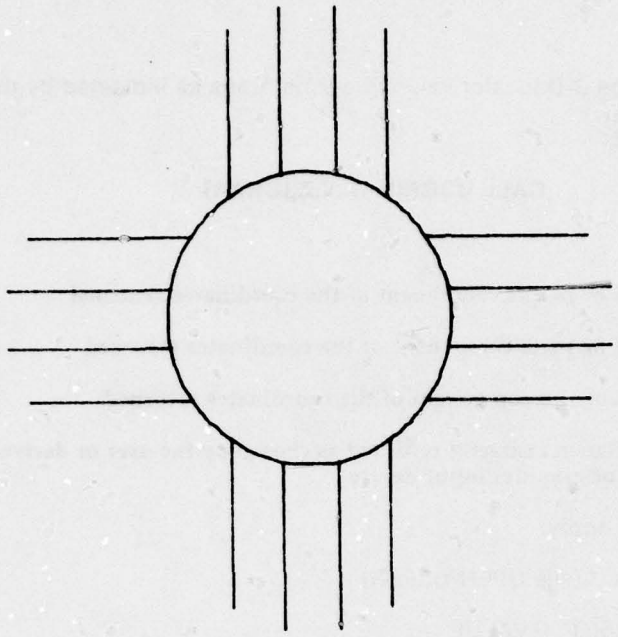
OPTIONS:

Pen Coordinate Options (see U3PEN)

COMMENTS:

This routine is equivalent to using U3PEN with line option 'LNULL'. The current setting of the line option has no effect on this routine.

Programming Notes:



ZIA Sun Symbol

```
      DIMENSION X(16),Y(16)
      DATA X/20.,20.,0.,0.,0.,0.,-20.,-20.,-20.,-20.,0.,0.,0.,0.
     &      ,20.,20./
      DATA Y/0.,0.,20.,20.,20.,20.,0.,0.,0.,0.,-20.,-20.,-20.,-20.
     &      ,0.,0./
      CALL USTART
      CALL UPSET ('TERMINATOR',',')
      CALL UCOSYS (50.,50.,1.,1.,0.)
      CALL USET ('POLAR')
      CALL UMOVE (17.5,0.)
      DO 10 I = 1, 120
   10 CALL UDRAW (17.5,FLOAT(I)*3.)
      THETA = 11.5
      DO 20 I = 1, 16
      CALL USET ('POLAR')
      CALL USET ('ABSOLUTE')
      CALL UMOVE (17.5,THETA)
      CALL USET ('RECTANGULAR')
      CALL USET ('RELATIVE')
      CALL UDRAW (X(I),Y(I))
   20 THETA = 11.25 + 22.5 * FLOAT(I)
      CALL USET ('LARGE')
      CALL USET ('CJUSTIFICATION')
      CALL USET ('ABSOLUTE')
      CALL UPRINT (0.,-45.,'ZIA S>UN <S>YMBOL;')
      CALL UEND
      STOP
      END
```

Subroutine U3GRIN

FUNCTION:

This routine returns the 3-D locator value in current units as indicated by the user.

CALLING SEQUENCE:

CALL U3GRIN (X,Y,Z,ICHAR)

Where

X	**is the X or radius component of the coordinates returned**
Y	**is the Y or theta component of the coordinates returned**
Z	**is the Z or rho component of the coordinates returned**
ICHAR	**is a Hollerith character returned as chosen by the user or derived from the actions of graphics input device**

OPTIONS which may apply:

All coordinate Options: (see UPEN/U3PEN)

Default ZVALUE: 'UPSET', 'ZVALUE'

Graphics Input Device: 'CURSOR', 'JOYSTICK', 'LIGHTPEN', 'TABLET', 'KEYBOARD', 'FUNCTION KEY', 'MOUSE', 'TRACKBALL'

COMMENTS:

If only a 2-D locator is available, the third component will be obtained from the current Z-VALUE setting.

Programming Notes:

FUNCTION:

This routine applies a general three-dimensional image transformation to the indicated segment/frame.

CALLING SEQUENCE:

CALL U3IMAG (X,Y,Z,SX,SY,SZ,RX,RY,RZ,SEGID)

Where

X,Y,Z	**is the new position of the segment in current 3D device units.**
SX,SY,SZ	**is the scale factor to be applied along each axis of the display surface.**
RX,RY,RZ	**in the rotation in current angular units to be applied arund each axis of the display surface.**
SEGID	**is the identifier of a "retained" segment/frame which was UOPENed for general 3D image transformation.**

OPTIONS which may apply:

Device Units: **'INCHES',** 'CENTIMETERS', 'RASTERUNITS', 'FONTUNITS', 'SPECIFICATION UNITS', 'PERCENT UNITS'

Specification Unit Size (UPSET): 'SPECIFICATION UNITS', 'XSPECIFICATION UNITS', 'YSPECIFICATION UNITS', 'ZSPECIFICATION UNITS'

Angular Units: **'DEGREES',** 'RADIANS', 'PIRADIANS', 'GRADS', 'MILS'

Segment Identifier Mode: **'FNAME',** 'FNUMBER', 'SNAME', 'SNUMBER'

COMMENTS:

The image transformation is applied in the order X-rotation, Y-rotation, Z-rotation, scaling, and translation. If the resultant image should exceed the display surface address space, the result is undetermined.

Image transformations are only applied if supported on the current display surface. Requests for image transformation will be ignored if the display device does not support this facility.

Programming Notes:

FUNCTION:

This routine creates a line in 3-D space by connecting an array of coordinates with lines drawn with the current line option.

CALLING SEQUENCE:

CALL U3LINE(X,Y,Z,PTS)

Where

X	**is an array of length PTS containing the X-components of the coordinates of the points to be connected.**
Y	**is an array of length PTS containing the Y-components of the coordinates of the points to be connected.**
Z	**is an array of length PTS containing the Z-components of the coordinates of the points to be connected.**
PTS	**is a real variable which specifies the number of points to be connected.**

OPTIONS which may apply:

All pen-related options (see U3PEN)

COMMENTS:

This routine will move to the first point specified and then draw a line to succeeding points until the last point is reached. The pen or beam will be left at the last point.

Programming Notes:

FUNCTION:

This routine draws an invisible vector in 3-D space from the current pen position to the position specified by the input arguments.

CALLING SEQUENCE:

CALL U3MOVE(X,Y,Z)

Where

X,Y,Z **are the coordinates of the end point of the line in current units.**

OPTIONS:

Pen Coordinate Options (see U3PEN)

COMMENTS:

This routine is equivalent to using U3PEN with line option 'NNULL'. The current setting of line option has no effect on this routine.

Programming Notes:

FUNCTION:

This routine draws a 3D vector of the type indicated by the current line option from the current beam position (vector tail) to the location specified by the arguments (vector head). The current beam position is updated to point to the head of the vector.

CALLING SEQUENCE:

CALL U3PEN (X,Y,Z)

Where

X	**is the X or RADIUS coordinate component of the head of the vector in current user units.**
Y	**is the Y or THETA coordinate component of the head of the vector in current user units.**
Z	**is the Z or RHO coordinate component of the head of the vector in current user units.**

OPTIONS which may apply:

Line Definition Options

Line Options

Tic Interval: UPSET('TICINTERVAL',value)

Tic Mark Size Options: UPSET('TICPLUS',value) UPSET('TICMINUS',value)

Dash Specification: UPSET('SETDASH',value)

System Character Setting: UPSET('MARKER',value)

Character Description Options: see U3PRNT.

Symbol/Marker Selection: UPSET('MARKER',value) UPSET('SYMBOL',value)

Symbol/Marker Size: UPSET('SZMARKER',value)

Coordinate Output Type: **'2DCOORDINATES'**, '3DCOORDINATES'

Gapped Line Mode: 'GAPPED', **'UNINTERRUPTED'**

Mapping/Projection Options: **'PERSPECTIVE'**, 'ORTHOGRAPHIC'

Pen Coordinate Options

Type: **'RECTANGULAR'**, 'POLAR', 'CYLINDRICAL', 'SPHERICAL'

Mode: **'ABSOLUTE'**, 'RELATIVE', 'XRELATIVE', 'YRELATIVE' 'ZRELATIVE', 'XABSOLUTE', 'YABSOLUTE', 'ZABSOLUTE', 'XYRELATIVE', 'XZRELATIVE', 'YZRELATIVE', 'XYABSOLUTE', 'XZABSOLUTE', 'YZABSOLUTE'

Space:	**'VIRTUAL'**, 'DEVICE'
Device Space Units:	'PERCENT', **'INCHES'**, 'CENTIMETERS', 'RASTER', 'FONT', 'SPECIFICATION'
Specification Unit Size:	UPSET('SPECIFICATION',value) UPSET('XSPECIFICATION',value) UPSET('YSPECIFICATION',value) UPSET('ZSPECIFICATION',value)
Coordinate System Selection:	**'SYSTEM'**, **'WORLD'**, 'USER', 'MODELLING'
User Coordinate System Type:	'REFERENCE', **'WORKING'**
Logarithmic Transforms:	'LOGARITHMIC', **'NOLOGARITHMS'**, 'XLOGARITHMIC', 'YLOGARITHMIC', 'ZLOGARITHMIC', 'XYLOGARITHMIC', 'XZLOGARITHMIC', 'YZLOGARITHMIC', 'XYZLOGARITHMIC'
Logarithm Application Time:	'LOGOBJECT', 'LOGUSER', **'LOGSYSTEM'**
Logarithm Base:	UPSET('BASE',value), UPSET('XBASE',value), UPSET('YBASE',value), UPSET('ZBASE',value)
ATTRIBUTES:	
Intensity	'BRIGHT', 'DIM', **'NORMAL'**, UPSET('BRIGHTNESS',value)
Line Width	**'THIN'**, 'WIDE', UPSET('WIDTH',value)

COMMENTS:

U3PEN is the central pen/beam movement subroutine of the three dimensional version of GCS. Not only is it called frequently by users in creating graphics images, it is also used extensively within the higher-level GCS routines.

To provide its power and flexibility, a large number of USET and UPSET options may effect any particular call to U3PEN. However, this should not be a concern for the less-experienced user since the defaults for these options will produce the effect he most likely desires, thus relieving him of the requirement to set any options but those which apply to this specific program.

These options have been classified into several major categories, each of which will be explained in detail. The visible output from U3PEN will also be effected by the viewing environment if plotting is occurring in virtual space. A full description of the concepts of the viewing environment can be found under UVIEW and UWINDO.

Programming Notes:

LINE DEFINITION OPTIONS:

This options apply to only the physical aspects of the lines created by U3PEN. Attributes which apply to both lines and characters are described under Attributes.

LINE OPTIONS:

The 40 available line options can be easily remembered if each is divided into its two component parts, the line type and the terminator type. The following table lists the available line and terminator types. Any line type may be combined with any terminator type to create a particular line option. This is done by prefixing the name of the terminator with the first character of the name of the line type. For example, to create a dashed arrow the first character of the line type "dashed" is "D". This letter is affixed to the word "arrow" (the name of the terminator) and the resulting USET line option is "DARROW". A ticked arrow would result if a "T" was substituted for the "D" to yield "TARROW". Calling UPEN with line option "NNULL" will yield identical results to calling UMOVE. Examples of the different line options are shown in Figure 1.

Line Type	Terminator Type
Null	NULL (absence of a terminator)
Line	POINT
Dashed	ARROW
Ticked	BACKARROW
Alpha	CHARACTER
	SYMBOL
	COORDINATES

Also available for the convenience of the user are synonyms for the most frequently selected line option. A table of these synonyms with their corresponding line/terminator specifications is included below:

Synonym	Line/Terminator Equivalent
ALPHALINE	ANULL
ARROW	LARROW
BACKARROW	LBACKARROW
CHARACTER	NCHARACTER
DASH	DNULL
DIMENSIONLINE	LDOUBLEARROW
DOUBLEARROW	LDOUBLEARROW
LINE	LNULL
MOVE	NNULL
NO LINE	NNULL
NO MARK	NNULL
NOMARK	NNULL
NOLINE	NNULL
PENDOWN	LNULL
PENUP	NNULL
POINT	NPOINT
SYMBOL	NSYMBOL
TICLINE	TNULL

Several other options may effect the appearance of some line options. These are tic-interval settings, dash specifications and system character settings. Each of these is described in more detail below.

TIC-INTERVAL SETTINGS:

The tic interval to be used when drawing ticked lines can be set by the UPSET option

"TICINTERVAL". The UPSET parameter value should be the distance between tic values in current user units (see Pen Coordinate Options below.) The user is cautioned that if he changes his space from device to virtual or visa versa then this may invalidate his tic-interval setting. Figure 2 shows the use of the "TICINTERVAL" UPSET parameter.

TIC MARK SIZE OPTIONS:

The size of the individual tic marks may be specified. GCS allows the tic marks to be of different sizes on each side of the line. If the line along which the tic marks are drawn is considered to be an X-axis and is rotated about its tail in the XY plane. When viewed from the positive Z-direction, the side towards positive angles of rotation is considered the 'TICPLUS' side and the side towards negative angles of rotation is considered the 'TICMINUS' side. For 2D applications, the 'TICPLUS' side is in the positive Y direction and the 'TICMINUS' side is towards the negative Y direction if the line is considered to be an X-axis. The tic mark sizes may then be set by the following UPSET options:

UPSET('TICPLUS',value) — The 'TICPLUS' size is set to the length value specified in current units.

UPSET('TICMINUS',value) — The 'TICMINUS' size is set to the length value specificed in current units.

DASH SPECIFICATION:

The dash line specification which UPEN will use can be adjusted to produce a wide variety of dashed lines. This adjustment is made by the UPSET option 'SETDASH' where the parameter value is a number which defines the dashed line as follows. Single digits allow selection of hardware-generated dashed lines if that capability exists. The digits 1-8 will produce up to 8 different dashed lines depending on the capability of the hardware. The digit 9 will produce a hardware-dotted line. Should hardware dashed lines not be available on the terminal being used, a default software dashed line will be generated.

Software dashed line specifications consist a floating print number, made of two or more digits which are evaluated left to right, digit by digit. Each digit represents a line segment which is either visible or invisible of varying length as specified in the following table.

Digit	Length (inches)	Visibility
1	.0366	Visible
2	.0366	Invisible
3	.0733	Visible
4	.0733	Invisible
5	.1831	Visible
6	.1831	Invisible
7	.3662	Visible
8	.3662	Invisible
9	.0073 (dot)	Visible

When the entire integer has been evaluated, scanning restarts at the left most digit. For example, dash code 34. will result in a visible line of .0733 inches followed by an invisible line of .0733 inches followed by a visible line of .0733 inches, etc., continuing until the head of the vector is reached. The last segment may be a shortened visible or invisible segment. If, however, the next UPEN movement is a dashed line starting at the head of the previous vector, the last segment of this previous vector will be completed and the sequence continued for the new vector.

The complexity of the dashed line specification is limited only by the number of digits of precision available in the computer floating point numbers. Moreover, visible line

segments need not be alternated with invisible line segments. Visible line segments may be combined with other visible line segments to create line segments of length not available through single digits. Similarily, this applies to invisible line segments.

SYSTEM CHARACTER SETTING:

The system character is used to create ALPHA lines and to indicate which character should be used to terminate UPEN operations when the line option specifies the 'CHARACTER' terminator. The system character may be set by a call to USET where the argument is the single character to become the system character. For example, the following statement would set the system character to the dollar sign:

```
CALL UPSET ('CHARACTER','$')
```

Should it also be disired to change the line type as well as the system character, the character to be specified in the USET argument may be prefixed with one of the five available line types. The following statement would change the system character to "+", and would also change the line option to 'DCHARACTER':

```
CALL USET ('D+')
```

The default system character is the asterisk (*).

Also to be considered is the case in which GCS is set at the time the system character is set. The default case is UPPERCASE. However, if calls to UPRINT have left GCS in LOWERCASE mode, any attept to set the system character will generate a lower case character. The desired case may be set by using one of the following

```
CALL USET('UPPERCASE')
CALL USET('LOWERCASE')
```

See UPRINT for a complete description of case shifting.

CHARACTER TERMINATOR:

The character terminator which can be attached to any of the line types will appear as either a hardware character, a simulated hardware character, or a software character depending on the current user setting of 'HARDWARE', 'SIMULATED' or 'SOFTWARE'. Hardware characters and simulated hardware characters will have their lower-left corner on the head of the vector. Software characters will have the head of the vector at the center of the character. The appearance of hardware, simulated hardware, or software characters may be affected by the setting of the italicization and character font option if the feature is available and has been implemented. Regardless to which type of character is specified, at the end of the UPEN operation, the beam on pen will be located at the head of the vector. See U3PRNT for a more detailed description of the character options.

CHARACTER DESCRIPTION OPTIONS:

The character description options are fully described in U3PRNT. Most character description options apply when using character or coordinate line terminators.

SYMBOL/MARKER TERMINATOR:

The symbol terminator is available so that the location of the head of each line may be uniquely marked. There are two types of symbols. The first type are GCS-defined symbols. The second type are installation-defined or device-defined symbols. The symbol is always displayed centered at the head of the line and the resulting beam/pen position is at the head of the line. Should a marker protrude over a display surface edge

or the edge of a window, the results are undefined (i.e., the marker may be partially slipped or may wraparound).

SYMBOL/MARKER SELECTION:

Symbols are selected by UPSET('MARKER',index) or UPSET('SYMBOL',index) where index is a non-negative real number whose integer value is used to select the symbol. Specifications of an index outside of the range of valid indecies will produce symbol X.

SYMBOL/MARKER SIZE:

The size of symbols/markers may be specified by UPSET('SZMARKER',value) where value is a positive number in current device units. Markers generated by hardware symbol generators may be restricted as to size variations. If so, the closest size (perhaps the only size) will be used.

COORDINATE TERMINATORS:

The coordinate terminators will generate a set of coordintes equivalent to the location value of the head of the vector in current user units enclosed in parentheses and separated by a comma. The result will be in the same format as if the coordinates had been produced by a call to UPRINT. For a more complete discussion of this format see UPRINT.

COORDINATE OUTPUT TYPE:

The dimensionality of coordinate output is determined by the setting of the USET options '2DCOORDINATES' and '3DCOORDINATES'. If '2DCOORDINATES' are specified, then only the (X,Y) *or* (R,THETA) components are displayed. With '3DCOORDINATES', all three coordinate components are displayed.

GAPPED LINE MODE:

If 'GAPPED' lines are specified, successfive calls to U3PEN will alternate drawing lines of the current line option with 'NNULL' (invisible) lines. This feature can be deactivated by specifying 'UNINTERRUPTED' lines which are the default.

MAPPING/PROJECTION OPTIONS:

The user may specify the type of mapping to be applied in projecting three-dimensional lines onto the viewport. The two available mapping are 'ORTHOGRAPHIC' in which the lines of projection are parallel and perpendicular to the projection plane, and 'PERSPECTIVE' in which line length diminishes as the distances from the viewing position (view point) becomes greater. 'PERSPECTIVE' is the default. The difference between these two options is shown in Figure 9.

PEN COORDINATE OPTIONS:

The GCS Pen Coordinate Options have been designed to provide the flexibility which will allow the user to work in his own coordinate type and unit while retaining the simplicity of operation and specification which the infrequent and/or unsophisticated user will find convenient. Thus, the Pen Coordinate Options have been divided into several classes each of which allows setting of its options completely independently of any effect on any of the others. The following paragraphs will discuss each class in detail.

COORDINATE TYPE:

This indicates the kind of coordinates the user will specify and how they will be interpreted for display.

'RECTANGULAR' — Coordinates are of the standard forms (X,Y) or (X,Y,Z) where X is the number of units along the X-axis, Y is the number of units along the Y-axis, and Z is the number of units along the Z-axis. This is the standard default.

'POLAR', 'CYLINDRICAL' — Coordinates are of the forms (R,THETA) or (R,THETA,Z) where R is the number of units of radius in the XY-plane, THETA is the number of angular units from the X-axis in the XY-plane, and Z is the number of units along the Z-axis. (See UARC for a description of angular units.)

'SPHERICAL' — Coordinates are of the form (RHO, THETA, PHI) where RHO is the number of units of radius, THETA is the number of angular units from the X-axis around the Z-axis in the XY-plane, and PHI is the number of angular units from the Y-axis around the X-axis.

COORDINATE MODE:

This indicates where the base point or origin of the coordinate units is located. Since the conversion from relative to absolute occurs first, the (X,Y,Z) components described below may be (R,THETA,Z) or (RHO,THETA,PHI) components if the user has specified a coordinate which is not rectangular.

'ABSOLUTE' — All coordinate components are indicated based on the origin of the current coordinate system. This is the default case

'RELATIVE' — All coordinates are indicated with respect to the current beam/pen position. At the completion of any beam/pen movement, the beam will be at relative coordinates (0,0,0).

'XABSOLUTE', 'YZRELATIVE' — The X-component is absolute and the Y- and Z-components are relative.

'YABSOLUTE', 'XZRELATIVE' — The Y-component is absolute and the X- and Z-components are relative.

'ZABSOLUTE', 'XYRELATIVE' — The Z-component is absolute and the X- and Y-components are relative.

'XRELATIVE', 'YZABSOLURE' — The X-component is relative and the Y- and Z-components are absolute.

'YRELATIVE', 'XZABSOLUTE' — The Y-component is relative and the X- and Z-components are absolute.

'ZRELATIVE', 'XYABSOLUTE' — The Z-component is relative and the X- and Y-components are relative.

NOTE: Use of relative coordinates will not necessarily generate device commands using relative coordinates.

COORDINATE SPACE:

This indicates the address space to which the coordinates refer.

'VIRTUAL' — The coordinates indicate a point in virtual space. Whether this point is visible on the display surface will depend on the current viewing environment (for a complete discussion of the viewing environment, see UVIEW and UWINDO). For 2-D applications, only the UWINDO settings need be considered. These coordinates can encompass any of the real numbers within the capbilities of the computer system. No meaning is attached to these numbers by GCS; therefore, the user is free to assign whatever meaning he desires to the numbers.

'DEVICE' — The coordinates indicate a point on the 'DISPLAY' physical device surface. Specification of points not on the display surface may yield indeterminate and/or spurious results. Since the lower left front corner of the display surface always has address (0,0,0), all negative coordinates may yield spurious results. The units which these numbers represent are described in the next paragraph.

DEVICE SPACE UNITS:

This indicates the interpretation to be given to the numbers used in plotting in 'DEVICE' space (see previous paragraph). These options have no meaning and are not considered when plotting in 'VIRTUAL' space.

'INCHES' — The coordinate values refer to actual inches on the display surface. This will be true regardless of the device being used. These units are device-dependant since the size of the display surface can vary from device to device.

'CENTIMETERS' — The coordinate values refer to actual centimeters on the display surface. This will be true regardless of the device being used. These units are device-dependent since the size of the display surface can vary from device to device.

'RASTER UNITS' — The coordinate values refer to the individual addressable positions on any particular device. These units are device-dependant and may not be transferable to other terminal devices since the size of the device address space may vary from device to device.

'FONTUNITS' — The coordinate values refer to the lower left front corner of individual character positions. The size of a character position is dependent upon the current size of the hardware, simulated hardware, or software characters (whichever is currently selected). It should be noted that these units need not be integers. On most terminals, characters can be placed anywhere on the display surface. These units are device-dependent since the size of hardware characters and the size of the display surface can vary from device to device. For a complete description of hardware, simulated hardware, and software characters, see U3PRNT.

'PERCENTUNITS' — The coordinate values refer to hundredth increments of the display surface address space. This is truly a device-dependent unit. Moving an image created on one device in 'PERCENTUNITS' to another device will display an image using the same proportional amount of the display

surface. It should be noted that if a device does not have a square display surface (aspect ratio = 1), the value of a 'PERCENTUNIT' in the X-direction will differ from the value of a 'PERCENTUNIT' in the Y-direction. See UASPCT for a description of the effect on 'PERCENTUNITS' of setting the aspect ratio of the display surface.

'SPECIFICATION' UNITS' — The coordinate values refer to user-specified increments of the display surface address space. These units are device-dependent and function similarly to the 'PERCENTUNITS' described above. 'SPECIFICATION UNITS' are defined as described in the next paragraph. Setting the 'SPECIFICATIONUNITS' to 100. in each direction results in 'SPECIFICATIONUNITS' which are identical to 'PERCENTUNITS'. The default units are 1000., along each display surface address space axis.

SPECIFICATION UNIT SIZE:

The number of specification units which occupy the display address space may be set by the following UPSET options. The default for each is 1000 units.

UPSET('SPECIFICATIONUNITS',value) — This sets the number of specification units along each axis of the display surface to the value specified.

UPSET('XSPECIFICATIONUNITS',value) — This sets the number of specification units along the X-axis of the display surface to the value specified. The number of units along the Y- and Z- axis remain uneffected.

UPSET('YSPECIFICATIONUNITS',value) — This sets the number of specification units along the Y-axis of the display surface to the value specified. The number of units along the X- and Z-axis remain uneffected.

UPSET('ZSPECIFICATIONUNITS',value) — This sets the number of specification units along the Z-axis of the display surface to the value specified. The number of units along the X- and Y-axis remain uneffected.

COORDINATE SYSTEM:

There are two types of coordinate systems available in GCS. One of these, the 'SYSTEM' or 'WORLD' coordinate system, is defined by GCS. The other, the 'USER' or 'MODELLING' coordinate system, is defined by the user. The 'USER' coordinate system is defined based on the 'WORLD' coordinate system or on other 'USER' coordinate systems. The 'USER' coordinate system can be cumulative or non-cumulative depending on whether the 'USER' coordinate system being created is to be a new 'REFERENCE' system or is to be a 'WORKING' system respectively. See U3CSYS and UCOSYS for a complete description of the 'USER' coordinate system facility. While 'USER' coordinate systems may be applied in 'DEVICE' space, the user is warned that a 'USER' coordinate system constructed in one coordinate space may not be correct if used in the other space.

'SYSTEM', 'WORLD' — If in 'DEVICE' space, GCS defines this coordinate system to refer to the unrotated, unscaled system inherent in the display surface addressing scheme. This normally places the origin at the lower left front corner of the display surface. If in 'VIRTUAL' space, this coordinate system is

unrotated with the origin located in virtual space at a position represented the computer system real numbers (0.,0.,0.). The 'SYSTEM' or 'WORLD' coordinate system is the default setting. The 'WORLD' coordinate system may be USET to be either 'LEFTHANDED' or 'RIGHTHANDED'. Default is 'RIGHTHANDED'.

'USER', 'MODELLING' — The user may define his own coordinate system based on the currently specified coordinate system whether it be 'SYSTEM' or 'USER'. The default 'USER' coordinate systems are identical to the 'SYSTEM' coordinate system. GCS maintains two 'USER' coordinate system: a 'REFERENCE' system and a 'WORKING' system. These are defined below.

USER COORDINATE SYSTEM TYPE:

This specifies which of the two GCS-maintained 'USER' coordinate systems are to be used.

'REFERENCE' — This selects the 'REFERENCE' user coordinate system. The effect of creating a new 'REFERENCE' system is cumulative as it is based on the previous 'REFERENCE' system. See UCOSYS and U3CSYS for details. This setting has no effect when using the 'SYSTEM' coordinate system.

'WORKING' — This selects the 'WORKING' user coordinate system. The effect of creating a new 'WORKING' system is non-cumulative as it is based on the current 'REFERENCE' system. See UCOSYS and U3CSYS for details.

LOGARITHMIC SCALING:

This option allows the application of a logarithmic scale factor along any of the coordinate components. Any logarithmic base is supported and may be individually selected for each component (see below). Note that if the coodinate type is other than 'RECTANGULAR', the logarithmic scale factor may be applied to the angular unit components if desired (e.g., 'YLOGARITHMIC' would apply this factor to the THETA component of 'CYLINDRICAL' or 'POLAR' coordinates). See the Logarithmic Application Time options described below.

Application of a logarithmic scale factor will modify the size of the plot in relation to he window since the window is now looking at exponents rather than actual values. This may be adjusted for by some of the higher-level graphing routines (see U3AXIS).

'LOGARITHMIC'
'XYZLOGARITHMIC' — A logarithmic scale factor is applied along each component

'NOLOGARITHMS' — Logarithmic scaling is disabled. This is the default.

'XLOGARITHMIC'
'YLOGARITHMIC'
'ZLOGARITHMIC'
'XYLOGARITHMIC'
'XZLOGARITHMIC'
'YZLOGARITHMIC' — The appropriate logarithmic scale factor is applied to the component indicated. Other components remain linear. 2-D applications should use 'XYLOGARITHMIC' to avoid conflicts with the default Z-value of 0.

LOGARITHM APPLICATION TIME:

The effect of logarithmic scaling is dependent upong the state of the coordinates at the time the scaling is applied. Three options allow the user to direct the time at which the scaling is applied as follows:

'LOGOBJECT' — Logarithmic scaling is applied to the coordinates provided by the user after conversion to 'ABSOLUTE' but prior to any other transformations. In this mode, log scaling may be applied to the angle and/or radius components of 'CYLINDRICAL', 'POLAR', or 'SPHERICAL' coordinates.

'LOGUSER'
'LOGMODELLING' — Logarithmic scaling is applied after conversion to 'ABSOLUTE', 'RECTANGULAR', 'USER' coordinates but before conversion to 'SYSTEM' coordinates. In this mode, logarithmic scaling will be applied along the current 'USER' coordinate system axes if 'USER' is specified. If 'SYSTEM' is specified, 'LOGUSER' is equivalent to 'LOGSYSTEM'.

'LOGSYSTEM'
'LOGWORLD' — Logarithmic scaling is applied after conversion to 'SYSTEM' coordinates. In this mode, the logarithmic scaling will be applied along the 'SYSTEM' coordinate system axes. This is the default mode.

LOGARITHM BASE:

The base to be used for logarithmic scaling can be selected for all coordinate components or individually for each coordinate component by using the following UPSET options. The value can be any positive real number or the character strings "E" or "e".

UPSET('BASE',value) — The base for logarithmic scaling along each component is set to the value specified.

UPSET('XBASE',value) — The base for logarithmic scaling along the X-component is set to the value specified. The other bases remain uneffected.

UPSET('YBASE',value) — The base for logarithmic scaling along the Y-component is set to the value specified. The other bases remmain uneffected.

UPSET'ZBASE',value) — The base for logarithmic scaling along the Z-component is set to tha value specified. The other bases remain uneffected.

INTENSITY:

The brightness of the output on the display surface may be specified if the device has variable intensity settings. Intensity settings are specified by UPSET('BRIGHTNESS',value) where the value is a percentage along the range of allowable intensity settings from minimum to maximum. This 0. is the lowest intensity setting and 100. is the highest intensity setting. Intensities may also be specified by the following USET options. Default intensity level is 60%.

'DIM' — The intensity is set to the 10% level.

'NORMAL' — The intensity is set to the 60% level.

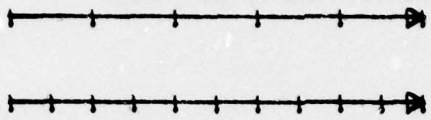
'BRIGHT' — The intensity is set to the 100% level.

LINE WIDTH:

The line width (sometimes referred to as spot size) may be specified if the device has a variable line width. Line widths are specified by UPSET ('WIDTH',value) where the value represents the width of the line in current units. The line width may also be set using the following USET options. The default width is 'THIN'.

'THIN' — The line width is set to the thinnest line. If a simulator is being used, a single line is generated.

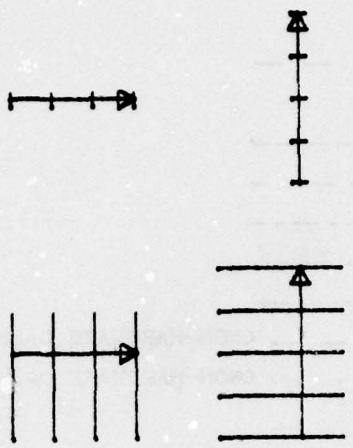
'WIDE' — The line width is set to the widest line. If a simulator is being used, a line approximately .1 inches in width is generated.



Tic Interval Specification

Figure 2 (USPEN)

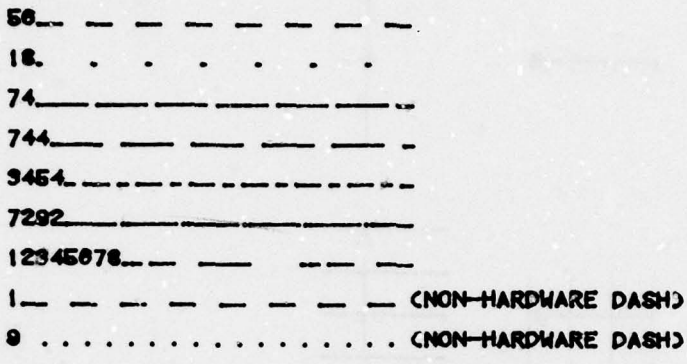
```
CALL USTART
CALL UPSET ('TERMINATOR',',')
CALL USET ('TARROW')
CALL UMOVE (0.,50.)
CALL UPSET ('TICINTERVAL',10.)
CALL UPEN (50.,50.)
CALL UMOVE (0.,40.)
CALL UPSET ('TICINTERVAL',5.)
CALL UPEN (50.,40.)
CALL USET ('ACENTER')
CALL USET ('LARGE')
CALL UPRINT (25.,30.,'T>IC <I>NTERVAL <S>PECIFICATION<,')
CALL UPRINT (25.,25.,'F>IGURE 2 <(USPEN);')
CALL UEND
STOP
END
```



Tic Mark Sizes

Figure 3 (U3PEN)

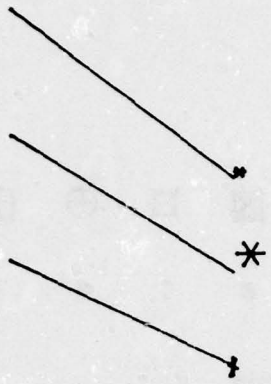
```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('TARROW')
CALL UPSET ('TICINTERVAL',5.)
CALL UMOVE (5.,70.)
CALL UPEN (20.,70.)
CALL UMOVE (40.,60.)
CALL UPEN (40.,60.)
CALL UPSET ('TICPLUS',10.)
CALL UPSET ('TICMINUS',5.)
CALL UMOVE (5.,40.)
CALL UPEN (20.,40.)
CALL UMOVE (40.,30.)
CALL UPEN (40.,50.)
CALL USET ('ACENTER')
CALL USET ('LARGE')
CALL UPRINT (25.,15.,'T>IC <M>ARK <S>IZES;')
CALL UPRINT (25.,10.,'<F>IGURE 3 <<U3PEN);')
CALL UEND
STOP
END
```



Examples of Dashed Lines

Figure 4 (U3PEN)

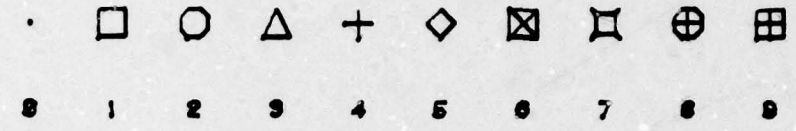
```
      DIMENSION DTABLE(8)
      DATA DTABLE/0.,1.,12345678.,7292.,3454.,744.,74.,18.,56./
      CALL USTART
      CALL UPSET ('TERMINATOR',';')
      CALL USET ('LARGE')
      CALL USET ('INTEGER')
      CALL USET ('DNULL')
      DO 10 I = 30,70,5
      Y = I
      J = (I-30) / 5 + 1
      CALL UPSET ('SETDASH',DTABLE(J))
      CALL UPRINT (0.,Y,DTABLE(J))
   10 CALL UPEN (50.,Y)
      CALL USET ('TEXT')
      CALL UPRINT (50.,35.,'(NON-HARDWARE DASH);')
      CALL UPRINT (50.,30.,'(NON-HARDWARE DASH);')
      CALL USET ('ACENTER')
      CALL UPRINT (50.,20.,'E>XAMPLES OF <D>ASHED <L>INES;')
      CALL UPRINT (50.,15.,'<F>IGURE 4 <U3PEN);')
      CALL UEND
      STOP
      END
```



Character Terminator Output

Figure 5 (U3PEN)

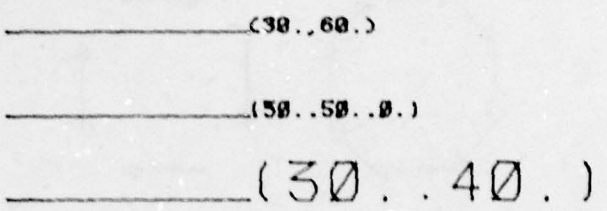
```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('PERCENT UNITS')
CALL USET ('LCHARACTERS')
CALL U3MOVE (10.,80.,0.)
CALL U3PEN (50.,80.,-50.)
CALL USET ('SIMULATED HARDWARE CHARACTERS')
CALL UPSET ('XSIZE',5.)
CALL UPSET ('YSIZE',7.)
CALL U3MOVE (10.,65.,0.)
CALL U3PEN (50.,65.,-50.)
CALL USET ('SOFTWARE CHARACTERS')
CALL U3MOVE (10.,50.,0.)
CALL U3PEN (50.,50.,-50.)
CALL USET ('ACENTER')
CALL USET ('HARDWARE CHARACTERS')
CALL USET ('LARGE')
CALL U3PRNT (25.,30.,0.,'C>HARACTER <T>ERMINATOR <O>UTPUT,')
CALL U3PRNT (25.,25.,0.,'<F>IGURE 5 <(U3PEN);')
CALL UEND
STOP
END
```



GCS-Defined Symbols

Figure 6 (USPEN)

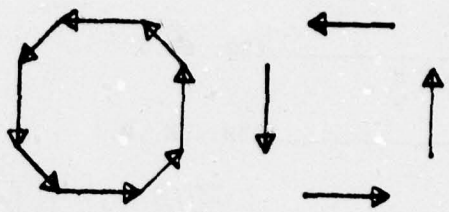
```
      CALL USTART
      CALL UPSET ('TERMINATOR',',')
      CALL USET ('NSYMBOL')
      CALL USET ('ACENTER')
      CALL USET ('INTEGER')
      CALL USET ('LARGE')
      CALL UPSET ('SZMARKER',.3)
      DO 10 I = 5,95,10
      X = I
      SYM = (I-5) / 10
      CALL UPSET ('SYMBOL',SYM)
      CALL UPEN (X,80.)
   10 CALL UPRINT (X,50.,SYM)
      CALL USET ('TEXT')
      CALL UPRINT (50.,40.,'GCS-D>EFINED <S>YMBOLS;')
      CALL UPRINT (50.,35.,'<F>IGURE 6 <(USPEN);')
      CALL UEND
      STOP
      END
```



Coordinate Terminators

Figure 7 (U3PEN)

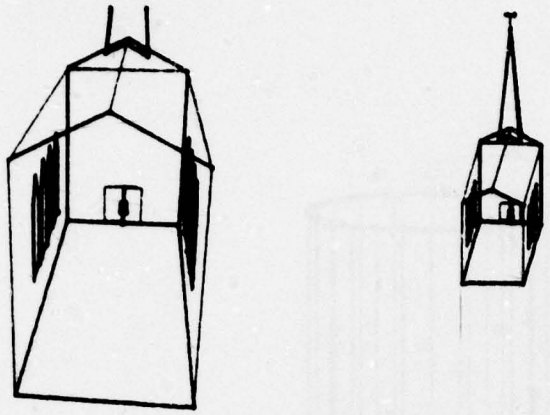
```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('LCOORDINATES')
CALL USET ('LARGE')
CALL UMOVE (0.,60.)
CALL UPEN (30.,60.)
CALL USET ('SIMULATED HARDWARE')
CALL USET ('3DCOORDINATES')
CALL UMOVE (0.,50.)
CALL UPEN (30.,50.)
CALL USET ('SOFTWARE')
CALL USET ('2DCOORDINATES')
CALL UMOVE (0.,40.)
CALL UPEN (30.,40.)
CALL USET ('ACENTER')
CALL USET ('HARDWARE')
CALL UPRINT (30.,30.,'C>OORDINATE <T>ERMINATORS;')
CALL UPRINT (30.,25.,'<F>IGURE 7 <(U3PEN);')
CALL UEND
STOP
END
```



Gapped Lines

Figure 6 (USPEN)

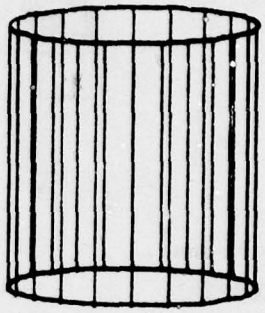
```
      CALL USTART
      CALL UPSET ('TERMINATOR',',')
      CALL USET ('LARROW')
      CALL USET ('UNINTERRUPTED')
      ILOOP = 0
      XLOC = 25.
   10 IF (ILOOP .EQ. 2) GO TO 20
      CALL UMOVE (XLOC+5.,30.)
      CALL UPEN (XLOC+15.,30.)
      CALL UPEN (XLOC+20.,35.)
      CALL UPEN (XLOC+20.,45.)
      CALL UPEN (XLOC+15.,50.)
      CALL UPEN (XLOC+5.,50.)
      CALL UPEN (XLOC,45.)
      CALL UPEN (XLOC,35.)
      CALL UPEN (XLOC+5.,30.)
      ILOOP = ILOOP + 1
      XLOC = 55.
      CALL USET ('GAPPED')
      GO TO 10
   20 CALL USET ('ACENTER')
      CALL USET ('LARGE')
      CALL UPRINT (50.,20.,'G>APPED <L>INES,')
      CALL UPRINT (50.,15.,'<F>IGURE 6 <<USPEN),')
      CALL UEND
      STOP
      END
```



Projection Modes

Figure 9 (U3PEN)

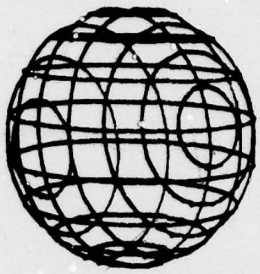
```
CALL ATTACH (2,'/TEK3DEMO/SAVE;',3,0,IST,)
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL UASPCT (1.)
CALL UPSET ('LIBRARYFILE',1.)
CALL UTILTY ('LOAD',2.)
CALL USET ('ACENTER')
CALL USET ('LARGE')
CALL UPRINT (50.,25.,'P>ROJECTION <M>ODES;')
CALL UPRINT (50.,20.,'<F>IGURE 9 <<U3PEN>;')
CALL USET ('NOCENTER')
CALL USET ('REFERENCE SYSTEM')
CALL UVIEW (-40.,200.,70.,-20.,20.,0.)
CALL UWINDO (-100.,100.,-100.,100.)
CALL UVWPRT (150.)
CALL USET ('PERSPECTIVE')
CALL USET ('PERCENT UNITS')
CALL UDAREA (0.,50.,25.,75.)
CALL U3CALL (-50.,20.,0.,1.,1.,1.,90.,0.,0.,'CHURCH')
CALL USET ('ORTHOGRAPHIC')
CALL UDAREA (50.,100.,25.,75.)
CALL U3CALL (-50.,20.,0.,1.,1.,1.,90.,0.,0.,'CHURCH')
CALL UEND
STOP
END
```



Cylindrical Coordinates

Figure 18 (U3PEN)

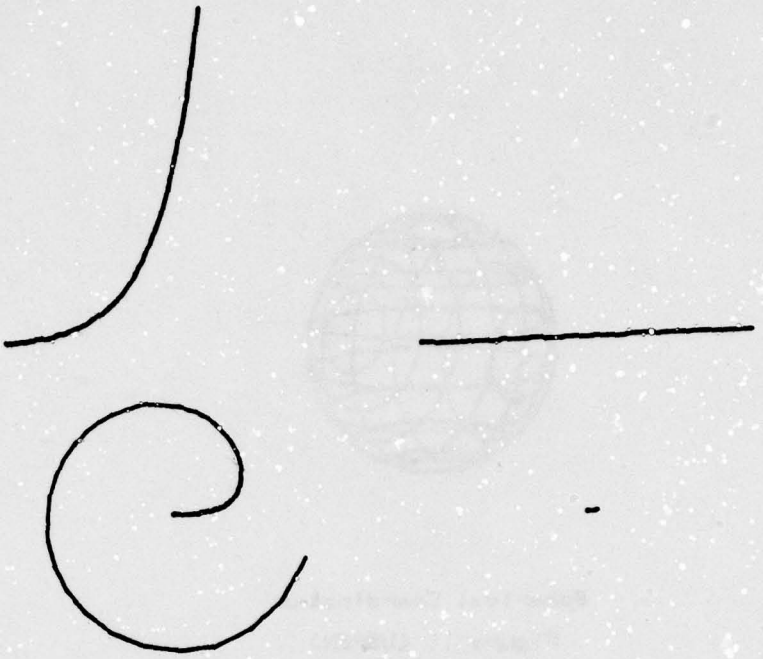
```
      CALL USTART
      CALL UPSET ('TERMINATOR',',')
      CALL USET ('LARGE')
      CALL USET ('ACENTER')
      CALL UPRINT (50.,20.,'C>YLINDRICAL <C>OORDINATES;')
      CALL UPRINT (50.,15.,'<F>IGURE 18 <(U3PEN);')
      CALL UVIEW (150.,150.,30.,0.,0.,30.)
      CALL UWINDO (-100.,100.,-100.,100.)
      CALL USET ('CYLINDRICAL')
      CALL UCRCLE (0.,0.,30.)
      CALL UPSET ('ZVALUE',60.)
      CALL UCRCLE (0.,0.,30.)
      DO 10 I=1,346,15
      THETA = I - 1
      CALL U3MOVE (30.,THETA,0.)
   10 CALL U3PEN (30.,THETA,60.)
      CALL UEND
      STOP
      END
```



Spherical Coordinates

Figure 11 (U3PEN)

```
      CALL USTART
      CALL UPSET ('TERMINATOR',';')
      CALL USET ('ACENTER')
      CALL USET ('LARGE')
      CALL UPRINT (50.,20.,'S>PHERICAL <C>OORDINATES;')
      CALL UPRINT (50.,15.,'<F>IGURE 11 <(U3PEN);')
      CALL UVIEW (150.,150.,30.,0.,0.,0.)
      CALL UWINDO (-100.,100.,-100.,100.)
      CALL USET ('SPHERICAL COORDINATES')
      RADIUS = 30.
      ILOOP = 0
   10 IF (ILOOP .EQ. 2) GO TO 20
      DO 15 I = 1, 181, 20
      RHO = I - 1
      CALL U3MOVE (RADIUS,0.,RHO)
      DO 15 J = 1, 361, 10
      THETA = J - 1
   15 CALL U3PEN (RADIUS,THETA,RHO)
      CALL U3MOVE (0.,0.,0.)
      CALL U3ROTA (0.,90.,0.)
      ILOOP = ILOOP + 1
      GO TO 10
   20 CONTINUE
      CALL UEND
      STOP
      END
```



Logarithmic Scaling

Figure 12 (USPEN)

```
      DIMENSION X(101),Y(101)
      CALL USTART
      CALL UPSET ('TERMINATOR',',')
      CALL UASPCT (1.)
      DO 10 I = 1, 101
      X(I) = I
   10 Y(I) = 10.**(X(I)/30.)
      CALL USET ('PERCENT UNITS')
      CALL UDAREA (5.,45.,60.,100.)
      CALL UMOVE (X(1),Y(1))
      DO 20 I = 2, 101
   20 CALL UPEN (X(I),Y(I))
      CALL USET ('Y_LOGARITHMIC')
      CALL UDAREA (55.,95.,60.,100.)
      CALL UMOVE (X(1),Y(1))
      DO 30 I = 2, 101
   30 CALL UPEN (X(I),Y(I))
      DO 40 I = 1, 101
      X(I) = FLOAT(I) / 1.2
   40 Y(I) = 2.**(X(I) / 10.)
      CALL USET ('POLAR')
      CALL UWINDO (-100.,100.,-100.,100.)
      CALL USET ('NOLOGARITHMS')
      CALL UDAREA (5.,45.,20.,60.)
      CALL UMOVE (X(1),Y(1))
      DO 50 I = 2, 101
   50 CALL UPEN (X(I),Y(I))
      CALL UDAREA (55.,95.,20.,60.)
      CALL USET ('XYLOGARITHMIC')
      CALL UPSET ('BASE',2.)
```

```
      CALL USET ('LOGOBJECT')
      CALL UMOVE (X(1),Y(1))
      DO 60 I = 2, 101
   60 CALL UPEN (X(I),Y(I))
      CALL USET ('NOLOGARITHMS')
      CALL USET ('RECTANGULAR')
      CALL UWINDO (0.,100.,0.,100.)
      CALL UDAREA (0.,100.,0.,100.)
      CALL USET ('LARGE')
      CALL USET ('ACENTER')
      CALL UPRINT (50.,20.,'L>OGARITHMIC <S>CALING,')
      CALL UPRINT (50.,15.,'<F>IGURE 12 <CUSPEN),')
      CALL UEND
      STOP
      END
```

FUNCTION:

This routine sets the USET option provided, draws the indicated 3D line using the current line option, and then restores the value changed by the USET option.

CALLING SEQUENCE:

CALL U3PEN1(X,Y,Z,OPTION)

Where

X	**is the X (RADIUS) coordinate of the head of the beam/pen movement**
Y	**is the Y (THETA) coordinate of the head of the beam/pen movement**
Z	**is the Z (PHI) coordinate of the head of the beam/pen movement**
OPTION	**is the USET option to be set only for the execution of the subroutine**

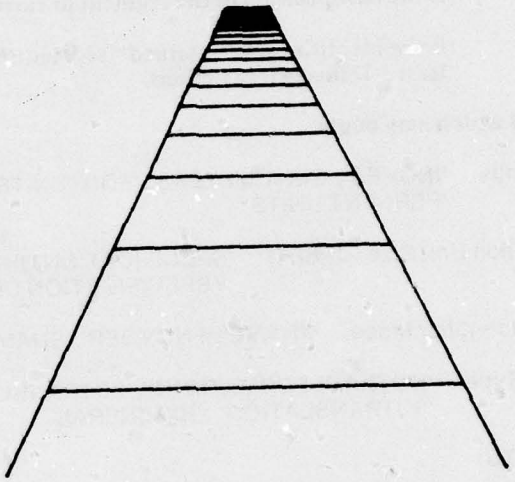
OPTIONS which may apply:

Any USET option.
See UPEN/U3PEN for a description of options effective pen/beam movements.

COMMENTS:

U3PEN1 is convenient for setting options which will apply only to one pen/beam movement. The GSA setting modified by OPTION will be restored prior to returning to the calling program.

Programming Notes:



```
      CALL USTART
      CALL UVIEW (0.,0.,10.,0.,10.,10.)
      CALL UWINDO (-10.,10.,-12.,6.)
      CALL U3MOVE (5.,5.,0.)
      CALL U3PEN (5.,100.,0.)
      DO 10 I = 10, 100, 5
      CALL U3MOVE (-5.,FLOAT(I-5),0.)
      CALL U3PEN (-5.,FLOAT(I),0.)
   10 CALL U3PEN1 (10.,0.,0.,'RELATIVE')
      CALL UEND
      STOP
      END
```

FUNCTION:

This routine applies a 3-D image translation transformation to the indicated segment/frame.

CALLING SEQUENCE:

CALL U3PLAC (X,Y,Z,SEGID)

Where

X,Y,Z	**is the new position of the segment in current 3-D device units.**
SEGID	**is the identifier of a "retained" segment/frame which was UOPENed for at least 3-D image translations.**

OPTIONS which may apply:

Device Units: **'INCHES',** 'CENTIMETERS', 'FONTUNITS', 'SPECIFICATION UNITS', 'PERCENT UNITS'

Specification Unit Size (UPSET): 'SPECIFICATION UNITS', 'XSPECIFICATION UNITS', 'YSPECIFICATION UNITS', 'ZSPECIFICATION UNITS'

Segment Identifier Mode: **'FNAME',** 'FNUMBER', 'SNAME', 'SNUMBER'

Segment Type: **'NOTRANSFORMATIONS',** '2DTRANSLATION', '2DGENERAL', '3DTRANSLATION', '3DGENERAL'

COMMENTS:

The image transformation is applied to the specified segment. If the resulting image exceeds the display dimensions, the result is undetermined.

Image transformations are only applied if supported on the current display surface. Requests for image transformations will be ignored if the display device does not support this facility.

Programming Notes:

FUNCTION:

This routine provides a general-purpose numeric 3-D plotting capability. Given three arrays of corresponding coordinates of one or more curves, it will scale and plot these points along with suitable axes and labels as specified by the user.

CALLING SEQUENCE:

CALL U3PLOT(X,Y,Z,CURVES,PTS,OPTS)

Where

X	**is a set of X or RADIUS COMPONENTS for the points for all the curves in current user units.**
Y	**is a set of Y or THETA components for the points for all the curves in current user units.**
Z	**is a set of Z or PHI components for the points for all the curves in current user units.**
CURVES	**is a single variable which indicates the number of curves to be plottes.**
PTS	**is an array which indicates how many points are in each curve.**
OPTS	**is an array which specifies which USET option will apply to ach curve as it is being plotted. One option must be specified for each curve and only the first four characters of the option name should be specified.**

OPTIONS which may apply:

All which apply to UPLOT
All which apply to U3AXIS

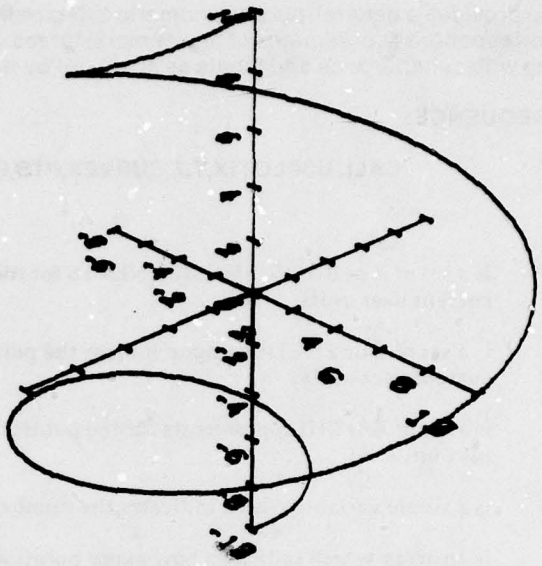
X-Component Repetition Option: 'XREPEAT', **'NOXREPEAT',** 'XCONSTANT'
Y-Component Repetition Options: 'YREPEAT', **'NOYREPEAT',** 'YCONSTANT'
Z-Component Repetition Options: 'ZREPEAT', **'NOZREPEAT',** 'ZCONSTANT'

COMMENTS:

Since the U3AXIS options apply, the user should become familiar with their effects.

The Component Repetition Options allow the user to avoid repetitious storing of components which retain the same value throughout a particular curve or over a group of curves. Selecting a 'CONSTANT' option indicates that that component does not vary during any of the drawing. Selecting a 'NOREPEAT' option indicates that a component is provided for every point in every curve. This is the default. Selecting a 'REPEAT' option indicates that one set of component values is provided which will be reused for each curve. The size of this set is always the number of points in the first curve. If subsequent curves have more points than the first curve, U3PLOT will start to resequence through the 'repeated' components.

Programming Notes:



SPIRAL ON A SPHERE

```
      REAL X(401), Y(401), Z(401), R
      R = 10.
      DO 10 I = 1, 401
      Z(I) = FLOAT(I-1) * R/200.0 - R
      T = SQRT(R**2 - Z(I)**2)
      X(I) = COS(Z(I) * 0.5) * T
      Y(I) = SIN(Z(I) * 0.5) * T
   10 CONTINUE
      CALL USTART
      CALL UPSET ('TERMINATOR',',')
      CALL UPSET ('HORIZONTAL',.6)
      CALL UPSET ('VERTICAL',1.2)
      CALL USET ('SOFT')
      CALL USET ('NOYLABELS')
      CALL UWINDO (-15.,15.,-15.,15.)
      CALL UVIEW (30.,-30.,20.,0.,0.,0.)
      CALL U3PLOT (X,Y,Z,1.,401.,'LNULL')
      CALL UVIEW (0.,0.,150.,0.,0.)
      CALL USET ('ACENTER')
      CALL U3MOVE (0.,-11.,0.)
      CALL UPRNTI ('SPIRAL ON A SPHERE;','TEXT')
      CALL UEND
      STOP
      END
```

FUNCTION:

This routine displays textual data justified at the 3D pen position specified. At termination, the pen will be positioned at the end of the text string ready for additional characters.

CALLING SEQUENCE:

CALL U3PRNT (X,Y,Z,DATA)

Where

X,Y,Z	**are the coordinates of the text string justification positioned in current units.**
DATA	**is a GCS Hollerith text string if 'TEXT' is specified.** **is a REAL varible if 'INTEGER' or 'REAL' is specified.** **is a REAL array of two elements if 'XYCOORDINATES' are specified.** **is a REAL array of three elements if 'XYZCOORDINATES' are specified.**

OPTIONS which may apply:

Alphanumeric Output Type: **'TEXT'**, 'INTEGER', 'REAL', 'XYCOORDINATES', 'XYZCOORDINATES'.

GCS Terminator Character: UPSET('TERMINATOR'),value)

Numeric Precision Option: UPSET('PRECISION',value)

Significant Zero Option: 'SIGNIFICANT ZEROES', **'NOSIGNIFICANTZEROES'**

Character Type Options: **'HARDWARE'**, 'SIMULATED HARDWARE', 'SOFTWARE'

Hardware Character Position Size: **'SMALL'**, 'MEDIUM', 'LARGE', 'EXTRALARGE', UPSET('XSIZE',value), UPSET('YSIZE',value)

Software Character Position Size: UPSET('HORIZONTALSIZE',value), UPSET ('VERTICALSIZE',value)

Character Position Occupancy: UPSET('XPERCENT',value), UPSET('YPERCENT', value), UPSET('ZPERCENT',value)

Alphabetic Case Shifting: 'UPPERCASE', 'LOWERCASE'

Case Shift Character Specification: UPSET('UPPERCASE',value), UPSET('LOWERCASE',value)

Orientation: UPSET('ORIENTATION',value)

Italicization Options: 'ITALICS', **'NOITALICS'**

Italic Slant Angle: UPSET('SLANT',value)

Coordinate Space Options: **'VIRTUAL'**, 'DEVICE'

Margin Boundaries: see UMARGN

Window Boundaries: see UWINDO

Alphanumeric Spacing Options: 'VERTICAL', **'HORIZONTAL'**

Alphanumeric Justification Options: **'NOCENTERING'**, 'ACENTERING', **'NOJUSTIFICATION'**, **'LJUSTIFICATION'**, 'CJUSTIFICATION', 'RJUSTIFICATION', 'BJUSTIFICATION', 'MJUSTIFICATION', 'TJUSTIFICATION'

Subscript Character: UPSET('SUBSCRIPT',value)

Superscript Character: UPSET('SUPERSCRIPT',value)

Subscript/Superscript Specification: 'SUBSCRIPT', 'SUPERSCRIPT', **'NOSCRIPT'**

Scripting Level: UPSET('SCRIPTLEVEL',value)

Pen Coordinate Option: see U3PEN

Coordinate System: **'SYSTEM'**, **'WORLD'**, 'USER', 'MODELLING'

User Coordinate System Type: 'REFERENCE', **'WORKING'**

Attributes: see U3PEN

COMMENTS:

The description of the alphanumeric output facilities is divided into three areas of format character descriptions and alphanumeric output, positioning each of these will be discussed individually.

Programming Notes:

FORMAT:

U3PRNT handles any of five types of alphanumeric opertions. The format of the input parameter DATA depends upon which type of operation is specified.

'INTEGER — The input is a single real variable whose integer value will be displayed at (X,Y,Z). Truncation will occur if a number which is not an integer is provided.

'REAL' — The input is a single real number whose value will be displayed at (X,Y,Z). The number of digits of precision can be adjucted by using UPSET ('PRECISION',value) where the value is the number of digits of precision desired. Numbers will be rounded to the least significant digit. The default precision is four. Display of significant trailing zeroes is controlled by the significant zero USET option. 'SIGNIFICANT ZEROES' specifies that significant zeroes are to be displayed; 'NOSIGNIFICANT ZEROES' suppresses their display. The default suppresses significant zeroes.

'XYCOORDINATES' — The input is a real array of two elements. The first element reflects an X- or RADIUS coordinate component at the second element. The second reflects a Y- or THETA coordinate component. The coordinates will be displayed at (X,Y,Z) enclosed in parentheses and separated by a comma.

'XYZCOORDINATES' — The input is a real array of three elements. The first element reflects an X- or RADIUS coordinate component; the second element, a Y- or THETA coordinate component; and the third element, a Z- or PHI coordinate component. The coordinates will be displayed at (X,Y,Z) enclosed in parentheses and separated by commas, e.g., (3.8765,1.5,0). The precision and significant zeroes options, described for 'REAL' numbers above, both apply to each component. Note that the coordinates displayed need not be the same values as the alphanumeric output justification position parameters (X,Y,Z).

'TEXT' — The input is a single variable or array containing a Hollerith Character string. The string may be as long as desired but must be terminated by the GCS terminator character. This character, which cannot be displayed while functioning as the terminator character, may be specified by UPSET ('TERMINATOR', value) where value is a one-character Hollerith string. The default GCS terminator character is the ASCII backslash (12-8-5) character. The input string may also contain upper and lower case shifting characters. The case shifting and scripting characters are described below.

CHARACTER DESCRIPTION OPTIONS:

These options specify the attributes of individual characters.

CHARACTER TYPE:

There are three basic types of characters in GCS. Each type will be described separately below. Figure 2 illustrates each type. The user can switch between any of the three types simply by invoking the appropriate USET options.

'HARDWARE' — These characters are generated when possible by a hardware character generator in the display device. The size of these character positions is frequently limited to several discrete sizes (see Hardware Character Position Size below). The characters are produced on the display surface plane. This is the default character type. Character spacing is exact. Hardware character output may be directed to either the 'MESSAGE DEVICE' or the 'PLOTDEVICE' if they are separable.

'SIMULATED HARDWARE' — These characters are produced via software. Although they act as hardware characters, they are generated to the exact hardware character position size (q.v.) specified. 'SIMULATED HARDWARE' characters use the 'SOFTWARE' characters descriptions. The characters are produced on the display surface plane. Character spacing is exact.

'SOFTWARE' — These characters are produced by software vectors in the current XY-plane. The line which connects the lower left corner of each character position in the string is parallel to the current X-axis. The string is drawn in the positive X-direction for horizontal spacing and the negative Y-direction for vertical spacing (see Character spacing). Both software character dimensions may be specified (see below).

HARDWARE CHARACTER POSITION SIZE — The size of 'HARDWARE' and 'SIMULATED HARDWARE' character positions may be set by either USET or UPSET options. An exact size may be specified by the following UPSET options. For hardware characters, the largest discrete size will be used which does not exceed the requested size. If all sizes exceed the requested size, then the smallest size is used.

UPSET('XSIZE',value) — The horizontal character position size is set to the value specified in current 'DEVICE' space units.

UPSET('YSIZE',value) — The vertical character position size is set to the value specified in current 'DEVICE' space units.

Four (4) discrete sizes are provided by USET options:

'SMALL' — The hardware character position size closest to typewriter size character positions. This is the default.

'MEDIUM' — This is the next larger size from 'SMALL'. Simulated 'MEDIUM' size is approximately twice the 'SMALL' size.

'LARGE' — This is the next larger size from 'MEDIUM'. Simulated 'LARGE' size is approximately three times the 'SMALL' size.

'EXTRALARGE' — This is the next larger size from 'LARGE'. Simulated 'EXTRALARGE' size is approximately four times the 'SMALL' size.

Additional discrete sizes may be specified by:

UPSET('SIZE',value) — The discrete size whose index is specified in the value is selected. Indexes must be positive values. 'SMALL' through 'LARGE' are represented by index values 1. through 4. Assignment of sizes to remaining index values (i.e., LT.4.) is installation dependent.

SOFTWARE CHARACTER POSITION SIZE:

The size of 'SOFTWARE' character positions may be set by the following two UPSET options. The visible size of software characters is affected by scaling factors created by U3CSYS calls and by the currently set viewing environment. Software character positions in device space are visibly the same size as software character positions in virtual space at the view plane.

UPSET('HORIZONTAL SIZE',value) — The horizontal size of software character positions is set to the value provided in current user units. Default value is 5.

UPSET('VERTICALSIZE',value) — The vertical size of software character positions is set to the value provided in current user units. Default value is 7.

CHARACTER POSITION OCCUPANCY:

The proportion of the software character positions to actually be occupied by the character may be specified as follows:

UPSET('XPERCENT',value) — The proportion of the width of the character position to be occupied by the character is set to the value provided. Valid values are greater than zero. A value of 1. specifies the entire width of the character position. Default value is .65.

UPSET('YPERCENT',value) — The proportion of the height of the character position to be occupied by the character is set to the value provided. Valid values are greater than zero. A value of 1. specifies the entire height of the character position. Default value is .65.

ALPHABETIC CASE SHIFTING:

In machines which have only six bits per character, it is still desirable in GCS to be able to specify both upper and lower case characters since the software character set can produce both. Therefore, a case shifting facility has been implemented which not only provides this service for the software characters but will also work for hardware characters on those devices which have hardware character generators which can produce both cases. Additionally, since case shifting applies only to upper case characters, when a program which does case shifting is executed on a computer which does not need case shifting, the desired upper and lower case character will still be produced.

To shift cases, it is necessary to insert special case shift characters in the GCS text string. GCS defaults to upper case. When the lower case shift character inserted in the

string, a shift to lower case will occur if GCS is currently in upper case. If GCS is already in lower case, the shift character will itself be generated. The opposite happens when the upper case shift character is inserted in the string. If GCS is in lower case mode, it will be placed in upper case. However, if GCS is already in upper case mode, the upper case shift character will be generated. See Figure 3 for an illustration of the use of case shifting.

When changing cases, all alphabetic characters will be generated in either UPPERCASE or LOWERCASE as appropriate. However, to provide the full complement of special characters available in the ASCII character set to persons with computers which handle only 6-bits per character internally, some special characters will be mapped into others when the case is changed.

The shift characters may be changed by the following two calls:

```
CALL UPSET('LOWERCASECHARACTER',CHAR)
```

or

```
CALL UPSET('UPPERCASECHARACTER',CHAR)
```

Where

CHAR — is a single character in Hollerith (left-justified, blank-fill) format. Thus, it may be a literal (quoted character string) or a variable containing a Hollerith value.

The user is warned that the shift character should not be set to be the line terminator character. If this happens, no case shifting will occur. The case in which GCS is currently set may be forced to either upper or lower case by the following:

```
CALL USET('UPPERCASE')
CALL USET('LOWERCASE')
```

This is convenient for setting the system character for use as CHARACTER terminators in the line option or in building ALPHA lines.

ORIENTATION:

As software characters are produced, they may be 'ORIENTED' (tilted) off the X-axis by the UPSET option described below. For a string of characters eminating from U3PRNT, the lower left corner of each character will touch a line which is parallel to the current X-axis. The bottom of the character will be rotated off this line by the number of angular units specified. An illustration of this feature is shown in Figure 4.

UPSET('ORIENTATION',value) — The geometric figure orientation parameter is set to the number of angular units specified in the value. Default value is 0. Note that the orientation parameter also applies to UPLYGN and URECT.

ITALIZATION:

Software characters may be 'ITALICIZED' by applying transformation which leans or slants the character away from the vertical as illustrated in Figure 5. This mode is selected as follows:

'ITALICS' — Software characters are to be slanted.

'NOITALICS' — Software characters are not to be slanted. This is the default.

ITALIC SLANT ANGLE:

The amount of slant to be applied to the characters may be specified by:

UPSET('SLANT',value) — The amount of slant from the vertical is set to the value provided in current angular units. Default value is approximately 17.5 degrees. Values provided must not be odd multiples of 90 degrees.

COORDINATE SPACE OPTIONS:

U3PRNT can display characters in either 'DEVICE' or 'VIRTUAL' space. If 'DEVICE' space is specified, all characters will be subject to margining; i.e., all characters will be produced within the specified margins (see UMARGN).

If 'VIRTUAL' space is specified, all characters will be clipped at the window boundaries (see UWINDO). For 'HARDWARE' and 'SIMULATED HARDWARE' characters, clipping will occur if any portion of the character position exceeds the window boundaries. Since 'SOFTWARE' characters are drawn with vectors, they will be clipped exactly at the window boundaries.

ALPHANUMERIC SPACING OPTIONS:

Adjacent character positions may be defined to proceed either horizontally or vertically. This option applies to all character types and is selected as follows:

'HORIZONTAL' — Adjacent character positions occur horizontally; i.e., character strings are produced in the positive X direction.

'VERTICAL' — Adjacent character positions occur vertically; i.e., character strings are produced in the negative Y direction.

ALPHANUMERIC JUSTIFICATION OPTIONS:

The arguments to U3PRNT contain a set of coordinates which indicate the justification position for the output string. Each justification option is illustrated in Figure 9. Justification options are as follows:

'LJUSTIFICATION'
'NOCENTERING'
'NOJUSTIFICATION' — The coordinates specify the position of the lower left corner of the first character position in the output string. This is the default.

'CJUSTIFICATION'
'ACENTERING' — The coordinates specify the position of the center of the output string. The output will be centered both horizontally and vertically. The centering is upon the unscripted string (see Subscript/Superscript Specification below).

'RJUSTIFICATION' — The coordinates specify the position of the lower left corner of the character position which follows the last character of the string. Note that this is also the lower right corner of the last character position of the string if 'HORIZONTAL' spacing is specified.

STRING WIDTH JUSTIFICATION:

'BJUSTIFICATION' — Justifies to the bottom of the text string for horizontal spacing and to the left edge of the text string for vertical spacing.

'MJUSTIFICATION' — Justifies to the midway point between the top and bottom of the text string for horizontal spacing and to the midway point between the left edge and right edge of the text string for vertical spacing.

'TJUSTIFICATION' — Justifies to the top of the text string for horizontal spacing and to the right edge of the text string for horizontal spacing.

SUBSCRIPT/SUPERSCRIPT SPECIFICATION:

The primary means of doing scripting is by the insertion in GCS text strings of subscript and superscript escape characters. Each occurrence of the subscript character will lower the position of succeeding characters 1/3 character position. Each occurrence of the superscript character will raise the position of succeeding characters 1/3 character position. When the GCS terminator character is encountered, the scripting level is returned to the zero position and the pen/beam position is set for additional unscripted characters. (See Figure 10.) Note that scripting positions will not be computed into center justification adjustments. The subscript and superscript escape characters may be specified by the following UPSET options:

UPSET('SUBSCRIPT',value) — The subscript escape character is set to the first character of the Hollerith string provided as a value. The default subscript character is an ASCII "pound sign ".

UPSET('SUPERSCRIPT',value) — The superscript escape character is set to the first character of the Hollerith string provided as a value. The default superscript character is an ASCII "underline sign".

An alternate method of performing scripting is provided if several successive output strings are to be scripted. The user may specify the type of scripting and the scripting level. The scripting level is defined to be the number of 1/3 character positions below the justification position for 'SUBSCRIPTING' and the number of 1/3 character positions above the justification position for 'SUPERSCRIPTING'.

'NOSCRIPTING' — Forced scripting is disabled. However, scripting by insertion of scripting escape characters can still occur. This is the default.

'SUBSCRIPTING' — The output string will be produced starting at the specified script level position below the justification position.

'SUPERSCRIPTING' — The output string will be produced starting at the specified script level position above the justification position.

Note that specification of the type of scripting still allows further modifications in the scripting by insertion of scripting escape characters. To set the scripting level, the following UPSET option should be used:

UPSET('SCRIPTLEVEL',value) — The scripting level is set to the integer value specified. Note that if a negative scripting level is provided, subscripting becomes superscripting and vice versa.

PEN COORDINATE OPTIONS:

All pen coordinate options apply to the justification position contained in the X,Y,Z parameters to U3PRNT. The actual starting location of the character string may be adjusted as required by the justification option described earlier. For a full description of the Pen Coordinate Options, see U3PEN.

COORDINATE SYSTEM SELECTION AND TYPE:

The full power of U3PRNT can only be realized through use of user coordinate systems. U3PRNT produces 'SOFTWARE CHARACTERS' on the XY-plane which passes through the justification point. Through use of user coordinate systems, the user can position and orient the XY-plane anywhere in 3-space. By suitable coordinate system definitions, 'SOFTWARE' character output can be made to recede into the distance or be viewed from behind. It should be noted that while 'HARDWARE' and 'SIMULATED HARDWARE' characters can be positioned anywhere in 3-space, they will still be produced in the display surface plane.

123456

3.142

(3.142,2.718)

(3.142,2.718,10.)

Example of 'TEXT' Output

Alphanumeric Output Formats

Figure 1 (U3PRNT)

```
DIMENSION COORD(3)
DATA COORD/3.1415926,2.71828,10./
CALL USTART
CALL UPSET ('TERMINATOR',',')
CALL USET ('LARGE')
CALL USET ('INTEGER')
CALL UPRINT (20.,50.,123456.)
CALL USET ('REAL')
CALL UPRINT (20.,45.,COORD(1))
CALL USET ('XYCOORDINATES')
CALL UPRINT (20.,40.,COORD)
CALL USET ('XYZCOORDINATES')
CALL UPRINT (20.,35.,COORD)
CALL USET ('TEXT')
CALL UPRINT (20.,30.,'E>XAMPLE OF <''TEXT'' O>UTPUT,')
CALL USET ('ACENTER')
CALL UPRINT (40.,20.,'<A>LPHANUMERIC <O>UTPUT <F>ORMATS,')
CALL UPRINT (40.,15.,'<F>IGURE 1 <<U3PRNT);')
CALL UEND
STOP
END
```

Hardware Characters

Simulated Hardware Characters

Software Characters

GCS Character Types

Figure 2 (U3PRNT)

```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('ACENTER')
CALL UROTAT (10.)
CALL USET ('EXTRALARGE')
CALL UPRINT (50.,50.,'H>ARDWARE <C>HARACTERS;')
CALL USET ('SIMULATED HARDWARE CHARACTERS')
CALL UPSET ('XSIZE',.25)
CALL UPSET ('YSIZE',.4)
CALL UPRINT (50.,40.,'<S>IMULATED <H>ARDWARE <C>HARACTERS;')
CALL USET ('SOFTWARE CHARACTERS')
CALL UPSET ('HORIZONTAL SIZE',4.5)
CALL UPSET ('VERTICAL SIZE',7.)
CALL UPRINT (50.,25.,'<S>OFTWARE <C>HARACTERS;')
CALL USET ('HARDWARE CHARACTERS;')
CALL USET ('LARGE')
CALL UPRINT (45.,15.,'<GCS C>HARACTER <T>YPES;')
CALL UPRINT (45.,10.,'<F>IGURE 2 <(U3PRNT);')
CALL UEND
STOP
END
```

This is case shifting in GCS

Upper/Lower Case Shifting

Figure 9 (USPRNT)

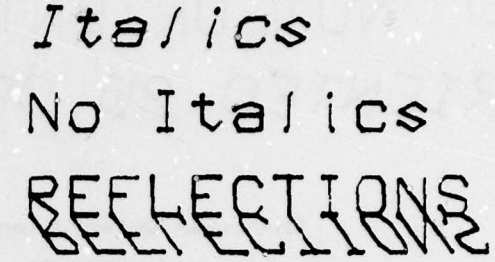
```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('ACENTER')
CALL USET ('LARGE')
CALL UPRINT (50.,30.,'T>HIS IS CASE SHIFTING IN <GCS;')
CALL UPRINT (50.,20.,'U>PPER/<L>OWER <C>ASE <S>HIFTING;')
CALL UPRINT (50.,15.,'<F>IGURE 9 <(USPRNT);')
CALL UEND
STOP
END
```

NOT ORIENTED
ORIENTED 20 DEGREES

Character Orientation

Figure 4 (U3PRNT)

```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('SOFTWARE')
CALL USET ('ACENTER')
CALL UPRINT (50.,50.,'NOT ORIENTED;')
CALL UPSET ('ORIENTATION',20.)
CALL UPRINT (50.,40.,'ORIENTED 20 DEGREES;')
CALL USET ('HARDWARE')
CALL USET ('LARGE')
CALL UPRINT (50.,25.,'C>HARACTER <O>RIENTATION;')
CALL UPRINT (50.,20.,'<F>IGURE 4 <(U3PRNT);')
CALL UEND
STOP
END
```



Examples of Italicization

Figure 5 (U3PRNT)

```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('ITALICS')
CALL USET ('SOFTWARE')
CALL UPRINT (25.,50.,'I>TALICS;')
CALL USET ('NOITALICS')
CALL UPRINT (25.,40.,'<N>O <I>TALICS<;')
CALL USET ('NOCENTER')
CALL UPRINT (25.,30.,'REFLECTIONS;')
CALL UCOSYS (25.,30.,1.,-1.,0.)
CALL UPSET ('SLANT',40.)
CALL USET ('ITALICS')
CALL UPRINT (0.,0.,'REFLECTIONS;')
CALL USET ('WORLD COORDINATES')
CALL USET ('HARDWARE')
CALL USET ('ACENTER')
CALL USET ('LARGE')
CALL UPRINT (50.,15.,'E>XAMPLES OF <I>TALICIZATION;')
CALL UPRINT (50.,10.,'<F>IGURE 5 <(U3PRNT);')
CALL UEND
STOP
END
```

This is
 an exa
mple of
 margin
ing.

Margining

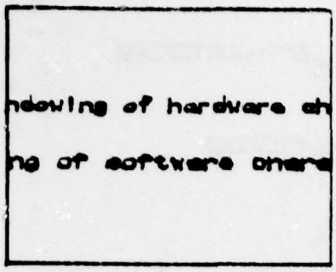
Figure 6 (USPRNT)

```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('PERCENT UNITS')
CALL UMARGN (45.,55.,30.,50.)
CALL USET ('LARGE')
CALL USET ('DEVICE')
CALL UPRINT (40.,45.,'T>HIS IS AN EXAMPLE OF MARGINING.;')
CALL USET ('ACENTER')
CALL UMARGN (0.,100.,0.,100.)
CALL UPRINT (50.,15.,'<M>ARGINING;')
CALL UPRINT (50.,10.,'<F>IGURE 6 <(USPRNT);')
CALL UEND
STOP
END
```

Vertical Spacing by using Margining

Figure 7 (USPRNT)

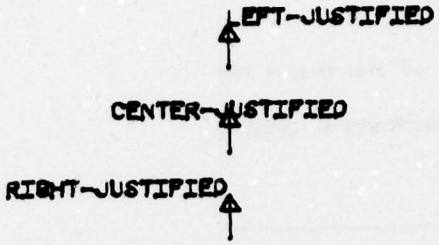
```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('PERCENT UNITS')
CALL USET ('DEVICE')
CALL UMARGN (50.,50.5,20.,60.)
CALL USET ('LARGE')
CALL UPRINT (50.,60.,'MORE MARGINING;')
CALL USET ('ACENTER')
CALL UMARGN (0.,100.,0.,100.)
CALL UPRINT (50.,15.,'V>ERTICAL <S>PACING BY USING <M>ARGINING,
CALL UPRINT (50.,10.,'<F>IGURE 7 <(USPRNT);')
CALL UEND
STOP
END
```



Examples of Windowing of Characters

Figure 8 (USPRNT)

```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('PERCENT UNITS')
CALL UDAREA (35.,65.,35.,65.)
CALL UOUTLN
CALL USET ('EXTRALARGE')
CALL UPRINT (-10.,60.,'W>INDOWING OF HARDWARE CHARACTERS;')
CALL USET ('SOFTWARE CHARACTERS')
CALL USET ('ACENTER')
CALL UPRINT (45.,40.,'<W>INDOWING OF SOFTWARE CHARACTERS;')
CALL USET ('DEVICE')
CALL USET ('HARDWARE CHARACTERS')
CALL USET ('LARGE')
CALL UPRINT (50.,25.,'<E>XAMPLES OF <W>INDOWING OF <C>HARACTERS
CALL UPRINT (50.,20.,'<F>IGURE 8 <(USPRNT);')
CALL UEND
STOP
END
```



Justification Options

Figure 9 (USPRNT)

```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('LARGE')
CALL USET ('LARROW')
CALL UMOVE (50.,45.)
CALL UPEN (50.,50.)
CALL UPRINT (50.,50.,'LEFT-JUSTIFIED,')
CALL USET ('CJUSTIFICATION')
CALL UMOVE (50.,35.)
CALL UPEN (50.,40.)
CALL UPRINT (50.,40.,'CENTER-JUSTIFIED,')
CALL USET ('RJUSTIFICATION')
CALL UMOVE (50.,25.)
CALL UPEN (50.,30.)
CALL UPRINT (50.,30.,'RIGHT-JUSTIFIED,')
CALL USET ('CJUSTIFICATION')
CALL UPRINT (50.,15.,'J>USTIFICATION <O>PTIONS,')
CALL UPRINT (50.,10.,'<F>IGURE 9 <(USPRNT),')
CALL UEND
STOP
END
```

3D Character String 4

3D Character String 1

3D Character String 2

3D Software Character String Orientation

Figure 11 (U3PRNT)

```
CALL USTART
CALL UPSET ('TERMINATOR',';')
CALL USET ('LARGE')
CALL USET ('ACENTER')
CALL UPRINT (50.,15.,'3D S>OFTWARE <C>HARACTER <S>TRING <O>RIEN
&TATION;')
CALL UPRINT (50.,10.,'<F>IGURE 11 <(U3PRNT);')
CALL USET ('SOFTWARE CHARACTERS')
CALL UWINDO (-100.,100.,-100.,100.)
CALL U3PRNT (0.,0.,0.,'3D C>HARACTER <S>TRING 1;')
CALL U3PRNT (0.,-30.,-50.,'<3D C>HARACTER <S>TRING 2;')
CALL U3CSYS (-30.,40.,30.,1.,1.,1.,45.,75.,45.)
CALL U3PRNT (0.,0.,0.,'<3D C>HARACTER <S>TRING 3;')
CALL U3CSYS (50.,50.,-10.,-1.,-1.,1.,0.,0.,-20.)
CALL U3PRNT (0.,0.,0.,'<3D C>HARACTER <S>TRING 4;')
CALL UEND
STOP
END
```

FUNCTION:

This routine builds a new coordinate system with origin at the current pen position rotated by the specified amount about each axis.

CALLING SEQUENCE:

CALL U3ROTA(RX,RY,RZ)

Where

RX,RY,RZ **indicates the amount of rotation around each of the axes in current angular units.**

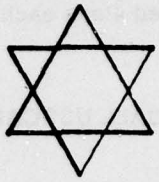
OPTIONS:

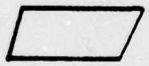
Rotation Application Order: See U3CSYS

COMMENTS:

This routine causes a new coordinate system to be built with origin at the current beam position and with a scale factor of 1 along each axis. A full description of the user coordinate system facility can be found in the U3CSYS writeup and in the UCOSYS writeup.

Programming Notes:





```
CALL USTART
CALL UMOVE (50.,75.)
CALL UROTAT (0.)
CALL UPLYGN (0.,0.,3.,10.)
CALL UROTAT (180.)
CALL UPLYGN (0.,0.,3.,10.)
CALL USET ('WORLD COORDINATES')
CALL UMOVE (25.,25.)
CALL USROTA (75.,0.,0.)
CALL UPLYGN (0.,0.,4.,10.)
CALL USET ('WORLD COORDINATES')
CALL UMOVE (75.,25.)
CALL USROTA (0.,45.,0.)
CALL UPLYGN (0.,0.,4.,10.)
CALL UEND
STOP
END
```

FUNCTION:

This routine composes a new coordinate system with the origin at the current beam position and scaled as specified along each axis.

CALLING SEQUENCE:

CALL U3SCAL(SX,SY,SZ)

Where

SX,SY,SZ **are the multiplicative scale factors along each axis respectively.**

OPTIONS:

None.

COMMENTS:

The rotation factors used in composing the new scaled coordinate system are 0. around each axis. See the U3CSYS and UCOSYS writeup for a full description of the user coordinate system facility.

Programming Notes:

FUNCTION:

This routine returns to the user the limits of his 3-D U3WNDO and U3AREA.

CALLING SEQUENCE:

CALL U3STUD (ARRAY)

Where

ARRAY **is an array of at least twelve words to contain the current settings.**

OPTIONS which may apply:

Device Units: **'INCHES'**, 'CENTIMETERS', 'PERCENT', 'FONT', 'RASTER', 'SPECIFICATION'

COMMENTS:

A call to this routine will return to the user the limits of his 3-D virtual window and display area in the order:

ARRAY (1) = virtual X minimum boundary
ARRAY (2) = virtual X maximum boundary
ARRAY (3) = virtual Y minimum boundary
ARRAY (4) = virtual Y maximum boundary
ARRAY (5) = virtual Z minimum boundary
ARRAY (6) = virtual Z maximum boundary
ARRAY (7) = device X minimum boundary
ARRAY (8) = device X maximum boundary
ARRAY (9) = device Y minimum boundary
ARRAY(10) = device Y maximum boundary
ARRAY(11) = device Z minimum boundary
ARRAY(12) = device Z maximum boundary

To recover only the 2-D boundaries, see USTUD.

Programming Notes:

FUNCTION:

This routine returns the three-dimensional coordinates of the pen position in current units.

CALLING SEQUENCE:

CALL U3WHER(X,Y,Z)

Where

X,Y,Z	**will contain the pen position coordinates in current units upon exit from this rotation.**

OPTIONS:

Pen Coordinate Options: see U3PEN

COMMENTS:

The coordinates returned will always indicate 'ABSOLUTE' position in the current coordinate system since the 'RELATIVE' position is always (0.,0.,0.).

Programming Notes:

FUNCTION:

This routine specifies the boundaries of the user window in virtual 3-D space.

CALLING SEQUENCE:

CALL U3WNDO(XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX)

Where

XMIN	is the minimum X-boundary of the desired new window, given in eye coordinates.
XMAX	is the maximum X-boundary of the desired new window, given in eye coordinates.
YMIN	is the minimum Y-boundary of the desired new window, given in eye coordinates.
YMAX	is the maximum Y-boundary of the desired new window, given in eye coordinates.
ZMIN	in the minimum Z-boundary of the desired new window, given in eye coordinates.
ZMAX	is the maximum Z-boundary of the desired new window, given in eye coordinates.

OPTIONS which may apply:

Z-axis clipping: **'NOZCLIPPING'**, 'ZCLIPPING'

COMMENTS:

The concept of windowing into virtual space expands greatly the capabilities of the user. In three-dimensional space, the user is assumed to be located somewhere in virtual space (i.e., at the view point) and looking in some direction (i.e., toward the view site) (see UVIEW). Normally, he is only interested in a particular subset of all the space within his field of view. This subset is specified by the U3WNDO routine. The portion of the user's picture which is within the boundaries of the window will be projected onto a plane perpendicular to the viewing vector and in front of the viewer. This plane is known as the view plane (See UVWPLN). The X and Y window boundaries refer to lines on the view plane which form a rectangle. The Z window boundaries indicate the portion of space in front of the viewer to be projected onto the plane if 'ZCLIPPNG' has been specified. Normally, 'NOZCLIPPING' is specified, in which case all lines which are projected onto the view plane within the X and Y window boundaries are visible.

Subroutine U3WNDO will generate an error condition if the maximum boundary is specified less than or equal to the minimum boundary and the previous setting of the window will be retained.

Programming Notes:

FUNCTION:

This routine displays textual output at the pen position specimen in 3-D space. The pen will be repositioned to the location it was at prior to the call to U3WRIT.

CALLING SEQUENCE:

CALL U3WRIT(X,Y,Z,DATA)

Where

X,Y,Z	**are the coordinates of the beginning of the alphanumeric output in current units.**
DATA	**is a GCS Hollerith character string terminated by the GCS terminator character if 'TEXT' is specified.**
	is a REAL variable or literal if 'INTEGER' or 'REAL' is specified.
	is a REAL array of two elements if 'XYCOORDINATES' is specified.
	is a REAL array of three elements if 'XYZCOORDINATES' is specified.

OPTIONS:

Coordinate Space Options:	See U3PRNT
Alphanumeric Character type:	See U3PRNT
Alphanumeric Output type:	See U3PRNT
Alphanumeric Spacing Option:	See U3PRNT

COMMENTS:

This routine only differs from U3PRNT in the position of the pen at return from the routine. U3PRNT leaves the beam positioned at the end of the text output. U3WRIT returns the pen to the position it held before entry to U3WRIT. For a full discussion of the various U3WRIT options, see U3PRNT.

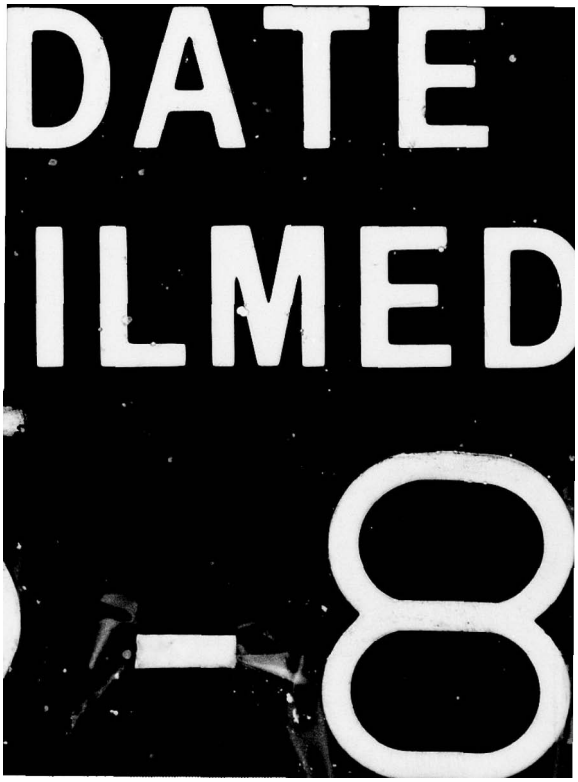
Programming Notes:

U.S. GOVERNMENT PRINTING OFFICE 1980-640-280/37

In accordance with letter from DAEN-RDC, DAEN-ASI dated 22 July 1977, Subject: Facsimile Catalog Cards for Laboratory Technical Publications, a facsimile catalog card in Library of Congress MARC format is reproduced below.

Westinghouse Word Processing Center, Pittsburgh.
Graphics Compatibility System (GCS) programmer's reference manual / by Westinghouse Word Processing Center, Pittsburgh, Pa. Vicksburg, Miss. : U. S. Waterways Experiment Station ; Springfield, Va. : available from National Technical Information Service, 1979.
ii, 280 p. : ill. ; 27 cm. (Miscellaneous paper - U. S. Army Engineer Waterways Experiment Station ; 0-79-5)
Prepared for Office, Chief of Engineers, U. S. Army, Washington, D. C., under Contract No. DACW39-78-M-2676.

1. Computer graphics. 2. Graphics Compatibility System. 3. Manuals. 4. Programmer's reference manual. I. United States. Army. Corps of Engineers. II. Series: United States. Waterways Experiment Station, Vicksburg, Miss. Miscellaneous paper ; 0-79-5.
TA7.W34m no.0-79-5


DATE
ILMED
-8