MEMORANDUM
RM-5660-PR
DECEMBER 1968

THE INTEGRATED GRAPHICS SYSTEM FOR THE S-C 4060: I. USER'S MANUAL

G. D. Brown, C. H. Bush and R. A. Berman

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA • CALIFORNIA

MEMORANDUM

RM-5660-PR

DECEMBER 1968

THE INTEGRATED GRAPHICS SYSTEM FOR THE S-C 4060: I. USER'S MANUAL

G. D. Brown, C. H. Bush and R. A. Berman

This research is supported by the United States Air Force under Project RAND—Contract No. F44620-67-C-0045—monitored by the Directorate of Operational Requirements and Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF. Views or conclusions contained in this study should not be interpreted as representing the official opinion or policy of the United States Air Force.

DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

The RAND Corporation

1700 MAIN ST • SANTA MONICA • CALIFORNIA • 90406

This study is presented as a competent treatment of the subject, worthy of publication. The Rand Corporation vouches for the quality of the research, without necessarily endorsing the opinions and conclusions of the authors.

Published by The RAND Corporation

PREFACE

The Integrated Graphics System (IGS) is a machine- and device-independent graphics system developed at The RAND Corporation. This Memorandum describes the S-C 4060 version of this system, which currently runs on IBM 7044 and IBM 360 computers.

This Memorandum is the user's manual. A companion Memorandum is the system programmer's guide: *The Integrated Graphics System for the S-C 4060: II. System Programmer's Guide*, The RAND Corporation, RM-5661-PR, January 1969. A version of IGS for the IBM 2250 is described in Gary D. Brown and Charles H. Bush, *The Integrated Graphics System for the IBM 2250*, The RAND Corporation, RM-5531-ARPA, October 1968.

SUMMARY

This Memorandum describes the Integrated Graphics System (IGS) used at The RAND Corporation on IBM 360 and IBM 7044 computers to support an S-C 4060 stored-program recording system. Graphic applications may be programmed in a wide variety of languages, including FORTRAN, PL/I, SIMSCRIPT 1.5, and assembly language.

IGS is currently used with an S-C 4060. However, since IGS is independent of its specific hardware considerations, the system can easily be adapted to other graphic devices.

IGS is composed of subroutines that draw lines and circles, plot points, and display characters and numeric data. Subroutines are also provided to draw, label and title graphs.

ACKNOWLEDGMENTS

The authors wish to thank D. T. Rumford of The RAND Corporation and M. J. Kaitz of Visual Computing Corporation for their help in designing the graphics system. We are also grateful to our RAND colleagues J. E. Rieber for programming assistance and Linda Colbert for her competent editorial assistance.

CONTENTS

PREFACE iii

SUMMARY v

ACKNOWLEDGMENTS vii

FIGURES xi

Section
I. INTRODUCTION 1

II. THE GRAPHICS SYSTEM HARDWARE 2

III. DESCRIPTION OF THE GRAPHICS SYSTEM 6
Mode Sets 6
Subject and Object Space 7
Character Display 8
Plotting 10
Typing 10
Subroutine-Naming and -Coding Conventions 13
The Structure of the Graphics System 14
Housekeeping Subroutines 14
Graphic Output Subroutines 14
Graph Subroutines 15
Conversion Subroutines 15

IV. DESCRIPTION OF THE GRAPHIC SUBROUTINES 16
Brief List of Graphic Subroutines 16
1) Housekeeping Subroutines 16
2) Graphic Output Subroutines 16
3) Graph Subroutines 17
4) Conversion Subroutines 17
Format of Graphic Subroutine Write-Ups .. 17
Purpose 17
Usage 17
Standard Calls 17
Special Calls 18
Modifications 18
Miscellaneous Information 18
Housekeeping Subroutines 19
MODESG: Initializes the System 21
SUBJEG: Sets Up Subject Space 22
OBJCTG: Sets Up Object Space 24
SETSMG: Sets Mode Values 26
GETSMG: Retrieves Mode Values 27
RSETMG: Resets all Default Values in Mode Set Array 28

TABSG: Sets Typing Tabs 29
PAGEG: Controls Forms Flash and Frame Advance 30
VECIG: Initializes a Vector Character Font 31
EXITG: Closes Down the Graphic Devices 33
Graphic Output Subroutines 34
CIRARG: Draws a Circle or an Arc 35
LEGNDG: Types Characters 37
LINESG: Draws Joined Lines 42
MLTPLG: Draws Multiple Line Segments .. 45
NUMBRG: Displays Numeric Data 47
POINTG: Plots Points 51
SEGMTG: Draws Line Segments 54
TEXTG: Types Characters 56
Graph Subroutines 57
GRIDG: Draws a Grid 58
LABELG: Labels a Grid 62
TITLEG: Titles a Grid 68
SETUPG: Computes Appropriate Arguments for the Grid Routines 72
GRAPHG: Constructs a Complete Graph ... 76
Conversion Subroutines 78
CONVTG: Converts Characters to Numeric Data 79
FMTSG: Converts Numeric Data to Characters 80

Appendix
A. IGS MODE SET ARRAY 83
B. IGS ERROR CODES 89
C. SPECIAL CONTROL CHARACTERS 91
D. IGS CHARACTER CASES 92
E. SC-4060 CHARACTRON CHARACTER SET 93
F. SUMMARY OF S-C 4060 GRAPHIC FEATURES 94
G. SAMPLE PROGRAM 96
H. SAMPLE VECTOR CHARACTER USAGE 98
I. SAMPLE VECTOR CHARACTERS 100
J. TECHNIQUES FOR SAVING CORE STORAGE 101
K. SCORS COMPATIBILITY 102
L. PL/I COMPATIBILITY 103

REFERENCES 105

ALPHABETIC LIST OF SUBROUTINES 107

FIGURES

1. Schematic of S-C 4060 Operation 5
2. Subject and Object Space 9
3. The Character Envelope 11
4. Line Orientation 12
5. Character Orientation 12
6. Skew Angle 12
7. Graphic Output Subroutines 34
8. Graph Subroutines 57
9. GRIDG Subroutine 59
10. LABELG Subroutine 63
11. TITLEG Subroutine 69
12. Output from Sample Program 97

I. INTRODUCTION

This Memorandum describes the machine- and device-independent Integrated Graphics System (IGS) developed and currently used at The RAND Corporation. The system now supports an S-C 4060 stored-program recording system with output generated by either an IBM 360 or an IBM 7044 computer.

IGS is a collection of subroutines that instruct the computer to create graphic orders that are transmitted to the S-C 4060. The graphic orders direct the S-C 4060 to produce graphic output on film or hardcopy.

IGS consists of subroutines called by the user to perform such graphic functions as drawing lines, plotting points, displaying characters, and drawing graphs. The user may code in a variety of languages--FORTRAN, PL/I, assembly language, SIMSCRIPT 1.5, or any other language with standard OS/360 linkages. IGS relieves the user of the intricate, specific hardware considerations of the S-C 4060, allowing him to concentrate on his displays.

II. THE GRAPHICS SYSTEM HARDWARE

Graphic orders are produced by a computer program that calls IGS subroutines. When this program runs on a general-purpose computer (called the *host computer* to distinguish it from the computer that drives the S-C 4060), the IGS subroutines produce the graphic orders in a form acceptable to the S-C 4060. This byte-oriented output, called the *meta-language*, is transmitted to the S-C 4060 either by tape or directly through an on-line connection. The S-C 4060 then processes the meta-language and creates the display. Reference 1 details the meta-language.

The S-C 4060 consists of a computer, a cathode ray tube (CRT), a camera, a film developer, and a hardcopy unit. The computer, called the Product Control Unit (PCU), reads input from tape or from an on-line connection and converts the input data into graphic commands. These commands direct a small beam of light across the screen of a high-precision CRT, drawing the picture. The beam may be either on (unblanked) or off (blanked) while it is being moved. However, images appear only if the beam is moved in the unblanked mode.

The usable portion of the CRT screen is defined by a matrix (4096,3072) of addressable point positions. The distance between any two adjacent addressable points (in either the x or y direction) on the display area of the CRT is defined as a *raster unit*.

The origin (coordinate 0,0) begins at the lower left corner of the screen, then extends horizontally to the right in the x-direction and vertically in the y-direction, so that the coordinates (4095,3071) at the top right corner represent the maximum boundary of the screen.

The beam in the CRT is shaped by a 119-character stencil through which it passes. This combination of CRT and stencil is called the CHARACTRON® tube. Characters

may be displayed at any addressable location on the CRT. Each character may be displayed in four sizes: .75, 1.0, 1.25, and 1.5 times normal. Appendix E, p. 93, lists the 119 characters in the CHARACTRON matrix.

Four of the CHARACTRON characters are plotting points. Their sizes are normal (four raster-units wide), .5, 2.0, and 4.0 times normal. Since each point is a character, it may also be displayed in four sizes: .75, 1.0, 1.25, and 1.5 times normal, for a total of 16 different point sizes. Lines may be drawn in four different widths by passing the beam, in the unblanked mode, through one of these points, then moving the beam between any two addressable positions on the screen. Special circuitry alternates the beam between the blanked and unblanked modes to draw dashed lines. The four dashed-line sizes are 32, 64, 128, and 256 raster units. The beam also moves at two speeds, normal and fast, to draw lines in two densities.

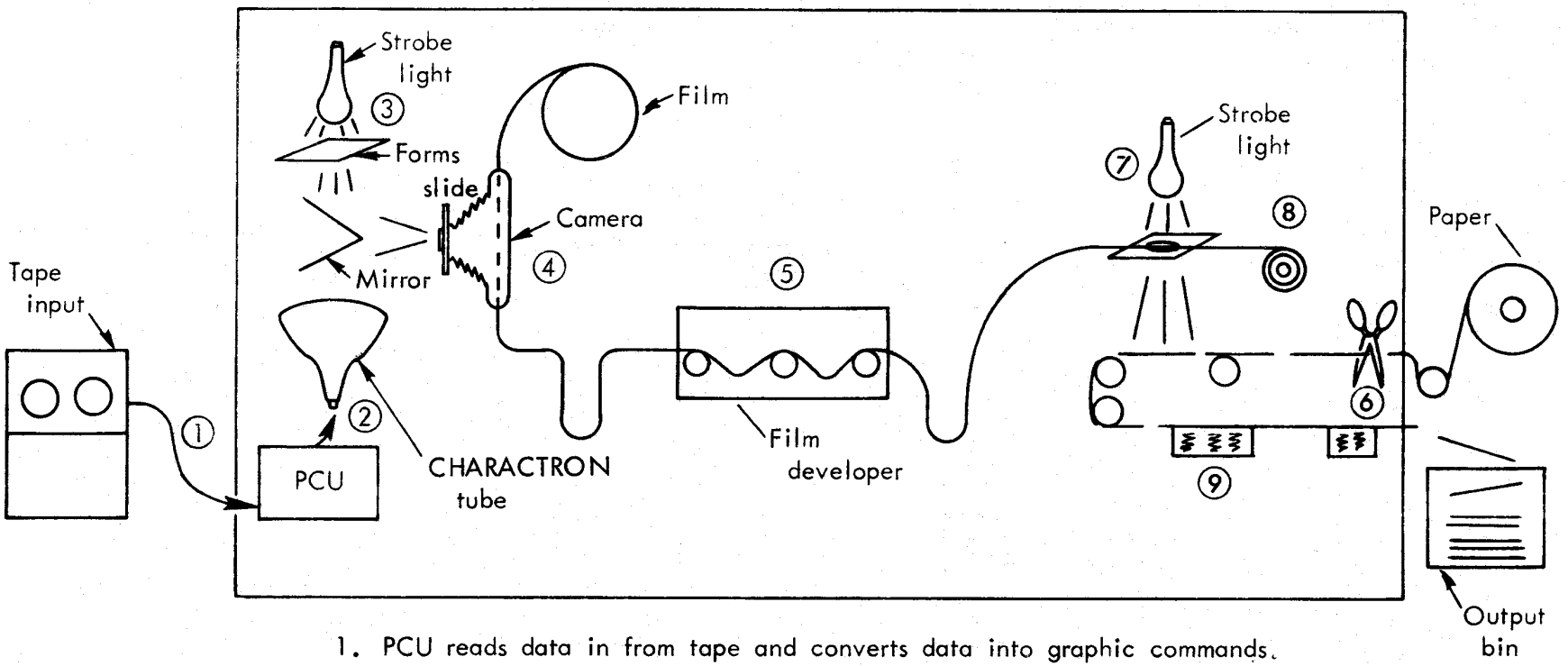
Characters may also be drawn with the stroke generator, a special S-C 4060 hardware feature. However, since IGS does not now support the stroke generator, it will be discussed no further.

When making movies, the operator can manually rotate the CRT 90^{o} counterclockwise, so the images appear on the film end-to-end as in normal movie film rather than side-by-side as in comic strips.

A 35-mm camera records CRT screen images by opening its shutter while the beam draws the displays. The operator can also mount a forms slide--a small plastic plate etched with an appropriate background display or form. The user then directs the S-C 4060 to flash a strobe light through the forms slide to record the form on film. When the display is complete, the camera shutter closes and the film is advanced. The film is then developed by the S-C 4060, or the operator may remove the film to be developed off-line.

Hardcopy is produced by flashing the film image onto specially treated paper, which is then fused and cut to the appropriate size. The paper can be cut to 8 1/2 × 11 or 11 × 14 inches, or can be advanced without being cut to produce a *strip chart*. A strip chart consists of an 11-inch-wide strip of paper of any length.

Figure 1 illustrates the S-C 4060 operation; Appendix F, p. 94, summarizes the S-C 4060 graphic features; and Ref. 2 details the S-C 4060 hardware.



1. PCU reads data in from tape and converts data into graphic commands.
2. Graphic commands direct a beam of light in CRT tube, drawing the picture.
3. Strobe light flashes through a plastic plate etched with background form.
4. Images are recorded on film in a camera.
5. Film is drawn through a developer.
6. Paper is cut to appropriate size.
7. Strobe light flashes film image onto specially treated paper.
8. Film is wound up on a spool.
9. Paper is fused and image is made permanent on the paper.

Fig. 1—Schematic of S-C 4060 operation

III. DESCRIPTION OF THE GRAPHICS SYSTEM

The graphics system consists of subroutines that draw lines, plot points, and display characters. These superficially simple functions are actually complex because of the many possible variations. For example, lines can be drawn in four weights, four dash sizes, and two densities. Mode sets allow the user to execute such simple actions as drawing lines without becoming confused by all the variations possible.

MODE SETS

Parameters are passed to IGS subroutines in two ways: as arguments in the subroutine calls and as values stored in the *mode set array*. The mode set array is defined as a 200-word, single-dimensioned, real array in the user's program and must appear as an argument in each call to an IGS subroutine. Default values are stored in this array when the system is initialized. Appendix A, pp. 83-88, lists the default values in the mode set array.

Parameters that change with each call are included as arguments in the subroutine calls; all others are held in the mode set array. Limiting the number of arguments necessary in the subroutine calls facilitates their use.

For example, to display characters, the user need specify only the characters themselves, the x,y location of the first character, and the number of characters. However, the mode set array automatically provides many other default parameters--character size, line spacing, margins, line orientation, etc. The user alters such values by calling subroutine SETSMG to store a new value in the array. These alterations will remain in effect until the user changes them again.

SUBJECT AND OBJECT SPACE

IGS automatically scales from user coordinates to absolute rasters. (If the user wishes to address absolute rasters himself, he can order the system to do no conversion.) The user must specify two items of information before the system can set up scaling factors. The first item is the size of the user's *subject space*, i.e., the coordinate system that bounds his data. The system will assume it to be the same as the raster size of the display surface, 4096.0 × 3072.0, unless otherwise informed through a call to subroutine SUBJEG.

Subject space is useful in both plotting--where the data's physical units may not correspond to the raster address of the S-C 4060--and textual displays. For example, by letting the x-axis range from 1 to 72 (the number of normal-sized characters per line) and the y-axis range from 1 to 52 (the number of lines per page), lines and character positions may be addressed directly, as on a typewriter.

The second item is the *object space*, i.e., the portion of physical display surface used. It is expressed in "normalized" units to differentiate it from raster units. A value of 1.0 is the smaller dimension of the display surface in either the x or y axis. For the S-C 4060, this is 3072 raster units. The system will assume it to be the full display surface size, 1.3333 × 1.0, unless otherwise informed through a call to subroutine OBJCTG. The ability to subdivide the display surface enables the user to put a margin around his picture, or to display multiple pictures simultaneously.

When an x or y coordinate is scaled into an area outside the physical screen, an error message is printed. The coordinate is set to zero if it is below or to the left of the screen, and to 3071 if above or 4095 if to the right.

Figure 2 illustrates the distinction between subject and object space.

CHARACTER DISPLAY

IGS permits two methods of displaying characters: CHARACTRON characters drawn by passing the beam through the stencil in the CRT, and vector characters drawn as a series of line segments. The S-C 4060 can display a maximum of 119 CHARACTRON characters in a maximum of four different sizes: .75, 1.0, 1.25, and 1.5 times the normal size of 31 by 51 raster units. These CHARACTRON characters may also be rotated 90^{0} counterclockwise, so an A would appear as ≺. The S-C 4060 can type at a rate of 40,000 characters per second. Although the CHARACTRON tube cannot display characters in varying levels of brightness, the same effect can be achieved by displaying them more than once at the same spot.

Each vector character is drawn as a series of line segments. The advantages of vector characters are that an unlimited number of characters and fonts (i.e., style of type) may be drawn in any size or orientation with any of the four line weights, two line densities, or four dashed-line sizes, and they may be skewed. Their disadvantages are that they require more time to draw on the S-C 4060, more elaborate tables, more time in the host computer, and the characters within the font must be designed by hand.

The quality of vector characters depends largely on the detail put into the design of the character fonts. CHARACTRON characters usually have better quality. IGS supplies one vector character font (FONT2) corresponding to the CHARACTRON characters. Reference 1 describes the means of building vector character fonts. Appendix I, p. 100, illustrates vector characters.

IGS features two kinds of character display: plotting and typing. The distinction allows the user to treat them

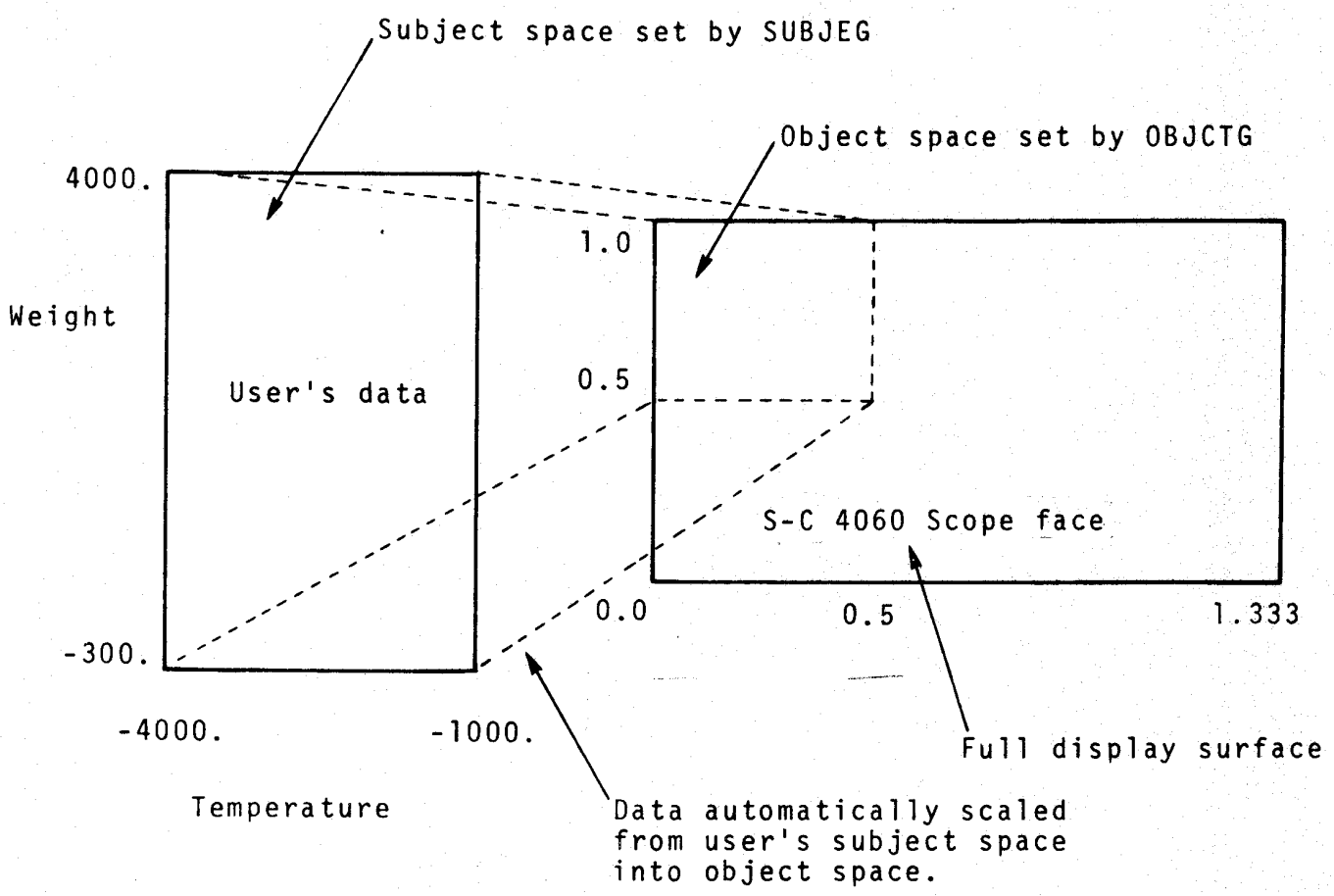


Fig. 2--Subject and Object Space

separately. For example, he may want to type with a large character size and plot with a small size. Plotting and typing require different mode sets.

The character size is always given in terms of its "envelope." Figure 3 illustrates the difference between the physical character size and its envelope. Note that the location of a character is its center.

PLOTTING

Plotting is the act of placing characters at random x, y coordinates on the screen. The character positions are random in that one character position does not depend on other character positions. IGS currently allows the user to plot with only CHARACTRON characters--119 characters in four sizes and two orientations provide ample plot symbols for most applications. However, the user may plot with vector characters by typing them one-at-a-time, with a penalty in host computer time.

TYPING

Typing is more difficult than plotting, because it involves more options for the user to consider. First, the character position depends on the previous position; normally, it is one to the right.

Second, typing offers many additional mode sets. The user may specify left, right, top, and bottom margins. He may command IGS to eject to a new page as his text reaches the bottom or top margin. He may change the character size, spacing, and orientation, and the line spacing and orientation. (Figures 4, 5, and 6 illustrate some of these options.) He may also specify character case, character font, and tab sets. Although determining these mode sets can be quite involved, IGS assumes default conditions, so the user need worry about them only when performing an unusual operation.

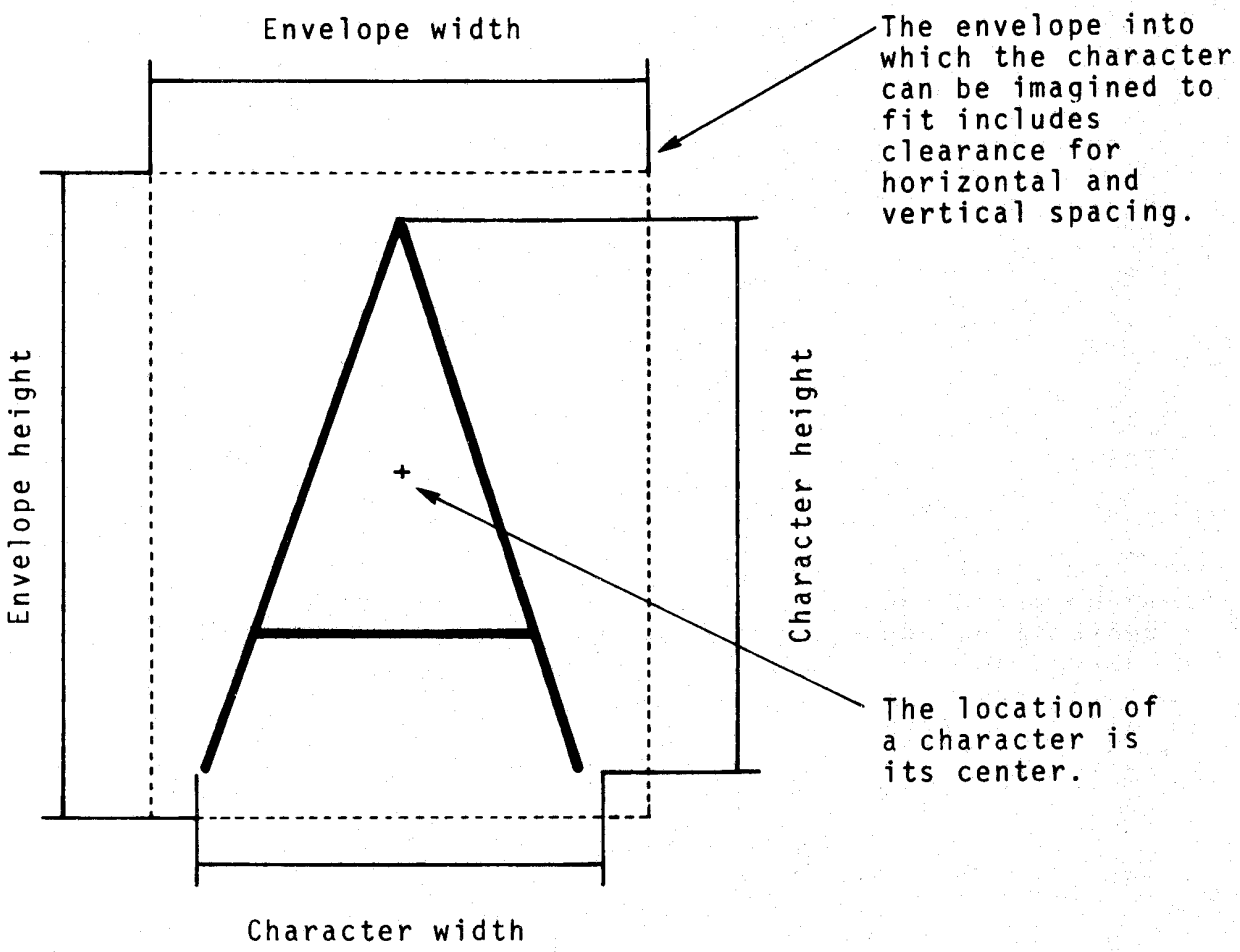
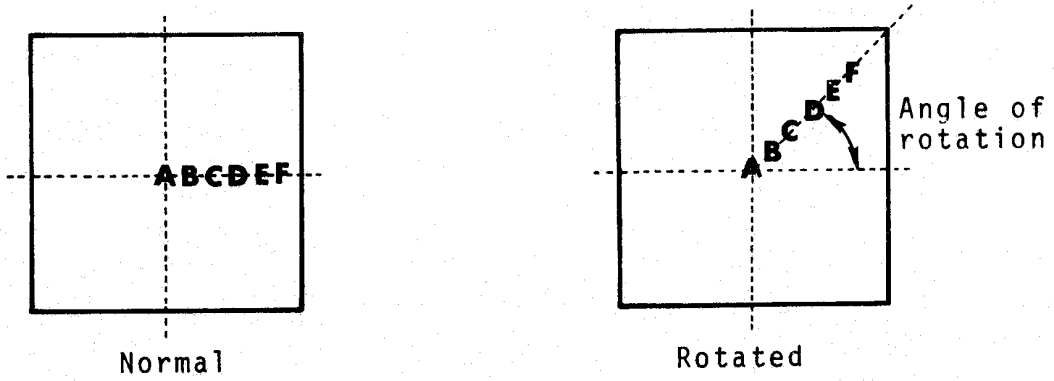
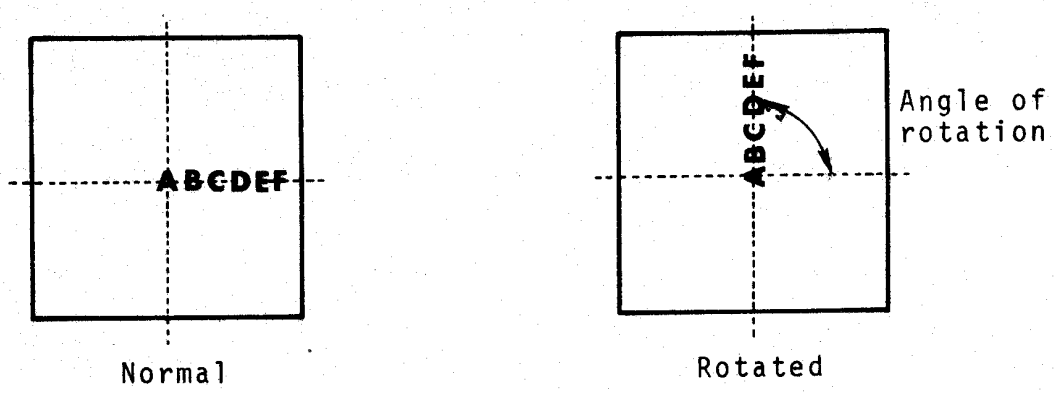


Fig. 3--The Character Envelope



NOTE--Changing the line orientation does not affect the character orientation.

Fig. 4--Line Orientation



NOTE--Changing the character orientation also changes the line orientation.

Fig. 5--Character Orientation

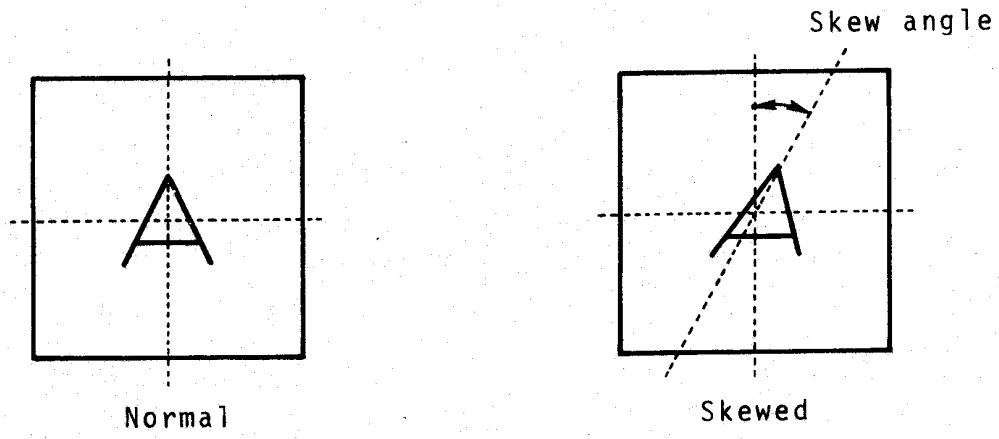


Fig. 6--Skew angle

The user may type with either CHARACTRON or vector characters; IGS treats them exactly alike. He indicates through a mode set which type he wants displayed; the default will be CHARACTRON characters.

The S-C 4060 has a 119-character set while most keypunches have only 48 or 63 characters. Special control characters, a single letter preceded by a dollar sign ($), may be punched directly in a character string to shift to upper, lower, or special case (upper case might be a B, lower case a b, and special case a β), so all the CHARACTRON characters may be used conveniently. For example, if the user wants lower case characters displayed, he precedes them with a $L. There is also a special character for upper and special case, capitalize, backspace, new page, line eject, tab, superscript, subscript, and a null. Control characters may be used for both CHARACTRON and vector characters. Appendix C, p. 91, describes all control characters; and Appendix D, p. 92, illustrates the character cases.

The user types with two subroutines--LEGNDG and TEXTG. LEGNDG begins typing at a specified x,y coordinate; TEXTG begins typing at the current point position, i.e., the position at which the CRT beam was left by the previous display. This may be the end of a line, the center of a plotted character, or one character space to the right of a typed character--depending on what was last displayed.

SUBROUTINE-NAMING AND -CODING CONVENTIONS

The following conventions have been adopted for the design and coding of IGS.

1) All general graphic subroutine names end with a G.
2) All system subroutine names not of general interest end with ZZ.
3) All function names end with a Z.
4) The first argument of each call to a graphic subroutine must be the mode set array defined in the user's program.

5) All variables in the mode set array are real.
6) Coordinate values in the calls to graphic subroutines are real.
7) Variable-length calls are not used unless one of the arguments contains the number of arguments in the call.
8) The subroutines are coded in ASA standard FORTRAN where possible.
9) The Operating System need not be modified.

THE STRUCTURE OF THE GRAPHICS SYSTEM

IGS is composed of several classes of subroutines: housekeeping, graphic output, graph, and conversion.

Housekeeping Subroutines

The user starts the graphics system with a call to MODESG, and stops it with a call to EXITG. After starting the system, he may call SUBJEG to tell it what units his data will occupy; the system will then calculate the scaling factors needed. He may also call OBJCTG to specify the portion of the S-C 4060 screen he wants to use. This enables him to subdivide the screen so he has, in effect, more than one scope in front of him. Both calls are optional. SETSMG and GETSMG set and retrieve mode set values, respectively. RSETMG sets all modes to their default values. TABSG sets typing tabs, PAGEG controls the forms flash and frame advance, and VECIG initializes a vector character font.

Graphic Output Subroutines

Graphic output subroutines display lines (LINESG, SEGMTG); characters (LEGNDG, TEXTG); circles or arcs (CIRARG); points (POINTG); numbers (NUMBRG) in I, E, F, or A format; and multiple line segments (MLTPLG) for drawing grids or crosshatching.

Graph Subroutines

Subroutines draw (GRIDG), label (LABELG), and title (TITLEG) grids. GRAPHG plots an entire graph, complete with grid, labels, titles, and data. An additional subroutine (SETUPG) computes aesthetically pleasing grid and label parameters for GRIDG and LABELG.

Conversion Subroutines

One of the two conversion subroutines converts numeric data to characters (FMTSG); the other converts characters to numeric (CONVTG).

IV. DESCRIPTION OF THE GRAPHIC SUBROUTINES

BRIEF LIST OF GRAPHIC SUBROUTINES

The following subroutines constitute the graphics system. A complete description of each subroutine follows this outline.

1) Housekeeping Subroutines

a) MODESG Initializes the graphics system. (This is the first graphic subroutine call.) (p. 21.)
b) SUBJEG Sets up the user's coordinate system (subject space) (pp. 22-23).
c) OBJCTG Sets up the portion of the S-C 4060 screen to be used (object space) (pp. 24-25).
d) SETSMG Stores a value in the mode set array (p. 26).
e) GETSMG Retrieves a value from the mode set array (p. 27).
f) RSETMG Resets all modes to their default values (p. 28).
g) TABSG Sets typing tabs (p. 29).
h) PAGEG Controls forms flash and frame advance. (This must be called to advance a frame.) (p. 30.)
i) VECIG Initializes a vector character font (pp. 31-32).
j) EXITG Terminates IGS. (This is the last call to the graphics system.) (p. 33.)

2) Graphic Output Subroutines

a) CIRARG Draws circles or arcs (pp. 35-36).
b) LEGNDG Displays characters or text at a given x,y location (pp. 37-41).

c) LINESG Draws joined lines (pp. 42-44).
d) MLTPLG Draws multiple line segments (pp. 45-46).
e) NUMBRG Displays numeric data (pp. 47-50).
f) POINTG Plots symbols or points (pp. 51-53).
g) SEGMTG Draws noncontiguous line segments (pp. 54-55).
h) TEXTG Displays characters or text from current point (p. 56).

3) Graph Subroutines

a) GRIDG Draws a grid (pp. 58-61).
b) LABELG Labels a grid (pp. 62-67).
c) TITLEG Titles a grid (pp. 68-71).
d) SETUPG Computes aesthetically pleasing grid and label parameters (pp. 72-75).
e) GRAPHG Draws, labels, and titles a complete graph (pp. 76-78).

4) Conversion Subroutines

a) CONVTG Converts characters to numeric (p. 79).
b) FMTSG Converts numeric data to characters (pp. 80-81).

FORMAT OF GRAPHIC SUBROUTINE WRITE-UPS

The graphic subroutine descriptions conform to the following format:

PURPOSE

Purpose of the subroutine.

USAGE

Standard Calls

Details of the subroutine calling sequence. FORTRAN notation indicates integer or real arguments. Arguments

beginning with A through H or O through Z are real; those beginning with I through N are integer.

Underscoring an argument designates it as an input value, either constant or variable, to the called subroutine. Overscoring an argument designates it as an output variable returned from the called subroutine. A line both above and below an argument indicates that the argument is both an input and an output variable. For example,

CALL SUBR(A,$\overline{B}$,$\overline{C}$)

1) A will contain an input value for SUBR.
2) B will return with a value from SUBR.
3) C will contain an input value for SUBR, and will return with a value from SUBR.

Special Calls

Special calls to the subroutine.

Modifications

Modifications of the subroutine through mode sets.

MISCELLANEOUS INFORMATION

Language in which the subroutine is coded and any additional information not covered above.

HOUSEKEEPING SUBROUTINES

This section briefly describes the housekeeping subroutines. The arguments in the calling sequences are detailed in the individual subroutine descriptions.

```
C     THE USER FIRST DIMENSIONS THE MODE SET ARRAY AND
C     INITIALIZES THE SYSTEM BY CALLING MODESG BEFORE
C     REQUESTING GRAPHIC OUTPUT.  MODESG SETS ALL DEFAULT
C     VALUES IN THE MODE SET ARRAY AND OPENS THE OUTPUT
C     FILES.  THE MODE SET ARRAY MUST BE 200 WORDS LONG.
          DIMENSION Z (200)
          CALL MODESG (Z, 0)
C     HE MAY THEN OPTIONALLY CHANGE THE COORDINATE SYSTEM
C     BY CALLING SUBJEG TO CHANGE THE SUBJECT SPACE, OR
C     OBJCTG TO CHANGE THE OBJECT SPACE.
          CALL SUBJEG (Z, XMIN, YMIN, XMAX, YMAX)
          CALL OBJCTG (Z, XMIN, YMIN, XMAX, YMAX)
C     SETSMG MAY BE CALLED TO CHANGE THE DEFAULT MODE SET
C     VALUES.
          CALL SETSMG (Z, NO, VALUE)
C     THE USER IS NOW READY TO CREATE DISPLAYS.  TO RESET
C     THE MODE SET ARRAY TO ITS DEFAULT VALUES, HE MAY
C     MAKE EITHER INDIVIDUAL CALLS TO SETSMG OR A SINGLE
C     CALL TO RSETMG.
          CALL RSETMG (Z)
C     GETSMG RETRIEVES MODE SET VALUES.
          CALL GETSMG (Z, NO, VALUE)
C     WHEN A DISPLAY IS COMPLETE, HE MUST CALL PAGEG TO
C     PRODUCE THE OUTPUT.
          CALL PAGEG (Z, 0, 1, 1)
C     THE USER MAY ESTABLISH SOME TAB SETS BY A CALL TO
C     TABSG.
          CALL TABSG (Z, NTABS, TABS)
```

```
C     EXITG SIGNALS THE END OF GRAPHIC OUTPUT.  IT MUST
C     BE CALLED BEFORE THE JOB IS TERMINATED.
           CALL EXITG (Z)
C     THE USER TERMINATES HIS PROGRAM IN THE NORMAL MANNER.
           CALL EXIT
           END
```

MODESG: Initializes the System

PURPOSE

Initializes the graphics system, opens files, and sets all default values in the mode set array. It must be the first graphic subroutine called.

USAGE

Standard Calls

CALL MODESG(Z,ITAPE)

1) Z must be a 200-word array defined in the user's program. All mode sets and system communication pointers are stored in this array.
2) ITAPE specifies the FORTRAN unit on which to write the S-C 4060 output. If ITAPE = 0, the default unit is selected.

Example

To initialize IGS with the default output unit,

```
CALL MODESG(Z,0)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV.

SUBJEG: Sets Up Subject Space

PURPOSE

Establishes the limits of the user's coordinate system (subject space), then calculates the scale factors.

USAGE

Standard Calls

CALL SUBJEG(Z,XMIN,YMIN,XMAX,YMAX)

1) Z is the mode set array.
2) XMIN is the minimum x-coordinate of the user's data.
3) YMIN is the minimum y-coordinate of the user's data.
4) XMAX is the maximum x-coordinate of the user's data.
5) YMAX is the maximum y-coordinate of the user's data.

Example

1) To set the subject space to range from 100 to -100 in the x-axis and -0.2 to 0 in the y-axis,

```
CALL SUBJEG(Z,100.,-0.2,-100.,0.)
```

2) To set the subject space so characters may be addressed directly, with 52 lines per page and 72 characters per line,

```
CALL SUBJEG(Z,1.,52.,72.,1.)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV. XMIN, YMIN, XMAX, and YMAX specify the boundaries of the user's data. XMIN and YMIN are not necessarily less than XMAX or YMAX. However, XMIN must not equal XMAX, or YMIN equal YMAX. SUBJEG recomputes the page margins so their relative positions in the new subject space are the same as in the old.

OBJCTG: Sets Up Object Space

PURPOSE

Establishes the size of the display scope (object space), then calculates the scale factors.

USAGE

Standard Calls

CALL OBJCTG(Z,XMIN,YMIN,XMAX,YMAX)

1) Z is the mode set array.
2) XMIN is the minimum x-coordinate of the screen in normalized object space.
3) YMIN is the minimum y-coordinate of the screen in normalized object space.
4) XMAX is the maximum x-coordinate of the screen in normalized object space.
5) YMAX is the maximum y-coordinate of the screen in normalized object space.

Special Calls

Two functions are provided to convert subject space coordinates into normalized object space.

1) To convert x-axis subject space to normalized object space, use the following function:

   x normalized object space = XNORMZ(Z,X)

2) To convert y-axis subject space to normalized object space, use the following function:

   y normalized object space = YNORMZ(Z,Y)

Modifications

Normalized object space is based upon a unit value of 1.0 in the x- and y-axes. The user may change these values by making a mode set call. The new normalized object space coordinates must be used wherever normalized object is required. The mode set calls are as follows:

1) To change the maximum x-axis normalized object space,

   CALL SETSMG(Z,19,XMAX)

   XMAX is the new normalized x-axis object space. XMAX = 1.333 is the default value.

2) To change the maximum y-axis normalized object space,

   CALL SETSMG(Z,20,YMAX)

   YMAX is the new normalized y-axis object space. YMAX = 1.0 is the default value.

Example

To set the object space so only the upper left quarter of the screen is used,

```
CALL OBJCTG(Z,0.0,.5,.6666,1.)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV. XMAX and YMAX must be greater than XMIN and YMIN, respectively.

SETSMG: Sets Mode Values

PURPOSE

Sets a single value in the mode set array. Mode sets stay in effect until specifically reset, or until the system is re-initialized. All mode sets have a unique default value set by the initialization call to MODESG and by calls to RSETMG.

USAGE

Standard Calls

CALL SETSMG(Z,NO,VALUE)

1) Z is the mode set array.
2) NO is a number identifying the particular mode set.
3) VALUE is the mode set value. See Appendix A (pp. 83-88) for a list of the individual mode sets possible.

Example

To set the typing character size to 1.5 times normal,

```
CALL SETSMG(Z,45,1.5)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV.

GETSMG: Retrieves Mode Values

PURPOSE

Retrieves a single value from the mode set array.

USAGE

Standard Calls

CALL GETSMG(Z,NO,VALUE)

1) Z is the mode set array.
2) NO is a number identifying the particular mode set.
3) VALUE will contain the mode set value on return. See Appendix A (pp. 83-88) for a description of the individual mode sets possible.

Example

To retrieve the number of lines per page,

```
CALL GETSMG(Z,65,VALUE)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV.

RSETMG: Resets all Default Values in Mode Set Array

PURPOSE

Resets all values in the mode set array to their proper default values.

USAGE

Standard Calls

CALL RSETMG(Z)

Z is the mode set array.

MISCELLANEOUS INFORMATION

Written in FORTRAN IV.

TABSG: Sets Typing Tabs

PURPOSE

Establishes tabs similar to those on a typewriter. The main differences are that the tabs are set to an x-coordinate location rather than to a character position, and the tab is actuated by a special tab character ($T) rather than by manually hitting a tab key.

USAGE

Standard Calls

CALL TABSG(Z,NTABS,TABS)

1) Z is the mode set array.
2) NTABS is the number of tabs--a maximum of ten is allowed.
3) TABS is an array of NTABS x-coordinates specifying the tabs to set.

Example

To set typing tabs at x = 20, 30, and 40,

```
CALL TABSG(Z,3,TAB)
DIMENSION TAB(3)
DATA TAB/20.,30.,40./
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV. Each tab character encountered in a character string causes a jump to the next tab set. If a tab character is encountered when no more tabs are set on a line, the carriage skips to the right margin and the next character encountered will cause an ejection to a new line. Each new line resets the tab counter as a carriage return restarts the tabbing sequence on a typewriter.

PAGEG: Controls Forms Flash and Frame Advance

PURPOSE

Controls the forms flash, produces hardcopy, and advances the frame. The forms flash exposes the forms overlay on the current frame. Frame advance signals the completion of the current picture and advances the film.

USAGE

Standard Calls

CALL PAGEG(Z,NFORM,NCOPY,NADVAN)

1) Z is the mode set array.

2) NFORM indicates the forms flash.

   If NFORM = 0, no forms flash.
   > 0, flash the forms NFORM times.

3) NCOPY specifies the number of hardcopies of the frame to make. If NCOPY = 0, no copies are made.

4) NADVAN specifies the number of frames to advance the film. If NADVAN = 0, the frame is not advanced. (One advance is normal.)

Example

To advance the film with no forms flash and obtain one hardcopy,

```
CALL PAGEG(Z,0,1,1)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV.

VECIG: Initializes a Vector Character Font

PURPOSE

Before using vector characters, call VECIG to initialize the font and SETSMG to change the character-font mode set to command LEGNDG and TEXTG to produce vector characters.

A vector character font is an array of data, usually compiled as a separate control section. Reference 1 describes the method for creating a font. Since vector characters are not subject to the limitations imposed on CHARACTRON characters, the user may alter their size, orientation, width, and height freely, or skew them. (See Figs. 4, 5, and 6.) He may also use vector characters to type characters that do not exist in the CHARACTRON set (e.g., Cyrillic, Hebrew, or Chinese) or whose style is different (e.g., English Script).

USAGE

Standard Calls

CALL VECIG(Z,FONT,NTYPE)

1) Z is the mode set array.
2) FONT is the name of the font, i.e., the name of the array containing the font data.
3) NTYPE specifies whether the font is internal or external to the calling program.

   If NTYPE = 0, the font is stored as an external BLOCK DATA subprogram, and the font must be defined as an EXTERNAL within the calling program.

   = 1, the font is stored as an internal array in the calling program.

Modifications

1) To inform the system that LEGNDG and TEXTG are to produce vector characters from the initialized font,

```
CALL SETSMG(Z,51,1.)
```

2) To reset to the default condition of CHARACTRON characters,

```
CALL SETSMG(Z,51,0.)
```

Example

To initialize FONT2, the IGS vector character font, and set the mode for vector characters,

```
CALL VECIG(Z,FONT2,0)
CALL SETSMG(Z,51,1.)
EXTERNAL FONT2
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV. Only one font is effective at any one time. To change to a new font, recall VECIG. Appendix H contains examples of vector character usage.

EXITG: Closes Down the Graphic Devices

PURPOSE

Turns off the graphic devices and closes the output files.

USAGE

Standard Calls

CALL EXITG(Z)

Z is the mode set array.

Special Calls

A call is provided to terminate the user's program abnormally without knowing the user's mode set array. The call is the same as that above, minus the reference to the mode set array:

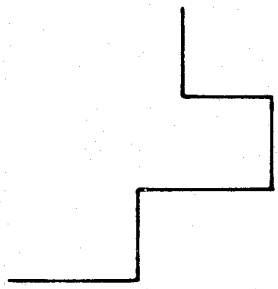
CALL EXITG(0)

MISCELLANEOUS INFORMATION

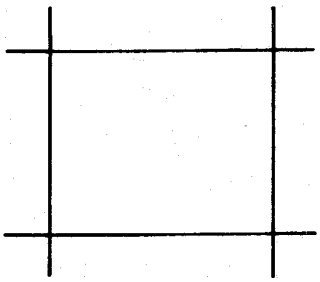
Written in FORTRAN IV.

GRAPHIC OUTPUT SUBROUTINES

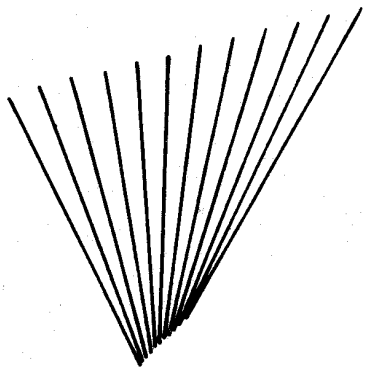
Figure 7 illustrates the graphic output subroutines.



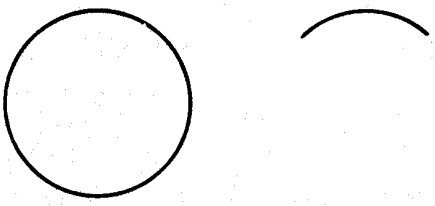
Draws joined lines--LINESG



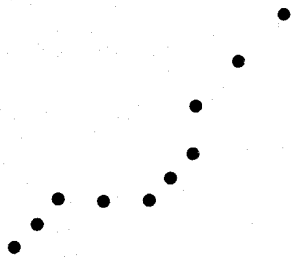
Draws line segments--SEGMTG



Draws multiple lines--MLTPLG



Draws circles and arcs--CIRARG



Plots points--POINTG

A
BC
DEfg
h*/K

Types characters--LEGNDG,TEXTG

27
36.293
0.263E+06
$0.263 \times 10^{+06}$

Displays numeric data--NUMBRG

Fig. 7--Graphic Output Subroutines

CIRARG: Draws a Circle or an Arc

PURPOSE

Draws a circle or arc. Since the S-C 4060 can not draw a curved line, the circle or arc is approximated by drawing short line segments.

USAGE

Standard Calls

CALL CIRARG(Z,XC,YC,RADIUS,START,ARC)

1) Z is the mode set array.
2) XC,YC is the x,y location of the center point of the arc.
3) RADIUS is the radius of curvature. It should be in the same units as the x coordinates.
4) START is the angle, in degrees, at which to start drawing. Counterclockwise angles are positive (0^{o} is horizontal and to the right).
5) ARC is the number of degrees of arc to display. For example, an arc of 720^{o} would cause two complete circles to be drawn, one on top of the other.

Modifications

Use mode sets to modify CIRARG as follows:

1) To change the tangent angle between two successive points,

   CALL SETSMG(Z,83,ANGLE)

ANGLE is the tangent angle in degrees between two points on the arc. ANGLE = 5.0^{o} is the default value.

2) To draw the arc with points rather than line segments,

CALL SETSMG(Z,82,1.)

To reset to draw the arc with line segments,

CALL SETSMG(Z,82,0.)

When an arc is drawn with points, all the mode sets described in POINTG are in effect. Points will be plotted at each end of an arc. Thus, if a 360^{o} arc is specified, points will be displayed at 0^{o} and at 360^{o}, which turn out to be the same location.

Example

1) To draw a circle with radius of 50 at location 100,100,

CALL CIRARG(Z,100.,100.,50.,0.,360.)

2) To draw a 90^{o} arc with radius of 100 in the third quadrant at location 1000,2000,

CALL CIRARG(Z,1000.,2000.,100.,180.,90.)

MISCELLANEOUS INFORMATION

Written in FORTRAN IV. All mode sets described in LINESG will be in effect if the circle is drawn with line segments, and all mode sets described in POINTG will be in effect if the circle is drawn with points.

LEGNDG: Types Characters

PURPOSE

Types characters, starting at a given x,y location. If the characters extend beyond the right margin, the line is ejected to the left margin; if beyond the bottom margin, the line is ejected to the top.

USAGE

Standard Calls

CALL LEGNDG(Z,X,Y,N,CHAR)

1) Z is the mode set array.
2) X,Y is the location of the center of the first character.
3) N is the number of characters to type ($N \geq 1$).
4) CHAR is the name of a variable or an array that contains the characters to type.

Modifications

Use mode sets to modify LEGNDG as follows:

1) To change the character size,

   CALL SETSMG(Z,45,SIZE)

   SIZE specifies the size of the characters relative to normal size; 0.75, 1.0, 1.25, and 1.5 are allowed for the S-C 4060. SIZE = 1.0 is the default value.

2) To change the character width (vector characters only),

CALL SETSMG(Z,42,W)

W is the character width in subject space units. W = 31.0 is the default value.

3) To change the character height (vector characters only),

CALL SETSMG(Z,43,H)

H is the character height in subject space units. H = 51.0 is the default value.

4) To change the line and character orientation,

CALL SETSMG(Z,46,ORIEN)

ORIEN designates the angle in degrees at which the line and characters are to be displayed. (0^{o} is horizontal and to the right.) Counterclockwise angles are positive. Only 0^{o} and 90^{o} are allowed for CHARACTRON characters. ORIEN = 0.0 is the default value.

5) To change the character spacing,

CALL SETSMG(Z,47,+-SPACE)

SPACE specifies the character spacing (positive to the right, negative to the left) for typing. SPACE = 1.0 is the default value.

6) To specify the number of lines to space up or down if the characters exceed the right margin, causing the line to be ejected to the left margin,

CALL SETSMG(Z,49,+-SPACE)

SPACE specifies the number of lines to space up or down. This is equivalent to

setting a typewriter carriage return for the number of lines to space. A minus spacing causes the line to be spaced down the page; and a positive spacing, up. SPACE = -1.0 is the default value.

7) To change the line orientation,

CALL SETSMG(Z,50,ANGLE)

ANGLE designates the angle in degrees at which the line of characters is to be displayed (0^{o} is horizontal and to the right). Counterclockwise angles are positive. This call does not change the character orientation. ANGLE = 0.0 is the default value.

8) To specify the margins of the page,

CALL SETSMG(Z,60,XLEFT)

XLEFT designates the left margin of the page in subject space coordinates. XLEFT = 0.0 is the default value.

CALL SETSMG(Z,61,RIGHT)

RIGHT specifies the right margin of the page in subject space coordinates. RIGHT = 4095.0 is the default value.

CALL SETSMG(Z,62,BOTTOM)

BOTTOM stipulates the bottom margin of the page in subject space coordinates. BOTTOM = 0.0 is the default value.

CALL SETSMG(Z,63,TOP)

TOP fixes the top margin of the page in subject space coordinates. TOP = 3071.0 is the default value.

9) To specify the form of the x,y coordinates used in the call,

CALL SETSMG(Z,14,TYPE)

TYPE fixes the form of the x,y coordinates.

If TYPE = 0.0, x,y are real subject space units.
= 1.0, x,y are real absolute raster units.
= 2.0, x,y are integer absolute raster units.
= 3.0, x,y are real normalized object space units.

10) To change the character font,

CALL SETSMG(Z,51,FONT)

If FONT = 0.0, the font is CHARACTRON characters.
= 1.0, the font is vector characters.
= 2.0,3.0,...,N.0, the font will be defined later.

All mode sets described in LINESG are in effect for vector characters. FONT = 0.0 is the default value.

11) To set the skew angle (vector characters only),

CALL SETSMG(Z,131,ANG)

ANG is the skew angle in degrees and must be between $+90^{o}$ and -90^{o}. 0^{o} designates no skew, positive skew is to the left, and negative to the right. ANG = 0.0 is the default value.

12) To specify the character case by a mode set rather than by the special control characters listed in Appendix C, p. 91,

CALL SETSMG(Z,52,CASE)

If CASE = 0.0, upper case.
= 1.0, lower case.
= 2.0, special case.
CASE = 0.0 is the default value.

13) To specify a frame advance if the line exceeds the top or bottom margins,

CALL SETSMG(Z,44,1.)

To reset to the default condition when no frame advance is specified,

CALL SETSMG(Z,44,0.)

Example

To type the sentence "This illustrates UPPER and lower case." in the lower left corner of the page,

```
CALL LEGNDG(Z,0.,0.,44,44HT$LHIS ILLUSTRATES
$UUPPER $LAND LOWER CASE.)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV.

LINESG: Draws Joined Lines

PURPOSE

Draws connected line segments, joining points in an x,y array.

USAGE

Standard Calls

CALL LINESG(Z,N,X,Y)

1) Z is the mode set array.
2) N is the number of x,y coordinates. N - 1 joined line segments will be drawn.
3) X,Y are arrays of x,y coordinates. Lines are drawn between consecutive pairs of points in the arrays.

Special Calls

1) To draw a line from the current point position to an x,y location,

   CALL LINESG(Z,1,X,Y)

   a) Z is the mode set array.
   b) X,Y is the x,y position to which to draw the line.

2) To position the current point to an x,y location without displaying anything,

   CALL LINESG(Z,0,X,Y)

   a) Z is the mode set array.
   b) X,Y is the x,y point at which to position the beam.

Modifications

Use mode sets to modify LINESG as follows:

1) To specify the line width,

CALL SETSMG(Z,30,TIMES)

TIMES designates the line width relative to the normal line width of 4 raster units.

If TIMES = 0.5, line width is one-half times normal (2 raster units).
= 1.0, line width is normal (4 raster units).
= 2.0, line width is two times normal (8 raster units).
= 4.0, line width is four times normal (16 raster units).

TIMES = 1.0 is the default value.

2) To specify that a dashed line be drawn,

CALL SETSMG(Z,31,DASH)

DASH specifies the dash size.

If DASH = 0.0, dash makes solid line.
= 1.0, dash size is 32 raster units.
= 2.0, dash size is 64 raster units.
= 3.0, dash size is 128 raster units.
= 4.0, dash size is 256 raster units.

DASH = 0.0 is the default value.

3) To specify the line density,

CALL SETSMG(Z,94,DEN)

If DEN = 0.0, line density is normal.
= 1.0, line density is light.

DEN = 0.0 is the default value.

4) To specify the form of the x,y coordinates used in the call,

CALL SETSMG(Z,14,TYPE)

TYPE designates the form of the x,y coordinates.

If TYPE = 0.0, x,y are real subject space units.
= 1.0, x,y are real absolute raster units.
= 2.0, x,y are integer absolute raster units.
= 3.0, x,y are real normalized object space units.

Example

To draw a rectangle encompassing the entire screen,

```
CALL LINESG(Z,5,X,Y)
DIMENSION X(5),Y(5)
DATA X/0.,4095.,4095.,0.,0./,Y/0.,0.,3071.,3071.,0./
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV.

MLTPLG: Draws Multiple Line Segments

PURPOSE

Draws multiple line segments. First, two line segments are drawn. Next, a given number of line segments are drawn between the two lines, in a direction similar to that of the original lines. MLTPLG facilitates drawing grid lines, crosshatching, and shading. It can draw lines at any orientation to each, not just parallel.

USAGE

Standard Calls

CALL MLTPLG(Z,N,X1,Y1,X2,Y2,X3,Y3,X4,Y4)

1) Z is the mode set array.
2) N specifies the number of lines to draw between the two given lines. If N = 0, only the two given lines will be drawn.
3) X1,Y1 and X2,Y2 are two points specifying a line.
4) X3,Y3 and X4,Y4 are two points specifying a line.

Modifications

All mode sets described in LINESG are applicable to this subroutine.

Example

1) To draw four horizontal, parallel lines encompassing the entire screen,

```
CALL MLTPLG(Z,2,0.,0.,4095.,0.,0.,3071.,4095.,3071.)
```

2) To draw a rectangle encompassing the entire screen,

```
CALL MLTPLG(Z,0,0.,0.,4095.,0.,4095.,0.,4095.,3071.)
CALL MLTPLG(Z,0,4095.,3071.,0.,3071.,0.,3071.,0.,0 )
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV. MLTPLG has many uses not at first apparent. The two delimiting lines need not be parallel, as illustrated in Fig. 7. It is a convenient means of drawing any two line segments with a single call. If the delimiting line segments are very short, the vectors drawn will appear as points. This allows the user to plot many points with a single call--exactly what one needs for shading.

NUMBRG: Displays Numeric Data

PURPOSE

Displays integer or real numbers.

USAGE

Standard Calls

1) To display an *integer* number in *I format*,

   CALL NUMBRG(Z,X,Y,IFMT,INTG)

   a) Z is the mode set array.

   b) X,Y is the x,y position at which to display the number, i.e., the center of the leftmost character.

   c) IFMT is an integer indicating the number of digits to display.

   d) INTG is the number to display.

2) To display a *real* number in *F format*,

   CALL NUMBRG(Z,X,Y,FMT,REAL)

   a) Z is the mode set array.

   b) X,Y is the x,y position at which to display the number, i.e., the center of the leftmost character.

   c) FMT is a real number that specifies the format. It is of the form w.d, where w specifies the field width and d the number of decimal places ($d < 10$). Use the special call if $d \geq 10$.

   d) REAL is the number to display.

3) To display a *real* number in *E format*,

   CALL NUMBRG(Z,X,Y,-FMT,REAL)

a) Z is the mode set array.

b) X,Y is the x,y position at which to display the number, i.e., the center of the leftmost character.

c) -FMT is a real number that specifies the format. It is of the form w.d, where w is the field width and d the number of decimal places (d < 10). Use the special call if d ≥ 10. Four places must be provided for the exponent and two places for the sign and digit to the left of the decimal place. In general, w = d + 7.

d) REAL is the number to display.

4) To display *characters* in *A format* (this call is equivalent to a call to LEGNDG),

CALL NUMBRG(Z,X,Y,-IFMT,CHARS)

a) Z is the mode set array.

b) X,Y is the x,y position at which to display the number, i.e., the center of the leftmost character.

c) -IFMT is an integer indicating the number of characters to display.

d) CHARS is the name of a variable or array that contains the character string to display.

Special Calls

A special call allows the format to be specified by mode sets:

CALL NUMBRG(Z,X,Y,0,VALUE)

VALUE is the name of the variable to display. The format must be specified by the following mode set calls prior to the call to NUMBRG:

1) CALL SETSMG(Z,77,WIDTH)

   WIDTH is the field width.

2) CALL SETSMG(Z,78,DEC)

   DEC is the number of decimal places.

3) CALL SETSMG(Z,79,FMT)

   FMT is the format in which to display the number.

   If FMT = 1.0, I format.
   = 2.0, F format.
   = 3.0, E format.
   = 4.0, A format.

Modifications

All mode sets described in LEGNDG are also applicable to this subroutine. Use mode sets to modify NUMBRG as follows:

1) To display numbers in scientific notation rather than E format,

   CALL SETSMG(Z,85,1.)

2) To reset to the default condition of E format,

   CALL SETSMG(Z,85,0.)

The number -.00116 would appear as -0.116E-02 in E format and as -0.116×10^{-02} in scientific notation.

Example

1) To display the variable N in I6 format at location 0,0,

   CALL LEGNDG(Z,0.,0.,6,N)

2) To display the number 3.1416 in F6.4 format at location 0,100,

```
CALL LEGNDG(Z,0.,100.,6.4,3.1416)
```

3) To display the same number at location 0,200 in E12.5 format so it appears as 0.31416E+01,

```
CALL LEGNDG(Z,0.,200.,-12.5,3.1416)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV.

POINTG: Plots Points

PURPOSE

Plots arrays of points.

USAGE

Standard Calls

CALL POINTG($\underline{Z}$,$\underline{N}$,$\underline{X}$,$\underline{Y}$)

1) Z is the mode set array.
2) N is the number of points ($N \geq 1$).
3) X,Y are arrays of N x,y coordinate pairs to plot.

Modifications

Use mode sets to modify POINTG as follows:

1) To specify a new plotting symbol (the default character is a normal-sized point),

   CALL SETSMG($\underline{Z}$,$\underline{84}$,$\underline{VALUE}$)

   VALUE must contain the character, left-justified, to use for plotting.

   To reset to the default value of a point,

   CALL SETSMG($\underline{Z}$,$\underline{84}$,$\underline{0.}$)

2) To change the plot-character size,

   CALL SETSMG($\underline{Z}$,$\underline{53}$,$\underline{SIZE}$)

   SIZE specifies the size of the characters relative to normal (.75, 1.0, 1.25, and 1.5 are allowed). SIZE = 1.0 is the default value.

3) To change the plot-character orientation,

CALL SETSMG(Z,46,ANGLE)

ANGLE designates the angle in degrees at which the character is to be displayed. (0^{o} is horizontal and to the right.) Counterclockwise angles are positive. Only 0^{o} and $+90^{o}$ are allowed. ANGLE = 0.0 is the default value.

4) To change the plot-character case,

CALL SETSMG(Z,55,CASE)

If CASE = 0.0, upper case.
= 1.0, lower case.
= 2.0, special case.

CASE = 0.0 is the default value. Appendix D, p. 92, illustrates the character cases.

5) To specify the form of the x,y coordinates used in the call,

CALL SETSMG(Z,14,TYPE)

TYPE designates the form of the x,y coordinates.

If TYPE = 0.0, x,y are real subject space units.
= 1.0, x,y are real absolute raster units.
= 2.0, x,y are integer absolute raster units.
= 3.0, x,y are real normalized object space units.

Example

To plot an array of 100 points contained in the arrays XVAL and YVAL with a lozenge (⌑) as a plot symbol,

```
CALL SETSMG(Z,55,2.)
CALL SETSMG(Z,84,1HU)
CALL POINTG(Z,100,XVAL,YVAL)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV.

SEGMTG: Draws Line Segments

PURPOSE

Draws noncontiguous line segments. Arrays X1,Y1 specify the initial x,y coordinates of each segment; arrays X2,Y2 specify the terminal x,y coordinates.

USAGE

Standard Calls

CALL SEGMTG(Z,N,X1,Y1,X2,Y2)

1) Z is the mode set array.
2) N is the number of line segments to draw. If N > 1, X1,Y1 and X2,Y2 must be arrays of points.
3) X1,Y1 are arrays containing the starting points of each line.
4) X2,Y2 are arrays containing the terminal points of each line.

Modifications

All mode sets described in LINESG are also applicable to this subroutine.

Example

To draw four line segments to form a rectangle encompassing the entire screen,

```
CALL SEGMTG(Z,4,X1,Y1,X2,Y2)
DIMENSION X1(4),Y1(4),X2(4),Y2(4)
DATA X1/0.,4095.,4095.,0./,Y1/0.,0.,3071.,3071./,
X2/4095.,4095.,0.,0./,Y2/0.,3071.,3071.,0./
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV.

TEXTG: Types Characters

PURPOSE

Types characters, beginning at the current point position. If the characters extend beyond the right margin, the line is ejected to the left margin. TEXTG resembles LEGNDG, but begins typing where the previous display ended, while LEGNDG begins typing at a specified x,y coordinate.

USAGE

Standard Calls

CALL TEXTG(Z,N,CHAR)

1) Z is the mode set array.
2) N is the number of characters to type ($N \geq 1$).
3) CHAR is the name of the variable or array that contains the characters to type.

Modifications

All mode sets described in LEGNDG are also applicable to this subroutine.

Example

To type the text "Horizontal Line" at the end of a horizontal line,

```
CALL SEGMTG(Z,1,0.,0.,200.,0.)
CALL TEXTG(Z,19,19HH$LORIZONTAL $CLINE)
```

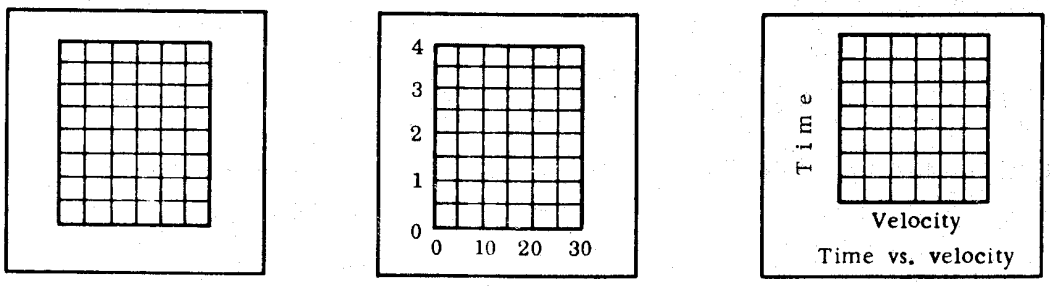
MISCELLANEOUS INFORMATION

Written in FORTRAN IV. LINESG may be used to position the beam to any specified location.

GRAPH SUBROUTINES

A graph, in addition to the data plotted on it, is composed of three separate parts: a grid of lines, labels of the major axes, and titles. Three corresponding subroutines produce the graphs. The user may then plot data on the graphs with the graphic output subroutines such as POINTG or LINESG.

A special subroutine, SETUPG, aids the user in computing arguments that produce aesthetically pleasing graphs. These subroutines are combined in subroutine GRAPHG to draw, label, title, and plot an entire graph.



Grid drawn by GRIDG

Labels made by LABELG

Titles made by TITLEG

Fig. 8--Graph Subroutines

GRIDG: Draws a Grid

PURPOSE

Constructs cartesian grids--draws all lines and emphasizes the major grid divisions. GRIDG assumes that the user has established his subject space with SUBJEG and his object space with OBJCTG. The object space should provide a margin because the grid will fill the entire object space. Figure 9 illustrates the grid.

USAGE

Standard Calls

CALL GRIDG(Z,DX,DY,IXTH,JYTH)

1) Z is the mode set array.

2) DX,DY are the x,y grid intervals in user coordinates. If DX or DY = 0.0, no grid lines are drawn for that axis. DX controls spacing between the vertical grid lines and DY the horizontal. If the user's x-axis subject space decreases from left to right, or the y-axis subject space decreases from bottom to top, DX or DY must be negative, respectively. If the zero-axis lies within the grid, DX or DY are spaced from the zero-axis outward. Otherwise, they are spaced from left to right or from bottom to top. The grid lines are drawn with 0.5 times normal line width.

3) IXTH,JYTH specify the i^{th} x-grid line and the j^{th} y-grid line to emphasize. If IXTH or JYTH = 0.0, no lines are emphasized for it. The emphasis lines are drawn with twice the normal line weight.

CALL GRIDG(Z,DX,DY,IXTH,JYTH)

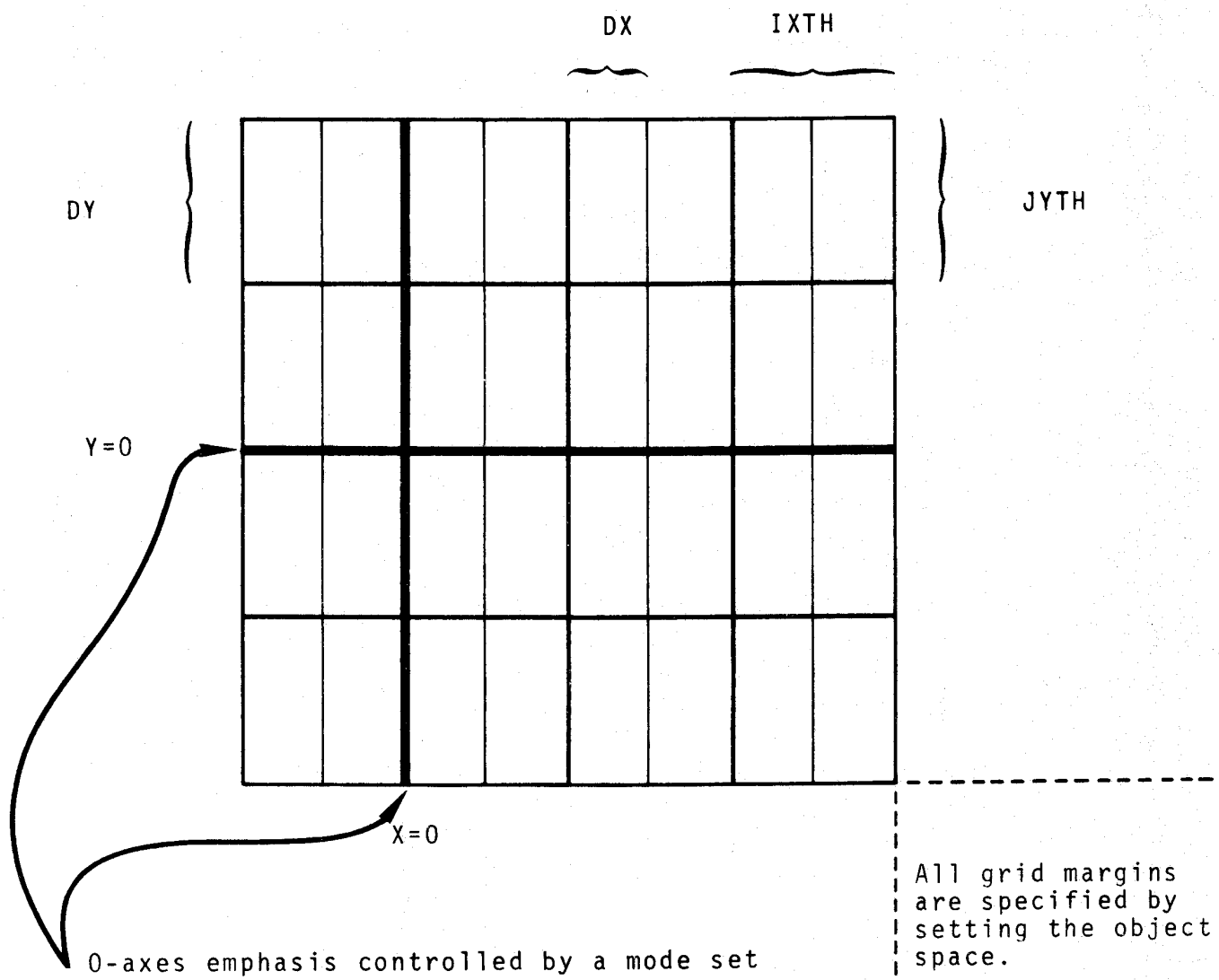


Fig. 9--GRIDG Subroutine

Special Calls

A log grid can be drawn by making one of the mode sets described below. The subject space must be positive and should be a power of ten. To draw a log grid, make the indicated mode set call and then call GRIDG. The arguments have the following special meanings for the log grids:

1) DX or DY is ignored and may be a dummy argument if the grid is log in x or y.

2) IXTH,JYTH specify whether the major cycle lines are to be emphasized. If zero, no emphasis occurs; if greater than zero, emphasis occurs.

Modifications

Use mode sets to modify GRIDG as follows:

1) To control the emphasis of the major axes,

CALL SETSMG(Z,100,EMPH)

EMPH specifies what is to be emphasized.
EMPH = 0.0 is the default value.

If EMPH = 0.0, emphasize x = 0, y = 0 axes.
= 1.0, emphasize x = 0 axis only.
= 2.0, emphasize y = 0 axis only.
= 3.0, no emphasis.

The emphasis lines are drawn with four times the normal line weight. They will not be drawn if the zero-axis lies outside the grid. For nonlinear grids, the major axes are defined as the X = 1, Y = 1 axes.

2) To draw a log grid in the x-axis,

CALL SETSMG(Z,23,1.)

To return to the default condition of a linear grid,

CALL SETSMG(Z,23,0.)

Set up the subject space for the log grid before making this mode set call.

3) To draw a log grid in the y-axis,

CALL SETSMG(Z,24,1.)

To return to the default condition of a linear grid,

CALL SETSMG(Z,24,0.)

Set up the subject space for the log grid before making this mode set call.

Example

To draw the grid in Fig. 9, assume the x-axis ranges from -20 to 60 and the y-axis from -100 to 100; leave a 0.2-normalized-object-space margin around the grid; then

```
CALL SUBJEG(Z,-20.,-100.,60.,100.)
CALL OBJCTG(Z,.2,.2,1.133,.8)
CALL GRIDG(Z,10.,50.,2,1)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV.

LABELG: Labels a Grid

PURPOSE

Labels the axes of a grid drawn by GRIDG. The user may label both the x and y axes with numeric or alphanumeric labels. The y-axis will be labeled to the left of the grid and the x-axis below it. Figure 10 illustrates the labels.

USAGE

Standard Calls

1) To create alphameric labels,

CALL LABELG($\underline{Z}$,$\underline{IAXIS}$,$\underline{DLXY}$,$\underline{NCHAR}$,$\underline{CHARS}$)

a) Z is the mode set array.

b) IAXIS specifies the axis to label.

If IAXIS = 0, label the x-axis.
= 1, label the y-axis.

c) DLXY fixes the x-interval if IAXIS = 0, or the y-interval if IAXIS = 1, in user co-ordinates (subject space) between each label. If the x-axis decreases from left to right, or the y-axis from bottom to top, then DLXY must be negative. For labeling log grids, DLXY specifies the cycle lines to label. For example, DLXY = 1.0 stipulates the labeling of each cycle line and DLXY = 3.0 every third line.

d) NCHAR specifies the number of characters to display for each cycle of the label.

e) CHAR contains the characters used in the label. CHAR must contain enough characters for all labels. For example, if NCHAR = 3

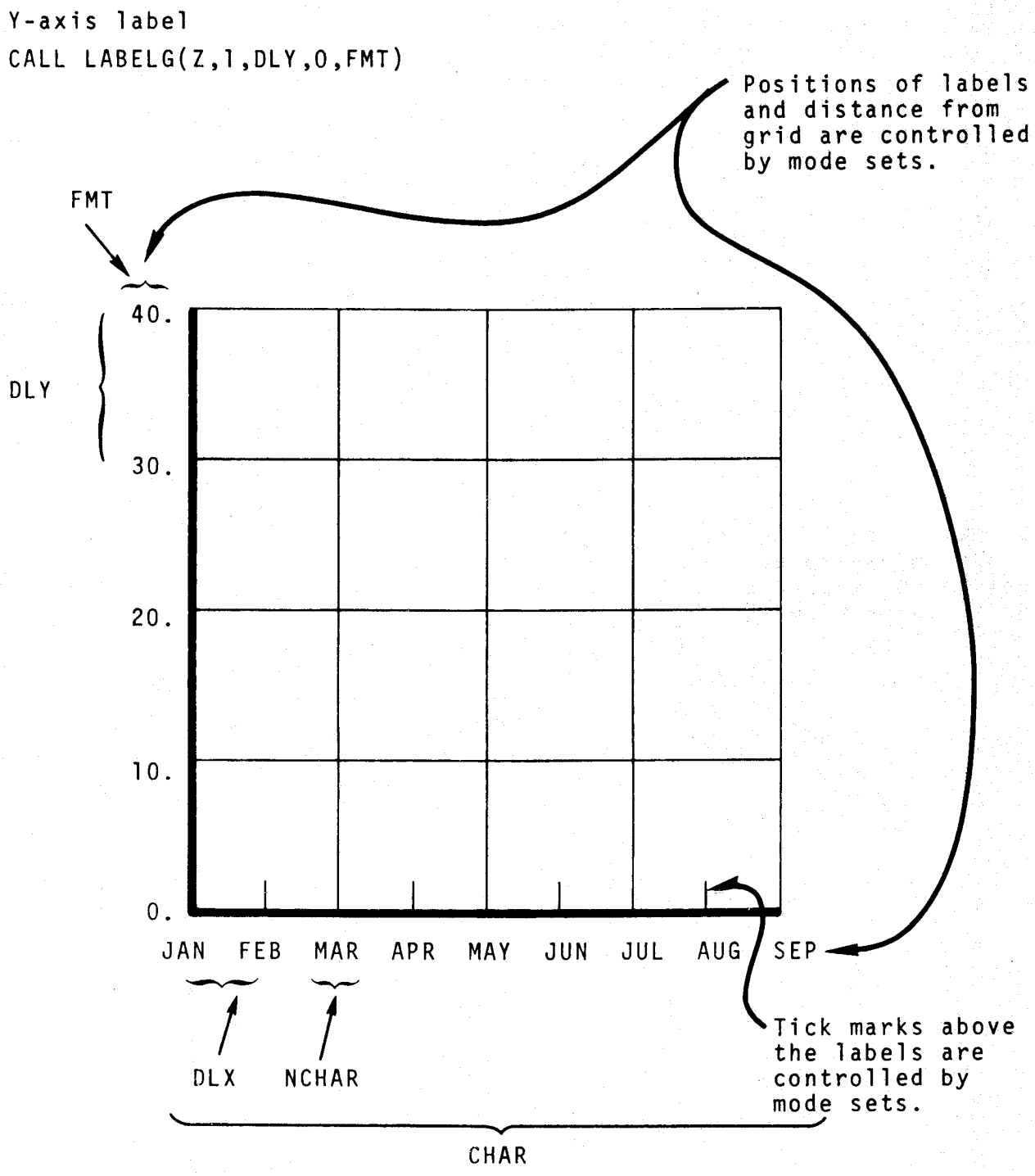


Fig. 10--LABELG Subroutine

and four lines are to be labeled, CHAR must contain twelve characters.

2) To create integer numeric labels in I format,

CALL LABELG(Z,IAXIS,DLXY,0,IFMT)

a) Z is the mode set array.

b) IAXIS specifies the axis to label.

If IAXIS = 0, label the x-axis.
= 1, label the y-axis.

c) DLXY fixes the x-interval if IAXIS = 0, or the y-interval if IAXIS = 1, in user co-ordinates (subject space) between each label. If the x-axis decreases from left to right, or the y-axis from bottom to top, then DLXY must be negative. For labeling log grids, DLXY specifies the cycle lines to label. For example, DLXY = 1.0 stipulates the labeling of each cycle line and DLXY = 3.0 every third line.

d) IFMT is an integer fixing the number of digits to display.

3) To create real numeric labels in F format,

CALL LABELG(Z,IAXIS,DLXY,0,FMT)

a) Z is the mode set array.

b) IAXIS specifies the axis to label.

If IAXIS = 0, label the x-axis.
= 1, label the y-axis.

c) DLXY fixes the x-interval if IAXIS = 0, or the y-interval if IAXIS = 1, in user co-ordinates (subject space) between each label. If the x-axis decreases from left to right,

or the y-axis from bottom to top, then DLXY must be negative. For labeling log grids, DLXY specifies the cycle lines to label. For example, DLXY = 1.0 stipulates the labeling of each cycle line and DLXY = 3.0 every third line.

d) FMT is a real number that specifies the format. It is of the form w.d, where w specifies the field width and d the number of decimal places (d < 10).

4) To create real numeric labels in E format,

CALL LABELG(Z,IAXIS,DLXY,0,-FMT)

a) Z is the mode set array.

b) IAXIS specifies the axis to label.

If IAXIS = 0, label the x-axis.
= 1, label the y-axis.

c) DLXY fixes the x-interval if IAXIS = 0, or the y-interval if IAXIS = 1, in user co-ordinates (subject space) between each label. If the x-axis decreases from left to right, or the y-axis from bottom to top, then DLXY must be negative. For labeling log grids, DLXY specifies the cycle lines to label. For example, DLXY = 1.0 stipulates the labeling of each cycle line and DLXY = 3.0 every third line.

d) -FMT is a real number that specifies the format. It is of the form w.d, where w is the field width and d the number of decimal places. Four places must be provided for the exponent and two for the sign and digit to the left of the decimal place. In general, w = d + 7.

Modifications

Use mode sets to modify LABELG as follows:

1) To specify that a tick mark be drawn above the x-axis labels or beside the y-axis labels,

CALL SETSMG(Z,102,XSIZE)

XSIZE specifies the length of the tick mark for the x-axis label in normalized object space. If XSIZE = 0.0, no tick mark is drawn. XSIZE = 0.0 is the default value.

CALL SETSMG(Z,103,YSIZE)

YSIZE fixes the length of the tick mark for the y-axis label in normalized object space. If YSIZE = 0.0, no tick mark is drawn. YSIZE = 0.0 is the default value.

All modes described in LINESG are in effect when the tick marks are drawn. If XSIZE or YSIZE are positive, the tick marks are drawn inside the grid; if minus, outside. To draw tick marks alone without a label, set NCHAR = 1, and define CHARS to contain as many Hollerith blanks as there are labels.

2) To specify the locations at which to label,

CALL SETSMG(Z,104,XPLACE)

XPLACE defines the distance from the x-axis to the label as a multiple of character height. XPLACE = -1.0 is the default value.

If XPLACE = 0.0, label at y = 0 major axis. (This may cause the label to fall inside the grid.)

> 0.0 (+), label the x-axis above the grid.
< 0.0 (-), label the x-axis below the grid.

CALL SETSMG(Z,105,YPLACE)

YPLACE defines the distance from the y-axis to the label as a multiple of character width. YPLACE = -1.0 is the default value.

If YPLACE = 0.0, label at x = 0 major axis. (This may cause the label to fall inside the grid.)
> 0.0 (+), label the y-axis on the right of the grid.
< 0.0 (-), label the y-axis on the left of the grid.

3) To change the character size or font, give the mode set calls described in LEGNDG.

Example

To label a grid as shown in Fig. 10,

```
CALL LABELG(Z,0,1.,3,27HJANFEBMARAPRMAYJUNJULAUGSEP)
CALL LABELG(Z,1,10.,0,3.0)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV.

TITLEG: Titles a Grid

PURPOSE

Titles the x and y axes and the graph, and automatically centers the titles. TITLEG provides a quick, easy method of titling a graph. For full control over titling, use LEGNDG rather than TITLEG. Figure 11 illustrates the titles.

USAGE

Standard Calls

CALL TITLEG(Z,NX,XCHAR,NY,YCHAR,NT,TCHAR)

1) Z is the mode set array.
2) NX specifies the number of characters in the x-axis title, centered below the grid. If NX = 0, the x-axis is not titled.
3) XCHAR contains the NX characters for the x-axis title. If NX = 0, XCHAR should be a dummy argument.
4) NY specifies the number of characters in the y-axis title, vertically centered left of the grid. The line and character orientation will be 90^{o}. If NY = 0, the y-axis is not titled.
5) YCHAR contains the NY characters for the y-axis title. If NY = 0, YCHAR should be a dummy argument.
6) NT specifies the number of characters in the graph title, centered below the x-axis title. If NT = 0, the graph is not titled.
7) TCHAR contains the NT characters for the graph title. If NT = 0, TCHAR should be a dummy argument.

CALL TITLEG(Z,NX,XCHAR,NY,YCHAR,NT,TCHAR)

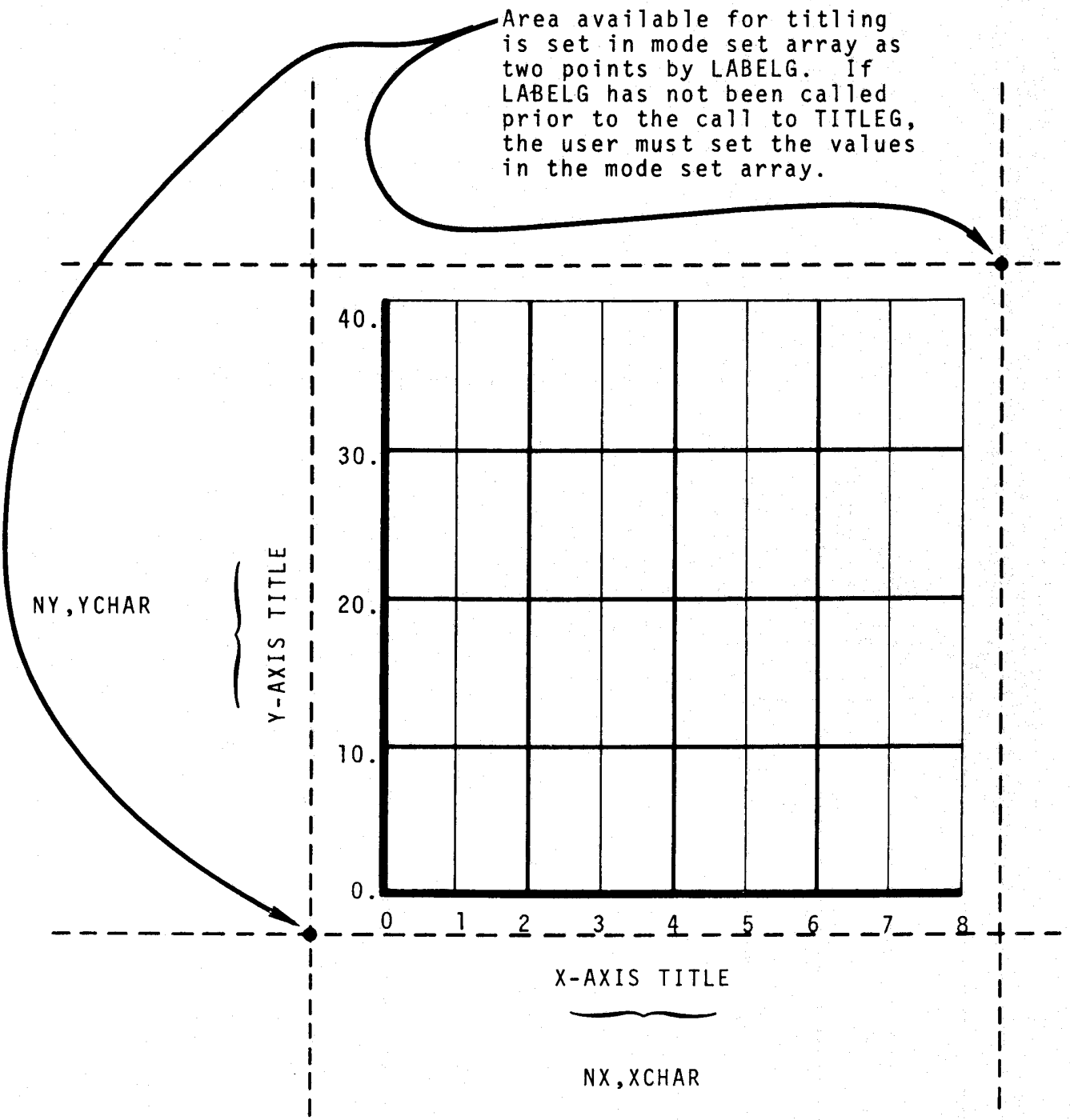


Fig. 11--TITLEG Subroutine

Modifications

Use mode sets to modify TITLEG as follows:

1) The mode set array contains the area available for titling. Normally, these values are stored in the mode set array by GRIDG or LABELG. However, if neither GRIDG nor LABELG has been called, the user must supply the values himself with one of the following mode set calls:

CALL SETSMG(Z,113,XMIN)

XMIN specifies the minimum x-location in normalized object space occupied by the graph, including labels.

CALL SETSMG(Z,114,YMIN)

YMIN designates the minimum y-location in normalized object space occupied by the graph, including labels.

CALL SETSMG(Z,115,XMAX)

XMAX specifies the maximum x-location in normalized object space occupied by the graph, including labels.

CALL SETSMG(Z,116,YMAX)

YMAX stipulates the maximum y-location in normalized object space occupied by the graph, including labels.

2) To specify the position of the title,

CALL SETSMG(Z,104,XPLACE)

XPLACE designates the distance from the grid to the x-axis title as a multiple of character height. It also fixes the distance from the x-axis title to the graph title. XPLACE = -1.0 is the default value.

If XPLACE > 0.0 (+), title above the grid.
< 0.0 (-), title below the grid.

CALL SETSMG(Z,105,YPLACE)

YPLACE specifies the distance from the y-axis to the title as a multiple of character width. YPLACE = -1.0 is the default value.

If YPLACE > 0.0 (+), title on the right side of the grid.
< 0.0 (-), title on the left side of the grid.

3) Change the character size or font with the mode set calls described in LEGNDG.

Example

To title a grid as shown in Fig. 11,

```
CALL TITLEG(Z,12,12HX-AXIS TITLE,12,12HY-AXIS TITLE,
11,11HGRAPH TITLE)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV.

SETUPG: Computes Appropriate Arguments for the Grid Routines

PURPOSE

Computes values to be subsequently included in calls to GRIDG and LABELG. SETUPG allows the user to adjust his subject and object space to produce a more aesthetic graph. Altering the object space reduces the size of the display surface, creating a margin around the grid. Such alteration also causes the scaling factors to be recomputed so that all subsequent calls to graphic output subroutines will result in the data being scaled into the area of the grid. If, after calling SETUPG, the user wants to draw outside the grid, he must either call OBJCTG to reset the display surface, or use the special mode set indicating absolute raster units.

In many instances, particularly where the range of the data is unknown, it is difficult to determine in advance the arguments for the grid routines. SETUPG analyzes the range of the user's data and computes appropriate arguments for the grid routines. SUBJEG must be called before SETUPG to establish the subject space so that SETUPG will know the limits of the data. However, the subject space must not be negative; the x-axis must increase from left to right, and the y-axis from bottom to top. SETUPG computes values for both linear and nonlinear grids.

USAGE

Standard Calls

CALL SETUPG(Z,MODE,DX,DY,IXTH,JYTH,DLX,DLY,XFMT,YFMT)

1) Z is the mode set array.
2) MODE indicates whether the user's subject space and the grid margins are to be adjusted.

If MODE = 0, no adjustment is necessary.
= 1, adjust both the subject and object space.
= 2, adjust only the subject space.
= 3, adjust only the object space.

The user must call SUBJEG, giving the minimum and maximum limits of his data, before calling SETUPG.

3) DX,DY contain the x,y grid intervals used in subsequent calls to GRIDG.

4) IXTH,JYTH contain numbers, used in calls to GRIDG, specifying the x,y grid lines to emphasize.

5) DLX,DLY contain the x and y intervals, used in calls to LABELG, between the axes labels.

6) XFMT,YFMT contain the numeric formats for labels, also used in calls to LABELG.

Modifications

Use mode sets to modify SETUPG as follows:

1) To force SETUPG to compute values that will result in a square grid,

CALL SETSMG($\underline{Z}$,$\underline{110}$,$\underline{1.}$)

To reset the mode to its default value so that a square grid is not mandatory,

CALL SETSMG($\underline{Z}$,$\underline{110}$,$\underline{0.}$)

2) To adjust the density of the grid lines,

CALL SETSMG($\underline{Z}$,$\underline{111}$,$\underline{XDEN}$)

XDEN specifies the density, expressed as a minimum distance between grid lines in normalized object space, of the x-axis (vertical) grid interval lines. XDEN = .01009 is the default value (about one character width).

CALL SETSMG(Z,112,YDEN)

YDEN specifies the density, expressed as a minimum distance between grid lines in normalized object space, of the y-axis (horizontal) grid interval lines. YDEN = .016607 is the default value (about one character height).

3) To reset the grid margins,

CALL SETSMG(Z,106,XLEFT)

XLEFT is the size of the left margin, in normalized object space. XLEFT = .15 is the default value.

CALL SETSMG(Z,107,RIGHT)

RIGHT is the size of the right margin, in normalized object space. RIGHT = .15 is the default value.

CALL SETSMG(Z,108,BOTTOM)

BOTTOM is the size of the bottom margin, in normalized object space. BOTTOM = .15 is the default value.

CALL SETSMG(Z,109,TOP)

TOP is the size of the top margin, in normalized object space. TOP = .15 is the default value.

Example

To compute, draw, and label a grid for a subject space computed to range from 0.16 to 92.34 in the x-axis and 0.01001 to 0.05628 in the y-axis,

```
CALL SUBJEG(Z,.16,.01001,92.34,.05628)
CALL SETUPG(Z,1,DX,DY,IXTH,JYTH,DLX,DLY,XFMT,YFMT)
```

```
CALL GRIDG(Z,DX,DY,IXTH,JYTH)
CALL LABELG(Z,0,DLX,0,XFMT)
CALL LABELG(Z,1,DLY,0,YFMT)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV.

GRAPHG: Constructs a Complete Graph

PURPOSE

Constructs an entire graph, complete with grid, labels, titles, and plotted data. It requires only minimum information from the user and makes all decisions itself. GRAPHG normally resets both the subject and object space. This allows subsequent calls to POINTG or LINESG for plotting more than one graph variable. The subject space will not be reset if NO, the number of points to plot, is zero. The object space will not be reset if the grid margins are set to zero in the mode set array.

USAGE

Standard Calls

CALL GRAPHG($\underline{Z}$,NO,$\underline{X}$,$\underline{Y}$,NX,XCHAR,NY,YCHAR,NT,TCHAR)

1) Z is the mode set array.
2) NO specifies the number of x,y coordinates to plot. If NO = 0, the graph is drawn but no points are plotted. This also prevents the subject space from being reset. NO = 1 is not legal.
3) X,Y are arrays of NO x,y coordinates to plot. If NO = 0, X and Y should be dummy arguments.
4) NX specifies the number of characters in the x-axis title, centered below the grid. If NX = 0, the x-axis is not titled.
5) XCHAR contains the NX characters to use for the x-axis title. If NX = 0, XCHAR should be a dummy argument.

6) NY specifies the number of characters in the y-axis title, oriented at 90^{o} and vertically centered to the left of the grid. If NY = 0, the y-axis is not titled.

7) YCHAR contains the NY characters used in the y-axis title. If NY = 0, YCHAR should be a dummy argument.

8) NT specifies the number of characters in the graph title, centered below the x-axis title. If NT = 0, the graph is not titled.

9) TCHAR contains the NT characters used in the graph title. If NT = 0, TCHAR should be a dummy argument.

Modifications

All the sets described in SETUPG, GRAPHG, LABELG, TITLEG, and POINTG are also applicable to this subroutine.

Example

To draw a complete graph, plotting 100 points contained in the arrays XVAL,YVAL,

```
CALL GRAPHG(Z,100,XVAL,YVAL,1,1HX,1,1HY,15,15HTHIS
IS A GRAPH)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV. GRAPHG resets the grid margins in the mode set array to zero because GRAPHG sets the object space based upon the current object space and the amount of grid margin specified in the mode set array. If GRAPHG did not set the grid margins to zero, subsequent calls to it would continue to decrease the size of the object space, resulting in smaller and smaller graphs.

CONVERSION SUBROUTINES

One of the two conversion routines converts character strings to numeric data (CONVTG); the other converts numeric data to a character string (FMTSG).

CONVTG: Converts Characters to Numeric Data

PURPOSE

Converts a character string to a real or integer number.

USAGE

Standard Calls

CALL CONVTG(Z,IFMT,IW,IN,OUT)

1) Z is the mode set array.
2) IFMT specifies the conversion to be done.

   If IFMT = 1, convert to integer.
   = 2, convert to real.
   = 3, convert to real if a decimal point or E is encountered; otherwise, convert to integer.

3) IW specifies the number of characters in the input string.
4) IN is the address of the character string to convert.
5) OUT will contain the converted integer or real number.

Example

To convert CHAR, a variable containing two left-justified characters, to integer and store the result in I,

```
CALL CONVTG(Z,1,2,CHAR,I)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV.

FMTSG: Converts Numeric Data to Characters

PURPOSE

Converts a number, either real or integer, into a character string.

USAGE

Standard Calls

CALL FMTSG(Z,IFMT,IW,ID,VALUE,CHARS)

1) Z is the mode set array.

2) IFMT specifies the type of number to convert.

   If IFMT = 1, convert an integer number to I format.
   = 2, convert a real number to F format.
   = 3, convert a real number to E format.

3) IW stipulates the field width of the converted number.

4) ID designates the number of decimal places (ID = 0 if IFMT = 1).

5) VALUE is the number to convert. It may be either real or integer, depending upon the value of IFMT.

6) CHARS contains the character string upon return. CHARS must be an array large enough to contain IW characters.

Modifications

1) To convert to scientific notation rather than E format,

   CALL SETSMG(Z,85,1.)

2) To reset to the default E format,

CALL SETSMG(Z,85,0.)

The number 263,000 would appear as 0.263E+06 in E format and as $0.263 \times 10^{+06}$ in scientific notation.

Example

To convert VAR, a floating-point variable, to characters in F8.3 format and store them in CHARS,

```
CALL FMTSG(Z,2,8,3,VAR,CHARS)
DIMENSION CHARS(2)
```

MISCELLANEOUS INFORMATION

Written in FORTRAN IV. If the number can not be converted, CHARS will contain asterisks (*) on return.

Appendix A

IGS MODE SET ARRAY

This section describes the contents of the mode set array. SETSMG does most mode setting, but RSETMG, SUBJEG, and OBJCTG can also do it.

NO	FUNCTION	DESCRIPTION	DEFAULT VALUE	SET BY
1	INITIALIZATION	TELLS WHETHER ARRAY IS INITIALIZED.	-	SYSTEM
2	MIN-X	SUBJECT SPACE.	0.	SUBJEG
3	MIN-Y	SUBJECT SPACE.	0.	SUBJEG
4	MAX-X	SUBJECT SPACE.	4095.	SUBJEG
5	MAX-Y	SUBJECT SPACE.	3071.	SUBJEG
6	MIN-X	NORMALIZED OBJECT SPACE.	0.	OBJCTG
7	MIN-Y	NORMALIZED OBJECT SPACE.	0.	OBJCTG
8	MAX-X	NORMALIZED OBJECT SPACE.	1.333	OBJCTG
9	MAX-Y	NORMALIZED OBJECT SPACE.	1.	OBJCTG
10	X-OFFSET	SCALING FACTORS COMPUTED BY SYSTEM.	0.	SYSTEM
11	Y-OFFSET	SCALING FACTORS COMPUTED BY SYSTEM.	0.	SYSTEM
12	X-SCALE	SCALING FACTORS COMPUTED BY SYSTEM.	1.	SYSTEM
13	Y-SCALE	SCALING FACTORS COMPUTED BY SYSTEM.	1.	SYSTEM
14	SCALE FLAG	TYPE OF SCALING TO DO. 0.-REAL SUBJECT SPACE UNITS. 1.-REAL ABSOLUTE RASTER UNITS. 2.-INTEGER ABSOLUTE RASTERS. 3.-REAL NORMALIZED OBJECT SPACE.	0.	SETSMG
15	ERROR COUNT	NUMBER OF ERRORS FOUND BY IGS.	0.	SYSTEM
16	ERROR MESSAGES	TREATMENT OF ERROR MESSAGES. < 0., DO NOT PRINT MESSAGES. = 0., PRINT ALL MESSAGES. > 0., STOP PRINTING ERROR MESSAGES WHEN ERROR COUNT REACHES THIS NUMBER.	0.	SETSMG
17	LAST X	CURRENT POINT POSITION IN SUBJECT SPACE. (NOTE-THIS VALUE WILL CONTAIN THE X LOCATION OF THE FIRST CHARACTER WHEN CHARACTRON CHARACTERS ARE DISPLAYED. IT WILL CONTAIN THE LOCATION OF THE 'NEXT' CHARACTER WHEN VECTOR CHARACTERS ARE DISPLAYED.)	-	SYSTEM
18	LAST Y	CURRENT POINT POSITION IN SUBJECT SPACE. (NOTE-THIS VALUE WILL CONTAIN THE Y LOCATION OF THE FIRST CHARACTER WHEN CHARACTRON CHARACTERS ARE DISPLAYED. IT WILL CONTAIN THE LOCATION OF THE 'NEXT' CHARACTER WHEN VECTOR CHARACTERS ARE DISPLAYED.)	-	SYSTEM
19	X-NORM. FACTOR	SCOPE SIZE IN RASTERS/MAX NORM.	1.333	SETSMG
20	Y-NORM. FACTOR	SCOPE SIZE IN RASTERS/MAX NORM.	1.0	SETSMG
21	MAX-X	SCOPE SIZE-IN ABSOLUTE RASTER UNITS	4095.	SETSMG
22	MAX-Y	SCOPE SIZE-IN ABSOLUTE RASTER UNITS	3071.	SYSTEM
23	X-LINEAR FLAG	FLAG FOR LINEAR SCALING IN X-AXIS. 0.-LINEAR 1.-NONLINEAR	0.	SETSMG
24	Y-LINEAR FLAG	FLAG FOR LINEAR SCALING IN Y-AXIS.	0.	SETSMG

	0.-LINEAR 1.-NONLINEAR		
25 LOG(MIN-X)	LOG OF SUBJECT SPACE.	-	SYSTEM
26 LOG(MIN-Y)	LOG OF SUBJECT SPACE.	-	SYSTEM
27 LOG(MAX-X)	LOG OF SUBJECT SPACE.	-	SYSTEM
28 LOG(MAX-Y)	LOG OF SUBJECT SPACE.	-	SYSTEM
29 VACANT			
30 LINE WIDTH	NO. TIMES NORMAL WIDTH. .5 2 RASTERS 1.0 4 RASTERS 2.0 8 RASTERS 4.0 16 RASTERS	1.	SETSMG
31 LINE TEXTURE	SIZE OF DASHED LINE. 0.-SOLID 1.-32 RASTER UNITS. 2.-64 RASTER UNITS. 3.-128 RASTER UNITS. 4.-256 RASTER UNITS.	0.	SETSMG
32 VACANT			
. VACANT			
39 VACANT			
40 NORMAL WIDTH	NORMAL CHARACTER WIDTH IN NORMALIZED OBJECT SPACE.	.01009	SYSTEM
41 NORMAL HEIGHT	NORMAL CHARACTER HEIGHT IN NORMALIZED OBJECT SPACE.	.0166	SYSTEM
42 CURRENT WIDTH	CHARACTER WIDTH IN SUBJECT SPACE.	31.	SETSMG
43 CURRENT HEIGHT	CHARACTER HEIGHT IN SUBJECT SPACE.	51.	SETSMG
44 PAGE EJECTION	EJECT TO A NEW PAGE IF LINES RUN OFF THE BOTTOM OR TOP MARGINS. 0.-NO 1.-YES	0.	SETSMG
45 CHARACTER SIZE	NO. TIMES NORMAL SIZE FOR TYPING. .75 23X38 RASTERS 1.0 31X51 RASTERS 1.25 39X64 RASTERS 1.5 47X77 RASTERS	1.	SETSMG
46 TYPING ORIEN.	TYPING ORIENTATION IN DEGREES. BOTH CHARACTER AND LINE ORIENTATION ARE SET. 0. IS A VERTICAL CHARACTER AND A HORIZONTAL LINE TO THE RIGHT. COUNTER CLOCKWISE IS POSITIVE. 0. AND 90. ARE ALLOWED FOR CHARACTRON CHARACTERS.	0.	SETSMG
47 CHARACTER SPACING	NO. TIMES TO SPACE CHARACTERS.	1.0	SETSMG
48 VACANT			
49 LINE SPACING	NO. TIMES CURRENT CHARACTER HEIGHT TO SPACE LINE UPON EJECTION. + VALUE, SPACE LINE UP. - VALUE, SPACE LINE DOWN.	-1.	SETSMG
50 LINE ORIENTATION	ANGLE IN DEGREES AT WHICH A LINE OF CHARACTERS IS TO BE DISPLAYED. 0. IS HORIZONTAL, TO THE RIGHT. COUNTER CLOCKWISE ANGLES ARE POSITIVE.	0.	SETSMG
51 CHARACTER FONT	CHARACTER FONT TO USE FOR TYPING. 0.-CHARACTRON 1.-STANDARD VECTOR	0.	SETSMG
52 CHARACTER CASE	CHARACTER CASE FOR TYPING. 0.-UPPER CASE	0.	SETSMG

	1.-LOWER CASE 2.-SPECIAL CASE		
53 PLOT CHAR SIZE	NO. TIMES NORMAL SIZE FOR PLOTTING. .75 23X38 RASTER UNITS. 1.0 31X51 RASTER UNITS. 1.25 39X64 RASTER UNITS. 1.5 47X77 RASTER UNITS.	1.	SETSMG
54 PLOT CHAR ORIEN.	PLOT CHARACTER CRIENTATION IN DEGREES. 0. IS UPRIGHT. COUNTER CLOCKWISE IS POSITIVE. 0. - UPRIGHT 90. - HORIZONTAL	0.	SETSMG
55 PLOT CHAR. CASE	CHARACTER CASE FOR PLOTTING. 0.-UPPER CASE 1.-LOWER CASE 2.-SPECIAL CASE	2.	SETSMG
56 VACANT			
. VACANT			
59 VACANT			
60 LEFT MARGIN	MARGIN IN SUBJECT SPACE.	0.	SETSMG
61 RIGHT MARGIN	MARGIN IN SUBJECT SPACE.	4095.	SETSMG
62 BOTTOM MARGIN	MARGIN IN SUBJECT SPACE.	0.	SETSMG
63 TOP MARGIN	MARGIN IN SUBJECT SPACE.	3071.	SETSMG
64 CHARACTERS/LINE	NO. CHARACTERS PER LINE	132.	SYSTEM
65 LINES/PAGE	NO. LINES PER PAGE.	60.	SYSTEM
66 NO. TABS	NO. TABS IN EFFECT	0.	TABSG
67 TAB 1	TAB SET IN X-COORD. SUBJECT SPACE.	0.	TABSG
68 TAB 2	TAB SET IN X-COORD. SUBJECT SPACE.	0.	TABSG
. . .	TAB SET IN X-COORD. SUBJECT SPACE.	.	TABSG
. . .	TAB SET IN X-COORD. SUBJECT SPACE.	.	TABSG
76 TAB 10	TAB SET IN X-COORD. SUBJECT SPACE.	0.	TABSG
77 FORMAT WIDTH	FORMAT WIDTH-USED BY NUMBRG.	-	SETSMG
78 FORMAT DECIMALS	FORMAT DECIMAL-USED BY NUMBRG.	-	SETSMG
79 FORMAT	FORMAT-USED BY NUMBRG 1. - I FORMAT 2. - F FORMAT 3. - E FORMAT 4. - A FORMAT	-	SETSMG
80 PL/I FLAG	PL/I CALLS. 0.-NO 1.-YES	0.	SETSMG
81 CHARACTER ADDRESS	N'TH CHARACTER OF A STRING TO BEGIN ADDRESSING WITH.	1.	SETSMG
82 TYPE OF ARC	METHOD TO USE IN DRAWING AN ARC. 0.-LINES 1.-POINTS	0.	SETSMG
83 ROUNDNESS OF ARC	DEGREES OF TANGENT BETWEEN TWO ADJACENT POINTS OF AN ARC.	5.	S ESMG
84 PLOT CHARACTER	CHARACTER TO USE FOR PLOTTING. (LEFT JUSTIFIED)	POINT	SETSMG
85 SCIENTIFIC FMT	SPECIFIES SCIENTIFIC OR E FORMAT. 0.-E FORMAT 1.-SCIENTIFIC NOTATION	0.	SETSMG
86 VACANT			
87 RESERVED			
88 VOID MARK	PUT VOID MARK ON BAD OUTPUT. 0. - YES 1. - NO	0.	SETSMG
89 TAPE UNIT	OUTPUT TAPE NUMBER	10.	MODESG

90 HARDCOPY MODE	TYPE OF HARDCOPY TO PRODUCE. 0.-NONE (OBJECT SPACE IS RESET TO XMIN=0., XMAX=1.33333, YMIN=0., YMAX=1.) 1.-11 X 14 (OBJECT SPACE IS RESET TO XMIN=0., XMAX=1.33333, YMIN=0., YMAX=1.) 2.-8 1/2 X 11 (OBJECT SPACE IS RESET TO XMIN=0., XMAX =.72967, YMIN=0., YMAX=1.) 3.-STRIP CHART (OBJECT SPACE IS RESET TO XMIN=0., XMAX=1., YMIN=0., YMAX=1.)	1.	SETSMG
91 FILM ADVANCE	DISTANCE TO ADVANCE FILM IN PERFS.	6.	SYSTEM
92 FORMS FLASH	AUTOMATIC FORMS FLASH. 0.-OFF 1.-ON	0.	SETSMG
93 FRAME MARKS	AUTOMATIC FRAME MARKS. 0.-NONE 1.-CORNER MARKS 2.-ID 3.-CORNER MARKS AND ID	3.	SETSMG
94 LINE DENSITY	DENSITY OF LINES. 0.-NORMAL 1.-LIGHT	0.	SETSMG
95 FAST PLOT	SPEED TO PLOT CHARACTERS. 0.-NORMAL 1.-FAST	0.	SETSMG
96 FRAME COUNT	NO. FRAMES PRODUCED	0.	SYSTEM
97 VACANT			
98 VACANT			
99 VACANT			
100 AXES FLAG	EMPHASIS OF MAJOR AXES. 0.-EMPHASIZE BOTH 1.-EMPHASIZE X=0 ONLY 2.-EMPHASIZE Y=0 ONLY 3.-NO EMPHASIS	0.	SETSMG
101 VACANT			
102 X TICK MARKS	LENGTH OF X-AXIS TICK MARK IN NORMALIZED OBJECT SPACE.	0.	SETSMG
103 Y TICK MARKS	LENGTH OF Y-AXIS TICK MARK IN NORMALIZED OBJECT SPACE.	0.	SETSMG
104 X-AXIS LABEL	NO. TIMES CHARACTER HEIGHT TO POSITION X-AXIS LABEL. + LABEL ABOVE GRID - LABEL BELOW GRID 0. LABEL AT Y=0	-1.5	SETSMG
105 Y-AXIS LABEL	NO. TIMES CHARACTER WIDTH TO POSITION Y-AXIS LABEL. + LABEL RIGHT SIDE OF GRID - LABEL LEFT SIDE OF GRID 0. LABEL AT X=0	-1.5	SETSMG
106 LEFT MARGIN	RELATIVE GRID MARGIN IN NORMALIZED OBJECT SPACE.	0.15	SETSMG
107 RIGHT MARGIN	RELATIVE GRID MARGIN IN NORMALIZED	0.15	SETSMG

	OBJECT SPACE.		
108 BOTTOM MARGIN	RELATIVE GRID MARGIN IN NORMALIZED OBJECT SPACE.	0.15	SETSMG
109 TOP MARGIN	RELATIVE GRID MARGIN IN NORMALIZED OBJECT SPACE.	0.15	SETSMG
110 SQUARE GRID	FORCE VALUES FOR A SQUARE GRID TO BE CALCULATED. 0.-NOT FORCED TO BE SQUARE. 1.-MUST BE SQUARE.	0.	SETSMG
111 X-AXIS DENSITY	MIN DISTANCE BETWEEN X-AXIS GRID LINES IN NORMALIZED OBJECT SPACE.	.01009	SETSMG
112 Y-AXIS DENSITY	MIN DISTANCE BETWEEN Y-AXIS GRID LINES IN NORMALIZED OBJECT SPACE.	.01660	SETSMG
113 MIN X-TITLE	NORMALIZED OBJECT SPACE UNAVAILABLE FOR TITLES.	-	LABELG
114 MIN Y-TITLE	NORMALIZED OBJECT SPACE UNAVAILABLE FOR TITLES.	-	LABELG
115 MAX X-TITLE	NORMALIZED OBJECT SPACE UNAVAILABLE FOR TITLES.	-	LABELG
116 MAX Y-TITLE	NORMALIZED OBJECT SPACE UNAVAILABLE FOR TITLES.	-	LABELG
117 VACANT			
. VACANT			
129 VACANT			
130 VACANT			
131 SKEW ANGLE	VECTOR CHARACTER SKEW ANGLE IN DEGREES.	0.	SETSMG
132 X-MULT	VECTOR CHARACTER MULTIPLE OF STANDARD CHARACTER WIDTH.	1.	SYSTEM
133 Y-MULT	VECTOR CHARACTER MULTIPLE OF STANDARD CHARACTER HEIGHT.	1.	SYSTEM
134 SKEW FACTOR	TAN(SKEW ANGLE)	0.	SYSTEM
135 ROTATION ANGLE	VECTOR CHARACTER ROTATION ANGLE IN RADIANS.	0.	SYSTEM
136 FONT NUMBER	VECTOR CHARACTER FONT NUMBER.	-	SYSTEM
137 CHARS/CASE	NUMBER OF VECTOR CHARACTERS PER CASE	0.	SYSTEM
138 X-DISP	VECTOR CHARACTER ENVELOPE CENTER	-	SYSTEM
139 Y-DISP	VECTOR CHARACTER ENVELOPE CENTER	-	SYSTEM
140 WIDTH ADJ	VECTOR CHAR. WIDTH ADJUSTMENT	-	SYSTEM
141 HEIGHT ADJ	VECTOR CHAR. HEIGHT ADJUSTMENT	-	SYSTEM
142 VACANT			
143 NO. OF CASES	VECTOR CHARACTER FONT CASES.	-	SYSTEM
144 VACANT			
145 VACANT			
146 PACKING FLAG	VECTOR CHARACTER PACKING FLAG. 0. = 1 WORD/COORD. 1. = 1 BYTE/COORD.	-	SYSTEM
147 CHARACTER TYPE	TYPE OF CHARACTERS IN FONT. 0. - VECTOR 1. - STROKE	-	SYSTEM
148 VACANT			
149 VACANT			
150 POINTER	POINTER TO CURRENT VECTOR CHARACTER FONT.	-	SYSTEM
151 VACANT			
. VACANT			
175 VACANT			

176 INSTALLATION DEPENDENT MODE SETS

. . . . . .

200 INSTALLATION DEPENDENT MODE SETS

Appendix B

IGS ERROR CODES

The following table (p. 90) defines each IGS error message. When an error is detected by a graphic subroutine, ERRZZ is called to print the error message:

BAD BAD BAD, ERROR NO. xx, value I,
value F, value A FRAME nn

where xx is the error number, values I, F, and A are the values printed out under I, F, and A format, and nn is the frame number. The job continues in a normal manner after the message is printed.

TABLE OF ERROR MESSAGES

Number	Subroutine	Value	Description
1	GETSMG	NO	Illegal mode set number in call.
2	LEGNDG	N	Illegal character count in call.
3	LINESG	NO	Illegal number in call.
4	MODESG	ITAPE	Illegal tape number in call.
5	NUMBRG	FMT	Illegal format in call.
6	OBJCTG	-	Max x or y LE min x or y in call.
7	PAGEG	-	Illegal arguments in call.
8	POINTG	N	Illegal number in call.
9	SEGMTG	N	Illegal number in call.
10	TABSG	N	Illegal number in call.
11	MLTPLG	NLINES	Illegal number in call.
12	GRIDG	-	Grid too small to draw.
13	LABELG	-	Illegal format in call.
14	TITLEG	-	Illegal arguments in call.
15	SETUPG	-	Illegal arguments in call.
16	SUBJEG	-	Max x or y EQ min x or y in call.
17	LABELG	-	Illegal arguments in call.
18	LABELG	-	Grid too small to label.
19	LABELG	-	Zero subject space.
20	GRIDG	-	Illegal arguments in call.
21	SETUPG	-	Subject (object) space too small to draw a grid.
22	SETUPG	-	Density LE 0.
23	SETSMG	N	Illegal mode set number in call.
24	SETUPG	-	Grid will not fit on page.
25	TEXTG	N	Illegal character count in call.
26	LABELG	-	Labels will not fit on page.
27	PACKZZ	-	No initialization call to MODESG.
28	GRAPHG	N	Illegal argument in call.
29	SUBJEG	-	Minus value for log grid.
30	SETUPG	-	Too many cycles in log grid.
31	VECTZZ	-	No vector character font initialized.
32	SCALZZ	X	Bad x-coordinate.
33	SCALZZ	Y	Bad y-coordinate.
34	VECTZZ	CHAR	Character not in font.
35	TITLEG	-	Subject (object) space too small to title grid.
36	VECTZZ	CHAR	Request for nonexistent vector character case.

Appendix C

SPECIAL CONTROL CHARACTERS

The following special control characters may be typed within a character string to obtain the functions described. The list is expandable; new functions will be added as the need arises.

Control Character	Function	Description
$N	NULL	The $N is ignored. The NULL aids the user in typing any of the control characters. For example, to type the characters $L, he must separate them with the NULL as follows: $$NL.
$T	TAB	Causes a tab to the next tab position, or to the right margin if no more tabs are set.
$E	EJECT	Same as a carriage return on a typewriter. It ejects the line to the left margin.
$U	UPPER CASE[a]	Sets character case to upper.
$L	LOWER CASE[a]	Sets character case to lower.
$S	SPECIAL CASE[a]	Sets character case to special.
$C	CAPITALIZE	Upper case next character only.
$B	BACKSPACE	Same as backspace on a typewriter.
$P	NEW PAGE	Ejects to a new page.
$+	SUPERSCRIPT	Shift line up 1/2 space.
$-	SUBSCRIPT	Shift line down 1/2 space.

[a]Appendix D, p. 92, illustrates the character cases.

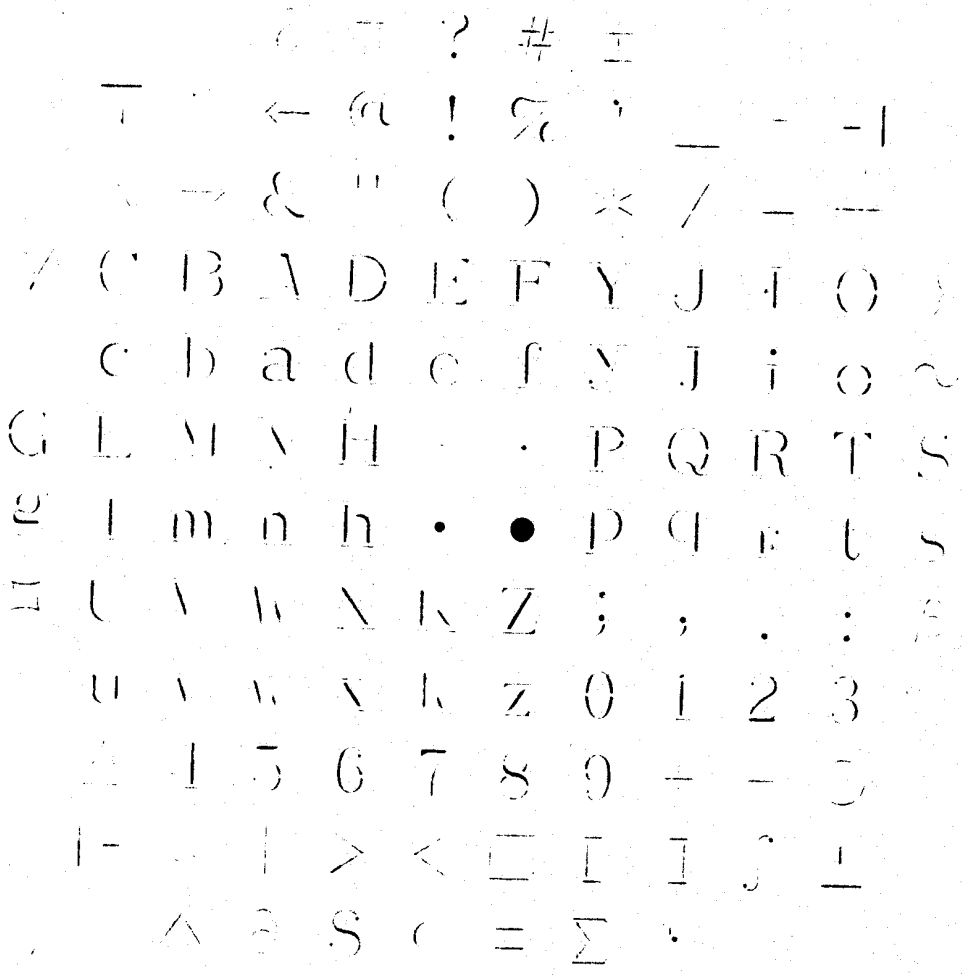
Appendix D

IGS CHARACTER CASES

Upper Case	Lower Case	Special Case	Upper Case	Lower Case	Special Case
A	a	α	Y	y	←
B	b	β	Z	z	→
C	c	^ (carat)	0 (zero)	0 (zero)	○ (circle)
D	d	δ	1	1	. (small point)
E	e		2	2	· (normal point)
F	f	%	3	3	• (medium point)
G	g	γ	4	4	● (large point)
H	h	>	5	5	
I	i	∫	6	6	
J	j	∂	7	7	
K	k	° (degree)	8	8	
L	l	<	9	9	
M	m	#	blank	blank	blank
N	n	¬ (not)	. (period)	. (period)	
O	o	| (and)	, (comma)	, (comma)	
P	p	π	' (close quote)	"	' (open quote)
Q	q	_ (under score)	$	¢	
R	r	□	(	[	{
S	s	∑	)	]	}
T	t	~ (tilde)	/	?	\
U	u	¤ (lozenge)	- (minus)	— (hyphen)	— (bar)
V	v	Δ	+	!	±
W	w		*	;	@
X	x		=	:	&

Appendix E

SC-4060 CHARACTRON CHARACTER SET



Appendix F

SUMMARY OF S-C 4060 GRAPHIC FEATURES

Film and Hardcopy Dimensions	Page Size		
	8 1/2 × 11"	11 × 14"	Strip Chart
Paper size	8 1/2 × 11"	11 × 14"	9.9 × 11"
Image size			
Hardcopy	7.2 × 9.9"	13.2 × 9.9"	9.9 × 9.9"
Film	.545 × .75"	1.00 × .75"	.75 × .75"
Film advance distribution	.75 × 0.0"	1.12 × 0.0"	.75 × 0.0"
Size in raster units	2241 × 3072	4096 × 3072	3072 × 3072
Characters/line, lines/page			
Small (.75 normal)	96 × 80	176 × 80	132 × 80
Normal (1.0 normal)	72 × 60	132 × 60	99 × 60
Medium (1.25 normal)	57 × 48	105 × 48	79 × 48
Large (1.5 normal)	48 × 40	88 × 40	66 × 40
	Raster Units		
CHARACTRON character sizes in raster units			
Small (.75 normal)	23 × 38 raster units = .075 × .113"		
Normal (1.0 normal)	31 × 51 raster units = .10 × .150"		
Medium (1.25 normal)	39 × 64 raster units = .125 × .188"		
Large (1.5 normal)	47 × 77 raster units = .15 × .225"		
Line weights			
Light	2 raster units = .0064" (on hardcopy)		
Normal	4 raster units = .013"		
Heavy	8 raster units = .026"		
Heaviest	16 raster units = .051"		
Plotting point diameters			
Small	2 raster units = .0064" (on hardcopy)		
Normal	4 raster units = .013"		
Medium	8 raster units = .026"		
Large	16 raster units = .051"		
Dashed-line sizes			
Small	32 raster units		
Normal	64 raster units		
Medium	128 raster units		
Large	256 raster units		

Summary of S-C 4060 graphic features

Line widths	Normal line density	Light line density
Light		
Normal		
Medium		
Heavy		

Line dash size

1 .....

2 - - -

Appendix G

SAMPLE PROGRAM

This section contains a simple sample program.

```
C SINCE THIS IS A SAMPLE PROGRAM, THERE WILL BE NO GREAT SIGNIFICANCE
C IN THE DATA THAT IS PLOTTED.
C DIMENSION THE MODE SET ARRAY.
      DIMENSION Z(200)
      DIMENSION X(51),Y(51)
C INITIALIZE IGS WITH A CALL TO MODESG.
      CALL MODESG(Z,0)
C SET FOR 8 1/2 X 11 OUTPUT.
      CALL SETSMG(Z,90,2.)
C COMPUTE SOME POINTS TO PLOT.
      DO 100 I = 1,51
      X(I) = (I-1)*10
  100 Y(I) = (I-1)*10
C THE DEFAULT PLOTTING SYMBOL IS A POINT.  I DON'T WANT A POINT FOR MY
C PLOT SYMBOL SO I WILL CHANGE IT WITH A MODE SET CALL.
C SET FOR NORMAL PLOT CHARACTER CASE.
      CALL SETSMG(Z,55,0.)
C AND THEN CHANGE THE PLOT CHARACTER.
      CALL SETSMG(Z,84,1H+)
C NOW I WILL PLOT +'S INSTEAD OF POINTS.
C I WILL NOW CALL ON GRAPHG TO DRAW, LABEL, AND TITLE THE GRAPH, AND
C PLOT 51 POINTS OF X,Y COORDINATES.
      CALL GRAPHG(Z,  51,X,Y,14,14HTIME--IN HOURS,18,18HDISTANCE--IN MIL
     1ES,16,16HTIME VS DISTANCE)
C COMPUTE SOME MORE POINTS TO PLOT ON THE SAME GRAPH.
      DO 200 I = 1,51
      X(I) = (51-I)*10
  200 Y(I) = (I-1)*10
C I WANT TO LABEL THE TOP AND RIGHT SIDE OF THE GRID.  I MUST MAKE THE
C APPROPRIATE MODE SETS FOR THE LABEL POSITION.
      CALL SETSMG(Z,104,1.)
      CALL SETSMG(Z,105,1.)
C I WILL NOW LABEL THE X-AXIS.
      CALL LABELG(Z,0,200.,0,4)
C AND LIKEWISE FOR THE Y-AXIS.
      CALL LABELG(Z,1,100.,0,4)
C THEN TO ADD SOME TITLES.
      CALL TITLEG(Z,21,21HDISTANCE--IN FURLONGS,19,19HTIME--IN FORTNIGHT
     1S,0,0)
C I SHOULD CHANGE THE PLOT SYMBOL SO I WON'T CONFUSE MY DATA.
      CALL SETSMG(Z,84,1HX)
C AND THEN PLOT THE DATA.
      CALL POINTG(Z,51,X,Y)
C NOW I WILL ADVANCE THE FRAME AND GET ONE COPY.
      CALL PAGEG(Z,0,1,1)
C I AM DONE.  I MUST CALL EXITG TO TERMINATE THE GRAPHIC OUTPUT.
      CALL EXITG(Z)
      CALL EXIT
      END
```

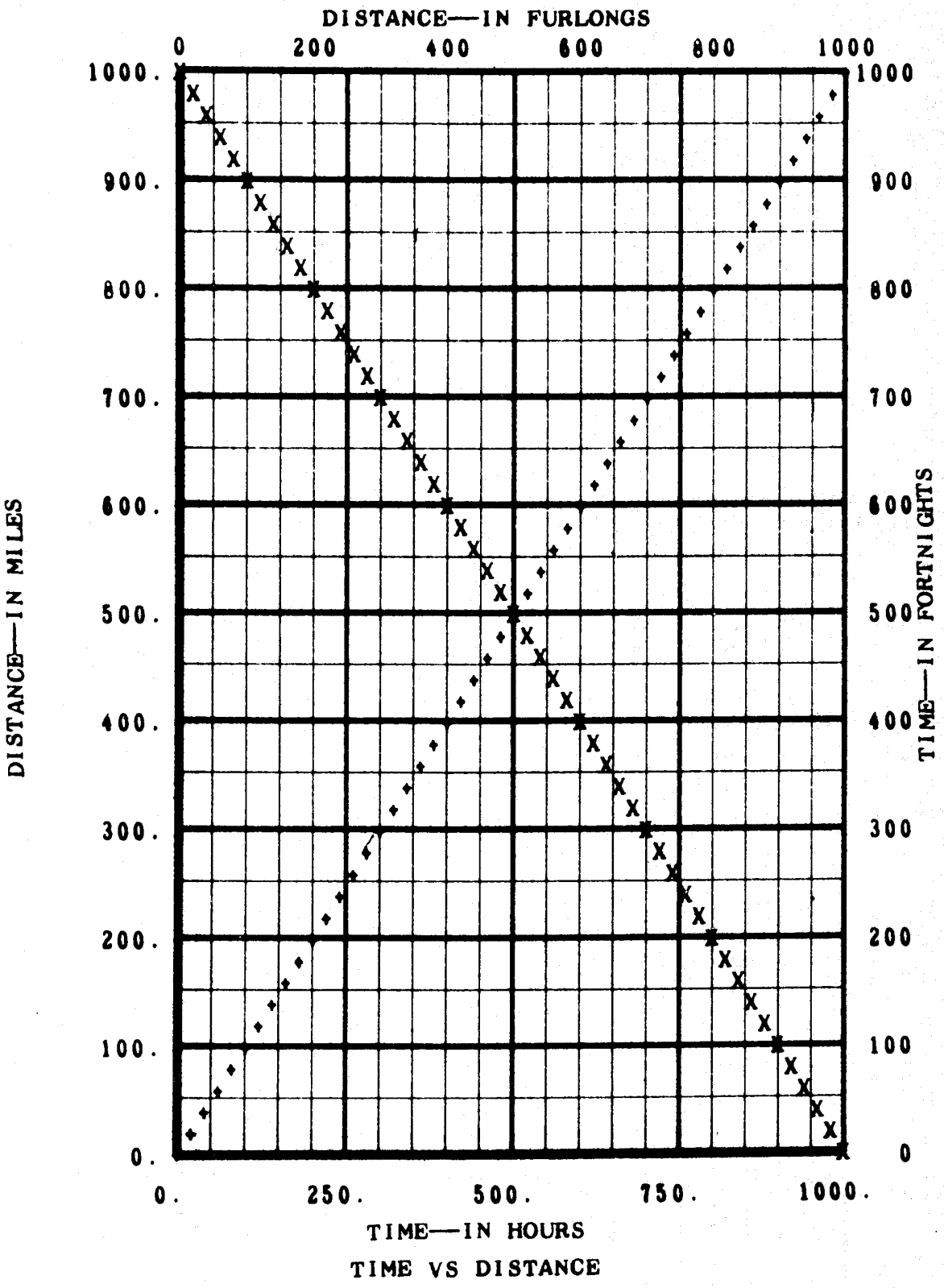


Fig. 12--Output from Sample Program

Appendix H

SAMPLE VECTOR CHARACTER USAGE

Sample Routine 1 for Vector Characters

This routine illustrates the methods for initializing a vector character font, alternating between CHARACTRON and vector characters, and switching from one vector character font to another.

```
      DIMENSION Z(200)
C
C EXTERNAL THE VECTOR CHARACTER FONT NAMES.
      EXTERNAL FONT2,FANCY
C INITIALIZE IGS
      CALL MODESG(Z,0)
C SET FOR NEW PAGE ON OVERFLOW
      CALL SETSMG(Z,44,1.)
C LABEL THE PAGE WITH CHARACTRON CHARACTERS
      CALL LEGNDG(Z,0.,3071.,31,31H S$LWAPPING VECTOR CHARACTERS$U)
C INITIALIZE VECTOR CHARACTER FONT 'FONT2'
      CALL VECIG(Z,FONT2,0)
C CHANGE TO USING VECTOR CHARACTERS
      CALL SETSMG(Z,51,1.)
C TYPE WITH VECTOR CHARACTER FONT 'FONT2'
      CALL LEGNDG(Z,31.,3000.,32,32HV$LECTOR CHARACTERS FROM $UFONT2)
C  NOTE THAT I MUST START HALF A CHARACTER WIDTH IN FROM THE MARGIN
C
C  INSTEAD OF CALCULATING HALF A CHARACTER WIDTH, I CAN LET
C   IGS DO IT FOR ME BY CALLING FOR A LINE EJECT($E)
      CALL LEGNDG(Z,0.,3000.,20,20H$ELINE EJECTED FONT2)
C CHANGE VECTOR CHARACTER FONT TO 'FANCY'
      CALL VECIG(Z,FANCY,0)
C TYPE SOME VECTOR CHARACTERS FROM 'FANCY' STARTING WHERE WE LEFT OFF
      CALL TEXTG(Z,22,22HFANCY FONT STARTS HERE)
C MORE FANCY FONT
      CALL LEGNDG(Z,0.,2000.,17,17H$EMORE FANCY FONT)
C BACK TO CHARACTRON
      CALL SETSMG(Z,51,0.)
      CALL TEXTG(Z,20,20H$EBACK TO CHARACTRON)
C THRU
      CALL EXITG(Z)
      CALL EXIT
      END
C
```

Sample Routine 2 for Vector Characters

This routine illustrates how to use vector characters.

```
      DIMENSION Z(200)
C EXTERNAL THE VECTOR CHARACTER FONT NAME.
      EXTERNAL FONT2
C INITIALIZE IGS
      CALL MODESG(Z,0)
C SET FOR NEW PAGE ON OVERFLOW
      CALL SETSMG(Z,44,1.)
C LABEL THE PAGE WITH CHARACTRON CHARACTERS
      CALL LEGNDG(Z,0.,3071.,28,28H U$LSING VECTOR CHARACTERS$U)
C INITIALIZE VECTOR CHARACTER FONT 'FONT2'
      CALL VECIG(Z,FONT2,0)
C CHANGE TO USING VECTOR CHARACTERS
      CALL SETSMG(Z,51,1.)
C CHANGE SIZE TO 2.7 TIMES NORMAL
      CALL SETSMG(Z,45,2.7)
      CALL TEXTG(Z,15,15H$ELARGE FONT2$E)
C RETURN SIZE TO NORMAL
      CALL SETSMG(Z,45,1.)
C MAKE CHARACTERS 2 TIMES NORMAL WIDTH
C  GET CURRENT WIDTH
      CALL GETSMG(Z,42,V)
      V2 = 2.* V
C  SET WIDTH AND TYPE MESSAGE
      CALL SETSMG(Z,42,V2)
      CALL TEXTG(Z,12,12H$ETWO WIDE$E)
C RETURN WIDTH TO NORMAL
      CALL SETSMG(Z,42,V)
C MAKE HEIGHT 3 TIMES NORMAL
C  GET CURRENT HEIGHT
      CALL GETSMG(Z,43,V)
      V3 = 3.* V
C  SET NEW HEIGHT AND TYPE MESSAGE
      CALL SETSMG(Z,43,V3)
      CALL TEXTG(Z,14,14H$ETHREE HIGH$E)
C  RETURN HEIGHT TO NORMAL
      CALL SETSMG(Z,43,V)
C ROTATE THE CHARACTERS 29 DEGREES CLOCKWISE.
      CALL SETSMG(Z,46,-29.)
      CALL TEXTG(Z,20,20H$EROTATED 29 DEGREES)
C  RESET ANGLE
      CALL SETSMG(Z,46,0.)
C SKEW CHARACTERS 30 DEGREES TO THE RIGHT (CLOCKWISE)
      CALL SETSMG(Z,131,-30.)
      CALL TEXTG(Z,23,23H$ESKEW 30 DEGREES RIGHT)
C THRU
      CALL EXITG(Z)
      CALL EXIT
      END
```

Appendix I

SAMPLE VECTOR CHARACTERS

VECTOR CHARACTERS HAVE THE FOLLOWING FEATURES.

Characters can be displayed in four line weights--

HEAVIEST HEAVY NORMAL LIGHT

DASHED LINES CAN BE USED TO DRAW VECTOR CHARACTERS.

VECTOR CHARACTERS HAVE UNLIMITED SIZE

CHARACTER WIDTH AND HEIGHT CAN BE ADJUSTED.

VECTOR CHARACTERS CAN *BE SKEWED.*

VECTOR CHARACTERS CAN BE ORIENTATED AT ANY ANGLE.

Appendix J

TECHNIQUES FOR SAVING CORE STORAGE

It is hoped that the amount of storage used by IGS will not inconvenience most users. However, a straightforward method exists for reclaiming some space from IGS.

Various subroutines are referred to in IGS and loaded into core on all jobs, though some jobs may not cause IGS to call them. This allows IGS to call subroutines automatically, thereby limiting the number of subroutines with which the user must be familiar. It is a simple matter to block the loading of any unnecessary subroutines by overriding it with a dummy subroutine as follows:

1) SUBROUTINE (put name of subroutine here)
2) END

For example, to block VECSZZ, the dummy subroutine would read:

```
SUBROUTINE VECSZZ
RETURN
END
```

These blocking decks will reclaim most of the storage required by each unnecessary subroutine.

The candidates for elimination are:

1) ERRZZ--block it if IGS error messages are not needed.
2) VECSZZ--block it if vector characters are not required.
3) XMODZ, YMODZ--block it if logarithmic grids are not required.

Appendix K

SCORS COMPATIBILITY

Although the design of IGS differs considerably from that of the S-C 4020 SCORS, the two graphic systems are somewhat *compatible*. That is, a few SCORS subroutines have been rewritten to produce S-C 4060 Meta-language input. Although the S-C 4060 can process the S-C 4020 SCORS input directly, an installation might prefer S-C 4060 Meta-language input for two reasons: First, the use of only one type of input facilitates the operation of the S-C 4060. Second, and more important, a large SCORS program need not be rewritten in IGS to access the S-C 4060 features not available on the S-C 4020, because IGS and SCORS calls can be intermixed to modify the output of the SCORS program, or separate parts of the program can be written in IGS, then added to an existing SCORS program. IGS calls within a SCORS program must refer to the SCORS mode set array, AMODES, kept in labeled common.

For the purpose of IGS compatability, SCORS will be defined exactly as detailed in Ref. 3. IGS cannot protect the programmer who capitalized on his knowledge of the coding of SCORS subroutines to accomplish functions not described in Ref. 3. Moreover, the SCORS compatability routines are written in FORTRAN; therefore, calls with implicit arguments, such as CALL FRAMEV, must be changed to specify their arguments explicitly (e.g., CALL FRAMEV(0)).

Appendix L

PL/I COMPATIBILITY

IGS is written primarily to be called from FORTRAN; however, it may be called from PL/I with a few changes in procedure necessary because of differences between the languages.

IGS must be informed with the following mode set call that it will be called by PL/I:

```
A = 1.0;
I = 80;
CALL SETSMG(Z(1),I,A);
```

In FORTRAN, the array name and the first element of the array are equivalent; in PL/I, they differ. Thus, a PL/I call to IGS must specifically refer to the first element of an array rather than the array name. Numeric literals cannot be used in PL/I calls--only variables. Thus a call that would be

```
CALL SETSMG(Z,80,1.)
```

in a FORTRAN program must be

```
A = 1.0;
I = 80;
CALL SETSMG(Z(1),I,A);
```

in a PL/I program. PL/I also differs from FORTRAN in that double-subscripted arrays are stored in row order rather than column order. Double-subscripted PL/I arrays must be transposed for calls to IGS.

The PL/I error routine that handles abnormal terminations will itself abnormally terminate if the original termination does not come from a PL/I subroutine. Since IGS is not written in PL/I, any abnormal terminations within IGS will be lost by the subsequent termination of the

PL/I error routine. The indicative dump will show only the PL/I error routine termination with no indication that the termination originated with IGS. It is hoped that there will be few abnormal terminations within IGS.

REFERENCES

1) Brown, G. D., C. H. Bush, and R. A. Berman, *The Integrated Graphics System for the S-C 4060: II. System Programmer's Guide*, The RAND Corporation, RM-5661-PR, January 1969.

2) *S-C 4060 Stored Program Recording System: Description and Specifications*, The Stromberg-Carlson Corporation, San Diego, California, Document 9500209, October 1966.

3) *Programmer's Reference Manual: S-C 4020 Computer Recorder*, The Stromberg-Carlson Corporation, San Diego, California, Document 9500056, October 1964 (Rev. August 1965).

4) *American Standard FORTRAN*, American Standards Association, New York, March 7, 1966.

5) *Graphic Specifications for Standard Graphic Subroutines*, SHARE Standard Graphic Output Language Committee, August 8, 1966.

6) *S-C 4020 Computer Recorder: Product Specification*, The Stromberg-Carlson Corporation, San Diego, California, Document 281001-241, May 1964.

7) Brown, Gary D., and Charles H. Bush, *The Integrated Graphics System for the IBM 2250*, The RAND Corporation, RM-5531-ARPA, October 1968.

ALPHABETIC LIST OF SUBROUTINES

CIRARG, pp. 35-36.
CONVTG, p. 79.
EXITG, p. 33.
FMTSG, pp. 80-81.
GETSMG, pp. 27-28.
GRAPHG, pp. 76-78.
GRIDG, pp. 58-61.
LABELG, pp. 62-67.
LEGNDG, pp. 37-41.
LINESG, pp. 42-44.
MLTPLG, pp. 45-46.
MODESG, p. 21.
NUMBRG, pp. 47-50.
OBJCTG, pp. 24-25.
PAGEG, pp. 30-32.
POINTG, pp. 51-53.
RSETMG, p. 28.
SEGMTG, pp. 54-55.
SETSMG, p. 26.
SETUPG, pp. 72-75.
SUBJEG, pp. 22-23.
TABSG, p. 29.
TEXTG, p. 56.
TITLEG, pp. 68-71.
VECIG, pp. 31-32.
XMODZ, pp. 24-25.
YMODZ, pp. 24-25.