CalComp Software Reference Manual

CALCOMP SOFTWARE REFERENCE MANUAL

October 1976

CALCOMP
California Computer Products, Inc.
2411 W. La Palma, Anaheim, CA 92801

Printed in the United States of America

INDEX

Introduction 1

Graphic Systems 2

Basic Software 3

Operating Considerations 4

Basic Software Design 5

Functional Software 6

Applications Software 7

PREFACE

This manual is intended to augment CalComp Programming Manuals for users who need more detailed knowledge of CalComp Graphic Systems. Table P-1 lists current manuals for CalComp software products. Section 1 summarizes the hardware/software of CalComp Graphic Systems. Sections 2 through 5 describe CalComp Graphic Products, the use of CalComp Basic Software, the operation of CalComp Graphic Systems, and the design of CalComp Basic Software.

Summaries of CalComp Functional and Applications Software are contained in Sections 6 and 7.

CONTENTS

Section		Page
1	INTRODUCTION TO CALCOMP	1.01
	General	1.01
	Graphic Products	1.01
	Graphic Systems	1.01
	CalComp Hardware	1.01
	System Configurations	1.02
	CalComp Plotters	1.02
	Drum Plotters	1.02
	Flatbed Plotters	1.04
	Electronic Plotter	1.04
	CalComp Drivers	1.04
	Offline Systems	1.05
	Online Systems	1.05
	CalComp Accessories	1.06
	Additional Peripherals	1.06
	CalComp Software	1.06
	Types of Software	1.06
	Basic Software	1.08
	Functional Software	1.08
	Applications Software	1.09
	CalComp Software Support Policy	1.09
	Compatibility Features	1.09
2	CALCOMP GRAPHIC SYSTEMS	2.01
	General	2.01
	System Specifications	2.01
	Accuracy and Repeatability	2.02
	Plotting Flexibility	2.02
	Plotting Speed	2.02
	Computer Time	2.02
	Types of Plotters	2.03
	Drum Plotters	2.03
	Repeatability	2.03
	Ballpoint/Liquid Ink Pens	2.03
	Accessories	2.04
	Flatbed Plotters	2.07
	Flatbed Plotter Drafting Tools	2.07
	Flatbed Accessories	2.07
	Electronic CRT Plotter	2.11
	1670 Options	2.13
	Drivers	2.14
	Model 905 Controller	2.16
	Model 925 Controller	2.16
	Online Interfaces	2.18
	Plotter Peripherals	2.18
	Model 942 Digitizer	2.18
	Digitizer Operation	2.19
	Model 2100 COM Printer	2.19

CONTENTS (cont)

Section		Page
3	BASIC SOFTWARE	3.01
	General	3.01
	Sample Program	3.02
	Host Computer Basic Software	3.05
	HCBS Subroutines	3.05
	PLOT Subroutine	3.05
	SYMBOL Subroutine	3.06
	NUMBER Subroutine	3.07
	SCALE Subroutine	3.07
	AXIS Subroutine	3.08
	LINE Subroutine	3.08
	Standard Calling Sequences	3.08
	Standard Basic Software	3.10
	CAL EDIT Basic Software	3.11
	Electronic Basic Software	3.12
	Sample Programs	3.13
	Program Number 1	3.13
	Program Number 2	3.26
	Accessory Support	3.32
	Model 942 Digitizer	3.33
	942 Digitizer Characteristics	3.34
	942 Operating Procedure	3.35
	Additional Programming Techniques	3.36
	Uses of INC in SCALE and LINE	3.36
	Shortcut Scaling	3.37
	Forcing SCALE to Choose Zero	3.38
	Scaling for Multiple Curves	3.39
	Data Point Formula	3.44
	Using LINE and SYMBOL with Data Files	3.44
	Drawing Nonstandard AXIS Constructions	3.45
	Drawing Nonstandard AXIS Divisions	3.47
	Using the WHERE Entry	3.48

CONTENTS (cont)

Section		Page
4	OPERATING CONSIDERATIONS	4.01
	General	4.01
	Plotter Initialization	4.01
	Limit Switches	4.01
	Accessory Checks	4.02
	Initial Alignment	4.02
	Controller Initialization	4.03
	Search Addresses	4.03
	Planning Your Graph	4.04
5	BASIC SOFTWARE DESIGN	5.01
	General	5.01
	Data Formats	5.01
	905 Tape Format	5.02
	900 Tape Format	5.05
	900 Online Formats	5.16
	CAL EDIT Format	5.18
	RS-232/936 Serial Format	5.23
	RS-232/960 Serial Format	5.26
	RS-232/836 Serial Format	5.31
	PLOT Subroutine Construction	5.31
	General Flow of PLOT Subroutine	5.31
	Line Approximation Algorithms	5.36
	8-Vector Algorithm	5.36
	Line Splitting Algorithm	5.41
	Velocity Calculations	5.41
	Output Processing	5.45
	Other PLOT Entry Points	5.46
	PLOTS Subroutine	5.46
	FACTOR Subroutine	5.47
	WHERE Subroutine	5.47
	NEWPEN Subroutine	5.47
	SYMBOL Subroutine	5.48
	Symbol Design	5.50
	Symbol Selection Tables	5.52
	Symbol Generation	5.53
	AXIS, LINE, NUMBER and SCALE Subroutine	5.57
	Graphic Controller Software Design	5.62
	Flow of GCS	5.62
	Executive Module	5.64
	READ Module	5.66
	PLOT Module	5.66
	SYMBOL Module	5.67
	Other GCS Modules	5.67

CONTENTS (cont)

Section		Page
6	FUNCTIONAL SOFTWARE	6.01
	General	6.01
	Host Computer Resident Software	6.01
	Standard Host Computer Subroutines	6.01
	Host Computer Symbol Sets	6.06
	Host Computer Photo Plotter Verifier	6.08
	Host Computer Assembler for CalComp Controllers	6.09
	Host Computer Microfiche Management Program	6.09
	Host Computer Digitizer Support Software	6.10
	Host Computer Polygon Painting Software	6.10
	Host Computer World Map Plotting Software	6.11
	Graphic Controller Resident Software	6.12
	Graphic Controller Scaling Software	6.12
	Graphic Controller Windowing Software	6.12
	Graphic Controller Line Printer Simulators	6.13
	Graphic Controller Emulators	6.14
	Graphic Controller Microfiche Management Software	6.14
	Graphic Controller Digitizer Support Software	6.15
	Graphic Controller Sizing Software	6.16
	Graphic Controller Circle Software	6.16
	Graphic Controller Forms Overlay Software	6.16
7	APPLICATIONS SOFTWARE	7.01
	GENERAL	7.01
	THREE-D - A Perspective Drawing Program	7.01
	GPCP - A General Purpose Contouring Program	7.01
	FLOWGEN/F - A FORTRAN Flowchart Generator	7.02
	AUTONET - An Automatic Network Display Program	7.03
	SAMPS - A Subdivision and Map Plotting System.	7.03
	DATAGRAPH - An Automatic Graph Generator	7.03
	PROVE-I - Programs for Numerical Control Verification	7.03
	SYMTRAN - A Symbol Generating and Diagramming Language	7.04
	A.S.A.P. - Automated Symbolic Artwork Program.	7.04
	CONTOUR - A Basic Contouring Program	7.04
	AUTOGANT - An Automatic Gantt Bar Chart Display Program	7.05
	FLOWGEN/C - COBOL Flowchart Generator	7.05

ILLUSTRATIONS

Figure		Page
1-1	System Configurations	1.03
1-2	Software Hierarchy	1.07
2-1	Drum Plotter Product Line	2.05
2-2	CalComp Flatbed Plotters	2.08
2-3	1670 Graphic COM Recorder	2.12
2-4	Magnetic Tape Drivers	2.17
2-5	Model 942 Digitizer	2.20
2-6	CalComp 2100 Configurations	2.21
3-1	Sample Basic Software Graph	3.03
3-2	Sample Plot (LINTYP = -2)	3.15
3-3	Samplt Plot (LINTYP = 0)	3.16
3-4	Sample Plot (LINTYP = +2)	3.17
3-5	Sample Plot, Angular Letter Test	3.20
3-6	Sample Plot, Average Car Value	3.23
3-7	Sample Plot, Multiple Axes and Lines	3.27
3-8	Scaling for Multiple Curves	3.40
3-9	Nonstandard Axes	3.46
3-10	Sample Use of WHERE	3.48
4-1	Plotting Conventions	4.04
5-1	905 Tape Format	5.03
5-2	900 Tape Format	5.07
5-3	ASCII Characters Available with the SYMBOL Routine	5.15
5-4	General Flow of PLOT Subroutine	5.35
5-5	8-Vector Structure	5.37
5-6	Solution of a Straight-Line Approximation . .	5.39
5-7	8-Vector Incremental Algorithm Functional Diagram	5.42
5-8	Line-Splitting Algorithm	5.44
5-9	SYMBOL Subroutine Flowchart	5.49
5-10	SYMBOL Grid System	5.51
5-11	SYMBOL Grid	5.55
5-12	AXIS Flowchart	5.58
5-13	LINE Flowchart	5.59
5-14	NUMBER Flowchart	5.60
5-15	SCALE Flowchart	5.61
5-16	GCS Flowchart	5.63

TABLES

Table		Page

Table		Page
2-1	Plotter/Driver Hardware	2.15
5-1	900 Format Control Codes	5.11
5-2	Delta Values Which May Be Held in Various Number of Characters	5.17
5-3	Pseudo Increments for Various Plotters . . .	5.17
5-4	CAL EDIT System Parameters	5.19
5-5	960 Characters	5.27
5-6	Value of IPEN	5.32
5-7	Controller Memory Layout	5.65

Table P-1. CalComp Current Software Manuals

Manual	Title
1001	GPCP - A General Purpose Contouring Program
1002	THREE-D, A Perspective Drawing Software System
1004	FLOWGEN/F FORTRAN Flowchart Generators
1005	CalComp Software Reference Manual
1006	Programming CalComp Electromechanical Plotters
1010	AUTONET - An Automatic Network Display Program
1011	CalComp Graphics Functional Software - Business Subroutines
1012	CalComp Graphics Functional Software - Drafting Subroutines
1013	CalComp Graphics Functional Software - General Subroutines
1014	CalComp Graphics Functional Software - CRVPT Program
1015	CalComp Graphics Functional Software - Scientific Subroutines
1021	CalComp Graphics Functional Software - FLOCT Program
1022	CalComp Graphics Functional Software - FORGN Program
1026	SAMPS - A Subdivision and Map Plotting System
1027	DATAGRAPH - An Automatic Graph Generator
1030	MASKGEN - An Integrated Circuit Mask Generator
1036	PROVE - Programs for Numerical Control Verification
1037	SYMTRAM - A Symbol Generating and Diagramming Language
1038	CalComp Graphics Functional Software - Drafting Letter and Symbol Set Subroutine
1039	Programming the CalComp Model 1675 COM System
1040	CalComp Graphics Functional Software - Block Letter and Symbol Set Subroutine
1048	GPCP-II, A General Purpose Contouring Program
1051	CalComp Graphics Functional Software - Drafting Letter and Symbol Set Subroutine with Lower Case Letters

Table P-1. CalComp Current Software Manuals (cont)

Manual	Title
1052	CalComp Graphics Functional Software - Photo Plotter Data Verifier and Lighthead Simulator
1056	A.S.A.P. - Automated Symbolic Artwork Program
1058	CAL EDIT User's Guide
1061	CALCOM Microfiche Management Program for the CalComp 21XX Printer
1062	Programming the CalComp 942 Digitizer
1063	CalComp 910/915 Controller Programmers' Reference Manual
1064	CalComp Graphics Functional Software - PAINT
1067	CalComp Microfiche Management Software (MMS) User's Manual
1068	CalComp 942 Digitizer Functional Software
1069	CalComp Applications Software - CONTOUR - A Basic Contouring Program
1074	World Map Plotting Software (WMAP)
1075	Microfiche Management Software (MMS-III) User's Manual
1077	AUTOGANT - An Automatic Gantt Bar Chart Program
1084	FLOWGEN/C - COBOL Flowchart Generator

SECTION 1
CONTENTS

INTRODUCTION TO CALCOMP

General

Graphic Products

Graphic Systems

CalComp Hardware

System Configurations
CalComp Plotters
CalComp Drivers
CalComp Accessories

CalComp Software

Types of Software
CalComp Software Support Policy
Compatibility Features

ILLUSTRATIONS

System Configurations
Software Hierarchy

SECTION 1
INTRODUCTION TO CALCOMP

GENERAL

California Computer Products, Inc., was founded in 1959 for the purpose of designing, manufacturing, and marketing a digital plotter. CalComp has developed many graphic products and today is the acknowledged world leader in computer graphics.

GRAPHICS PRODUCTS

CalComp's graphic products today consist of many different hardware and software items covering the complete range of computer graphics. Products may be configured to the particular needs of customer applications.

GRAPHIC SYSTEMS

Graphic systems are generally used as peripheral equipment to a general purpose digital computer. CalComp systems are supported by computers manufactured by major computer companies throughout the world. The general purpose computer used with a CalComp graphic system is referred to as the HOST computer.

CALCOMP HARDWARE

CalComp Graphic Systems are centered around a graphic output device (plotter) with controlling hardware and optional accessories.

System Configurations

A plotter must be part of a system in order to provide service. CalComp plotters are designed to operate both in offline and online configurations. Figure 1-1 illustrates these two configurations.

When the plotter is used in an offline system, the plotter data and control commands are generated by a host computer and output to a portable medium e.g., magnetic tape or punched cards. The data is input on a CalComp controller which converts the data into signals and drives the plotter.

When the plotter is used in an online system (see Figure 1-1), the plotter commands are output directly to a CalComp plotter through interface electronics supplied by CalComp or supplied by the host computer manufacturer. The plotter may be connected through a CalComp controller. The data may not go directly to the CalComp plotter in some systems, but may be "SPOOLED" or transmitted to a terminal to which the plotter is connected.

CalComp Plotters

CalComp produces the following basic types of plotters:

- Electromechanical ink-on-paper plotters
- Electronic/microfilm plotter

The electromechanical plotters include drum and flatbed plotters; each available in various sizes. The electronic/ microfilm plotter uses a CRT (cathode-ray tube) with a camera.

Drum Plotters

CalComp's drum plotters present computer output data in an uninterrupted manner. A continuous series of graphics, drawings, or maps can be completed in rapid sequence, with the paper advancing under computer control into position for each successive presentation. A series of pictures can be run several times for updating or addition of new information, because of the repeatability factor. Continuous plots up to 120 feet in length are produced by the rotary motion of the drum and lateral motion of the pen carriage.

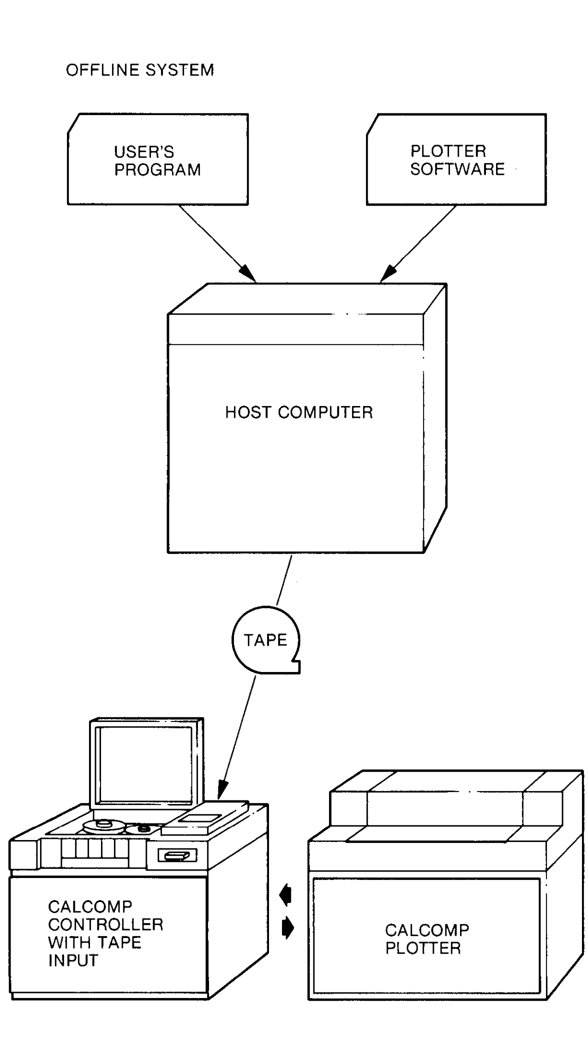


Figure 1-1. System Configurations

Flatbed Plotters

The exposed plotting surface of CalComp's flatbed plotters makes it possible to view the graphic presentation as it is being plotted. Flatbed plotters are well suited to online operation where real-time graphic output is required. The flatbed plotter is particularly adapted to automatic drafting applications. The 'flatbeds' can handle a variety of preprinted forms and special materials which normally are not practical for printing and punching in roll form. This facility allows for easy updating of previous work or additions to preprinted forms.

Electronic Plotter

The Calcomp electronic plotter is a true **digital** plotter, utilizing the same basic design principles developed, perfected, and patented by CalComp for all of its electromechanical (pen-type) plotters. CalComp Basic Software converts the user's data to commands which produce discrete, electron-beam deflections (relative to the X and Y axes) and intensity variations on the face of the cathode-ray tube. The CRT display is transmitted through the camera lens system and recorded on microfilm. The exposed film may then be processed to produce either positive or negative transparencies for direct viewing or for printing in a variety of processes.

CalComp Drivers

CalComp produces three types of drivers for CalComp plotters. These drivers may be either stand-alone offline systems or online systems connected directly (or remotely) to a host computer.

Offline Systems

CalComp drivers with magnetic tape units provide offline plotting capability for virtually all medium- and large-scale computers. Offline plotting provides the following operational advantages:

- Flexible scheduling and efficient use of computer time
- Independent operation of plotting systems (regardless of computer load)
- Permits several kinds of computer systems to be used with a given plotting system
- Permits repeat plotting without access to a computer

CalComp magnetic tape units have special features and the high reliability required for reading of industry-compatible tapes for operating CalComp plotters. Specific CalComp drivers can accept input in the form of punched cards or punched paper tape.

Online Systems

When a CalComp plotter is used in an online system, the plot data and control commands are supplied directly from the computer through a CalComp plotter controller or adapter. Interface units for CalComp plotters are available from the computer manufacturer, for some computers.

Online systems offer the advantage of graphic output in real time, and in most cases require minimum capital expenditure. These systems do not make optimum use of computer time unless output is spooled or a multitask system is used. The CalComp product line includes online interface units for a wide range of computers.

CalComp Accessories

CalComp supports graphic systems with a full range of optional accessories. Electromechanical plotters can use a wide variety of plotting materials with both ballpoint and liquid ink pens pressurized at high speeds. Flatbed plotters have accessories to scribe and cut materials and expose film with light sources. The electronic plotters employ a wide variety of cameras.

Additional Peripherals

Peripherals may be attached to certain controllers and include:

- Card Reader
- Paper Tape Equipment
- Teletypes

The CalComp product line includes a COM printer used as a direct line printer replacement or used offline in a full Microfiche Management System. A digitizer with punch card or paper tape output, may also be included.

CALCOMP SOFTWARE

CalComp has developed and maintains a library of software to support CalComp products on major computers throughout the world. The software is supported by a staff of development programmers and regional analysts. CalComp strives to maintain a high degree of compatibility between software and hardware.

Types of Software

CalComp provides three classifications of graphic software:

- Basic Software
- Functional Software
- Applications Software

Figure 1-2 shows CalComp software hierarchy.

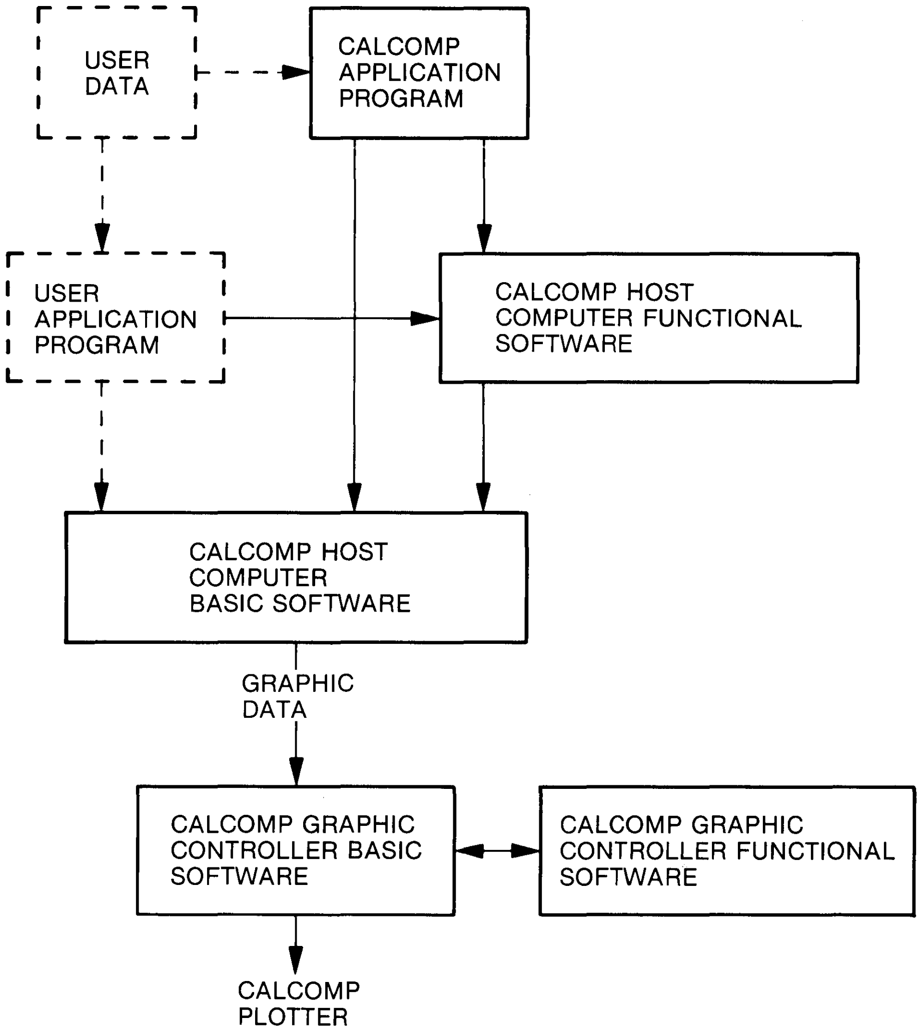


Figure 1-2. Software Hierarchy

Basic Software

Basic software is a set of closely related subroutines that generate output for controlling a digital graphic system. The programmer is not required to communicate with the hardware in its own data structure. Instead, he communicates with the set of subroutines in a manner such as:

"move the pen to specified coordinates"

"place some characters at a certain location on the page"

"draw an annotated axis"

"scale and draw a line through a series of points"

This kind of communication reduces the problem of formatting graphic output to the level of formatting the data for a printed report.

Basic Software consists of Host Computer Basic Software (HCBS) and Graphic Controller Software (GCS). GCS is used only in selected systems. HCBS consists of those subroutines which are resident on the user's computer. Output for CalComp graphic systems is produced by various types of HCBS. Standard HCBS produces a compact data format while CAL-EDIT HCBS produces an editable format; both used with CalComp electromechanical plotting systems. A COM HCBS produces a compact data format incorporating the additional capabilities of an electronic plotting system. GCS is the software which resides in the CalComp programmable controller, interprets the appropriate data format and drives the plotter.

Functional Software

The second level of CalComp software is Functional Software. Functional programs and subroutines perform plotting functions used in many different applications. Functional Software may reside in either the user's host computer or in the CalComp controller.

Applications Software

The highest level of software is composed of Applications programs. These programs are the problem solvers which determine the output to be graphically displayed. CalComp can provide these proprietary programs or the user may develop his own programs. Examples of Applications programs available from CalComp include:

"GPCP" - A General Purpose Contouring Program

"FLOWGEN/F" - A FORTRAN Flowchart Generator

CalComp Software Support Policy

CalComp software is an integral and essential part of the product line. Each software package is given a product number and is supported with appropriate literature and documentation. This is similar to the processing of individual hardware items. Because of the proprietary nature of CalComp software, the packages are leased rather than sold outright. The lease agreement allows unrestricted use of the software with the CalComp system, and the lease price covers this usage for as long as the system is retained.

Compatibility Features

The modular design of CalComp hardware and software allows the user to upgrade or modify CalComp plotting systems without the massive conversion problems usually associated with hardware modifications. A change in the user's computer or plotting system requires little or no modification to operational application programs, if they have been written in a high-level language such as FORTRAN.

SECTION 2

CONTENTS

CALCOMP GRAPHIC SYSTEMS

General

System Specifications

Accuracy and Repeatability
Plotting Flexibility
Plotting Speed
Computer Time

Types of Plotters

Drum Plotters
Flatbed Plotters
Electronic CRT Plotter

Drivers

Model 905 Controller
Model 925 Controller

Online Interfaces

Plotter Peripherals

Model 942 Digitizer
Model 2100 COM Printer

ILLUSTRATIONS

Drum Plotters
Flatbed Plotters
COM Recorder
Magnetic Tape Drivers
Model 942 Digitizer
CalComp 2100 Configurations

SECTION 2 (cont)
CONTENTS (cont)

TABLES

Plotter/Driver Hardware

SECTION 2
CALCOMP GRAPHIC SYSTEMS

GENERAL

An understanding of the software requirements for CalComp systems requires knowledge of the differences in models available as well as knowledge of the specifications for different system configurations. Section 2 provides descriptions of CalComp digital plotters, plotter drivers, and auxiliary hardware.

All CalComp digital plotters operate on the principle of moving the plotter in digital increments. Decoded input commands from the computer are used to produce increments of movements in either direction along either axis, or at some angle relative to the axes. The electromechanical ink-on-paper plotters generates a plot by movement of a pen relative to the surface of the recording paper. The electronic plotters produces a plot by movement of a cathode-ray tube electron beam with the plot recorded automatically on microfilm. Decoded computer commands are used to raise and lower the pen, or to blank and unblank the electron beam.

SYSTEM SPECIFICATIONS

CalComp's product line covers a wide range of plotters and drivers suitable to many applications. The best configuration for a particular application is dependent upon the required specifications. Specifications and characteristics to be considered are:

- Accuracy and Repeatability
- Plotting Flexibility
- Plotting Speed
- Computer Time

Accuracy and Repeatability

How accurate is the digital plotting system? Ideally, no information that the user is capable of reading from a graphic presentation should be lost in the digital plotting process. CalComp equipment can produce drawings with resolution to 0.0001 inch, with excellent repeatability. CalComp plotters can retrace a plot or curve without discernible deviation. Overplotted lines appear as one line. An unlimited number of plots or graphs can be generated from the same computer program with unvarying resolution.

Plotting Flexibility

What are the capabilities of the digital plotting system? CalComp plotters can produce any variety of types and sizes of symbols, at any angle or position, in addition to producing any type of complex curve or line. The system's capabilities are limited only by programmer ingenuity. This flexibility is enhanced by the availability of different types and colors of pens and inks, together with a wide variety of preprinted forms on continuous rolls.

Plotting Speed

How fast can the system produce a graph or drawing without sacrificing quality or accuracy? The 500 series plotters draw lines at the rate of 2.0 to 4.2 inches/second, depending on the angle of movement; a speed adequate for low-volume applications.

Some CalComp plotters can move pen over paper at speeds up to 40.0 inches/second. These plotters can produce a greater volume of complex, fully annotated graphic output in a given time than any other existing electromechanical system. CalComp also offers a cathode-ray tube/microfilm digital plotting system capable of plotting 400,000 increments/second, for high-volume plotting requirements. This plotting capacity is equivalent to 500 inches/second at standard magnification.

Computer Time

How much computer time is required to produce a digital plot of given complexity and size? Computer time is costly and should be kept to a minimum, consistent with accuracy and quality. For offline plotting, the amount of magnetic tape required to produce a given plot is the determining factor. For online plotting, the maximum plotter speed and the type of buffering are of paramount importance. CalComp offline systems can produce 2,000 linear inches of plot from a single inch of magnetic tape. CalComp online systems are capable of time-shared operation with other input/output equipment. When internal buffers are used, these systems can accept high input data rates.

TYPES OF PLOTTERS

The CalComp product line includes:

- Drum Electromechanical Plotters
- Flatbed Electromechanical Plotters
- Electronic CRT/Microfilm Plotters

Drum Plotters

Drum plotters present computer output data in an uninterrupted manner. A continuous series of graphs, drawings, or maps can be completed in rapid sequence, with the paper advancing into position under computer control for each successive presentation.

Repeatability

Repeatability allows a series of pictures to be run several times for updating or addition of new information. Continuous plots of 120 feet in length are produced by the rotary motion of the drum and the lateral motion of the pen carriage. Figure 2-1 shows and describes drum plotters.

Ballpoint/Liquid Ink Pens

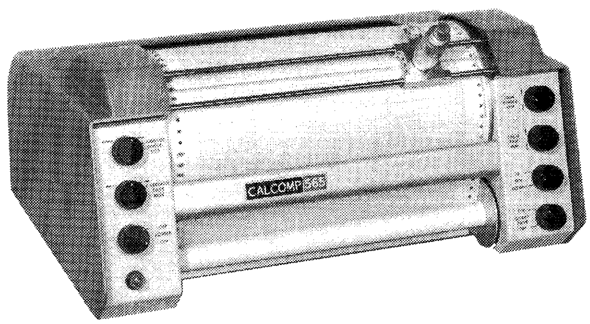
Drum plotters are available with either pressurized ballpoint pens or liquid ink pens in a variety of colors. Liquid ink pens are available in several widths. Graphic chart materials are available in the following forms:

- Translucent paper
- Vellum
- Rag paper
- Mylar (available with preprinted grids)

Accessories

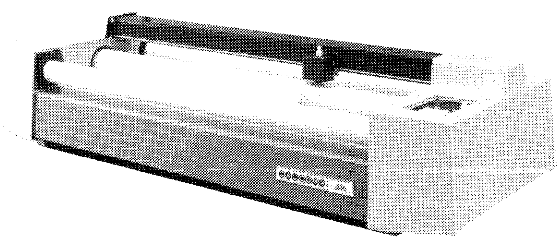
Drum plotter accessories include the following:

- Plot display attachments (for 500 series plotters for ease of viewing finished plots)
- Narrow drum kits (for wide plotters to allow use of narrow paper)
- Clear magnifying reticle (for accurate manual alignment of the plotter)



565 PLOTTER

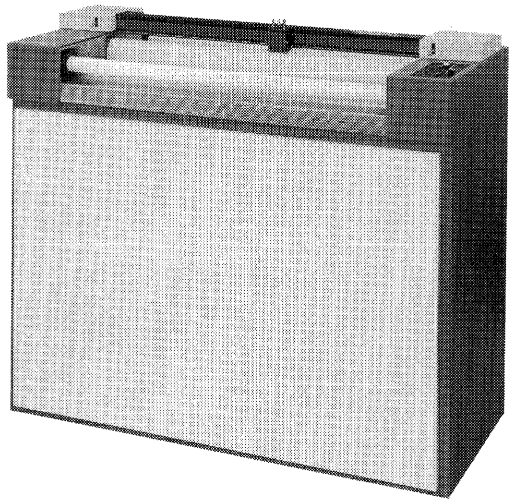
12-inch (30.48 cm) drum plotter.
Drawing speed: 3 inches/second, 1.5 inches/second, or 3 cm/second.
Resolution: 0.01 inches, 0.005 inches, or 0.1 millimeters.
Tabletop or rack mounted.
Easily interfaced to a variety of control units.



836 PLOTTER

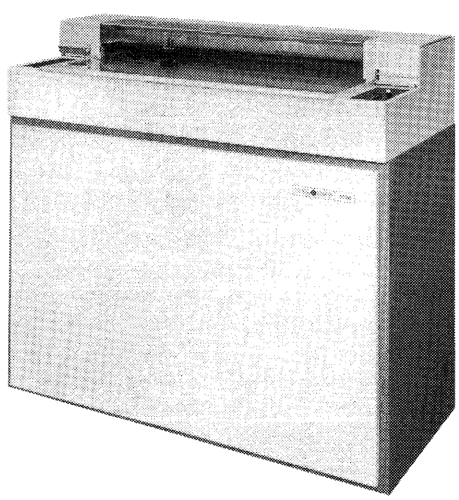
33.1-inch (84.1 mm) drum plotter.
Drawing speed: 50 mm/second (1.97 inches/second).
Resolution: 0.1 mm (0.004 inches).
Tabletop mounted.
Plug compatible with 500 series plotters or optionally available RS-232C connector.

Figure 2-1. Drum Plotter Product Line



936 PLOTTER

33.0-inch (83.8 cm) drum plotter with optional narrow paper adapter.
3.6 inches/second or 9 cm/second pen down speed.
5.0 inches/second or 12.5 cm/second pen up speed.
Throughput: 4.25 inches/second (10.80 cm) (typical).
Resolution: 0.002 inch or 0.05 mm.
Floor mounted.
Easily interfaced to a variety of control units.
3 program-selectable pens.



1036 PLOTTER

33.3-inch (83.8 cm) drum plotter with optional narrow paper adapter.
10-inch/second or 25 cm/second maximum pen down speed.
Throughput: 7 to 8 inches/second (17 cm to 20 cm) (typical).
Resolution: 0.002 inch or 0.05 mm.
Floor mounted.
Requires use of 925 Controller.
3 program-selectable pens.
Integral noise cover.

Figure 2-1. Drum Plotter Product Line (cont)

Flatbed Plotters

The exposed plotting surface of CalComp's flatbed plotters makes it possible to view the graphic presentation as it is being plotted. Flatbeds are well suited to online operation where real-time graphic output is required. The flatbed plotter is available in several sizes and is particularly adaptable to automatic drafting applications. Flatbeds are capable of handling a variety of preprinted forms and special materials which normally are not practical for printing and punching in roll form. This facility allows for easy updating of previous work or additions to preprinted forms. The larger size flatbeds include a 4-color pen assembly for automatic plotting of multicolor graphics. Figure 2-2 describes various flatbed plotters.

Flatbed Plotter Drafting Tool

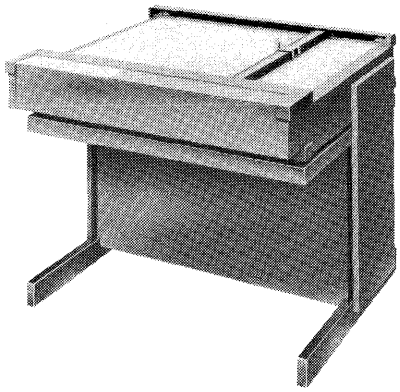
Flatbed plotters are available with either pressurized ball-point pens or liquid ink pens. The materials that may be used are extensive and include:

- Paper
- Vellum
- Mylar (for inking)
- Scribe coat (for scribing)
- Peel coat (for use with film cutters)
- Photographic film (for use with lightheads)

Flatbed Accessories

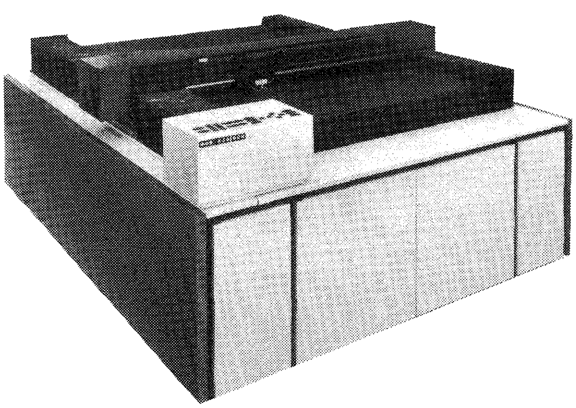
Flatbed plotter accessories include the following:

- 748 Pressurized Inking System: Each of the four liquid ink pen reservoirs are supplied with separate air pressure tubes. The electro-mechanical system senses plotter speed and controls the pressure to each pen as a function of plotter speed.



502 FLATBED PLOTTER

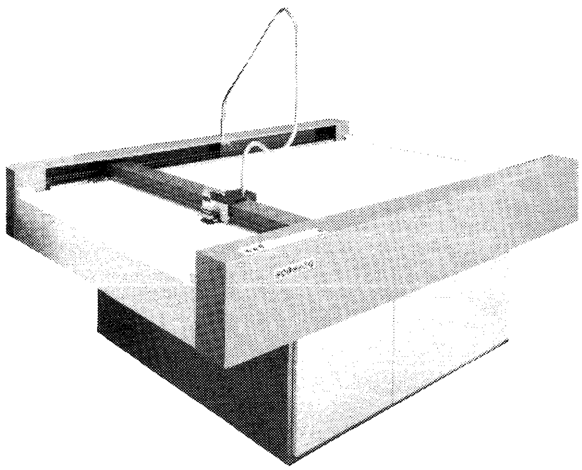
31 x 34 inch (78.74 cm x 86.36 cm) digital flatbed plotter, operating at a constant speed of 300 increments/second. Resolution (increment size) of 0.01 inch, 0.005 inch, or 0.002 inch (0.1 or 0.05 millimeter). Employs vacuum holddown, single pen, and may be vertically mounted.



745 FLATBED PLOTTER

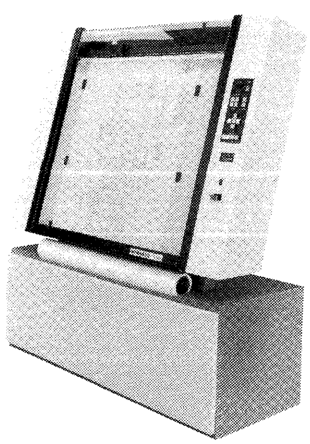
43 x 59 inch (109.22 cm x 149.86 cm) precision flatbed plotter with a single black, granite, plotting surface giving the plotting industry the highest quality (most accurate and best repeatability) plotter at a reasonable speed. Plotter resolution is 0.0001 inch (0.000254 cm) with speed ranges from 0.1 to 8.0 inches/second (0.254 to 20.30 cm). Four pens or drafting accessories are standard. Proven accessories include scribes, film cutters, and lightheads.

Figure 2-2. CalComp Flatbed Plotters



748 FLATBED PLOTTER

48 x 82 inch (122.0 cm x 208.0 cm) high performance flatbed plotter featuring extremely high speed, flexibility of speed control and quality inking through the use of a comprehensive pressurized inking system. Plot speed can be as high as 42 inches/second (107.0 cm), and acceleration as high as 1.4G. Four program-controlled pens are standard. Accessories include scribes and film cutters. It is a high resolution plotter with increment size of 0.005 millimeter.



960 PLOTTER

33 x 59 inch (841 mm x 1518 mm) digital plotter with vertical plotting surface. Plotter resolution is 0.0125 millimeter (0.005 inch) with plotting speed as high as 30 inches/second (75.0 cm) and acceleration of 4G. Two program-controlled pens are standard with liquid ink or ballpoint pens.

Figure 2-2. CalComp Flatbed Plotters (cont)

- Tangential Tool: The Tangential Tool Holder System consists of a tool holder head assembly which mounts on the plotter pen carriage, an electronic assembly, and a Tangential Tool Cable Assembly. A scribe dust vacuum pickup is provided. The Tangential Tool Holder System is designed to accommodate directional tool tips such as chisel-point scribing tools and strippable film cutting knives. The system indexes the tool through 360° of rotation in selectable angular increments under program control such that the tool cutting direction is approximately parallel to the direction of the plotter pen carriage movement. Scribing tools and film cutting tools are interchangeable. Tangential scribing GCS is required for scribing and tangential film cutting GCS is required for film cutting. Film cutting may be accomplished at speeds up to 30/ips (76.20 cm) axially. Scribing may be accomplished at speeds up to 3.2/ips (81.28 cm) axially.

- Stationary Scribe System: A tungsten carbide tool is held stationary in a tool holder for plotting precise line widths on negative scribe film. This material produces high contrast, accurate linework for mapping or numerical control optical-following machinery. Line widths range from 0.001 inch to 0.008 inch (0.00254 cm to 0.01932 cm). Two different systems are available, one for the large flatbed plotter and one for the small flatbed plotter.

Flatbed Accessories (cont)

- Film Cutter: This system consists of spade-shaped tungsten carbide cutting tools for straight (0, 180, ±45, ±90, or ±135 degrees) lines on strippable film. Strippable film is a transparent base with a photographically opaque coating that can be cut and peeled away. It is popular medium for integrated circuit manufacturers and cartographers (map makers).

- 7182 Optical Writing: Available only for the 745 Flatbed Plotter. Consists of a system capable of producing artwork directly on photographic film. It has eight program-selectable light pens consisting of fiber optics bundles that can produce a square, round, or rectangular image on film.

Electronic CRT Plotter

The CalComp Model 1670 Electronic Plotter produces high resolution microfilm in 16, 35, or 105 millimeter film using incremental dot writing on a high resolution CRT. The high speed of the 1670 plotter utilizes the digital incremental technology of CalComp electromechanical plotters to produce a high output volume.

Reference Figure 2-3.

STANDARD FEATURES

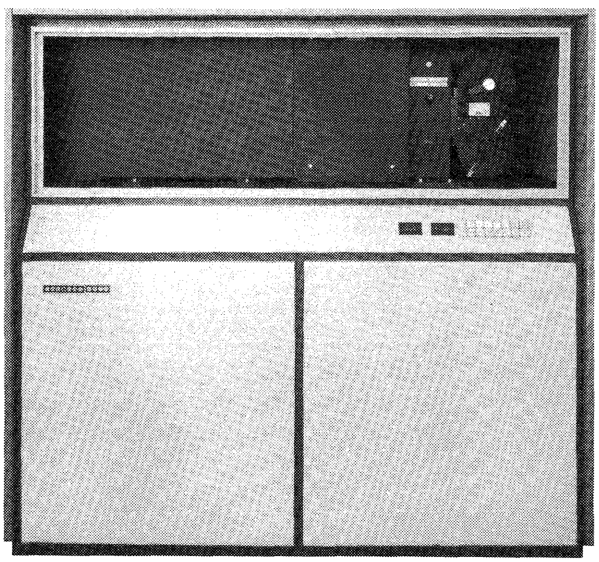
Linearity of better than ±0.5% over the entire frame.

Stability of better than ±0.1% over a continous 8-hour period.

Raster of 16,384 x 16,384 addressable spot positions.

0.0008-inch spot size or less on the CRT over the usable CRT area.

Thirty levels of program-controlled intensity producing 20 usable line weights (minimum). Normal intensity is set at hardware level 15.



Four program-selectable step sizes.

High speed hardware character generator producing 64 characters at an average character generation time of 25 microseconds. Provides high speed printing on Dacomatic A film.

Absolute repositioning (move from any raster position) in less than 150 microseconds.

Hardware line generator providing a maximum throughput of 400,000 increments/second and an average incremental rate of 2.5 microseconds/increment at hardware intensity level 15.

Hardware rotation providing either Cine or Comic presentation.

16mm, 35mm, and 105mm microfiche cameras are operator interchangeable.

Pushbutton selectable intensity for use with AHU or Dacomatic A films with no speed degradation.

Figure 2-3. 1670 Graphic COM Recorder

1670 Options

The 1670 is available with the following options:

- Cameras

  An operator can easily change cameras or lenses on the 1670. The exception is the 105mm full-frame camera which is factory installed.

  One 16mm or 35mm camera is included in each 1670 system (sprocketed or nonsprocketed). A 105mm microfiche camera with a 16mm film adapter or a 105mm full-frame camera is also available.

  Typical applications for various camera options are as follows:

  <u>16mm Nonsprocketed</u>

  Printing

  <u>16mm Sprocketed</u>

  Movies

  Business Graphics

  Frame-Butted Strip Charts

  <u>35mm Nonsprocketed</u>

  High Resolution Graphics e.g., seismic display

  Engineering Drawings

  Mapping

  <u>35mm Sprocketed</u>

  Printing/Publishing (used with Bruning 1800 platemaker)

  Frame-Butted Continuous Graphics

<u>105mm Microfiche</u> (with 16mm adapter)

Microfiche (with graphics) plus roll film capability

<u>105mm Full Frame</u>

Large Format High Resolution Graphics (maps or engineering drawings)

- Lenses

Optional 16mm and 35mm lenses plus 24X, 42X, and 48X lenses for the 105mm microfiche camera.

- Viewer

An optional Model 8350 CRT Viewer permits the user to view the CRT image for verification purposes. The viewer may be located up to 100 feet from the 1670 System.

- Hardware Character Generators

A 64-character, hardware character generator is standard. Printing speeds to 10,000 lines/minute are possible using this hardware character generator. An optional Katakana character set is available.

DRIVERS

The CalComp product line includes several plotter drivers. Drivers accept either input from magnetic tape or they are connected directly online. Drivers contain either a programmable controller or they are hard-wired controllers. Table 2-1 lists the drivers together with the plotters they can drive.

Table 2-1. Plotter/Driver Hardware

Plotters (drum)	Tape Drivers		Online Drivers			
	905	925	621	J1130B (IBM 1130)	RS-232	925 (P18/P34)
565	X	X	X			X
836	X	X	X		X	X
936	X	X		X	X	X
1036		X				X
Plotters (flat-beds)						
502	X	X	X			X
748		X				X
745		X				X
960		X			X	X
Plotters (electronic)						
1670		X				X

Model 905 Controller

The CalComp Model 905 Controller has been designed to provide low cost offline plotting. Input of plot data is prerecorded magnetic tape in IBM compatible NRZI or PE format. A read-only magnetic tape unit is integral to the 905 controller with a choice of 7-track NRZI (200, 556, and 800 bpi), 9-track NRZI (800 bpi), or 9-track PE (1600 bpi) read heads.

The 905 controller offers manual operator controls for the magnetic tape functions of TAPE DENSITY selection, TAPE LOAD, REWIND, BACKSPACE, FAST FORWARD, and RESET. Other manual controls/indicators include HALT, POSITION, SINGLE PLOT, MULTIPLE PLOT, and PARITY CHECK. There are also indicators for FINAL BLOCK ADDRESS RECORD and INTERMEDIATE BLOCK ADDRESS RECORD. Figure 2-4 shows and describes the Model 905 Controller.

A compact data structure has been developed to reduce the amount of data transfer and a fixed size character buffer is provided to allow high speed data transfer. A tape read speed of 37.5 inches/second and up to 200 inches of plotting can be generated by each inch of magnetic tape. A reduction of up to 98% over older CalComp systems can be obtained for many applications using the 905 controller.

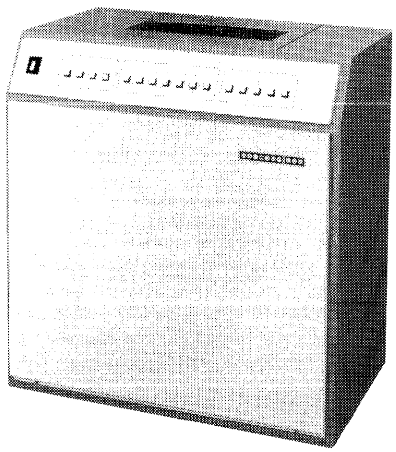
Model 925 Controller

The CalComp Model 925 Controller is a stored program device that has been designed to drive all CalComp plotters. A unique instruction set within the 925 controller optimizes data conversion, formatting, and data transfer to the graphics equipment. This controller is an 8,192 (9-bit binary) word unit that features a memory cycle time of 1.15 to 1.65 microseconds. Memory options of up to 32,768 words of core memory are available. The 925 controller includes a magnetic tape cartridge I/O unit for program loading.

Data can be input to the 925 controller several ways. An integral IBM compatible 1/2-inch magnetic tape unit is available in one of four options:

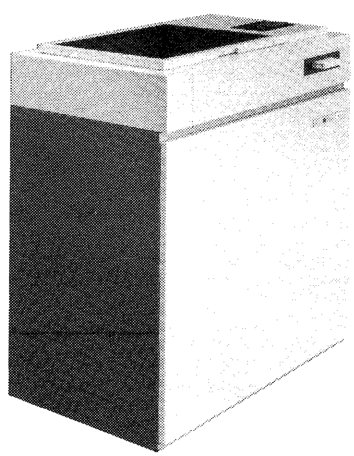
- 7-track NRZI with densities of 200, 556, or 800 bpi
- 9-track NRZI with 800 bpi density
- 9-track PE unit with 1600 bpi density
- A single unit incorporating the features of all of the above

The 925 controller can also use for input a card reader, an online adapter (IBM 360/370 system, or a serial line adapter to connect online to several popular minicomputers).



905 CONTROLLER

Integral Magnetic Tape Unit
Standard IBM Tape Format - 7/9 Track
Digital Plotting at Minimum Cost
Easy to Operate - No Special Skills Required
Compact Coding Techniques Reduce Data Requirements
Automatic Error Detection



925 CONTROLLER

Programmable Controller
Standard Integral Magnetic Tape Unit
Optional Card Reader or On-line Adapters
8K to 32K Words of 9-Bit Core Storage
Integral Tape Cartridge Program Load Unit
Versatile Options (Scaling and Windowing)

Figure 2-4. Magnetic Tape Drivers

Online Interfaces

The Model 621 Remote Data Phone Receiver Plotter Controller is used with a 500 series plotter and various RJE terminals. The 621 offers remote plotting at a low cost with a widely used terminal.

The J1130B online interface is used with a 936 plotter and an IBM 1130 computer. The J1130B/936 offers a higher speed and better resolution plotter than an IBM 1627 plotter.

The RS-232/936, RS-232/960, and RS-232/836 online interfaces are used with the plotters and any computer system equipped with an EIA RS-232 interface connection with 9600 BAUD transmission. The RS-232 interfaces are asynchronous devices and receive 8-bit words, serially transmitted. The RS-232/936 and RS-232/960 interfaces feature compact data transmission, and contain a buffer. These features provide full-speed plotting with low CPU interference.

The 925 controller may also be connected online to computers. This allows all CalComp plotters to be attached online. The core memory of the 925 acts as a buffer and allows fast transmission of data along with complete checking of data.

Other online interfaces supporting 565, 836, and 936 plotters are available from computer manufacturers.

PLOTTER PERIPHERALS

Items which complement the plotter line include film processors and duplicators. These items do not require software support. Items such as digitizers and COM printers do require software support.

Model 942 Digitizer

The 942 digitizer is a high precision coordinate measuring device that converts graphical data into digital form suitable for computer applications. This unit consists of a 42 x 60-inch/work surface (pedestal base) with an interface to either a teletype or keypunch. The work surface is comparable to a standard drafting table with adjustments for surface tilt and elevation. The cursor is a free moving type with a two-inch diameter glass window containing a precision cross-hair reticle. Five buttons on the cursor, control all digitizing functions. A digital display mounts to the top edge of the table and displays X-Y coordinates together with a sequence number. Figure 2-5 shows the 942 digitizer.

Digitizer Operation

The 942 output is fixed format. CalComp offers software that provides menu capabilities, coordinate transformations, variable resolution, and other functions. This simplifies digitizer operation while freeing the user from hardware limitations.

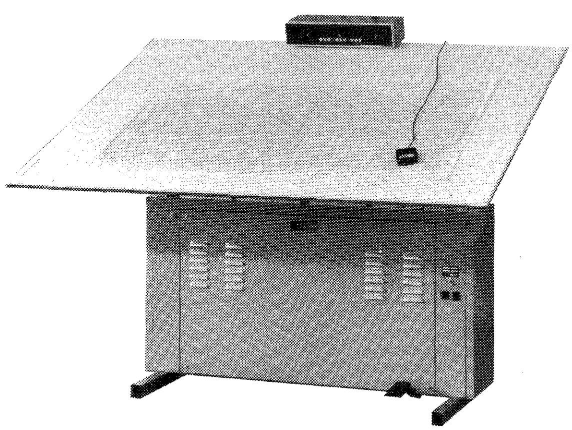
Model 2100 COM Printer

The 2100 COM printer systems produce alphanumeric computer output 25 times as fast as mechanical line printers. Instead of bulk, paper printouts, 2100 printers produce 16mm roll microfilm or 105mm microfiche which are easy to retrieve, inexpensive to duplicate and distribute, and compact to store. The 2100 COM printer is available in several configurations as shown in Figure 2-6. These configurations illustrate online or offline systems and 16mm roll or 105mm microfiche systems. The cameras use either 24X, 42X, or 48X lenses and the 105mm camera can handle several standard page formats.

2100 COM printer features are:

- Diode plugboard used as a carriage control tape
- Pushbutton selection of Cine or Comic image rotation
- Easily adjusted glass slide forms overlay
- Resettable frame and fiche counters

The CalComp 2100 COM system is supported by a complete line of software products in the functional software section.



STANDARD FEATURES

Work Surface
42 x 60 inches (106.68 cm x 152.40 cm)

Displays
4-digit sequence counter
5-digit X position plus sign
5-digit Y position plus sign

Cursor
Free, with precision cross-reticle and 5 buttons:

- 3 buttons for recording data
- 1 button for coordinate origin
- 1 button for advancing the sequence counter

Resolution
0.001 inch (0.00254 cm)

Accuracy
0.005 inch (0.01270 cm)

Table Adjustments
Height: 34 to 46 inches (86.36 cm x 116.84 cm)
Tilt: 0 to 90 degrees

Figure 2-5. Model 942 Digitizer

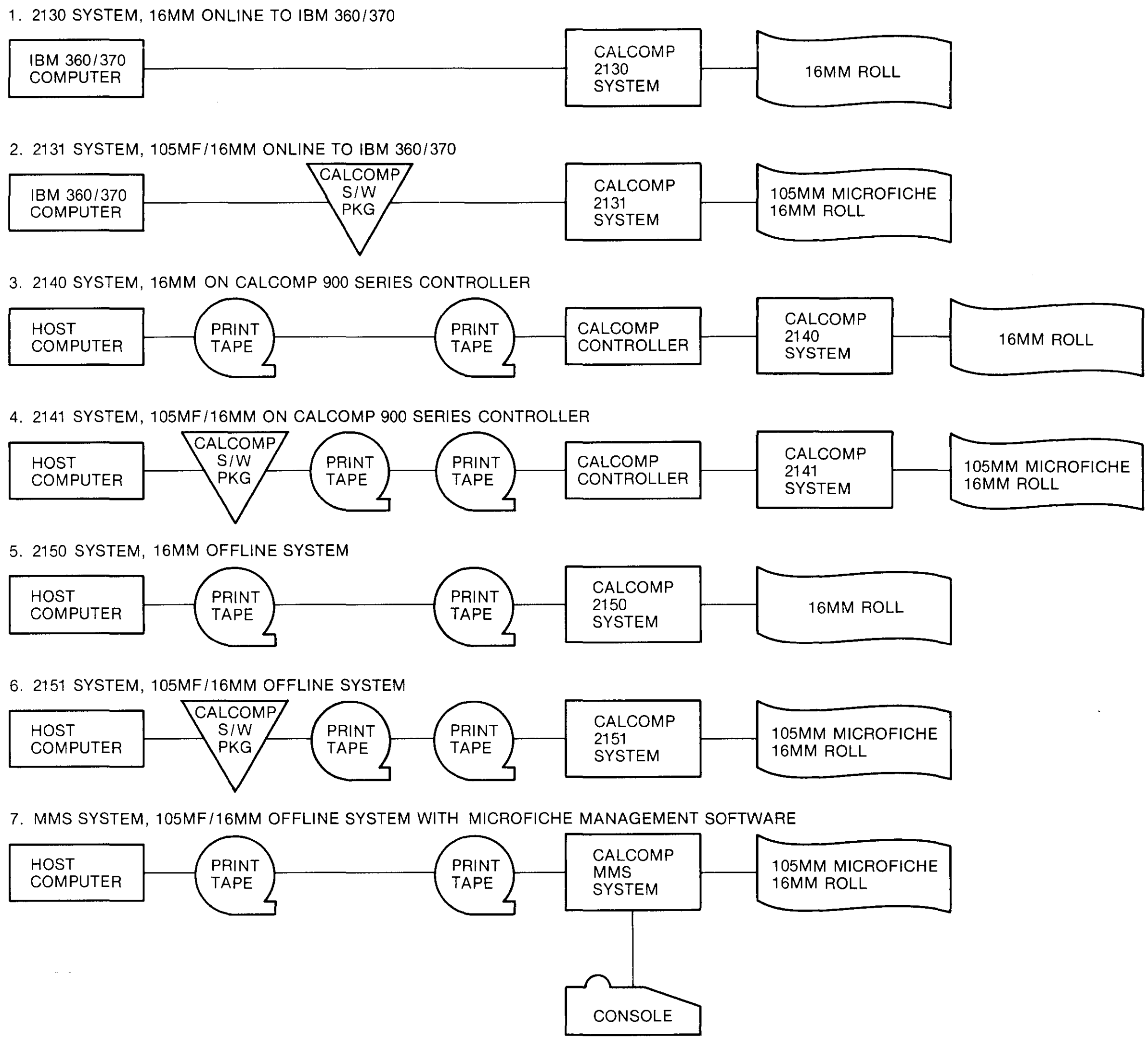


Figure 2-6. CalComp 2100 Configurations

SECTION 3
CONTENTS

BASIC SOFTWARE

General

Sample Program

Host Computer Basic Software

HCBS Subroutines
Standard Calling Sequences
Standard Basic Software
CAL EDIT Basic Software
Electronic Basic Software

Sample Programs

Program Number 1
Program Number 2

Accessory Support

Model 942 Digitizer

Additional Programming Techniques

Uses of INC in SCALE and LINE
Shortcut Scaling
Forcing SCALE to Choose Zero
Scaling for Multiple Curves
Data Point Formula
Using LINE and SYMBOL with Data Files
Drawing Nonstandard AXIS Constructions
Drawing Nonstandard AXIS Divisions
Using the WHERE Entry

CONTENTS (cont)

ILLUSTRATIONS

Sample Basic Software Graph
Sample Plot (LINTYP = -2)
Sample Plot (LINTYP = 0)
Sample Plot (LINTYP = +2)
Sample Plot, Angular Letter Test
Sample Plot, Average Car Value
Sample Plot, Multiple Axes and Lines
Scaling for Multiple Curves
Nonstandard Axes
Sample Use of WHERE

SECTION 3
BASIC SOFTWARE

GENERAL

CalComp Basic Software consists of a set of subroutines for the user's general purpose computing system (referred to as Host Computer Basic Software - HCBS). In some graphic systems, the software that resides in the CalComp programmable controller, referred to as Graphic Controller Software (GCS), is also part of the CalComp Basic Software.

CalComp Host Computer Basic Software is a set of subroutines callable from a user's program which permits the user to perform elementary operations on the plotter without concern for the details of actual plotter operation or output to the plotter. Basic software enables a user to program a plot simply specifying the lines and symbols to be drawn. The software then produces automatically and efficiently the codes necessary to draw these lines and symbols on the plotter and outputs them for the resultant graphic display.

In addition to drawing lines between specified points, alphanumeric and special characters; CalComp basic software includes subroutines that provide the following functions:

- To draw an annotated axis for a graph
- To scale user data to fit on a graph of a given size
- To connect a set of data points with straight lines
- To plot a floating point number in FORTRAN 'F' type format
- To scale the size of a plot

HCBS is available in three versions:

- Standard
- CAL EDIT
- Electronic

The standard HCBS is used on all CalComp electromechanical plotters (online/offline) with any controller. CAL EDIT HCBS is only available for offline electromechanical systems driven by a programmable controller with a console. The electronic HCBS is used with a 1670 COM System and is similar to standard HCBS with the addition of CRT functions and camera operations.

The GCS is the software that the user interfaces with when operating the plotting equipment. The GCS is directly related to the type of HCBS used. How to use the GCS is discussed in Section 4.

SAMPLE PROGRAM

To illustrate the use of CalComp Basic Software, the sample program that produced the graph in Figure 3-1 is shown with the following assumptions:

1. The 24 pairs of TIME and VOLTAGE data values are contained in a file of 24 records.

2. The plotting pen is initially positioned at the extreme -Y side of the plotter.

> NOTE
>
> Only 11 executable statements are required to complete the graph.

```
   DIMENSION XARRAY (26), YARRAY (26)
```

Reserve space for 24 data values plus two additional locations required by the SCALE, AXIS, and LINE subroutines.

```
10 CALL PLOTS (0,0,6)
```

Initialize the PLOT subroutine, giving the logical number for the output device.

```
20 READ 25, (XARRAY (I), YARRAY (I), I = 1, 24)
25 FORMAT (2F6.2)
```

Read 24 pairs of TIME and VOLTAGE from an input file into two arrays with names XARRAY and YARRAY.

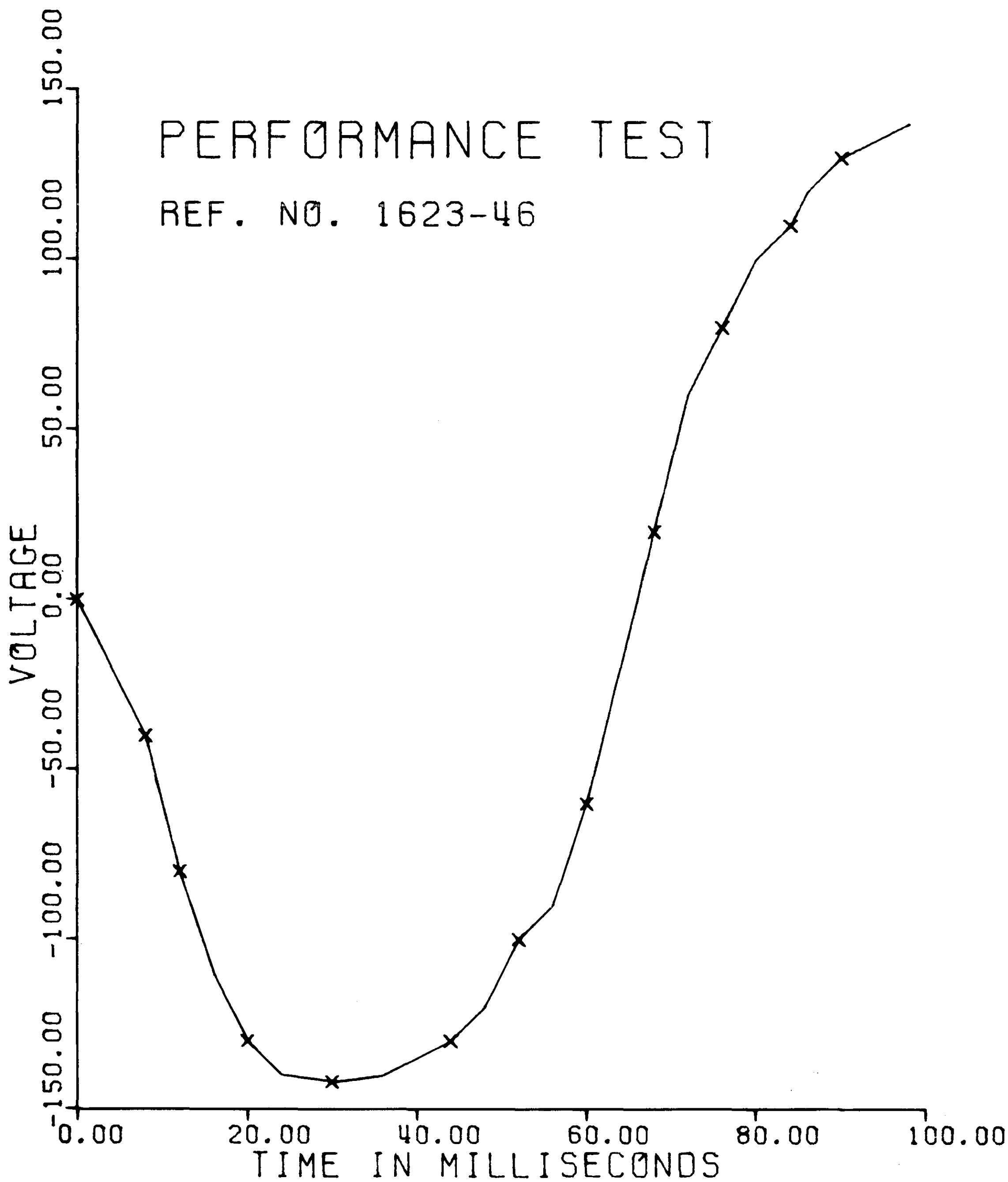


Figure 3-1. Sample Basic Software Graph

SAMPLE PROGRAM (cont)

30 CALL PLOT (0.0, 0.5, -3)

Establish a new origin 1/2-inch higher than the point where the pen was initially placed by the operator so that the annotation of the TIME axis will fit between the axis and the edge of the plotting surface.

40 CALL SCALE (XARRAY, 5.0, 24, 1)

Compute scale factors for use in plotting the TIME values within a 5-inch plotting area.

50 CALL SCALE (YARRAY, 6.0, 24, 1)

Compute scale factors for use in plotting the VOLTAGE data values within a 6-inch plotting area (i.e., the data pairs of TIME, VOLTAGE will plot within a 5 x 6-inch area).

60 CALL AXIS (0.0, 0.0, 20HTIME IN MILLISECONDS, -20, 5.0, 0.0, XARRAY (25), XARRAY (26))

Draw the TIME axis (5 inches long), using the scale factors computed in statement 40 to determine the milliseconds at each inch along the TIME axis.

70 CALL AXIS (0.0, 0.0, 7HVOLTAGE, 7, 6.0, 90.0, YARRAY (25), YARRAY (26))

Draw the VOLTAGE axis (6 inches long), using the scale factors computed in statement 50 to determine the voltage at each inch along the VOLTAGE axis.

80 CALL LINE (XARRAY, YARRAY, 24, 1, 2, 4)

Plot VOLTAGE vs TIME, drawing a line between each of the 24 scaled points and a symbol X at every other point.

90 CALL SYMBOL (0.5, 5.6, 0.21, 16HPERFORMANCE TEST, 0.0, 16)

Plot the first line of the graph title.

100 CALL SYMBOL (0.5, 5.2, 0.14, 16HREF. NO. 1623-46, 0.0, 16)

Plot the second line of the graph title.

SAMPLE PROGRAM (cont)

110 CALL PLOT (12.0, 0.0, 999)

Advance the pen beyond the current plotting area, write a terminating record, and close the plot output device.

120 STOP

Terminate program execution.

END

HOST COMPUTER BASIC SOFTWARE

CalComp's Host Computer Basic Software (HCBS) consists of a set of subroutines written in FORTRAN (and/or Assembly Language), which control elementary operations of the plotter and provide aids in plotting graphs. These subroutines are called by CalComp Applications Programs, CalComp Host Computer Functional Software, and user written programs. All output to the plotting system should go through the basic software package.

Although the subroutines included in the HCBS package are basically the same for all plotting system configurations, minor modifications or additional subroutines are provided in certain packages.

HCBS Subroutines

The subroutines included as part of the standard HCBS package are as follows:

PLOT Subroutine

The PLOT subroutine functions as the logical interface between the user's program and the plotting system. This subroutine actually produces commands for the plotter. The other support subroutines (SYMBOL, NUMBER, AXIS and LINE) eventually call the PLOT subroutine directly or indirectly. Four other functions closely associated with the PLOT operation are as follows:

PLOTS	Performs initialization.
FACTOR	Adjusts the overall size of a plot.
WHERE	Returns the current pen location.
NEWPEN	Selects pens.

These functions are programmed as separate entry points within the PLOT subroutine in some basic software packages. In other software packages, these functions are separate subroutines which call PLOT with a special value of IPEN to perform the necessary operation. In either case, the calling sequence and function is the same. The discussion in Section 3 assumes that they are entry points in the PLOT subroutine.

SYMBOL Subroutine

The SYMBOL subroutine produces plot annotation at any angle and in any size. There are two SYMBOL call formats:

- "Standard" call, used to draw text such as titles, captions, and legends.
- "Special" call, used to draw special centered symbols such as a box, octagon, triangle, etc., for plotting data points.

The standard characters available in SYMBOL include the letters A to Z, digits 0-9, and FORTRAN special characters:

blank	slash
equals	parenthesis (right and left)
plus	comma
minus	decimal point
asterisk	

Other characters are available in symbol routines for some host computer and plotting systems. The entire 64-character subset of ASCII will be available, whenever practical.

NUMBER Subroutine

NUMBER converts a floating-point number to the appropriate decimal equivalent so that the number may be plotted in the FORTRAN F-type format.

SCALE Subroutine

The user's program may accumulate plotting data in two arrays:

- An array of independent variables, X_i
- An array of dependent variables, $Y_i = F(X_i)$

It would be unusual if the range of values in each array corresponded exactly with the number of inches available in the actual plotting area. For some problems the range of data is predictable. The programmer can predetermine suitable conversion factors for use in drawing the axis scale values and plotting the data points on the graph. These factors are not usually known in advance.

The SCALE subroutine is used to examine the data values in an array. Both the starting value (minimum or maximum) and a scaling factor (positive or negative) must be determined as follows:

- The scale annotation drawn by the AXIS subroutine at each division will properly represent the range of real data values in the array.
- The data points will fit in a given plotting area, when plotted by the LINE subroutine.

These two values are computed and stored by SCALE at the end of the array.

AXIS Subroutine

Most graphs require axis lines and scales to indicate the orientation and values of the plotted data points. The most common type of scaled axis is produced by the AXIS subroutine which performs the following:

- Draws any length line at any angle.
- Divides the line into 1-inch segments (2-centimeter segments with metric software).
- Annotates the divisions with appropriate scale values.
- Labels the axis with a centered title.

When both the X and Y axes are needed, AXIS is called separately for each axis.

LINE Subroutine

The LINE subroutine produces a line plot of the pairs of data values in two arrays (X and Y). LINE computes the page coordinates of each plotted point according to the data values in each array and the respective scaling parameters. The data points may be represented by centered symbols and/or connecting lines between points.

The scaling parameters determined by the SCALE subroutine must immediately follow each array. If these parameters have not been computed by the SCALE subroutine, they must be supplied by the user.

Standard Calling Sequences

This list of calling sequences for CalComp Basic Software uses typical mnemonic names for the arguments. These arguments conform to standard FORTRAN conventions: if the first letter is I through N, then the argument must be an integer value; if it is any other letter, then the argument must be a real (floating point) value. The summary of arguments only provides gross definitions. Manual 1006, Programming CalComp Electromechanical Plotters, explains the full details of each subroutine and its arguments.

```
CALL PLOT (XPAGE,YPAGE,±IPEN,)
CALL PLOTS (IBUF,NLOC,LDEV)
CALL FACTOR (FACT)
CALL WHERE (RXPAGE,RYPAGE,RFACT)
CALL NEWPEN (IPEN)
```

Standard Calling Sequences (cont)

XPAGE,YPAGE are the X,Y coordinates of the terminal position of a pen movement, in inches, from the current origin.

IPEN specifies the pen up/down status during movement (up=3;down=2) and, if negative, establishes a new origin at the new position.

IBUF names a large output buffer area.

NLOC is the number of locations reserved for IBUF.

LDEV is the logical number of the plot output device.

FACT is a scale factor that determines the enlargement or reduction of the entire plot.

RXPAGE,RYPAGE.RFACT are the locations that will contain the current values of XPAGE, YPAGE, and FACT after WHERE is called.

IPEN is the number of the pen to be used.

```
CALL SYMBOL (XPAGE,YPAGE,HEIGHT,IBCD,ANGLE,
   +NCHAR)
CALL SYMBOL (XPAGE,YPAGE,HEIGHT,INTEQ,ANGLE,
   -ICODE)
CALL NUMBER (XPAGE,YPAGE,HEIGHT,FPN,ANGLE,
   ±NDEC)
```

XPAGE,YPAGE defined the relative origin of the character string (usually the lower left corner of the first character position).

HEIGHT is the height (and width), in inches, of a character position.

IBCD is the location of a character string.

ANGLE is the angle at which the string is to be rotated.

NCHAR is the number of characters in IBCD.

INTEQ is the integer equivalent of a special centered plotting symbol.

ICODE specifies the pen up/down status during movement to the relative origin.

FPN is a real (floating point) value.

NDEC specifies the number of decimal places to be printed.

```
CALL SCALE (ARRAY,AXLEN,NPTS,±INC)
CALL AXIS (XPAGE,YPAGE,IBCD,±NCHAR,AXLEN,
   ANGLE,FIRSTV,DELTAV)
CALL LINE (XARRAY,YARRAY,NPTS,INC,LINTYP,
   INTEQ)
```

ARRAY names an array of data values.

AXLEN is the length of the axis line.

NPTS is the number of entries in an array.

INC is the increment between entries in an array.

XPAGE,YPAGE is the relative origin of the axis line.

IBCD is the location of the alphanumeric axis title.

NCHAR is the number of characters in IBCD.

ANGLE is the angle of the axis line.

FIRSTV is the first scale value printed along the axis.

DELTAV is the increment between scale values on the axis.

XARRAY,YARRAY contain the pairs of data values to be plotted.

LINTYP specifies the type of line to be drawn through the data points.

INTEQ is the integer equivalent of a special centered plotting symbol.

Standard Basic Software

CalComp Standard Basic Software (online or offline) is available for all CalComp graphic plotting systems. It is described in more detail in Manual 1006, Programming CalComp Electromechanical Plotters, and consists of standard HCBS.

Standard Basic Software (Cont)

The Standard Graphic Controller Software which is used on CalComp controllers to process data generated by Standard HCBS, allows the operator to selectively plot portions of the data and control the various parameters of the plotting system. This may be augmented by various graphic controller functional software packages.

CAL EDIT Basic Software

CalComp CAL EDIT Basic Software is available only for offline graphic plotting systems driven with programmable controllers and is designed to give the plotter operator special editing controls.

The CAL EDIT software consists of Host Computer Basic Software (HCBS) and Graphic Controller Software (GCS) for creating and plotting data in a modified EIA RS-274 tape format. The system hardware includes the following:

- CalComp controller
- Input unit
- Communication console (teletype or equivalent)
- CalComp plotter

The Host Computer Basic Software consists of a set of FORTRAN subroutines that runs on the user's computer, and may be called by his applications programs to create plot data in a modified EIA RS-274 format. Currently existing plot programs (which call the standard CalComp basic software subroutines) may be rerun with these HCBS subroutines to produce a new plot tape, with little or no modification to the existing program. New features are provided with this HCBS which are not available with the Standard CalComp Basic Software.

The additional features include subroutines:

- To plot a smooth curve through an array of data points
- To draw a circle or circular arc
- To set scaling, rotation, and mirroring parameters
- To produce a formatted dump of the output tape
- To print messages to the plotter operator during plotting

CAL EDIT Basic Software (cont)

These features together with the Standard subroutines (see p. 3.05) are described in CAL EDIT USER'S GUIDE, Manual 1058.

The plot data is plotted using the Graphic Controller Software that operates on a CalComp Controller. The Graphic Controller Software provides for the following additional capabilities:

- Selective searching or plotting under teletype control
- Scaling, rotation, and mirror imaging of the plot
- Corrections and/or additions to be input directly from the console
- Windowing of the plotted data

The software has optional features which allow plots to be overlayed on previous plotted data. Plots larger than the plotter bed are also permitted.

Electronic Basic Software

CalComp Electronic Basic Software is available only for CalComp Graphic COM Systems. It is designed to take advantage of the specific features of COM Systems. The data format is compatible with standard basic software.

The Electronic HCBS differs from the Standard HCBS in that the PLOT subroutine and all its entries are replaced by a CALCMP subroutine.

The CALCMP subroutine performs the following functions:

- Initialization
- Moves beam with specified intensity
- Returns current beam position and intensity
- Advances film
- Sets plot origin at other than lower left corner of frame
- Sets standard intensity
- Sets scale factors
- Creates point plot

- Sets various increment sizes, rotation parameters
- Centers plot on frame
- Repeats frames

These features together with other features are described in Programming The CalComp Model 1675 COM System, Manual 1039.

The COM System is driven using the Graphic Controller Software that operates on a CalComp controller. The Electronic GCS provides for the following:

- Sets the magnification that is to be used
- Selects the camera type, step size, frame advance count, and the Absolute position option
- Selects intensities and centering values
- Performs scaling and windowing

SAMPLE PROGRAMS

The following paragraphs contain program listings and comments describing the Sample Program distributed with standard-inch HCBS (see Figures 3-2 through 3-7). Metric sample programs use different values.

Program Number 1

```
DIMENSION XARRAY (62), YARRAY (62)
```

Arrays are reserved for 60 abscissa and ordinate values. Space is reserved for the two additional cells required by SCALE and LINE.

```
CALL PLOTS (0, 0, 6)
```

Initialize the PLOT subroutine, giving the logical number for the output device (usually tape).

Program Number 1 (cont)

```
CALL PLOT (0.0,-.5,3)
```

Move pen to (0.,-.5) to ensure that Y=0 is 1/2-inch from edge of paper for AXIS. This will cause errors if closer than 1/2-inch from limit switch.

<u>Plot three graphs (Figures 3-2, 3-3, 3-4) illustrating SCALE, AXIS, and LINE</u>.

```
DELTAX=0.04
```

Initialize DELTA-X.

```
DO 110 I=1,3
```

Start loop to plot three complete graphs.

```
DELTAX=2.0*DELTAX
```

Set DELTA-X to 0.08, 0.16, 0.32, for three graphs.

```
XARRAY(1)=DELTAX
```

Initialize first element of X-ARRAY to DELTA-X.

```
DO 105 J=1,60
```

Loop to evaluate 60 points.

```
YARRAY(J)=XARRAY(J)**2-0.7*XARRAY(J)
**3+0.1*XARRAY(J)**4
```

Evaluate equation and place results in Y-ARRAY.

```
105 XARRAY(J+1)=XARRAY(J)+DELTAX
```

Set the next element in X-ARRAY to current one plus DELTA-X and end loop. (XARRAY (61) is filled but not used.)

```
CALL SCALE (XARRAY(1), 6.5,60,1)
```

Use SCALE subroutine to scale X-ARRAY over 6.5 inches with 60 points in a single array. The SCALE subroutine places an adjusted minimum in XARRAY(61) and an adjusted delta in XARRAY(62).

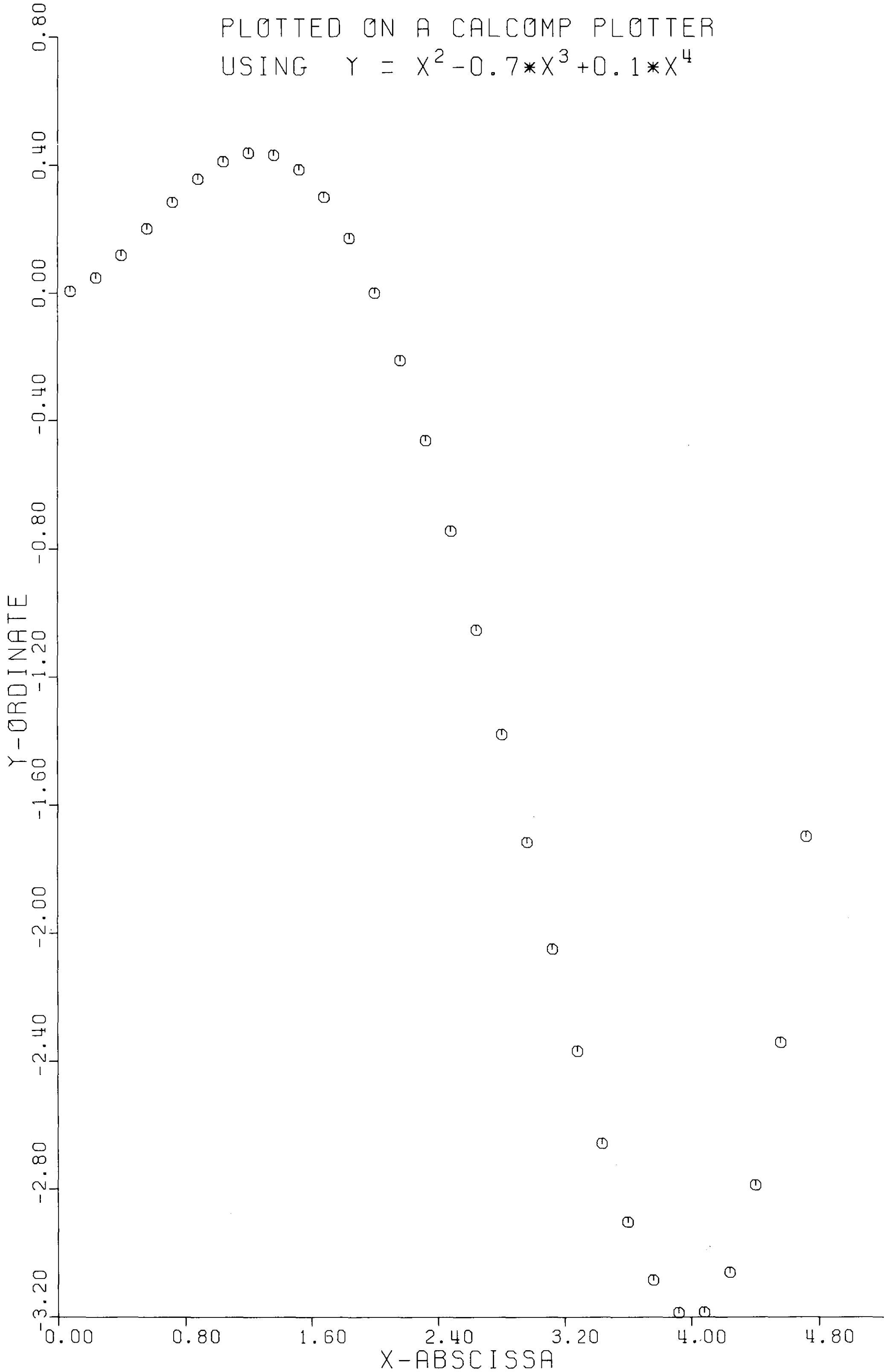


Figure 3-2. Sample Plot (LINTYP = -2)

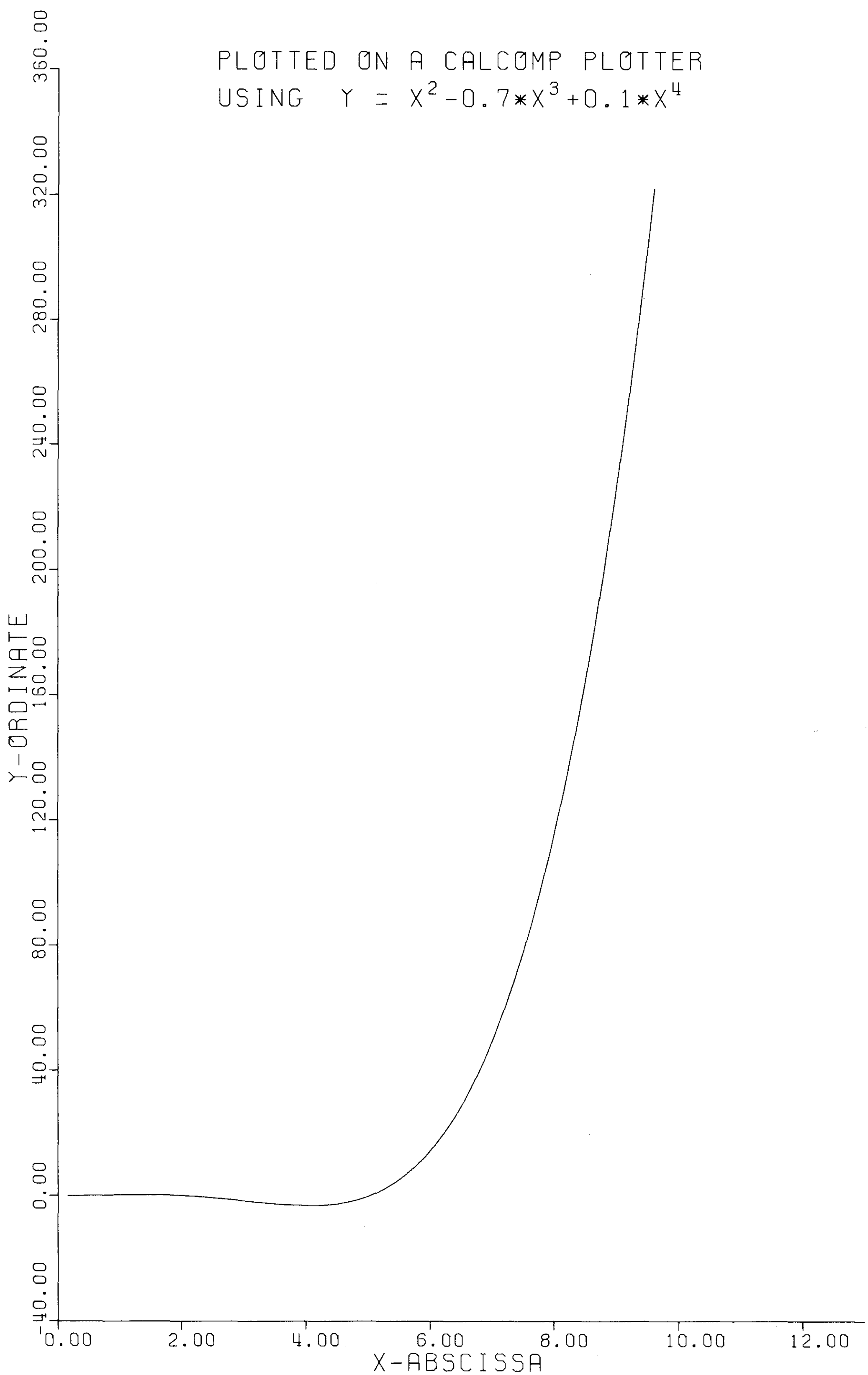


Figure 3-3. Sample Plot (LINTYP = 0)

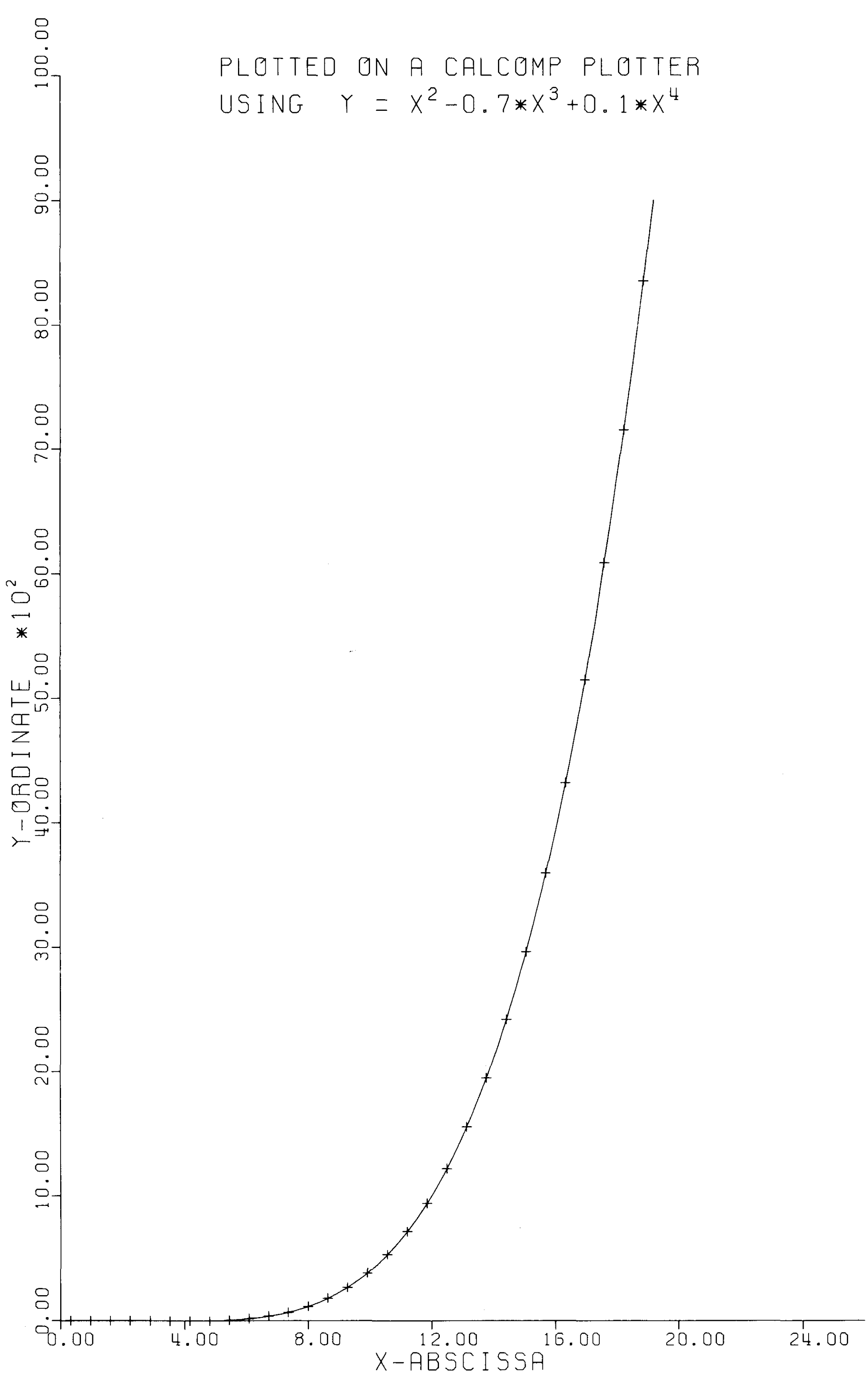


Figure 3-4. Sample Plot (LINTYP = +2)

Program Number 1 (cont)

```
CALL SCALE (YARRAY(1),10.0,60,1)
```

Use SCALE subroutine for Y-ARRAY over 10 inches.

```
CALL AXIS (0.,0.,10HX-ABSCISSA,-10,6.5,0.,
XARRAY(61),XARRAY(62))
```

Starting at (0,0), plot the X-axis (abscissa) 6.5 inches long, at 0 degrees, placing the annotation on the clockwise side of the axis, and using the values calculated by the SCALE subroutine to label the tick marks. Since the axis (including annotation) is 1/2-inch wide, the starting point (0.0) must be 1/2-inch from the edge of the paper (see CALL PLOT (0.0,-0.5,3) above).

```
CALL AXIS (0.,0.,10HY-ORDINATE,10,10.,90.,
YARRAY(61),YARRAY(62))
```

Plot a similar Y-axis, 10 inches long, at 90 degrees, with annotation on the counter-clockwise side.

> NOTE
>
> If the minimums calculated by the SCALE subroutine are used, the X-axis must start at (0,Y) and the Y-axis at (X,0).

```
CALL LINE (XARRAY(1), YARRAY(1), 60,1,
2*(I-2),I)
```

Plot the 60 data points from the X and Y arrays using different symbols and types of lines. The LINE subroutine uses the scale factors added to the arrays by the SCALE subroutine.

```
CALL SYMBOL (1.3,10.,.14,28HPLOTTED ON A
CALCOMP PLOTTER,0.,28)

CALL SYMBOL (1.3,9.7,.14,28HUSING  Y =
X -0.7*X +0.1*X  ,0.,28)

CALL NUMBER (2.98,9.8,.1,2.0,0.,-1)

CALL NUMBER (3.96,9.8,.1,3.0,0.,-1)
```

Program Number 1 (cont)

CALL NUMBER (4.94,9.8,.1,4.0,0.,-1)

Plot a title above each graph. The characters are 0.14 inches high at 0 degrees. The first line of 28 characters starts at (1.3,10.), centered over 6.5 inches. The second line of 28 starts at (1.3,9.7), 0.3 inches below the first.

Three calls to NUMBER are used to plot the exponents for the equation in the title.

110 CALL PLOT (10.0,0.0,-3)

Move carriage, with pen up, to (10,0), reset origin for the next graph, and end loop.

<u>Plot Angular Letter Test (Figure 3-5)</u>.

CALL PLOT (4.5,5.5,-3)

Move to (4.5 5.5) and reset origin at center of angular letter test.

ANGLE=0.0

Initialize ANGLE (in degrees) to 0.

HEIGHT=0.105

Initialize HEIGHT (size of annotation) to 0.105.

DO 120 I=1,8

Loop to plot eight sentences at different angles with increasing size.

RAD=0.0174533*ANGLE

Convert ANGLE to radians for sine and cosine routines.

XX=0.5*COS (RAD)

YY=0.5*SIN (PAD)

(XX,YY) is the starting point of the sentence, 0.5 inches from the origin.

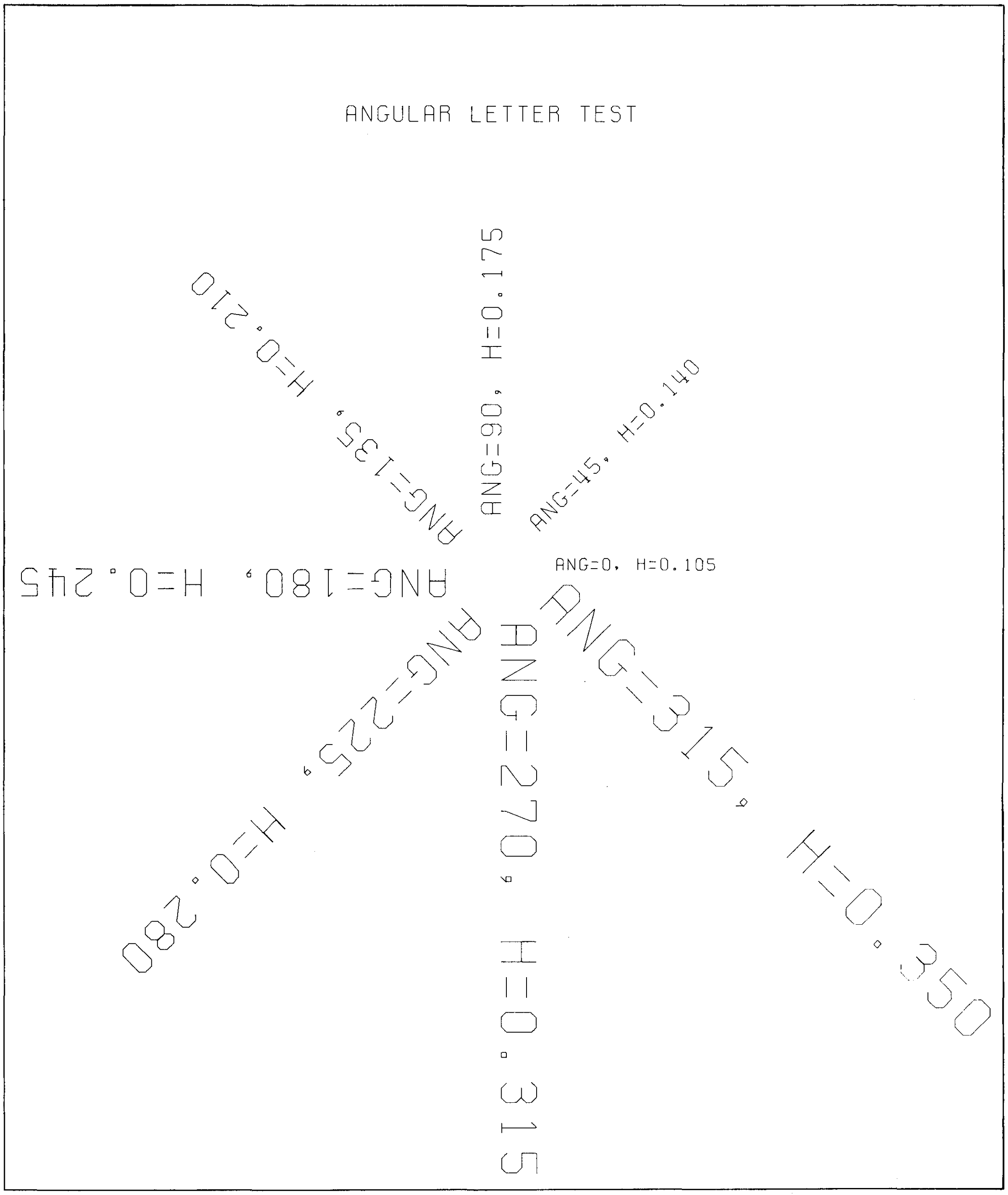


Figure 3-5. Sample Plot, Angular Letter Test

Program Number 1 (cont)

```
CALL SYMBOL (XX,YY,HEIGHT,4HANG=ANGLE,4)
```

Using the SYMBOL subroutine, plot the beginning of the sentence.

```
CALL NUMBER (999.0,999.0.HEIGHT,ANGLE,
ANGLE,-1)
```

Use the NUMBER subroutine to plot the value of ANGLE following ANG= by using the special values of 999.0 for X and Y.

```
CALL SYMBOL (999.0,999.0,HEIGHT,4H, H=,
ANGLE,4)
```

```
CALL NUMBER (999.0,999.0,HEIGHT, HEIGHT,
ANGLE,3)
```

Plot the value of HEIGHT, using the same technique.

```
HEIGHT=HEIGHT+0.035
```

Increase HEIGHT by 0.035 inches.

```
120 ANGLE=ANGLE+45.0
```

Increase ANGLE by 45 degrees and end loop.

```
CALL SYMBOL (-1.4,4.0,0.14,19HANGULAR
LETTER TEST,0.0,19)
```

Plot the title, centered above the test.

```
CALL PLOT ( 4.5,5.0,3)

CALL PLOT (-4.5,5.0,2)

CALL PLOT (-4.5,-5.5,2)

CALL PLOT ( 4.5,-5.5,2)

CALL PLOT ( 4.5, 5.0, 2)
```

Draw a 9 by 10.5 inch rectangle around the angular letter test.

```
CALL PLOT (6.5,-5.5,-3)
```

Move to (6.5,-5.5) and reset origin for next plot.

Program Number 1 (cont)

Plot Car Value Chart (Figure 3-6) without using SCALE, AXIS, or LINE.

```
X=1.0
```

Initialize X (car age in years and location of tick mark) to 1.0.

```
DO 130 I=1,7
```

Loop to plot a 7-inch axis with tick marks and annotation at every inch.

```
CALL PLOT (X-1.0,0.0,3)
```

Move carriage, with pen up, to last tick mark.

```
CALL PLOT (X  ,0.0,2)
```

Move carriage with pen down, to next tick mark.

```
CALL PLOT (X  ,-0.1,2)
```

Draw tick mark.

```
CALL NUMBER (X-.02,-0.25,0.1,X,0.0,-1)
```

Plot the value of X (car age in years) below tick mark.

```
130 X=X+1.0
```

Increase X by 1.0 and end loop.

```
CALL SYMBOL (2.0,-0.5,0.14,21HCAR MODEL AGE
(YEARS),0.0,21)
```

Plot axis title below the axis.

```
VALUE=1000.0
```

Initialize VALUE (in dollars) to 1000.0.

```
DO 140 I=1,6
```

Loop to plot a 9-inch axis with tick marks every 0.75 inches and annotation at every other tick mark.

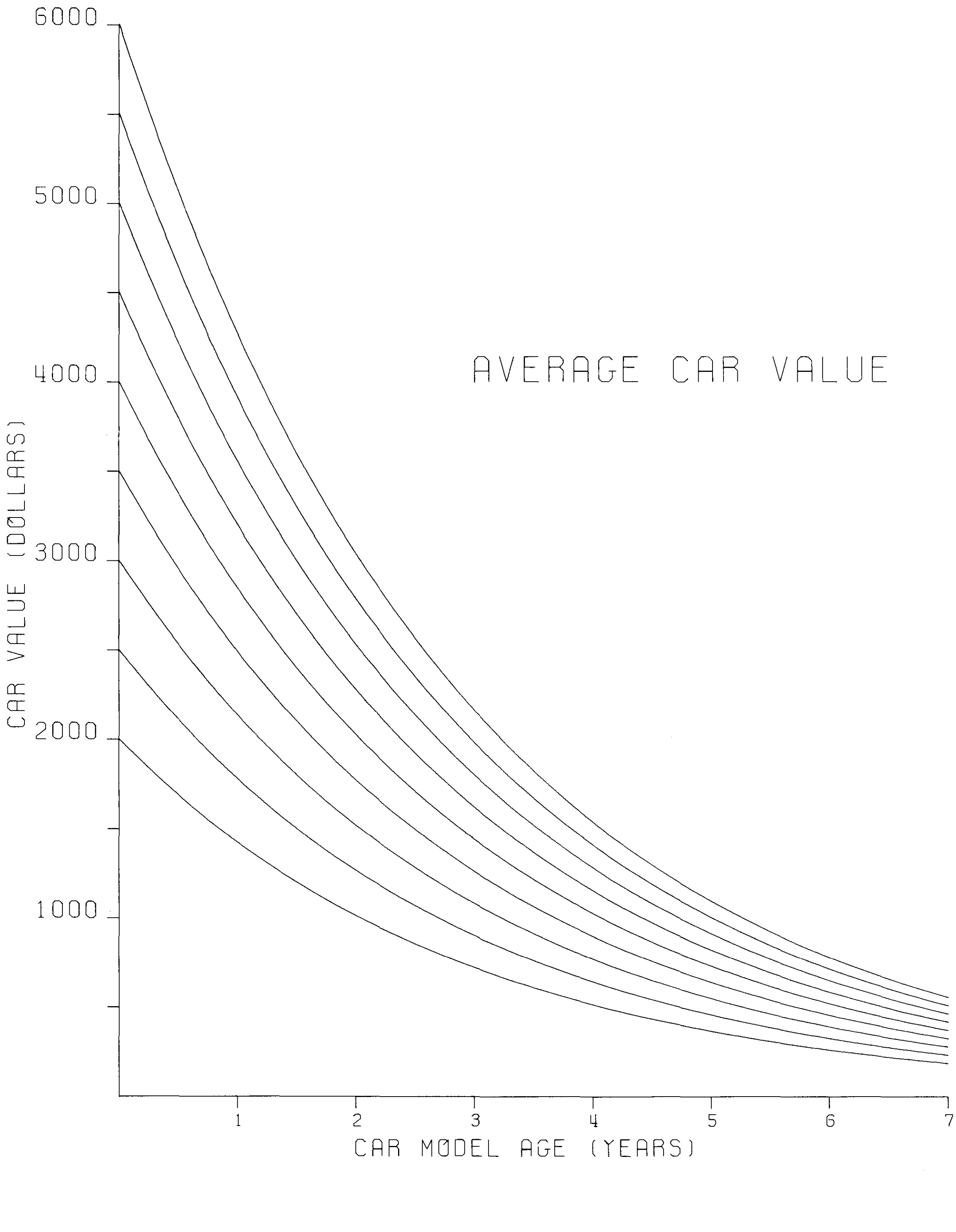


Figure 3-6. Sample Plot, Average Car Value

Program Number 1 (cont)

```
    Y=0.0015*VALUE

    Set Y (location of tick mark) to 1.5*VALUE/
    1000.

    CALL PLOT (0.0,Y-1.5,3)

    Move carriage, with pen up, to last tick
    mark.

    CALL PLOT (0.0,Y-.75,2)

    Move carriage, with pen down, to inter-
    mediate tick mark.

    CALL PLOT (-.1,Y-.75,2)

    CALL PLOT (0.0,Y-.75,2)

    Draw tick mark.

    CALL PLOT (0.0,Y   ,2)

    Move carriage, with pen down, to next
    tick mark.

    CALL PLOT (-.1,Y   ,2)

    Draw tick mark.

    CALL NUMBER (-0.7,Y,0 14, VALUE,0.0,-1)

    Plot VALUE to the left of tick mark.

140 VALUE=VALUE+1000.0

    Increase VALUE by 1000 dollars.

    CALL SYMBOL (-0.8,3.1,0.14,19HCAP VALUE
    (DOLLARS),90.0,19)

    Plot the axis title alongside the axis.

    DO 150 I,2000,6000,500

    Start loop to plot nine curves with values
    from 2000 to 6000 dollars.

    VALUE=I

    Initialize VALUE for each curve.
```

Program Number 1 (cont)

```
    AGE=0.0
```

Initialize AGE to 0 years.

```
    CALL PLOT (AGE,0.0015*VALUE,3)
```

Move carriage, with pen up, to start of curve.

```
    DO 150 J=1,84
```

Loop to plot 84 points along curve.

```
    VALUE=VALUE*0.972
```

Decrease VALUE by 2.8 percent.

```
    AGE=AGE+0.08333
```

Increase AGE by 1 month (0.0833 years).

```
150 CALL PLOT (AGE,0.0015*VALUE,2)
```

Draw line to new VALUE and end loops.

```
    CALL SYMBOL (3.0,6.0,0.21,17HAVERAGE CAR
    VALUE,0.0,17)
```

Plot graph title.

```
    CALL PLOT (9.0,0.0,-3)
```

Move pen to (9.,0) and reset origin for next plot.

```
    CALL PLOT (0.0,0.0,999)
    STOP
```

Call PLOT with IPEN=999 to end plotting and close output device.

```
    END
```

Program Number 2

CALL PLOTS (0,0,6)

Initialize the PLOT subroutine.

CALL PLOT (0.0,-0.5,3)

Move to (0,-.5) to ensure that origin is 1/2-inch from edge of paper.

Plot graph with multiple axes and lines (Figure 3-7).

X=0.0

Initialize X to 0 for first axis.

DO 210 I=1,10

Loop to plot a 10-inch axis with tick marks and annotation at every inch.

CALL PLOT (X,0.0,3)

Move carriage, with pen up, to start of axis segment.

X=X+1.0

Increase X by 1 inch.

CALL PLOT (X,0.0,2)

CALL PLOT (X,-.1,2)

Draw tick mark.

CALL NUMBER (X,-0.25,0.1,5.0*X,0.0,-1)

Plot values of WIDTH (5*X) under tick mark, and end loop.

CALL SYMBOL (4.0,-0.40,0.12,1,0.0,-1)

Plot Special Symbol 1 that will be used on lines corresponding to this axis.

CALL SYMBOL (4.2,-0.45,0.14,10HWIDTH (FT),0.0,10)

Plot the axis title below the axis.

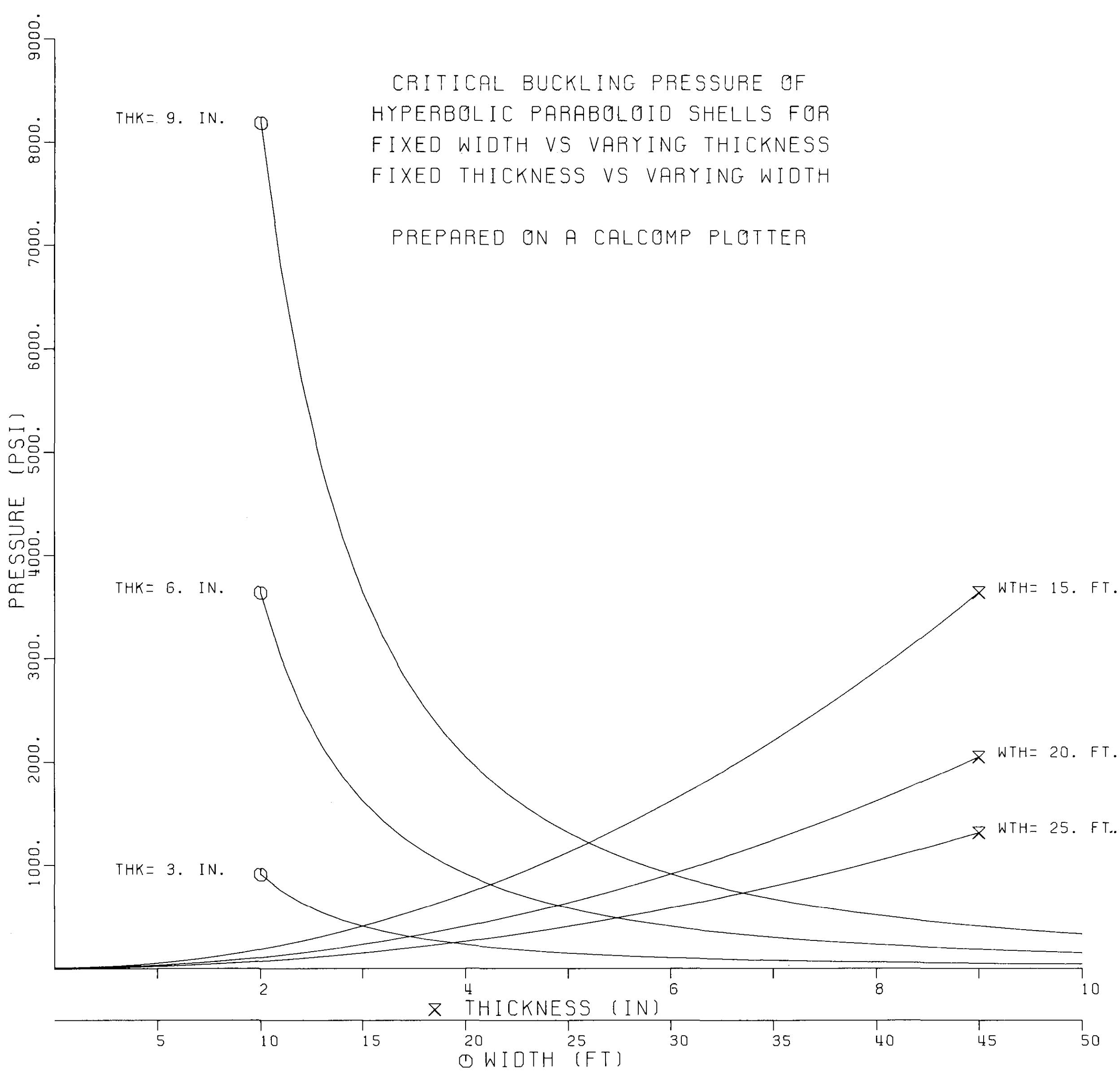


Figure 3-7. Sample Plot, Multiple Axes and Lines

Program Number 2 (cont)

```
CALL PLOT (0.0,0.5,-3)
```

Move pen to (0.0,.5), and reset origin for second axis.

```
X=0.0
```

Initialize X to 0 for second axis.

```
DO 220 I=1,5
```

Loop to plot a 10-inch axis with tick marks at every inch and annotation at every other inch.

```
CALL PLOT (X,0.0,3)
```

Move carriage, with pen up, to start of axis segment.

```
X = X + 1.0
```

Increase X by 1 inch.

```
CALL PLOT (X,0.0,2)

CALL PLOT (X,-.1,2)

CALL PLOT (X,0.0,2)
```

Draw intermediate tick mark.

```
X = X + 1.0
```

Increase X again

```
CALL PLOT (X,0.0,2)

CALL PLOT (X,-.1,2)
```

Draw another tick mark.

```
220 CALL NUMBER (X,-0.25,0.1,X,0.0,-1)
```

Plot values of THICKNESS (X) under second tick mark, and end loop.

```
CALL SYMBOL (3.7,-0.40,0.12,7,0.0,-1)
```

Plot Special Symbol 7 that will be used on lines corresponding to this axis.

```
CALL SYMBOL (4.0,-0.45,0.14,14HTHICKNESS
(IN), 0.0,14)
```

Program Number 2 (cont)

Plot the axis title below the axis.

```
Y = 0.0
```

Initialize Y to 0 for vertical axis.

```
DO 230 I=1,9
```

Loop to plot a 9-inch axis with tick marks and annotation at every inch.

```
CALL PLOT (0.0,Y,3)
```

Move carriage, with pen up, to start of axis segment.

```
Y = Y + 1.0
```

Increase Y by 1 inch.

```
CALL PLOT (0.0,Y,2)

CALL PLOT (-.1,Y,2)
```

Draw tick mark.

```
230 CALL NUMBER (-.15,Y-.2,0.1,1000.*Y,90.0,0)
```

Plot values of pressure (1000*Y) to the left of tick marks, and end loop.

```
CALL SYMBOL (-0.30,3.5,0.14,14HPRESSURE (PSI),90.0,14)
```

Plot the axis title alongside the axis.

```
THICK = 3.0
```

Initialize THICKNESS to 3 inches for fixed-thickness curves.

```
WIDTH = 25.0
```

Initialize WIDTH to 25 feet for fixed-width curves.

```
DO 260 I=1,3
```

Loop to plot three sets of curves.

```
TSQR = THICK*THICK
```

Initialize TSQR to THICKNESS squared.

```
WSQR = WIDTH*WIDTH
```

Initialize WSQR to WIDTH squared.

Program Number 2 (cont)

```
PSI = 100.99*TSQR
```

The equation for pressure (PSI) is 10099*
(THICKNESS**2)/(WIDTH**2). Calculate
initial PSI with WIDTH=10 feet for fixed-
thickness curve.

```
CALL SYMBOL (0.6,PSI/1000.0,0.1,5HTHK=,0.0,
5)

CALL NUMBER (999.0,999.0,0.10,THICK,0.0,0)

CALL SYMBOL (999.0,999.0,0.10,4H IN.,0.0,4)
```

Plot identification for fixed-thickness
curves, using the special X and Y values of
999.0 in symbol.

```
CALL SYMBOL (2.0,999.0,0.12,1,0.0,-1)
```

Plot the first point on the curve at WIDTH =
10. with Special Symbol 1.

```
DO 240 J=10,50
```

Loop to plot 40 points along curve from 10
to 50.

```
WX = J
```

Convert WIDTH to floating point (WX).

```
PSI=10099.0*TSQR/(WX*WX)
```

Calculate PSI using WIDTH (WX).

```
240 CALL PLOT (WX/5.0,PSI/1000.0,2)
```

Plot PSI using WIDTH, and end loop.

```
PSI = 10099.0*81.0/WSQR
```

Calculate initial PSI with THICKNESS=9
inches for fixed-width curve.

```
CALL SYMBOL (9.2,PSI/1000.0,0.1,5HWTH =,
0.0,5)

CALL NUMBER (999.0,999.0,0.10,WIDTH,0.0,0)
```

Program Number 2 (cont)

```
    CALL SYMBOL (999.0,999.0,0.10,4H FT.,0.0,
    4)
```

Plot identification for fixed-width curves.

```
    CALL SYMBOL (9.0,999.0,0.12,7,0.0,-1)
```

Plot the first point on the curve at THICKNESS = 9 inches with Special Symbol 7.

```
    DO 250 J=5,50
```

Loop to plot 45 points along curve from 9 to 0.5. (Curve is plotted backwards.)

```
    TX=J

    TX=(50.0-TX)/5.0
```

Convert THICKNESS to floating point and change to correct units (TX=(N-J)/5).

```
    PSI=10099.0*TX*TX/WSQR
```

Calculate PSI using THICKNESS (TX).

```
250 CALL PLOT (TX,PSI/1000.0,2)
```

Plot PSI versus THICKNESS, and end loop.

```
    THICK=THICK+3.0
```

Increase constant THICKNESS by 3 inches.

```
260 WIDTH=WIDTH-5.0
```

Decrease constant WIDTH by 5 feet, and end loop.

```
    CALL SYMBOL (3.3,8.5,.14,29HCRITICAL
    BUCKLING PRESSURE OF,0.,29)

    CALL SYMBOL (3.1,8.2,.14,32HHYPERBOLIC
    PARABOLOID SHELLS FOR,0.,32)

    CALL SYMBOL (3.1,7.9,.14,32HFIXED WIDTH
    VS VARYING THICKNESS,0.,32)
```

Program Number 2 (cont)

```
CALL SYMBOL (3.3,7.0,.14,29HPREPARED ON
A CALCOMP PLOTTER,0.,29)
```

Plot title for graph. Title has five lines, each centered over 11 inches.

```
CALL PLOT (12.0,-0.5,-3)
```

Move carriage, with pen up, to (12.,-.5) and reset origin for next plot.

```
CALL PLOT (0.0,0.0,999)

STOP
```

CALL PLOT with IPEN=999 to end plotting and close output device.

```
END
```

ACCESSORY SUPPORT

CalComp's accessories for their graphic systems allow a wider use of graphics which enables the user to perform his task easier and quicker. Most of the accessory functions are covered by the Basic Software and require no additional features or additional software. Examples of hardware that require additional software are:

- Model 942 Digitizer

Basic software for both of these items is described in the following paragraphs:

MODEL 942 DIGITIZER

The CalComp Model 942 Digitizer is a free cursor digitizer that writes coordinate data to either punched cards or paper tape. Data is written when one of three data select buttons on the cursor is pressed. CalComp Basic Software converts this raw coordinate data into coordinates transformed to a user coordinate system. This software recognizes menu accesses, transforms them into easily usable data, and performs error-checking on all input data.

Some features found on other digitizers are absent on the CalComp 942 Digitizer. The following features are implemented within the CalComp Basic Software:

- Coordinate rounding
- Scaling and rotation
- Alphanumeric keyboard
- Special function keys

The result is a saving in hardware complexity and a versatility unmatched by other digitizers with comparable hardware features.

The following paragraphs give a detailed description of the CalComp 942 Digitizer Basic Software. This software comprises five subroutines: DINIT, WAREA, MENU, GRID, and GETPT. Any of these subroutines can be called by the user's FORTRAN program, although some of these subroutines call upon each other for certain operations. The number of these subroutines used for a particular application depends on the user's requirements. The five subroutines for the 942 Digitizer Software are as follows:

DINIT — establishes the logical device number from which the raw digitizer coordinates are to be read and the user coordinate system to be used.

WAREA — establishes a certain area of the digitizer as a working area and causes all digitized points outside of the area to be flagged.

MENU — establishes the location on the digitizer surface and the number of rows and columns of a rectangular menu area.

GRID — allows the user to select the resolution at which user coordinates will be generated. The resolution may be varied independently in either axis.

GETPT — processes the raw coordinates read from the input device and returns to the calling program either a menu row and column number or a coordinate that has been transformed into the user coordinate system. The calling program is also informed which data button was depressed to digitize the point, and whether a sequence field error was detected.

942 Digitizer Characteristics

The user should be familiar with some of the characteristics of the CalComp 942 Digitizer, in order to understand the functions performed by the Basic Software. He should also be acquainted with certain conventions that are built into the Basic Software.

The digitizer coordinate system is oriented with the X-axis running parallel to the front edge of the table with X values increasing as one moves to the right. The Y-axis is parallel to the left edge of the table and Y values increase with movement toward the back edge of the table. Coordinates are displayed and punched in inches.

The resolution of the digitizer is 0.001 inches. This is the smallest distance that the digitizer can measure. The accuracy of the digitizer is 0.005 inches. This is the maximum error in either axis between a point's true coordinate value and its displayed coordinate value.

All digitizing should take place within the rectangle defined by the four black dots near each corner of the table, to ensure maximum accuracy.

942 Operating Procedure

The origin of the digitizer coordinate system may be placed anywhere on the table by moving the cursor to the desired origin position and pressing the red button on the cursor. This zeros the coordinate registers. The origin should always be set at the lower left corner of the table. A bracket is mounted at the lower left corner against which the cursor may be placed for the purpose of origining.

The identical origin should be reestablished using the position bracket if any of the following occurs:

- If the cursor is accidentally lifted or removed from the table surface
- If power is removed from the digitizer electronics
- If the user wishes to resume digitizing after an interruption

This ensures that subsequent coordinates are consistent with those coordinates previously digitized.

When resuming digitizing after an interruption, the sequence counter should be restored to the value punched on the last record output prior to the interruption. This may be accomplished by pressing the yellow button on the cursor. Holding the yellow button down for more than one second causes the sequence counter to advance rapidly. Advance the counter until it is five or ten less than the desired number. This will be shown on the display console at the back of the digitizer table. Then advance the counter to its desired value by quickly pressing the yellow button the required number of times.

The white, blue, and green buttons on the cursor are data buttons and are identified as A, B, and C, respectively (on the digitizer output medium). Pressing one of these three buttons causes the coordinate registers (as shown on the display console) to be recorded on the output device.

A blank in any one of the button identification fields, signals the end of data on a card. The Basic Software will ignore the remainder of the card and continue processing on the next card. Thus, digitizing may be terminated or interrupted at any point, (even if a card has not been completely filled), by ejecting the card from the punch station. Digitizing may be resumed starting on a new card. The Basic Software recognizes a card with 9999 punched in Columns 1-4, as the end of data.

ADDITIONAL PROGRAMMING TECHNIQUES

Some of the unusual and more sophisticated techniques of using the Basic Software subroutines have been developed through experience and need by CalComp programmers and users. These techniques are documented here for the benefit of users who may have similar problems.

Uses of INC in SCALE and LINE

Since the SCALE subroutine is used in conjunction with the LINE subroutine, both subroutines are discussed together to emphasize their close relationship. INC (the fourth argument in the calling sequence) refers to the subscripting increment between significant entries in the array. Generally, this value is one, but there are instances where the value is greater than one. If two or more arrays of data are intermixed, INC equals the number of arrays.

EXAMPLE: A DIMENSION statement may define a single array for convenience in inputting the data, but since the area actually contains two separate arrays, (X and Y), of 10 entries each, INC equals 2.

The statement

```
DIMENSION AXY(24)
```

reserves 20 words for the data and 4 words for the scaling parameters to be computed by SCALE (2 words for each array). Then, the statements

```
CALL SCALE (AXY(1),XAXLEN,10,2)

CALL SCALE (AXY(2),YAXLEN,10,2)
```

compute the scaling parameters. Notice that AXY(1) is the location of the first X-array entry and AXY(2) is the location of the first Y-array entry. Since the value of INC is 2, every other entry in the array AXY is scanned, starting with the first one named, either AXY(1) or AXY(2). The resulting array appears as follows:

```
AXY(1) = X1, AXY(2) = Y1, AXY(3) = X2, AXY(4) = Y2, . . .
AXY(19) = X10, AXY(20) = Y10, AXY(21) = XFIRSTV,
AXY(22) = YFIRSTV, AXY(23) = XDELTAV, AXY(24) = YDELTAV
```

Uses of INC in SCALE and LINE (cont)

Then, to draw the line, the statement

```
CALL LINE (AXY(1),AXY(2),10,2,LINTYP,
INTEQ)
```

specifies the beginning of each data array, and INC=2.

A similar situation arises when the array variables are defined as double-precision. Since the SCALE and LINE subroutines assume single-precision data, the user must change the value of INC to compensate.

EXAMPLE: DOUBLE PRECISION XDP(12)

Normally reserves 24 words for the array, but there are only ten entries as far as SCALE and LINE are concerned, plus two words for the computed scaling parameters. Therefore, INC should be given a value of 2 so that only the high-order part of each double-precision entry will be used by SCALE and LINE.

Shortcut Scaling

When the array contains data values arranged in the same order as they would appear on an axis (from minimum to maximum or vice-versa), it is not necessary to scan all of the entries in order to compute scaling parameters FIRSTV and DELTAV. The first value can be positioned next to the last value since the locations of the minimum and maximum values are at the extremes of the array. This, in effect, makes a new array of only two values.

EXAMPLE: Assume an X array (XA) of 50 entries arranged sequentially from a minimum value to a maximum value. Then,

```
DIMENSION XA(53)

XA(51)=XA(1)

CALL SCALE (XA(50), AXLEN, 2, 1)
```

will compute and store the scaling parameters in XA(52) and XA(53). These parameters can be referenced

Shortcut Scaling (cont)

directly when calling AXIS, but they must be shifted back one position in the array before calling LINE. Thus,

```
XA(51)=XA(52)    This is FIRSTV.

XA(52)=XA(53)    This is DELTAV.
```

This arrangement results in much faster execution of the SCALE subroutine. If the limits of the array values are known constants, FIRSTV and DELTAV can be predetermined by the programmer and stored as constants, without calling SCALE.

Forcing SCALE to Choose Zero

If all of the values in an array are positive, the user can force SCALE to choose zero for FIRSTV (the adjusted minimum) by including a zero in the array. If the zero entry is not to be plotted by LINE, it should be the first entry in the array and omitted when calling LINE.

EXAMPLE: Assume 20 points to be plotted from arrays XA and YA. All values in XA are positive, (not necessarily zero) and the X-axis scale must start with 0.00. The partial coding would be:

```
DIMENSION XA(23), YA(22)
```

Then compute values for XA(2) through XA(21) and YA(1) through YA(20).

```
XA(1)=0.0

CALL SCALE (XA,XAXLEN,21,1)
```

The zero value will be chosen for FIRSTV.

NOTE

NPTS=21. NPTS is the third argument in the calling sequence.

```
CALL SCALE (YA,YAXLEN,20,1)
```

Observe that NPTS now equals 20.

Forcing SCALE to Choose Zero (cont)

```
CALL LINE (XA(2),YA,20,1,LINTYP,INTEQ)
```

The zero value in XA(1) is excluded from the line plot.

This technique of forcing a constant value for FIRSTV works only for zero. A FIRSTV value based on a user-selected minimum other than zero may be changed by SCALE to a lower value to accommodate the adjusted DELTAV.

> EXAMPLE: If the user inserts a minimum value of 100 in the array (with DELTAV now 8), SCALE will set FIRSTV equal to 96 instead of 100.

Scaling for Multiple Curves

Drawing multiple curves on one graph requires special programming so that all curves are scaled the same with respect to one of the axes. A common example consists of two curves with a common X-axis and two separate Y-axes.

To ensure that both curves will fit on the X-axis, the X arrays of both curves must be scaled using (SCALE), and scaling parameters that will accommodate the widest range must be used by AXIS and LINE.

Assume that curves A and B (see Figure 3-8) are represented by arrays (XA,YA) and (XB,YB), and assume that the X and Y arrays of each curve have the same number of points. After all of the arrays have been examined by SCALE to compute the scaling parameters, replace the smaller parameters with the larger parameters. Assume that the scaling parameters of XA are:

AFIRST, ADELTA

Assume that the parameters of XB are:

BFIRST, BDELTA

Scaling for Multiple Curves (cont)

The amount of coding required to choose the best scaling parameters depends on some knowledge of the array values. If one of the X arrays is a subset of the other array (e.g., if the range of A is contained within the range of B), the coding logic is simple. If each array extends in opposite directions, the coding logic becomes complex.

In the simple case, the coding needs only to determine the larger range and replace the scaling parameters of the smaller range with those of the larger range.

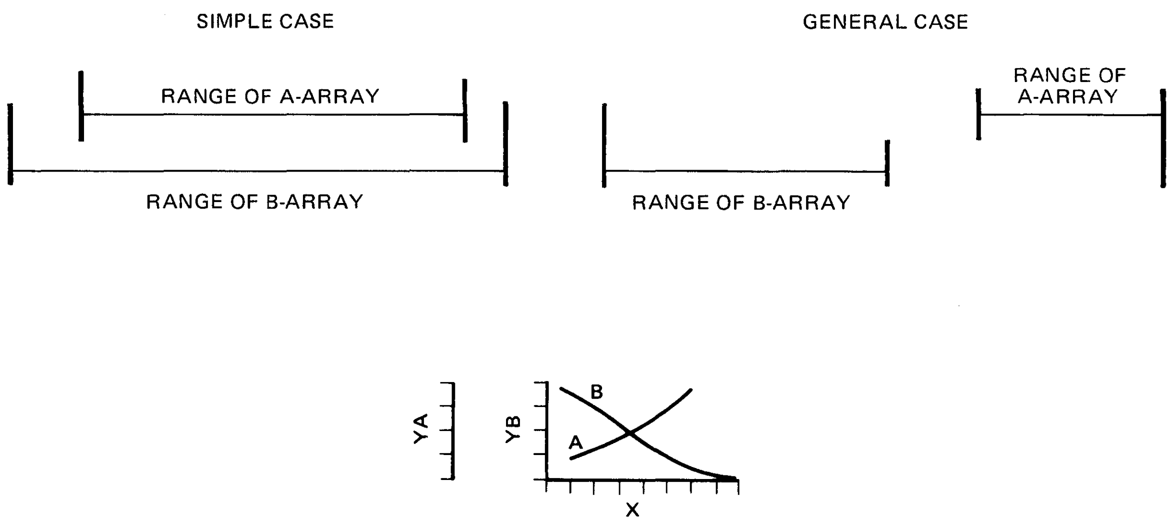


Figure 3-8. Scaling for Multiple Curves

```
      IF (ADELTA - BDELTA) 10,30,20
   10 AFIRST = BFIRST
      ADELTA = BDELTA
      GO TO 30
   20 BFIRST = AFIRST
      BDELTA = ADELTA
   30 CONTINUE
```

In the complex case, it cannot be assumed that the scaling parameters for either of the arrays are suitable for plotting both arrays together. A new set of scaling parameters must be computed which are a composite of the original. There are several ways of obtaining these parameters which are described as follows:

Method 1. It is important in each of the methods to find the minimum and maximum of all values in both arrays. Then, SCALE must determine the scaling parameters that will work for both arrays when they are plotted on a common axis. In Method 1, the two arrays are stored in consecutive locations so that SCALE can handle the arrays as if they were a single array. There must be a provision for inserting the two scaling parameters at the end of each array so LINE can function properly.

EXAMPLE: Assume two arrays, with 30 entries in array A and 40 entries in array B.

```
DIMENSION A(74), B(42)
EQUIVALENCE (A(33), B(1))
```

The arrays are now assigned consecutive locations with space for four scaling parameters.

```
A(31) = A(30)
A(32) = A(30)
```

The two locations reserved for A's scaling parameters are filled with dummy values. SCALE can process both arrays as one continuous array.

```
CALL SCALE (A,AXLEN,72,1)
```

The scaling parameters that apply for both arrays are stored following array B.

```
A(31) = B(41)

A(32) = B(42)
```

The common parameters are also stored following array A (overlaying the dummy values), and the arrays are then ready for AXIS and LINE.

Method 2. Methods 2 and 3 require the use of a third array composed of the minimum and maximum values of each data array. This composite array is then used by SCALE to compute the effective scaling parameters. Method 2 assumes that all entries in each array are in ascending order by their values, so that their minimum and maximum values can be found in the first and last locations of each array. These four values are placed in the third array, which is then examined by SCALE. The resulting scaling parameters are then moved to their respective positions following the data arrays. The coding is as follows, assuming the same arrays used in Method 1:

```
DIMENSION A(32), B(42), C(6)

C(1) = A(1)

C(2) = A(30)

C(3) = B(1)

C(4) = B(40)
```

The minimum and maximum values of each array are placed in the dummy area.

```
CALL SCALE (C, AXLEN, 4,1)

A(31) = C(5)

A(32) = C(6)

B(41) = C(5)

B(42) = C(6)
```

The common scaling parameters are moved, and the arrays are now ready for AXIS and LINE.

Method 3. If the minimum and maximum values of each array are not easily found, each array can be scaled separately and the scaling parameters can be used to compute the maximum values. The minimum and maximums are then stored in a third array so that SCALE can compute the common scaling parameters. The coding is as follows for arrays A and B:

```
DIMENSION A(32), B(42), C(6)

CALL SCALE (A, AXLEN, 30,1)

CALL SCALE (B, AXLEN, 40,1)
```

From the scaling parameters, the maximums can be computed.

```
C(1) = A(31)

C(2) = A(31) + A(32)*AXLEN

C(3) = B(41)

C(4) = B(41) + B(42)*AXLEN

CALL SCALE (C, AXLEN, 4, 1)

A(31) = C(5)

A(32) = C(6)

B(41) = C(5)

B(42) = C(6)
```

Data Point Formula

The programmer should be familiar with the formula used by LINE to position a data point on the graph. This formula can be useful when it becomes necessary to find a plotted point in order to draw additional annotation at or near the same place on a graph. The formula is:

$$\frac{\text{A(I)} - \text{FIRSTV}}{\text{DELTAV}} = \text{number of inches from axis origin,}$$

where A(I) is the subscripted value in the array, and FIRSTV and DELTAV are the scaling parameters for that array. The coordinate of the corresponding data value in the paired array would have to be computed in the same manner in order to locate both the X and the Y page coordinates of the plotted point.

Using LINE and SYMBOL with Data Files

In some applications, the data files are too large to be brought into a storage area. If the programmer can estimate the range of the data, he can predetermine scaling parameters FIRSTV and DELTAV without using SCALE. The constant scaling parameters can be referenced in a call to AXIS to draw the axis line, but the LINE subroutine cannot be utilized in the normal manner because the full array of data is not in storage. The alternative is to define for each data file (X,Y) a dummy array large enough for one data value and two scaling parameters. Data can then be read one item at a time, and LINE can be called to plot one point at a time. Connecting lines cannot be drawn between data points when this procedure is used. The coding is as follows:

```
      DIMENSION XA(3), YA(3)

   10 XA(2) = (FIRSTV constant for file X)

      XA(3) = (DELTAV constant for file X)

      YA(2) = (FIRSTV constant for file Y)

      YA(3) = (DELTAV constant for file Y)

      CALL AXIS (XPAGE, YPAGE, IBCDX,-NCHAR,XAXLEN,
      0.,XA(2),XA(3))

      CALL AXIS (XPAGE,YPAGE,IBCDY,+NCHAR,YAXLEN,
      90.,YA(2),YA(3))
```

Using LINE and SYMBOL with Data Files (cont)

```
20 READ (file X data value into XA(1))
   READ (file Y data value into YA(1))
   CALL LINE (XA,YA,1,1,-LINTYP,INTEQ)
   GO TO 20
```

If the application requires connection lines between plotted data points, the programmer must write a substitute routine for LINE, using the special call to SYMBOL. After calling AXIS as shown in the above coding, the following coding would read and plot the data points one at a time:

```
   ICODE = -1
20 READ XA(1)
   READ YA(1)
   XPAGE = (XA(1)-XA(2))/XA(3)
   YPAGE = (YA(1)-YA(2))/YA(3)
   CALL SYMBOL (XPAGE, YPAGE, HEIGHT, INTEQ, 0.,
   ICODE)
   ICODE = -2
   GO TO 20
```

The initial value of ICODE is set to -1 so that the pen is up when moving to the first point. From then on, ICODE = -2, causing a connecting line to be drawn from point to point.

Drawing Nonstandard AXIS Constructions

The typical graph is drawn with a horizontal X-axis at the bottom and a vertical Y-axis at the left side, adjoining the X-axis at its origin. Other axes constructions are shown in Figure 3-9.

In construction (A), the right side of the Y-axis is drawn by changing certain arguments for AXIS as follows:

```
XPAGE = X-axis length
NCHAR = -NCHAR
```

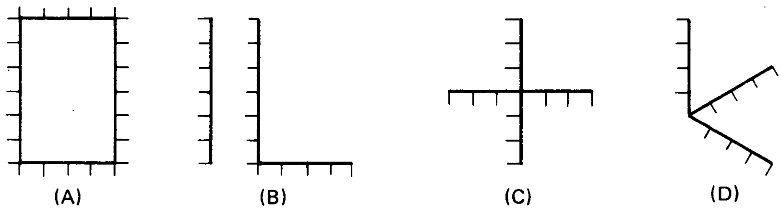


Figure 3-9. Nonstandard Axes

The top X-axis of construction (A) is drawn by making similar changes:

YPAGE = Y-axis length

NCHAR = -NCHAR (i.e., change the sign to plus)

In construction (B), the extreme left Y-axis is positioned at least 1/2-inch from the X-origin. Thus, in calling AXIS,

XPAGE = -0.5

In construction (C), the axes do not join at the lower left corner; they intersect at the axis scale value of zero, which may or may not be at the center of the axis. This construction is preferred when both arrays of data have positive and negative values.

The XPAGE,YPAGE coordinates for each axis are computed from the scaling parameters, assuming that array A corresponds to the horizontal axis and array B corresponds to the vertical axis.

To position the A-axis line at the B-axis 0.00 scale value:

XPAGE =0. and YPAGE = -BFIRSTV/BDELTAV

To position the B-axis line at the A-axis 0.00 scale value:

XPAGE = -AFIRSTV/ADELTAV and YPAGE =0.

Drawing Nonstandard AXIS Constructions (cont)

> **NOTE**
>
> Each computed page coordinate uses the scaling parameters of the other array.

In construction (D), AXIS will draw the desired axis lines at the various angles shown, but LINE cannot be used to plot the data points. The programmer should compute the page coordinates according to his application requirements. When drawing non-standard axes, LINE assumes that X-array page coordinates and X-axis line have the same physical X-origin. Y-array coordinates and the Y-axis line also have the same physical Y-origin. If this were not true, there could not be a proper correspondence between the axis scale values and the position of the plotted data point.

Drawing Nonstandard AXIS Divisions

The AXIS subroutine does not provide any means of dividing the axis line into other than 1-inch divisions. If the programmer needs a finer division than this, he may add coding that will draw-in the extra "tick marks", so long as the tick marks at 1-inch intervals are also acceptable. If the divisions desired are not based on the inch, he will have to write the coding to draw the entire axis, including the scale values at each division. The programmer must take care to adjust (or compute his own) scaling parameters for each array; otherwise irrational scale values might have to be used along the axis line.

EXAMPLE: Assume that a centimeter scale is wanted on axis line. When calling SCALE, the axis length is stated in inches. SCALE would compute a DELTAV that would produce convenient rational scale values at each inch on the axis. If the user's program draws the centimeter divisions on its own axis (i.e., without calling AXIS), it is quite unlikely that the centimeter scale values will also be rational. This scaling-value problem exists only if LINE is called for plotting the points. If the programmer does not use AXIS or LINE, he may state the axis length in centimeters when calling SCALE. He may employ the scaling parameters directly in computing the page coordinates used to plot the data points with SYMBOL.

Using the WHERE Entry

A typical use of the WHERE entry occurs in user-written subroutines that draw lines or figures based on data arrays. The array values can be accessed in either a forward or reverse manner (incrementing or decrementing the subscript). Computing and plotting time can be saved by starting the line or figure at the point that is closer to the current pen position.

EXAMPLE: The curve in Figure 3-10 might be represented by a pair of arrays each containing 100 data values. Assume that the pen is at the point (RX,RY) when the user's subroutine is called to draw this curve. By calling WHERE to locate the current pen position, and comparing these coordinates with the first and last coordinates of the arrays, the subroutine can choose the closer starting point. This starting point can be either (X1,Y1) or (X100,Y100), and thus minimize pen movement.

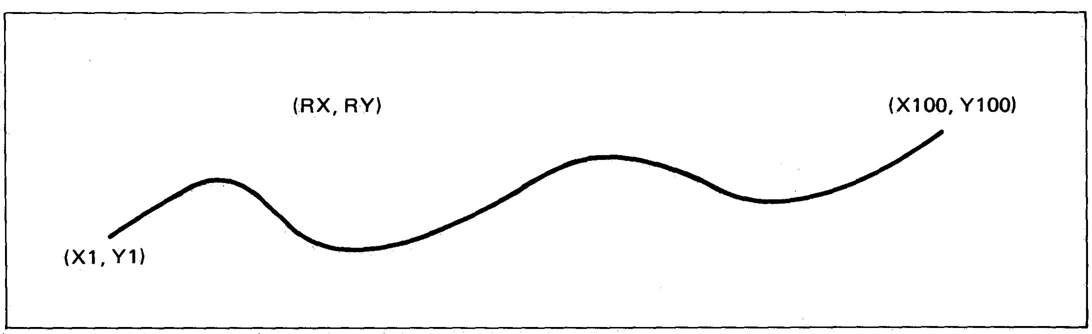


Figure 3-10. Sample Use of WHERE

SECTION 4
CONTENTS

OPERATING CONSIDERATIONS

Plotter Initialization
Controller Initialization
Search Addresses
Planning Your Graph

ILLUSTRATIONS

Plotting Conventions

SECTION 4
OPERATING CONSIDERATIONS

GENERAL

The programmer must consider the operating restrictions and aids when creating his output in order to effectively program CalComp graphic systems. Although CalComp systems allow a great deal of flexibility in operation, many installations impose restrictions in order to simplify and standardize operations. Different types of graphic systems have different operating considerations which are dependent on the specific model of the plotter/controller.

PLOTTER INITIALIZATION

The initialization of the plotter is an operator function for most plotting systems. This could be as simple as turning on power and waiting for the plotter (electronic CRT plotter) to be ready, or as precise as the alignment of a preprinted form on a flatbed plotter and adjustment of pen, cutter, or scribe forces.

All plotting software assumes that the initial origin of the plotter is where the operator has left the pen positioned.

Limit Switches

There are two basic types of electromechanical plotters:

- Flatbed Plotters
- Drum Plotters

Pen (or accessory) travel is limited in both the X and Y directions by limit switches on the flatbed plotters. Pen (or accessory) travel is limited only in the Y direction on the drum plotters. Trying to plot beyond these limits may cause a shift in origin or it may shut down the plotter, depending upon the plotter model.

Limit switches on several early plotter models were used effectively for initial positioning by the program and for repositioning. These practices are not effective on the newer, high-speed models. Limit switches are no longer accurate and may cause the plotter to abort the job.

Accessory Checks

Pens or accessories should be checked to ensure proper operation before alignment. If pens (especially liquid ink pens) are being used, the operator should test all pens and check that they are flowing properly. If multiple pens are being used irregularly in a given plot, the program might move a new pen to a margin area and draw small rectangles to ensure that the pen is flowing properly.

It is necessary to check the various accessories (cutters, scribes, optical writing systems etc.) to make sure that they are in adjustment before using.

Initial Alignment

The initial positioning of the pen by the operator depends on the type of material being used and the specific type of plotter. On a drum plotter (with plain paper), the operator positions the pen at the right edge of the drum, close to the limit switch. When preprinted chart paper is used, the operator positions the pen accurately (using a reticle) on a specified grid mark near the right edge of the drum. The operator positions the pen (or accessory) on a flatbed plotter (with unprinted material) near the lower-left corner of the material. When preprinted material is used on a flatbed plotter, initial alignment consists of not only positioning the pen accurately at some specified mark, but also consists of aligning the material squarely on the bed of the plotter. Special software may be used to measure the rotation and shrinkage of the material and may also be used to adjust the data.

The foregoing discussion assumes a single pen for initial positioning, whereas most plotters actually have multiple pens. Multiple pen plotters require the operator to align pen one. The operator should also exercise care when changing pens near the limit switches since this could cause the pen block to move into the limit switch.

It is a good practice in installations with drum plotters that run many jobs consecutively, to end your plot with the pen in the same Y position as when you started. You should also end far enough down the paper to be beyond all plotting.

CONTROLLER INITIALIZATION

Initialization is easily handled by the operator for most CalComp controllers. The initialization functions depend upon the type of controller involved and the complexity of the system being used. Online controllers may require no operator initialization, but stored program controllers may require the entering of numerous parameters.

Simple online or offline controllers usually require the following initialization:

- Turning on power
- Loading tapes
- Starting the system

Initialization is usually the same for a stored program controller using either standard or default parameters.

In more complex systems, one or more parameters may be entered to effect the plotted output or supply information about the data. After loading and starting the controller, the operator is offered the option of entering parameters or using the default values. Some of the parameters the operator may enter, depending upon the system, are as follows:

- Maximum velocity of plotter.
- Acceleration level of plotter.
- Scaling factors in the form N/D.
- Coordinates of windows.

SEARCH ADDRESSES

Search Addresses are short identification records within the plotting data that are used to identify different plots from one run or sections of longer complicated plots. Users should avoid excessively long plotting runs without some stopping points. One solution is the use of Search Addresses. The PLOT subroutine produces Search Addresses when any of the following events occur:

PLOTS	is called (Search Address 001 is output)
PLOT	is called with an -IPEN value (the next sequential search address is output)
PLOT	is called with IPEN = 999 (Search Address 999 or 9999 is output)

SEARCH ADDRESS (Cont)

All CalComp Magnetic Tape-Driven Controllers utilize Search Addresses in various forms. Some online and card-driven systems can utilize these Search Addresses.

The operator of most offline controllers can search for a given search address (forward or backward), plot from one search address to the next, or plot to selected search address. The operating instructions for each plotting system contains detailed information on which features are available and how to use them.

PLANNING YOUR GRAPH

Graphs, plots and computer listings, require some planning to achieve a pleasing and effective format. The following check-list of plotting conventions is letter-keyed to Figure 4-1 to help you in your planning.

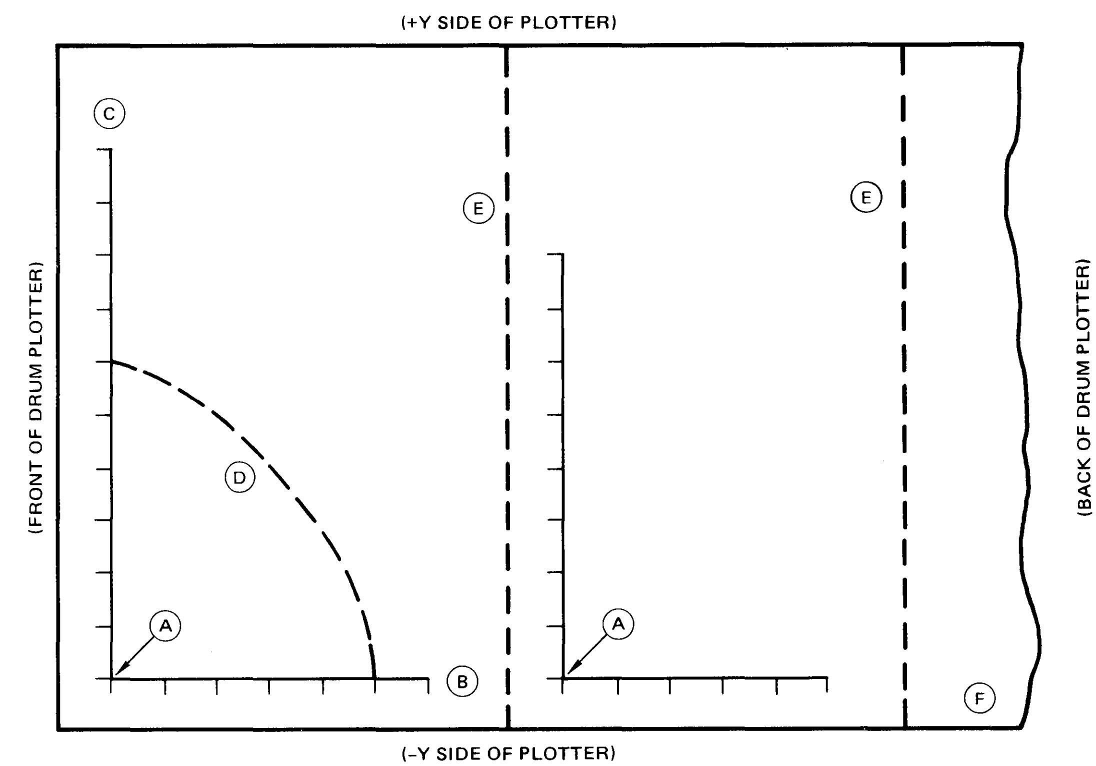


Figure 4-1. Plotting Conventions

PLANNING YOUR GRAPH (cont)

A. The initial position of the pen when the plotting operation begins is assumed to be the logical origin (X = 0). All pen movements are defined in X-coordinate and Y-coordinate inches from the current origin. Subsequent origins can be established at other positions by appropriate programming to provide new reference points. (See PLOT in Section 3.) When the next graph is started, a new origin should be defined far enough away to avoid any axis or annotation conflict with the previous graph.

B. The X-axis lies parallel to the side of the drum plotter, with the +X direction toward the back. The X-axis line should be at least 1/2-inch from the sides of the plotter to leave a margin for labeling.

   The maximum width of a graph (X-direction) is limited only by the length of the paper: On a drum plotter, it is usually unlimited; on flatbed plotters, it is limited to the size of the plotter.

C. The Y-axis is parallel to the width of the plotter. The Y-axis line should also be at least 1/2-inch from the sides of the plotter to leave a margin for the label and annotation.

   The maximum height of a graph (Y-direction) is determined by the pen carriage movement.

D. The angle of rotation about any point is determined by a vector which when extended in the +X direction represents 0 degrees. Any angle argument used in a calling sequence may be stated in plus or minus degrees relative to the X-axis.

E. When drawing several graphs in one program, it may be desirable to draw trim lines for the operator's convenience in separating the graphs.

F. After the last graph has been drawn, the pen should be moved to a position that permits easy removal of the graphs. On the drum plotters, this would be several inches beyond the end margin to allow the paper to be torn off the roll. On flatbed plotters, this position would be the extreme -X, -Y corner of the bed.

SECTION 5

CONTENTS

BASIC SOFTWARE DESIGN

General

Data Formats

905 Tape Format
900 Tape Format
900 Online Formats
CAL EDIT Format
RS-232/936 Serial Format
RS-232/960 Serial Format
RS-232/836 Serial Format

PLOT Subroutine Construction

General Flow of PLOT Subroutine
Line Approximation Algorithms
Output Processing
Other PLOT Entry Points

SYMBOL Subroutine

Symbol Design
Symbol Selection Tables
Symbol Generation

AXIS, LINE, NUMBER and SCALE Subroutines

Graphic Controller Software Design

Flow of GCS
Executive Module
PEAD Module
PLOT Module
SYMBOL Module
Other GCS Modules

CONTENTS (cont)

ILLUSTRATIONS

905 Tape Format
900 Tape Format
ASCII Characters Available with the SYMBOL Routine
General Flow of PLOT Subroutine
8-Vector Structure
Solution of a Straight-Line Approximation
8-Vector Incremental Algorithm Functional Diagram
Line-Splitting Algorithm
SYMBOL Subroutine Flowchart
SYMBOL Grid System
SYMBOL Grid
AXIS Flowchart
LINE Flowchart
NUMBER Flowchart
SCALE Flowchart
GCS Flowchart

TABLES

900 Format Control Codes
Delta Values which may be Held in Various Number of Characters
Pseudo Increments for Various Plotters
CAL EDIT System Parameters
960 Characters
Value of IPEN
925 Controller Memory Layout

SECTION 5
BASIC SOFTWARE DESIGN

GENERAL

CalComp creates Basic Software to support the complete line of graphic systems for many of the world's computers. This has established a large library of software devoted to graphic output. In order to maintain and expand this library, many standards have been developed. This section describes the standards of Basic Software. Due to limitations of certain hardware and the various languages, some standards must be compromised. These standards also change as better means to accomplish goals are developed and new ideas are tried and perfected.

CalComp strives to maintain compatability from one generation of hardware to the next generation of hardware. However, advances in hardware necessitate some changes in software which reduce compatability.

DATA FORMATS

The Basic Software standards are composed of various data formats. The data formats are generally dependent upon the type of hardware involved with some formats handling multiple hardware products. The more important features of the various formats are described in this section.

905 Tape Format

The CalComp Model 905 Tape Controller uses 7 or 9 track odd parity tapes. The controller uses the high order four bits of the available six or eight data bits together with the parity bit. The remaining two or four low order bits are usually zero. All codes shown are in hexadecimal.

Physical tape records may be of fixed or variable length, but the tape record must not contain more than 500 plotting characters. The first plotting character must be within the first 30 characters of the record.

There are three types of records used by the tape controller. Other records on tape (labels, filemarks, etc.) are ignored by the 905 controller. The three types of records are as follows:

Intermediate Block Address (IBA)	used for stopping while in single plot mode and for positioning. The first record on tape (following labels) should be an IBA record.
Final Block Address (FBA)	used for stopping while in the multiple plot mode. The last record on tape should be an FBA record.
Data Records	contain the Plot Data.

The format of each record is shown in Figure 5-1. The Data Characters are also shown in Figure 5-1. Data Characters consist of pen up and pen down codes, incremental codes, special functions, and delta moves. Pen up, pen down, and special function codes are used as needed. The End Of Record special function must appear once (and only once) in every tape record. All characters following the End Of Record are ignored by the 905. Each incremental code causes the plotter to move one increment in the direction indicated.

Delta moves consist of five consecutive characters on tape. These characters consist of a control character that gives direction of move (see Figure 5-1), two characters for the magnitude of delta X and two characters for the magnitude of of delta Y. The magnitude is limited to 0 through 255 and is represented by two Hex digits (00 to FF), one in each character.

905 Tape Format (cont)

The format of each record is as follows (all codes shown in Hex):

Intermediate Block Address 7 TK	Intermediate Block Address 9 TK	Final Block Address 7 TK	Final Block Address 9 TK	Data Record 7 TK	Data Record 9 TK
3C	F0	3C	F0	3C	F0
3C	F0	3C	F0	3C	F0
28	A0	28	A0	28	A0
0C	30	0C	30	0C	30
10	40	10	40	10	40
28	A0	28	A0	28	A0
0C	30	0C	30	0C	30
10	40	0C	30	14	50
				14	50
				•	•
				Data	Data
				•	•

The Data Characters are as follows:

7 TK	9 TK		
00	00	NO-OP (ignored)	
04	10	Raise Pen	
08	20	Lower Pen	
0C	30	Special Function (see explanation below)	
10	40	+Y+X	
14	50	-Y+X	
18	60	+Y-X	Delta Move (see explanation of Delta)
1C	70	-Y-X	
20	80	+Y	
24	90	+Y+X	
28	A0	+X	
2C	B0	-Y+X	
30	C0	-Y	Incremental Codes
34	D0	-Y-X	
38	E0	-X	
3C	F0	+Y-X	

Figure 5-1. 905 Tape Format

905 Tape Format (cont)

Legitimate special functions are as follows:

7 TK	9 TK	
0C,0C	30,30	End of record
0C,20,04,18	30,80,10,60	Select pen one
0C,20,04,04,18	30,80,10,10,60	Select pen two
0C,20,04,04,04,18	30,80,10,10,10,60	Select pen three

Other combinations are not currently used.

Figure 5-1. 905 Tape Format (cont)

EXAMPLE: The following five characters (shown as 9 track Hex) will cause a move of 95 increments in the +X direction and 129 increments in the -Y direction. These characters are: 50, 50, F0, 80, 10.

The CalComp Model 905 Tape Controller drives several models of CalComp incremental plotters. The distance moved by giving incremental codes or delta commands is a function of the increment size of the plotter.

900 Tape Format

The 900 Tape Format is used on the CalComp Model 925 Controller and its predecessors (Models 900, 910, and 915). The CalComp 925 Controller is a stored program device that drives all CalComp plotters from several types of data sources. The standard unit contains a magnetic tape drive that handles 7 or 9 track odd parity tapes. The format of these tapes is described below.

The 925 Controller is programmable and is capable of handling versatile tape formats. CalComp has developed an efficient data format to transmit plot information from a Host Computer to a CalComp 925 Controller. This tape format offers some distinct advantages over other formats, such as compressed deltas and the ability to plot the same tape on several different plotter models.

The 900 tape format is divided into two classes depending upon the character and the word size of the Host Computer.

- A 6 bit format is produced by computers with 6 bit characters e.g., Honeywell 6000 and Univac 1108.

- An 8 bit format by computers with 8 bit characters e.g., IBM 370 and Burroughs 3500.

Tape records can either be 7 or 9 track containing 6 or 8 bit characters. The data is packed on the tape with 6 bit characters on 9 track packed 4 characters per 3 tape frames. Eight bit characters are packed on 7 track tape with 3 characters per 4 tape frames. Six bit characters are packed on 7 track tape (or 8 bit characters on 9 track tape) with 1 character per tape frame. The maximum record size is 384 8-bit characters or 512 6-bit characters. The suggested minimum record length is 100 characters. Physical tape records may be fixed or variable length, but must not be blocked. Any filler may be used at the end of the record following the plot data. Up to 25 control characters may precede the plot data providing the maximum record size is not exceeded. Each plot record consists of the following:

- Four SYNC characters
- One to 507 plotting characters
- One end-of-plot character

The four SYNC characters must appear within the first 30 characters of the tape record. Any records which do not contain these SYNC characters are ignored when read by the controller. This enables the system to skip over tape labels and filemarks. The system also skips over characters that may be added to the beginning of all records by various operating systems and language processors.

The first plot record is a dummy record that is skipped by the controller, but the record contains the four SYNC characters. The second plot record contains search address 1 with the last record containing search address 9999.

Figure 5-2 shows the general tape format for 900 tapes. All codes shown are 8-bit characters (Hexadecimal). Except as noted, 6-bit characters are the same except the two high order bits are omitted.

The plotting characters are interpreted on the controller by examining a control character that specifies the operation and then processing the following characters according to the operation specified. When this processing is completed, the next control code is picked up and the operation is repeated until the end of plot character is encountered. An operation must be completely contained within one record. Table 5-1 summarizes the control characters as follows:

Several control characters specify 1, 2, or 3 characters that contain a binary number. This binary number is a two's complement number occupying all bits of the number of characters specified.

EXAMPLE: In three 8-bit characters, it is a 24-bit number. In two 6-bit characters, it is a 12-bit number. The number is right justified in the number of characters specified by the Control Code.

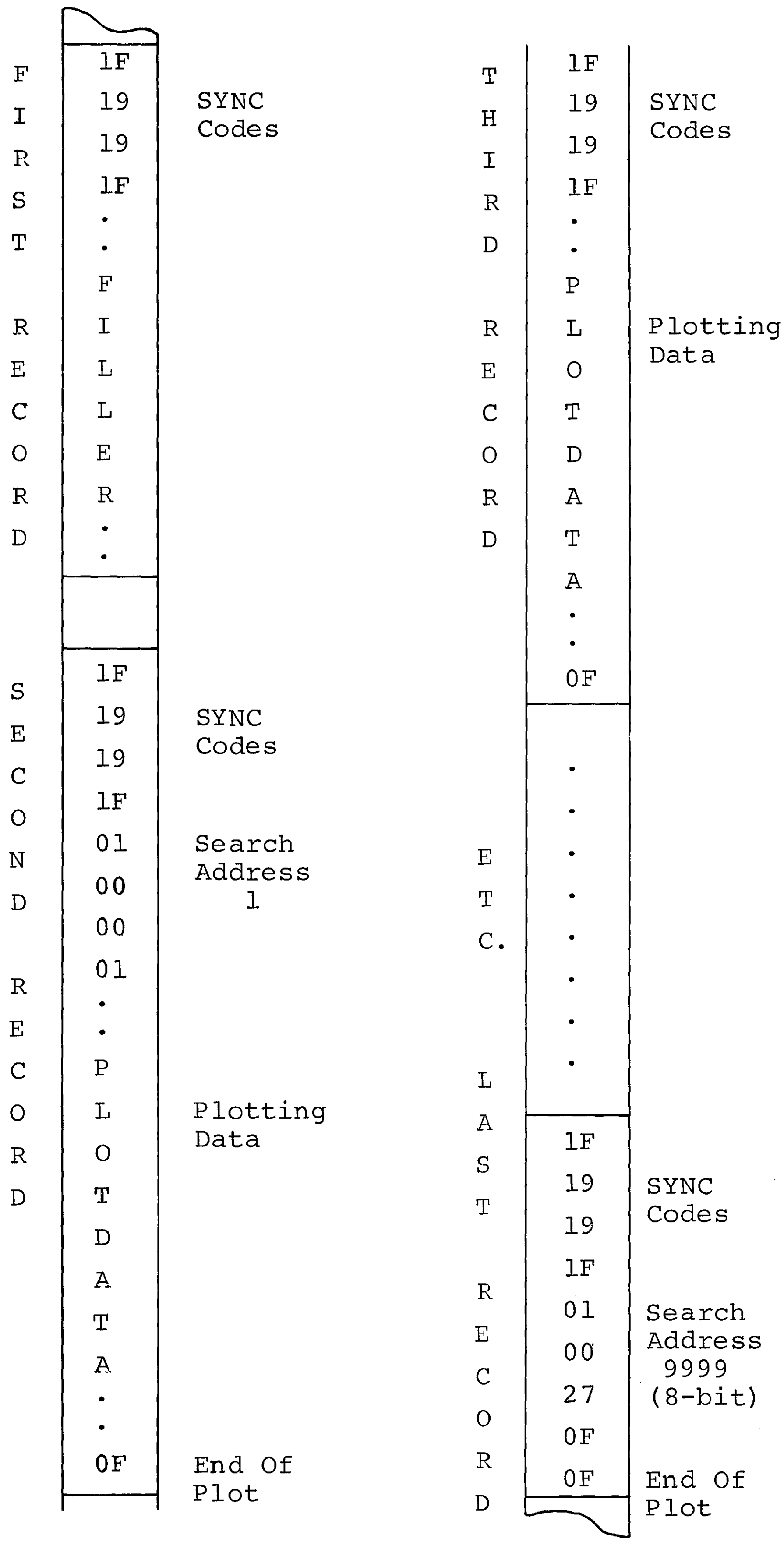


Figure 5-2. 900 Tape Format

Table 5-1. 900 Format Control Codes

Control Characters (Hex)	Additional Characters Number and Description	Function
00	None	NO-OP
01	3 - binary address number	Search Address
02	None	Pen Down/Beam On
03	None	Pen Up/Beam Off
04	1 - binary pen number or intensity level	NEWPEN
05	1 - count of characters (N)	SHIFT
	N - Secondary Control Characters	
06	1	not used
07	1	not used
08	1 - dummy value	1670 Centering
09	1 - binary stepsize	1670 Stepsize
0A	1 - binary advance value	1670 Advance
0B	2 - count of characters (N)	1670 Character Generator
	N - ASCII characters	
0C	1 - circle type flag	Circle Generator
0D	1	not used
0E	1 - binary form number	1670 Forms Selection
0F	None	Stop Plot

Table 5-1. 900 Format Control Codes (cont)

Control Characters (Hex)	Additional Characters Number and Description	Function
1X*	0 to 3 - binary X delta	Plot DELTA
	0 to 3 - binary Y delta	
2X*	0 to 3 - binary XAX value	Symbol Mode
	0 to 3 - binary YAY value	
	1 - binary count of characters (N)	
	N - ASCII characters	
3X*	0 to 3 - binary XAY value	Extended Symbol Mode
	0 to 3 - binary YAX value	

* X represents the number of characters in the X and Y values as follows: The first 2 bits represent the number of characters (00_2 to 11_2) in the X value, and the last 2 bits represent the Y value.

NO-OP	is ignored by the controller. This may be used as a filler code if fixed length records are written or word oriented output is being done.
SEARCH ADDRESS	is a binary number from 1 to 9999 which is the address of the following plot:
	The search address must always appear at the beginning of the plot record immediately following the SYNC codes (this does not mean that every record must have a search address). The first plot record contains search address 1 and the last plot record on a tape contains search address 9999.
PEN DOWN/ BEAM ON	is used to indicate a change in the state of the pen from pen up to pen down (writing) or the state of the beam from off to on (writing).
PEN UP/ BEAM OFF	is used to indicate a change in the state of the pen from pen down (writing) to pen up or the state of the beam from on to off.
NEWPEN	is used for plotting systems with multiple pens or variable intensities. The one character binary number following the control code specifies the the pen number to be selected. This number can range from 1 to 63 and will be masked into the range acceptable by a specific plotter.
SHIFT	is used by the 1675 COM System to signify a repeat count and by certain online systems to request start of plotting.
	1675 COM System repeat count is used to repeat the plotting from the search address indicated through the current search address the number of times indicated by the repeat count.

SHIFT (cont)	05	Shift Code
	05	Count of Characters Following
	XX XX XX	Three Character Binary Search Address
	NN NN	Two Character Binary Repeat Count

Start plotting command used in real time online plotting systems that force the system to start plotting.

05	Shift Code
01	Count of Characters Following
0A	Start Plotting Code

In other than these specific cases, the shift code is ignored along with the number of characters specified by the count in the character following the shift code.

1670 CENTERING CODE — is used to indicate that the next delta move is a centering move that represents the offset from the lower left corner of the 1670 CRT to the lower left corner of the PLOT.

1670 ADVANCE CODE — is used to indicate a film advance or feature selection on the 1670. The character following the advance code indicates the type of advance or selection of special features as follows:

00	Set rotation to 0°
01	Set rotation to 90°
08	Flash Form
20	Advance one frame with reset
21	Advance one frame without reset
22	Advance one fiche with reset
23	Advance one fiche without reset

1670 CHARACTER GENERATOR CODE	is used to indicate the selection of the 1670 Hardware Character Generator. The first two characters following the code are the binary count of characters to be printed and they are immediately followed by the ASCII characters to be printed.
CIRCLE GENERATOR CODE	specifies that a circle definition follows on the tape. The character following the code indicates the type of circle which is to be plotted as follows: 00 Counterclockwise solid circle 01 Counterclockwise dashed circle 02 Clockwise solid circle 03 Clockwise dashed circle Following the first Circle Control Code, one or more delta commands specify the vector from the starting point of the circle (or arc) to the center. A NO-OP Control Code follows signifying end of vector, then another group of delta commands specify the vector from the center to the ending point of the circle. This is again followed by a NO-OP to end the group of deltas.
1670 FORMS SELECTION	is used to specify the number of the form to flash when requested by the appropriate 1670 Advance Code.
STOP PLOT	signifies the end of plot data for the particular plot.
PLOT DELTA CONTROL CODES	used to specify pen (or beam) movements to the plotter. The 16 different control codes are used to specify the number of characters needed to express the X and Y component of the delta (0, 1, 2, or 3 characters for each component). Only those characters necessary to represent the value, need be used. Additional characters may be used to simplify programming. The number of characters required to represent delta values are given in Table 5-2.

Table 5-2. Delta Values Which May Be Held In Various Number Of Characters

Number of Characters	Range of Values	
	6-Bit	8-Bit
0	0	0
1	-32 to 31	-128 to 127
2	-33 to -2048 32 to 2047	-129 to -32767 128 to 32767
3	-2049 to -32767 2048 to 32767	N/A

Table 5-3. Pseudo Increments For Various Plotters

Plotter Type	Pseudo Increment	
	Per Inch	Per Centimeter
936, 1036 .002 inch	1000.0	393.7
936, 1036 metric	2032.0	800.0
960	2032.0	800.0
748	10160.0	4000.0
745	10000.0	3937.0
1670	1000.0	393.7

The maximum delta value allowed is 32,767. This limit is placed to allow freedom of computation within the controller without resorting to triple precision. The delta values are specified in pseudo increments dependent upon the plotting system that is used (see Table 5-3).

SYMBOL MODE CONTROL CODES	used to specify a symbol string, height and angular orientation of the characters. The 16 different control codes specify the number of characters needed to express the values of XAX and YAY (0, 1, 2, or 3 characters for each value). XAX and YAY are used to determine the height and angle of the characters drawn according to the following formulas:

XAX=HEIGHT*COS(ANGLE)*STEPS*XFACT*8/7

YAY=HEIGHT*SIN(ANGLE)*STEPS*YFACT*8/7

Where:

HEIGHT	is the height in inches (or centimeters) of the character string.
ANGLE	is the angle between the base line of the character string and the X axis of the plotter.
STEPS	is the number of pseudo increments per inch (centimeter) for the plotting system.
XFACT, YFACT	are the X and Y components of the scaling factor applied to the plot. XFACT and YFACT are the same for most systems supplied by CalComp.
8/7	is a factor used for precision and to represent one grid unit. This is replaced by 8/4 for centered symbols.

Only those characters necessary to represent the value, need be used (see Figure 5-3), however, additional characters may be used to simplify programming. The maximum value allowed is 10,922.

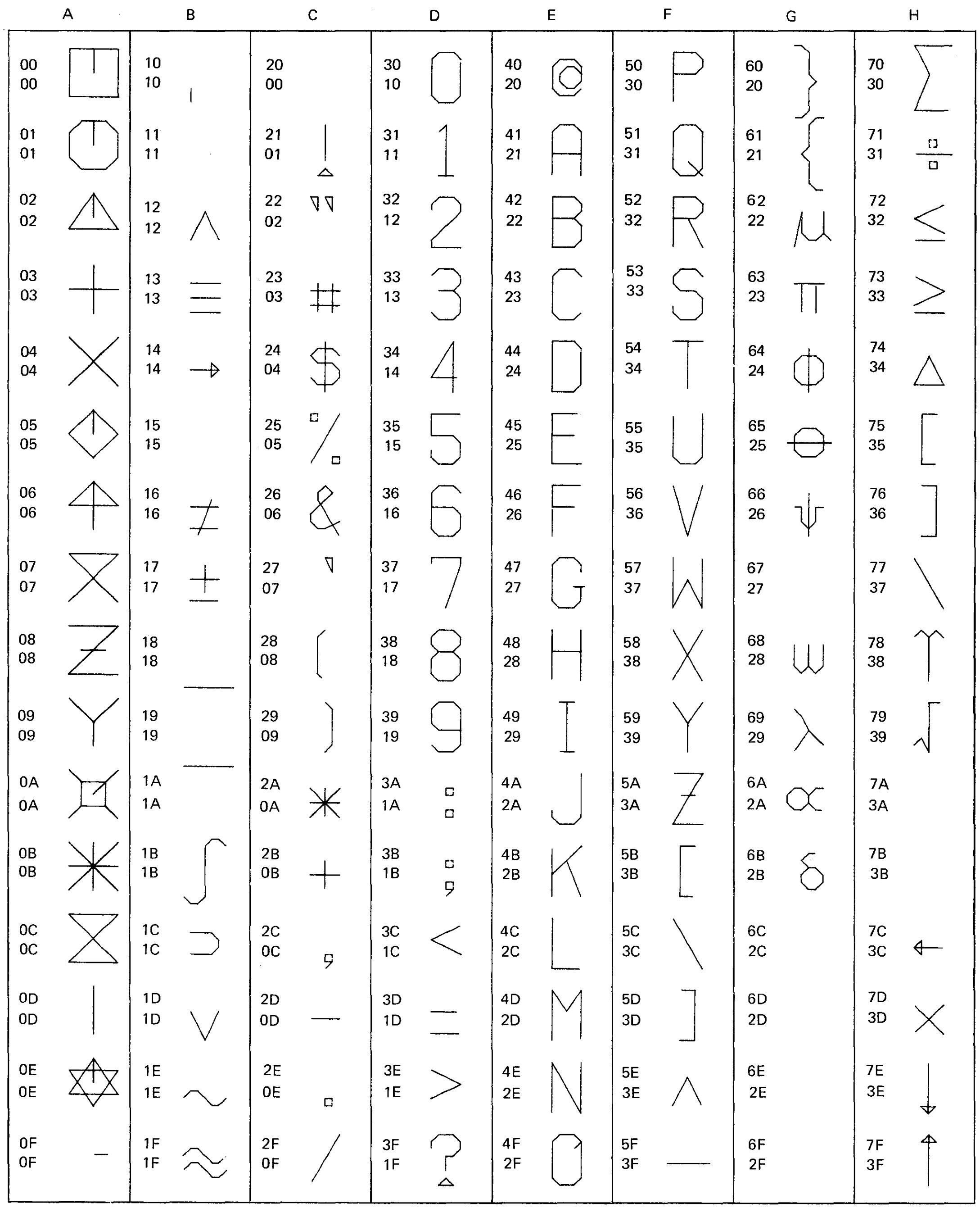
If the character string is drawn at the same height, angle, and factor as the previous string; then XAX and YAY values of zero may be used to eliminate recalculation of tables in the controller. This is often used and may be indicated with a control code of 20_{16}.

> NOTE
>
> The controller remembers the pen state and position prior to plotting symbols. Afterwards, it returns the pen to the original state and adjusts the next delta according to the accumulated distance moved while plotting the symbol string. Thus, the next delta should represent a move from the starting point of the symbol string.

The first character following the XAX and YAY values is a binary count specifying the number of characters to be plotted. The count can be from 1 to 63 (or 1 to 255 for 8 bit systems). A count of 0 is ignored in 8 bit systems, but the count of 0 specifies special characters in 6 bit systems.

The alpha characters are represented in ASCII with either 7 bits for 8 bit systems or a masked 6 bit subset for 6 bit systems. If the character count is zero, then in 6 bit systems a single 6 bit character follows representing one of the Special ASCII characters. The characters available with their Hex codes are shown in Figure 5-3.

EXTENDED SYMBOL MODE CONTROL CODES	used to augment the height and angular orientation information in the following symbol mode codes. The 16 different control codes are used to specify the number of characters needed to express the values of XAY and YAX (0, 1, 2, or 3 characters for each control code). XAY and YAX are used with XAX and YAY to determine the height and angle of the characters drawn according to the formulas:



NOTE: Codes to the left of each symbol are the Hex representation of the symbol for:

1 - (top) 8-bit systems
2 - (bottom) 6-bit systems

For 6-bit systems, center four columns (C,D,E,F) are available with positive count and outside columns (A,B,G,H) are available with zero count.

Figure 5-3. ASCII Characters Available with the SYMBOL Routine

XAY=HEIGHT*COS(ANGLE)*STEPS*YFACT*8/7

YAX=HEIGHT*SIN(ANGLE)*STEPS*XFACT*8/7

If XFACT equal YFACT, then XAY equals XAX and YAX equal YAY. In this case, the extended symbol mode does not have to be used since the controller assumes they are equal. If they are not equal, then any symbol mode code must be preceded by an extended symbol mode control code.

If previous values are used for XAY, YAX, XAX, and YAY; the extended symbol mode may be eliminated and the symbol mode may be represented by Control Code 20.

See the paragraph SYMBOL SUBROUTINE on page 5.48 for a detailed discussion of XAX, YAY, XAY, and YAX values.

900 ONLINE FORMATS

The CalComp Model 925 Controller can be connected online to several computers. Modifications to the Standard 900 Tape Format are required in order to make interfaces compatible to the controller. These modifications are tailored to individual interface requirements with some of the following major modifications.

The P-18 Online Interface is used to interface the CalComp 925 Controller with the IBM 360/370 Systems. The controller acts as a card punch for the IBM Central Processing Unit. It requires 80 byte data records. The data is similar to the data used for tape systems (8 bit systems) with the following exceptions:

- Plotting data occupies the first 72 bytes of each 80 byte record with the last 8 bytes containing Hollerith sequence data. The sequence field consists of a 4-digit search address and a 4-digit record sequence number.
- The first record (dummy record) contains the Hollerith data PLOT in the first 4 bytes and a sequence field of all zeros.

- A logical Plot Record may span up to 5 data records (5 x 72 = 360 bytes), but must contain a Stop Plot Control Code.
- No SYNC codes are used.

The P-34 RS-232C Online Interface is used to interface the CalComp 925 Controller with various computers. The controller is used as a serial output device and requires serial data. This data is similar to the data used for the P-18 Online Interface.

CAL EDIT FORMAT

Plot data created from CAL EDIT Host Computer Basic Software or specified by teletype input is in a modified EIA RS-274 format. The basic processing unit of this format is called a sentence. A sentence is a series of words which is terminated with a period. A word consists of a key letter specifying a particular function followed by numeric digits which are coordinate values or additional function information. The words may appear in a sentence in any order and may appear more than once with only the last word of a given type being effective. Actual plotting of the sentence does not begin until the terminal period is encountered. If a word necessary for plotting the sentence is not contained in the sentence, the last value given in a previous sentence is used. If no previous value is given, the default value shown in Table 5-4 is used. Blanks within a sentence are ignored, except within a character string.

Word Types

The maximum absolute numeric values for each word type are shown in Table 5-4. Signs may be used with unsigned numbers assumed positive. Leading zeros may be omitted. Coordinates are given as absolute values in units of plotter steps.

The word types implemented are as follows:

A, B	gives the length of pen down/pen up segments when using the G4 mode (dashed line).

Table 5-4. CAL EDIT System Parameters

Word Type	Default Value	Maximum Numeric Absolute Value
A	0.25 inches (0.5 centimeters)	67108863
B	0.25 inches (0.5 centimeters)	67108863
D	0	99
E	0	8191
F	0	8191
G	0	99
I	0	262143
J	0	262143
M	0	99
N	0	99999
P	10000	67108863
Q	0	67108863
R	0	67108863
S	10000	67108863
T	0	9999
U	0	67108863
V	0	67108863
X	0	67108863
Y	0	67108863

D	controls certain plotter operations and is followed by a 1 or 2-digit code:

D1 = pen down (G1 - G5 modes)

D2 = pen up (G1 - G5 modes)

Dnn = select pen nn (G50 mode)

E, F	are the values that control the height and angle of symbols within a character string as follows:

$$E = H * COS\ \theta * 8/N$$

$$F = H * SIN\ \theta * 8/N$$

Where:

H	is the height of the symbol in plotter steps.
θ	is the angle of rotation.
N	is the number of grid units on which the symbol is based (15 for standard symbols and 8 for centered symbols).

G	specifies the current mode of operation and is followed by a 1 or 2-digit code:

G1 = linear interpolation

G2 = circular interpolation - clockwise

G3 = circular interpolation - counterclockwise

G4 = linear interpolation - dashed line

G5 = spline interpolation

G25 = reorigin at current (X,Y) coordinate

G50 = pen selection (also requires Dnn code to select pen nn)

G52 = character string - standard symbols

G53 = single character - special symbols

G55 = operator message

I, J — gives the incremental distance from the current point to the center of a circle or circular arc.

M — controls stopping as follows:

M1 = temporary halt

M2 = final halt

N — specifies the Sentence Number.

P, Q, R, S — form a transformation matrix which is applied to all (X,Y) coordinates prior to plotting. If (X_i, Y_i) is the input coordinate and (X_f, Y_f) is the final plotted coordinate, then:

$$\begin{pmatrix} X_f \\ Y_f \end{pmatrix} = \begin{pmatrix} P & Q \\ R & S \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \end{pmatrix}$$

This matrix permits scaling, rotation, mirror imaging, and axis interchanging. P, Q, R, and S are defined as follows:

$P = F_x \cos\theta$

$Q = -F_x \sin\theta$

$R = F_y \sin\theta$

$S = F_y \cos\theta$

Where:

F_x, F_y — are the X and Y axes scale factors.

θ — is the angle of rotation.

The unit scale factor is 1 followed by 4 zeros. The range of scale factors is 0.0001 to 6710.8863.

To obtain a mirror image about the X axis, use:

PQRS = (1,0,0,-1)

To obtain a mirror image about the Y axis, use:

PQRS = (-1,0,0,1)

To interchange axes, use:

PQRS = (0,1,1,0)

> NOTE
>
> Independent X and Y scaling, rotating, and mirror imaging may also be specified at plot time by teletype command.

T — specifies a Search Address. The value following T is a standard CalComp Search Address and may range from 1 to 9999.

U, V — gives the origin offset in the X and Y directions, respectively. This origin is subtracted from each X and Y value before the PQRS transformation is applied.

X, Y — are the coordinate values for linear, circular, or spline interpolation. The values are absolute coordinates in units of plotter steps.

> NOTE
>
> On magnetic tape, each record contains a group of sentences or portions of sentences such that the total number of characters in a record is less than 512.

RS-232/936 SERIAL FORMAT

The CalComp 936 Plotter can be driven online to various computers using an RS-232 serial interface. The data sent to the plotter consists of a string of commands each containing one or more characters.

A character consists of 1 start bit, followed by 8 data bits, followed by 1 stop bit. The data rate is 9600 baud, with other rates available by special request.

The interface uses the following circuits in accordance with EIA Standard RS-232-C:

BA - Transmitted Data

CA - Request to Send

CB - Clear to Send

CC - Data Set Ready (Forced to ON by this interface)

AB - Signal Ground

CF - Received Line Signal Detector (Forced to ON by this interface)

CE - Ring Indicator (Forced to OFF by this interface)

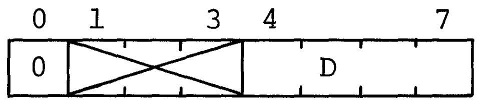
Commands sent by the computer to this interface are of two types:

- Direct Commands
- Delta Commands

The direct commands are 1 character (8 bits) long and provide for pen manipulation and single-step incremental motion, if desired.

The delta commands are 3 characters long and provide vector information in the form of magnitude and direction. Each new vector is begun at the point at which the previous vector terminated.

Direct Commands

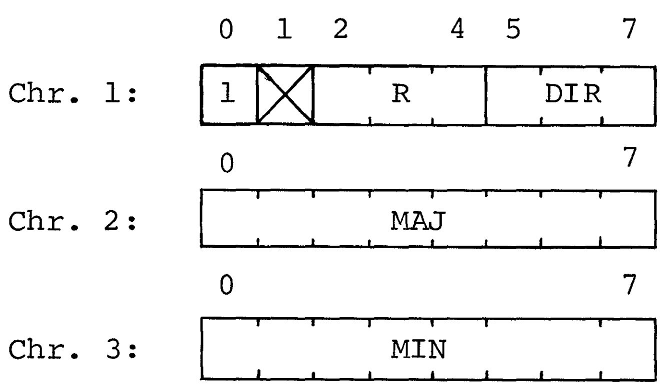


D	Command	D (Octal)	Command
0	+Y step	10	Enter Special Function
1	+Y, +X step	11	Pen Up
2	+X step	12	Pen Down
3	+X, -Y step	13	No Operation (NOOP)
4	-Y step	14	No Operation (NOOP)
5	-X, -Y step	15	No Operation (NOOP)
6	-X step	16	Leave Special Function
7	-X, +Y step	17	No Operation (NOOP)

Pen Up and Pen Down delays are built into the hardware and hence need not be provided in software unless additional delays are required. Pen Selection of Pen 1, 2, or 3 is accomplished by the following sequence of direct commands:

Select Pen 1	Select Pen 2	Select Pen 3
10	10	10
11	11	11
16	11	11
	16	11
		16

Delta Commands



Delta Commands (cont)

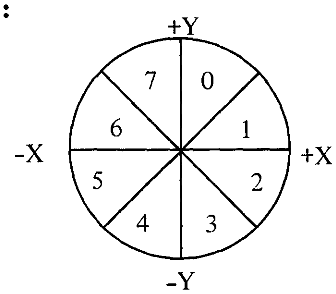
Where:

R — is the number of times this vector is to be repeated.

If R = 0, the vector is drawn once.

If R = 7, the vector is drawn 8 times.

DIR — is the direction of the vector, expressed as an octant (0-7) in which this vector is to be drawn. The octant numbers are assigned as follows:



If a vector is to be drawn which lies on an octant boundary, either octant number may be used.

MAJ — is the magnitude of the larger component of the vector (X component or Y component) expressed as a positive number of plotter steps from 0 to 377_8.

MIN — is the magnitude of the smaller component of the vector (X component or Y component). It represents a positive number of plotter steps from 0 to 377_8, but is expressed in one's complement form.

Deltas drawn on either the X or Y axis have a smaller component of 0. Deltas drawn at 45°, 135°, 225°, and 315° have equal components, e.g. MIN is equal to MAJ (before MIN is complemented).

MIN (before complementing) must always be equal to or less than MAJ or loss of origin will occur.

MAJ must not be 0 (although allowed in the command format) or loss of origin may occur.

Delta Commands (cont)

If the interface is sent an incomplete delta command (1 or 2 characters), it will wait indefinitely for the command to be completed. If it is necessary to restart the plotter from the computer when the plotter buffer contents are unknown, a sequence of 4 or more NO-OP's can be sent to reestablish synchronization.

RS-232/960 SERIAL FORMAT

The CalComp 960 Plotter can be driven online to various computers using an RS-232 serial interface. The data sent to the plotter consists of a string of commands each containing one or more characters.

A character consists of 1 start bit, followed by 8 data bits, followed by 1 stop bit. The data rate is 9600 baud, with other rates available by special request.

Commands sent by the computer to the plotter are of four types:

- Direct Commands
- Single Precision Delta Commands
- Double Precision Delta Commands
- Symbol String Commands

Direct Commands (Pen Commands, No-ops)

ID				COMMAND			
0	0	0	0	N	N	N	N

NNNN = 0000	Pen Up
NNNN = 0001	Pen Down
NNNN = 0011 through NNNN = 1111	NO-OP
NNNN = 0010	The next 8-bit character contains additional direct command information:

Table 5-5. 960 Characters

Character	Octal Code	Character	Octal Code
@	00	b̸	40
A	01	!	41
B	02	"	42
C	03	#	43
D	04	$	44
E	05	%	45
F	06	&	46
G	07	' (quote)	47
H	10	(	50
I	11	)	51
J	12	*	52
K	13	+	53
L	14	, (comma)	54
M	15	-	55
N	16	.	56
O	17	/	57
P	20	0	60
Q	21	1	61
R	22	2	62
S	23	3	63
T	24	4	64
U	25	5	65
V	26	6	66
W	27	7	67
X	30	8	70
Y	31	9	71
Z	32	:	72
[	33	;	73
\	34	<	74
]	35	=	75
↑	36	>	76
←	37	?	77

Direct Commands (cont)

A	A	A	A	A	A	A	A

AAAAAAAA = 01110000 Select Pen 1

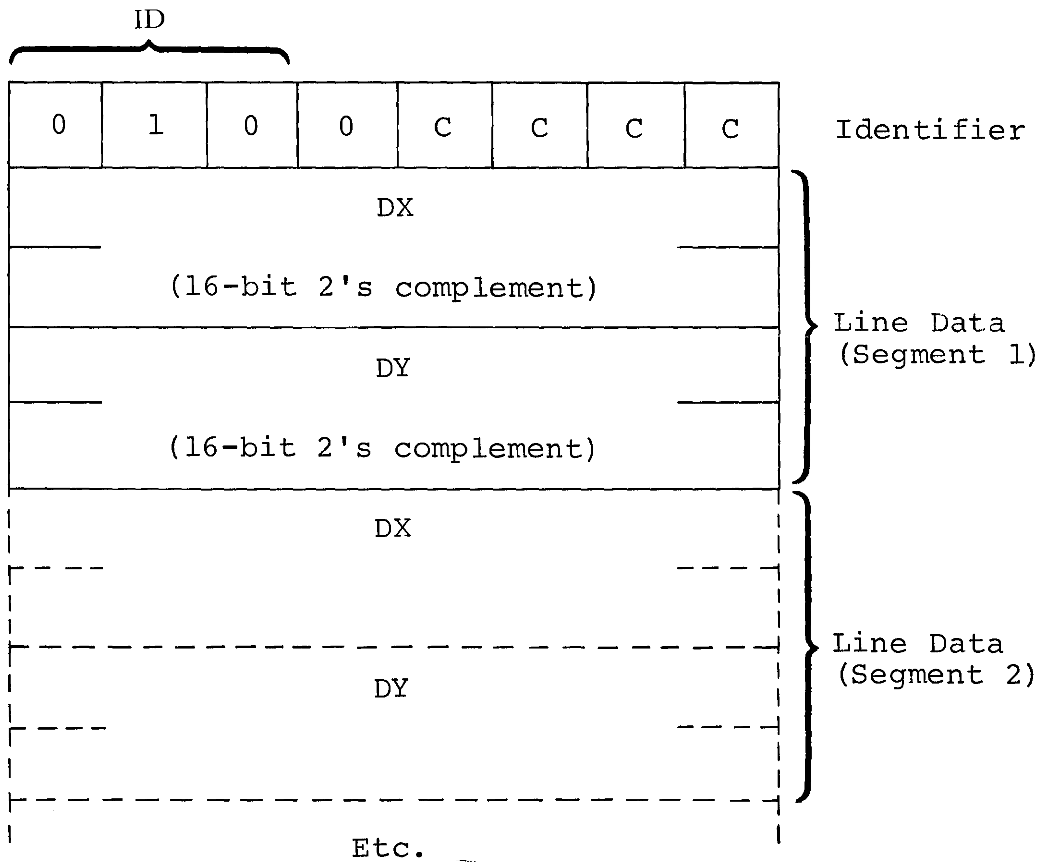
AAAAAAAA = 01110001 Select Pen 2

Single Precision Delta Commands

ID								
0	0	1	0	C	C	C	C	Identifier
DX (Eight-bit 2's complement)								Line Data (Segment 1)
DY (Eight-bit 2's complement)								
DX								Line Data (Segment 2)
DY								
Etc.								

CCCC is the segment count indicating one less than the number of X-Y pairs following the identifier byte (1 to 16 pairs). Line segments up to 127 steps in length may be plotted using single precision data. This is approximately 1.6 millimeters (0.06 inches).

Double Precision Delta Commands



CCCC is the segment count indicating one less than the number of X-Y pairs following the identifier byte (1 to 16 pairs). Line segments up to 32767 steps in length may be plotted using double precision data. This is approximately 400 millimeters (16 inches).

Symbol String Commands

<u>Long Form</u>

ID								
0	1	1	1	C	C	C	C	Identifier
XAX (16-bit 2's complement)								Height and Angle Data
YAY (16-bit 2's complement)								
0	0	6-bit character						Symbol Data
Etc.								

<u>Short (Continuation) Form</u>

ID								
0	1	1	0	C	C	C	C	Identifier
0	0	6-bit character						Symbol Data
Etc.								

CCCC indicates one less than the number of characters in the command (1 to 16 characters).

XAX and YAY define the height and orientation of all characters in the string as follows:

XAX = HEIGHT*STEPS*COS(ANGLE)

YAY = HEIGHT*STEPS*SIN(ANGLE)

Symbol String Commands (cont)

Where:

HEIGHT — is the height of the character in millimeters.

STEPS — is the number of steps/millimeter of the plotter (80).

ANGLE — is the rotation angle of the character from the positive X-axis.

The maximum value permitted for XAX and YAY is 16383 steps. This corresponds to a maximum character height of approximately 200 millimeters (8 inches).

If the XAX,YAY values for a symbol string will be the same as those previously output to the plotter, the Short Form of the command can be used, omitting XAX and YAY. The plotter will use the last XAX,YAY set which it received.

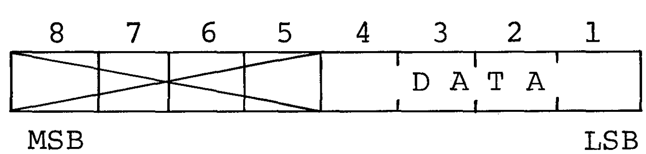
The plotter leaves the pen up at the end of all symbol strings. Table 5-5 indicates the available characters along with the 6 bit octal representation of each character. The octal code is a masked ASCII code.

RS-232/836 SERIAL FORMAT

The CalComp 836 Plotter can be driven online to various computers using an RS-232 serial interface. The data sent to the plotter consists of a serial string of characters. Each character consists of a one start bit, followed by eight data bits, followed by one stop bit. The maximum data rate is 4800 baud. Each character contains one plotter command.

Commands sent by the computer to the plotter are of the single character direct command type, incorporating incremental moves, pen commands and No-ops.

The least significant four bits of each character received by the plotter interface are interpreted as follows:



DATA

(BINARY)	(OCTAL)	FUNCTION
0000	00	+Y
0001	01	+X, +Y
0010	02	+X
0011	03	+X, -Y
0100	04	-Y
0101	05	-X, -Y
0110	06	-X
0111	07	-X, +Y
1000	10	NO-OP
1001	11	PEN UP
1010	12	PEN DOWN
1011	13	NO-OP
1100	14	↕
1101	15	
1110	16	
1111	17	NO-OP

Programming Considerations

- Pen Delays

  The user must provide pen delays external to the plotter. This can be accomplished by issuing a minimum of six (6) consecutive pen-up or pen-down commands whenever a pen operation is desired.

- Limit Switches, Out-of-Paper, and Manual Mode

  If the plotter is driven into a limit switch, the interface continues to accept characters in a normal manner, ignoring those commands which attempt to move in the limit direction. Any commands which direct the plotter to move out of the limit switch will be acted upon normally.

  If the Out-of-Paper condition occurs, or the plotter is placed in Manual mode, the interface will continue to accept characters in a normal manner.

The above conditions have no effect on any of the RS-232-C circuits - while the power is on, the interface will accept commands.

- Character Command Format

  Because the plotter interface uses only the four least significant bits of each command byte, the user may, if he chooses, map the 16 commands onto any ASCII, EBCDIC, or BCD characters which provide the required four bits. For example, the entire command set could be mapped onto the ASCII characters @ through O. On some computers, this may allow the use of existing teletype software drivers to communicate with the plotter.

PLOT SUBROUTINE CONSTRUCTION

The PLOT subroutine effectively functions as the logical interface between the user's programs and the plotting system. It is the only subroutine that actually produces commands to direct the plotter to move. The other support subroutines eventually call PLOT either directly or indirectly. In order to operate at maximum efficiency, this routine is often written in Assembler Language; and, many times, contains several support subroutines as auxillary entry points.

General Flow of PLOT Subroutine

PLOT subroutine consists of six major component sections:

- IPEN Analysis
- XPAGE,YPAGE Conversion
- Line Approximation
- Post Processing of IPEN
- Special Entries
- Output Processing

Table 5-6. Value of IPEN

Block diagram of Figure 5-4 which represents appropriate operation is:	OPERATION	0 1	* 2	* 3	4 to 9	10 11	12	13	14 to 19	20 21	22	23	24 to 29	30 31	32	33	34 to 39	40 to 998	* 999	≥ 1000
A	No change in pen up/down	±				±				±				±						Special Processing of Calls from Support Subroutines
	Lower pen before moving		±				±				±				±					
	Raise pen before moving			±	±			±	±			±	±			±	±	+	+	
B / C	Move to XPAGE, YPAGE	±	±	±	±					±	±	±	±	±	±	±	±	+	+	
	Move to XPAGE, YPAGE after adjusting by offsets					±	±	±	±											
D	Reorigin; set old X and Y to zero	−	−	−	−	−	−	−	−	±	±	±	±	−	−	−	−	+	+	
	Empty buffers if it is necessary	−	−	−	−	−	−	−	−	−	−	−	−	±	±	±	±	+	+	
	Write next consecutive search address	−	−	−	−	−	−	−	−	−	−	−	−	±	±	±	±			
	Write final search address either IBEN, 999, or 9999																	+	+	

Table 5-6. Value of IPEN (cont)

Block diagram of Figure 5-4 which represents appropriate operation is:	OPERATION	0 1	* 2	* 3	4 to 9	10 11	12	13	14 to 19	20 21	22	23	24 to 29	30 31	32	33	34 to 39	40 to 998	* 999	≥ 1000
D	Close output file																	+	+	Special Processing of Calls from Support Subroutines
	Undefined operation																	-	-	-

* These five values are supported by all PLOT subroutines; operations defined by other values of IPEN are not necessarily implemented on various sytems.

General Flow of PLOT Subroutine (cont)

These sections have the relationship indicated in Figure 5-4 and are discussed in the following paragraphs.

IPEN Analysis

The first section of the PLOT subroutine (block A) analyzes the IPEN parameter using its sign and value to determine whether the pen is up or down while moving and certain other operations. The various operations and values allowed for IPEN are shown in Table 5-6; however, only the values ±2, ±3, and 999 are always recognized in all CalComp PLOT subroutines. The other values are not implemented in most PLOT subroutines.

XPAGE, YPAGE Conversion

In block B, the XPAGE,YPAGE coordinates are multiplied by the current value of FACT, which is supplied through subroutine FACTOR. FACT is initially set to 1.0 until changed by the user's program. The coordinates are then converted from inches or centimeters to increments by multiplying by the constant STEPS (number of increments per inch or centimeter) and rounding to the nearest increment. An increment generally corresponds to one plotter increment or in some systems a fraction of the plotter increment. The incremental coordinates are then used with the previous incremental coordinates to determine the delta increments (Delta X,Delta Y), which represents the line segment to be drawn.

Line Approximation

Block C represents the heart of the PLOT subroutine, the Line Approximation Section. Since CalComp plotters are digital incremental plotters, each line segment must be plotted using some type of approximation. The next section covers some approximation techniques.

Post Processing

Block D represents the section of the PLOT subroutine which handles any postprocessing operations implied by the IPEN values. These include defining a new origin, filling out partial plotting records, writing a Search Address Record, and closing the output file when necessary.

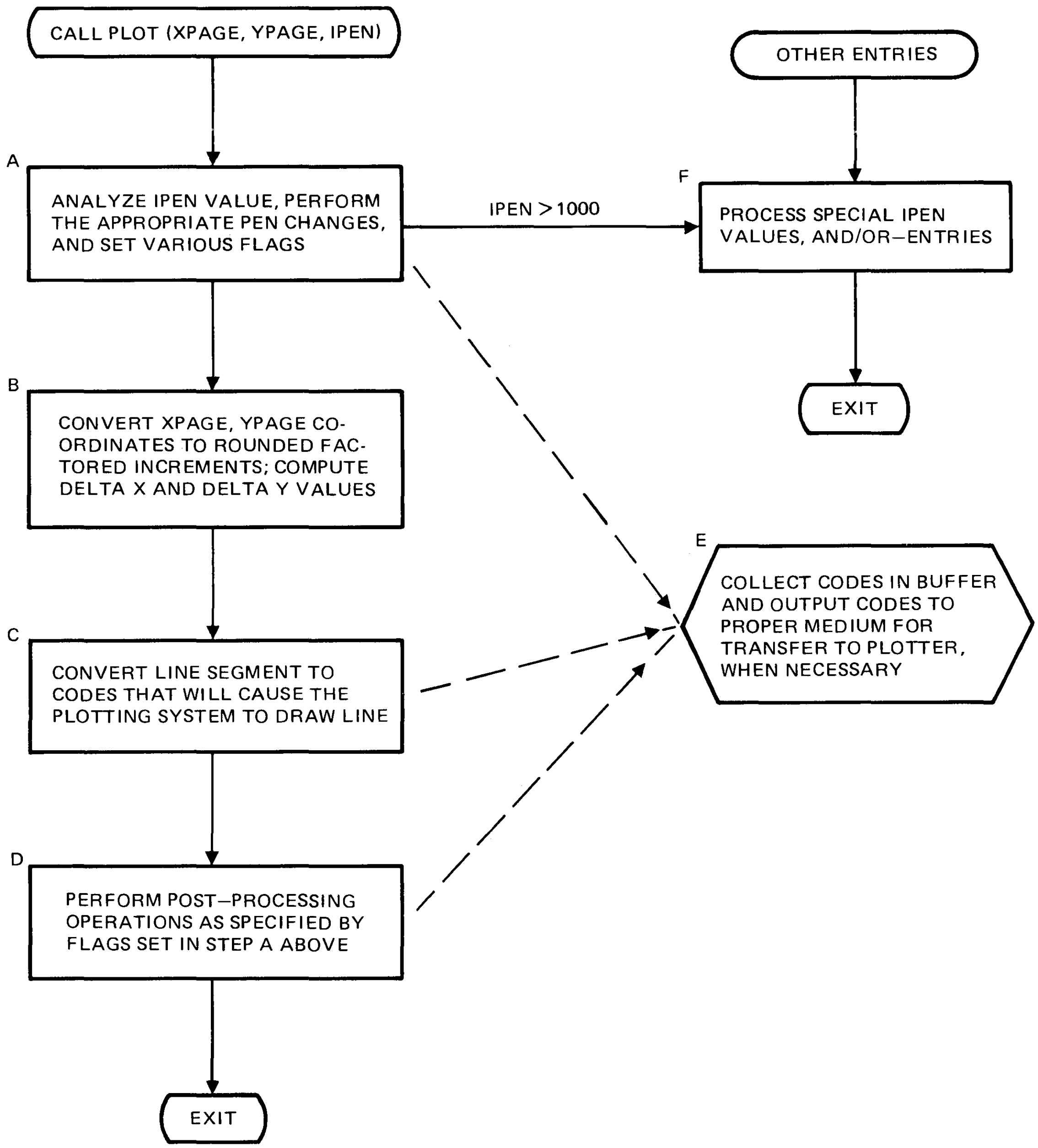


Figure 5-4. General Flow of PLOT Subroutine

Special Entries

Several of the Basic Software Subroutines need access to the PLOT subroutine other than generating a movement. This is handled by either having multiple entry points to the subroutine or assigning special values of IPEN to branch to various functions.

These special functions, represented by block E, include setup of factors, returning current coordinates, opening output files, and handling special delta values. The functions may use other portions of the PLOT subroutine.

Output Processing

The Output Processing component (block F) performs or calls upon Subroutine BUFF to perform the particular output method for the system configuration. This section can do buffering of data, code transactions, initiate output, service any interrupts, etc.

LINE APPROXIMATION ALGORITHMS

There are many line approximation algorithms used in CalComp Software. All the algorithms have the same basic goal and that is to move the pen (or beam of light) from one point to another in the smoothest, fastest possible manner. The particular algorithm used is dependent upon the data format used by the plotting system and the arithmetic capability of the computer which is generating the format. Three commonly used algorithms are described in the following paragraphs.

8-Vector Algorithm

The Basic Line Approximation Technique for all CalComp plotting systems is the 8-vector incremental algorithm. Some systems have this algorithm implemented in hardware or microprocessors. The command structure for a plotter allows it to move in one of eight directions with each increment. Figure 5-5 illustrates the 8-vector movements of the plotter. Any line can be plotted exactly if it is parallel to one of the basic plotting directions. At any other angle, a line must be approximated by combining a series of the two incremental vectors that lie on either side of the "true line". In the 8-vector structure, these two vectors are:

- Either a major ±X or ±Y vector or
- A diagonal vector

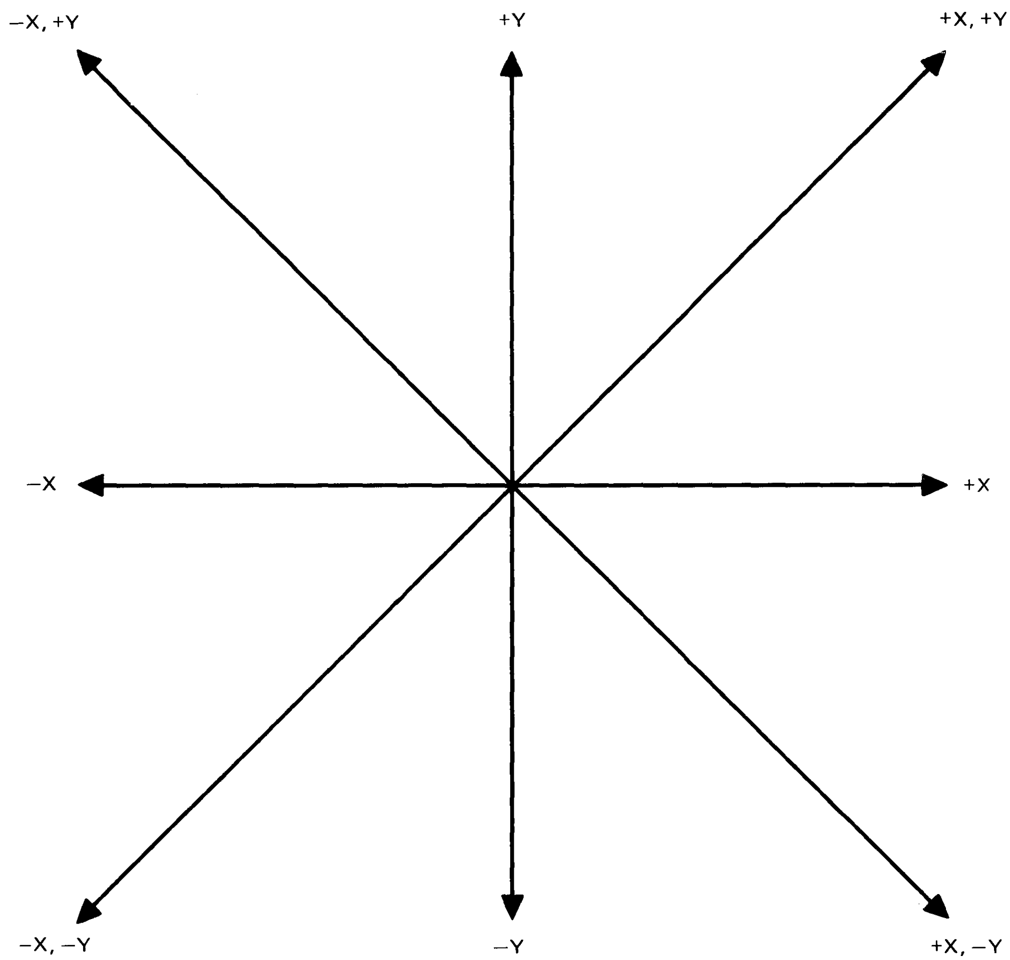


Figure 5-5. 8-Vector Structure

8-Vector Algorithm (cont)

Therefore, the plotted line consists of a series of only two basic vectors: one in a major direction and the other in a diagonal direction. In the worst case, the end point of the plotter line deviates from the true line by 1/2 of an increment. Consider the problem of moving from one point to another point. This will result in some vector DX,DY. The algorithm can be simplified by transforming this vector to the first octant (0° to 45°). This can be done by first determining the octant of the resultant vector of DX,DY. The appropriate PLOT command codes for the major axis pen movement and the diagonal pen movement are then preselected. If the signs or values of DX,DY are exchanged, the incremental algorithm can logically function as if the resultant vector were always between 0° and 45°. Thus, for a vector in the third octant (90° to 135°) the major axis command is +Y and the diagonal command is -X+Y. The major axis is the larger of the two vectors, (DY); the minor axis is the smaller vector (DX). The problem of moving from point (XA,YA) (the current position of the pen) to any point (XC,YC) can be reduced to moving to a point (XB,YB) with an angle less than 45°. Figure 5-6A illustrates this situation by an enlarged grid where:

- Each square (I) represents one increment of movement.
- DX and DY represent the total number of increments that must be taken on the X and Y axes.
- The vector diagram indicates the possible movements at each step.
- The true line is represented by the dashed line.

In this example, since DX is greater than DY, there is always an X-movement, but there is also a Y-movement (resulting in a diagonal XY-movement), whenever the combined movement is needed to keep the plotted line as close as possible to the true line. Figure 5-6C shows the sequence of moves (on an exaggerated scale) that approximates the true line.

The basic problem that is solved by the incremental algorithm is to determine the correct combination of major and diagonal movements that best approximates the line. This is determined at each step of the approximation by deciding whether a major move or a diagonal move will bring the pen closer to the true line. In Figure 5-6B, the pen has made four moves and is at point (XP,YP), ready for move 5 (step N). If D, which is the difference between YT_N (the Y value of the true line at step N) and YP is not greater than half of an increment (I/2), a major (X) move should be taken. If D is greater than I/2, a diagonal (XY) move should be made.

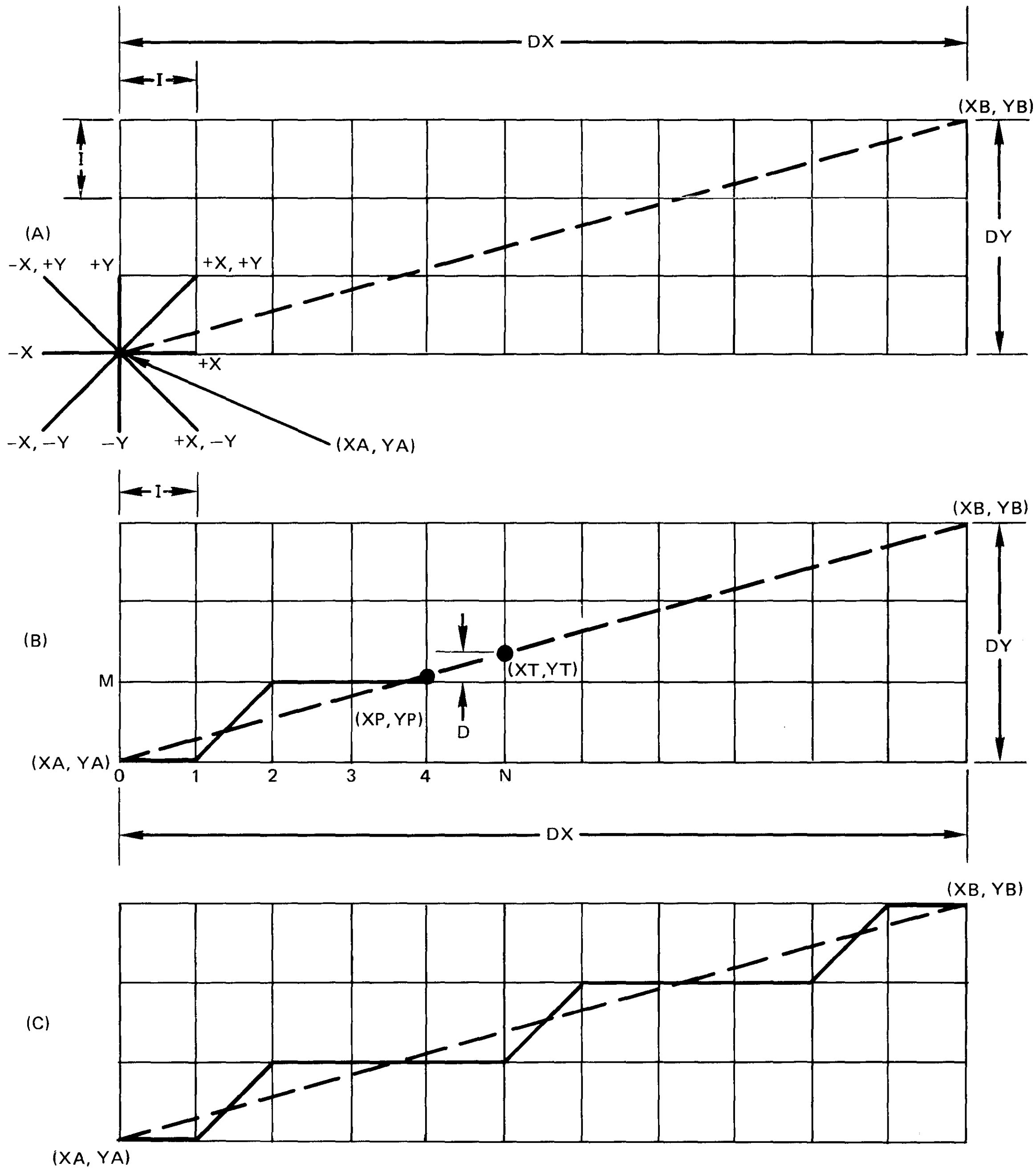


Figure 5-6. Solution of a Straight-Line Approximation

The evaluation of D versus I/2 is the key to the solution. That is, when $D \leq I/2$, a move should be taken in the major direction. Since:

$$D = YT_N - YP$$

we must first compute YT_N. This can be found using DY/DX as the slope of the true line, and using the number of major moves (N) times the increment size (I) as the X-distance of the true line:

$$YT_N = (N*I)\ \frac{DY}{DX}$$

YP is the sum of the diagonal moves (called M) that contain a Y component, times the increment size (I):

$$YP = M*I$$

Therefore:

$$D = (N*I)\ \frac{DY}{DX} - (M*I).$$

When this expression is compared to I/2, it may be reduced:

$$D : I/2$$

$$\frac{I\ (N*DY-M*DX)}{DX} : \frac{I}{2}$$

Which is:

$$N*2*DY-M*2*DX : DX.$$

Transposed:

$$0 : DX-N*2*DY+M*2*DX$$

When the evaluation of this expression is positive (zero), a major axis move is required. If the evaluation is negative, a diagonal move is required. In the coding, DX and DY represent the number of steps to be taken in the major and minor axes. N represents the cumulative number of major axis moves up to any given step. Since a diagonal move is composed of both major and minor axis moves, a major axis move occurs at every step.

Therefore:

$$N*2*DY$$

can be obtained by accumulating the sum of 2*DY (negatively, for the expression) at every step. M represents the cumulative number of minor axis moves up to any given step.

8-Vector Algorithm (cont)

Consequently:

$$M*2*DX$$

is obtained by accumulating the sum of 2*DX each time a diagonal move occurs.

> NOTE
>
> In the equations, an asterisk (*) is used to denote a multiplication operation.

The 8-vector Line Approximation Algorithm is summarized in the flow charts shown in Figure 5-7 (Sections A and B).

Line Splitting Algorithm

Most of the formats used to transmit data from the user's Host Computer to the CalComp Plotting Systems utilize some form of delta representation. The delta representations take on several forms depending upon the plotting system. Some representations accept standard two's complement DX,DY values, other representations require an octant number and positive major and minor axes values. Some representations accept a repeat count along with the deltas. The deltas are limited in magnitude by the format and at times by the word size of the Host Computer. Figure 5-8 shows the flow chart of a Line Splitting Algorithm that will convert a line segment into a series of smaller line segments, each equal to or less than the maximum. This algorithm would be modified if the segments should be of equal length or if the Host Computer has a large (greater than 24 bits) integer format.

Variations of this algorithm are found in the software for different CalComp Plotting Systems.

Velocity Calculations

Many CalComp Plotting Systems operate at varying velocities which are usually dependent upon the length of the line segment (the straight distance between two consecutive points or a single delta). Some systems with limited velocity control, handle each line segment independently i.e., each segment starts and ends at the minimum velocity and in between accelerates to the best velocity. Other systems (those with a large range between minimum and maximum velocity), operate too slowly in this mode. For these systems, velocity must be maintained higher than the minimum between line segments.

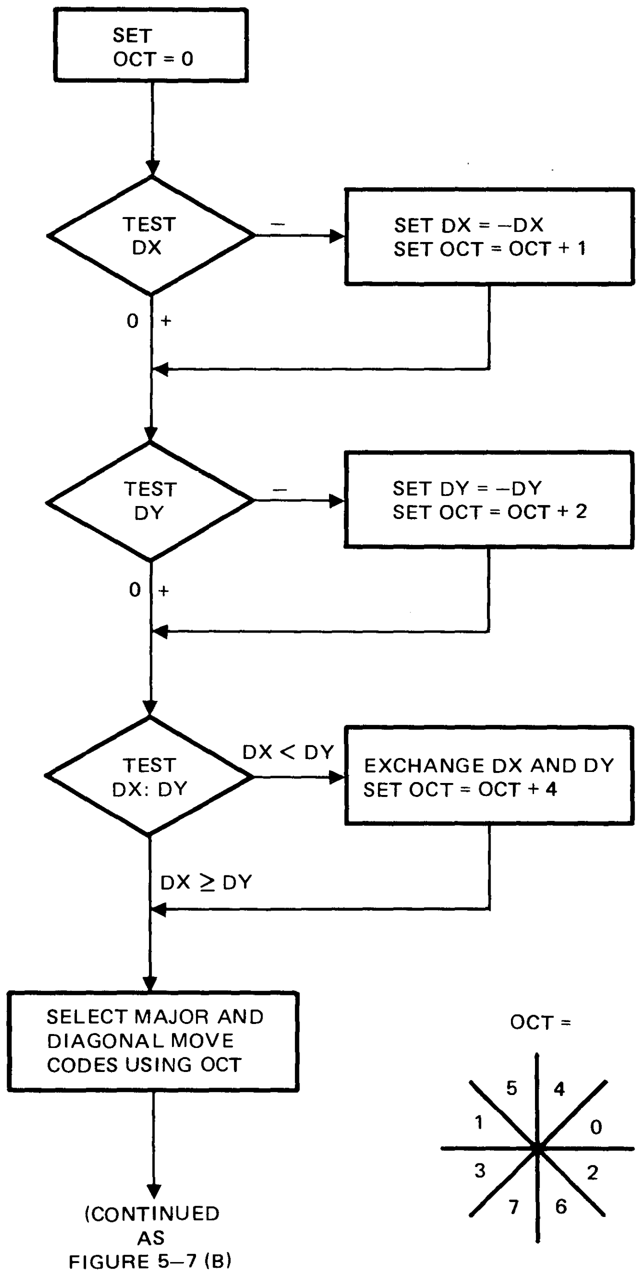


Figure 5-7(A). 8-Vector Incremental Algorithm Functional Diagram

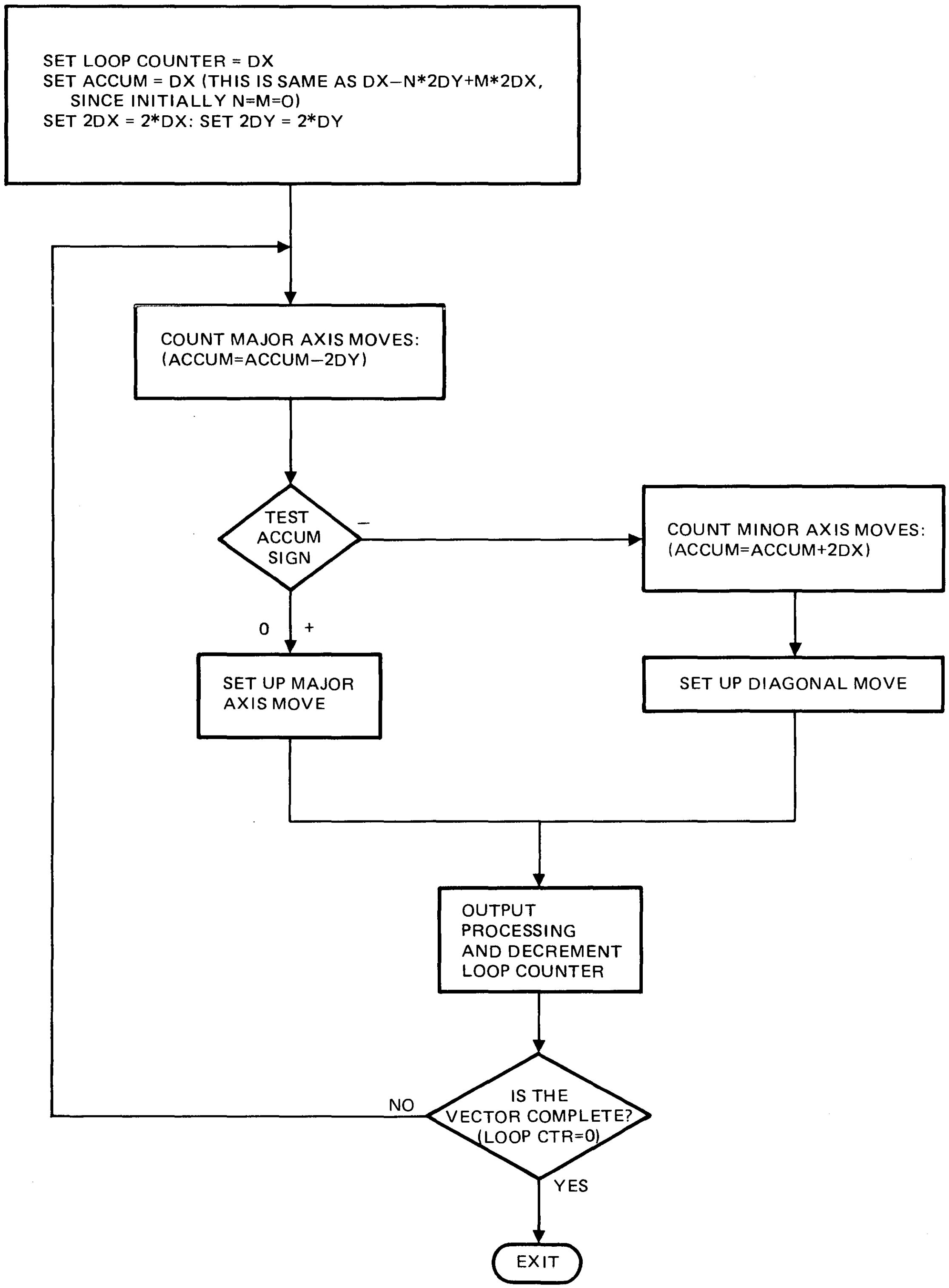
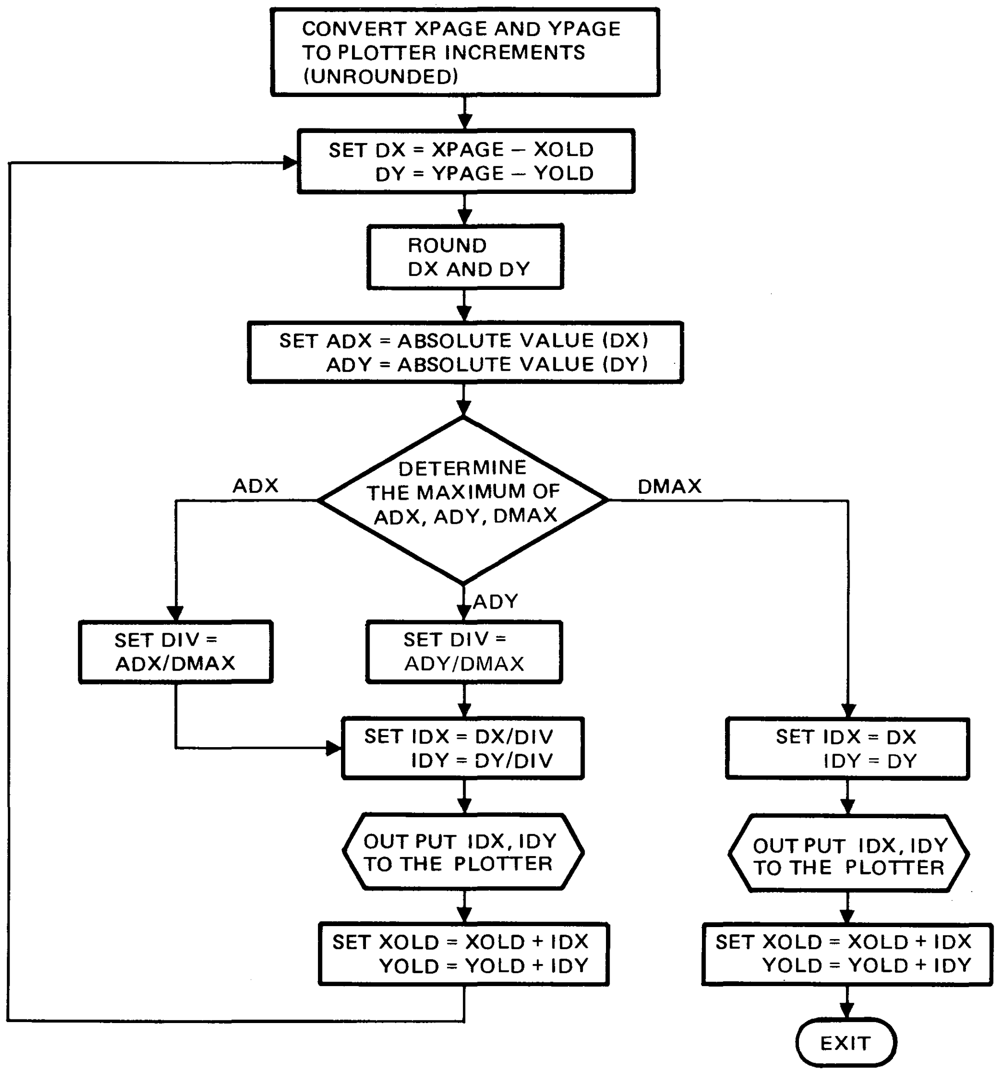


Figure 5-7(B). 8-Vector Incremental Algorithm Functional Diagram (cont)



NOTES:

1. XOLD AND YOLD ARE INITIALIZED TO ZERO AND REPRESENT THE TRUE INTEGRAL NUMBER OF STEPS TAKEN.

2. COMPUTE THE DELTA AS THE DIFFERENCE BETWEEN THE NEW POSITION AND THE CURRENT POSITION.

3. ROUNDING IS DONE BY ADDING $\pm$0.5.

4. DMAX IS THE LARGEST VALUE THAT CAN BE SENT TO THE PLOTTER OR CAN BE HELD IN AN INTEGER WORD.

5. IDX AND IDY ARE CONVERTED TO INTEGERS BY TRUNCATION.

6. THIS IS EITHER DIRECT OR THROUGH SOME OTHER ALGORITHM.

7. THE CURRENT POSITION IS UPDATED BY THE DELTA OUTPUT IN PREVIOUS STEP.

8. EITHER REPEAT THE OPERATION OR EXIT THE ROUTINE.

Figure 5-8. Line-Splitting Algorithm

Velocity Calculations (cont)

In order to maintain a high velocity along a series of line segments, the software collects and examines groups of line segments. The software considers such factors as:

- The angle between adjacent line segments
- The length of each line segment
- Points at which the plotter must halt (pen state change, end of buffer, etc.)

Various parameters of the plotting systems such as the velocities available, the acceleration (including radial acceleration) limits, and the distances needed to change velocities are also considered. The software determines the start, end, and maximum velocities of each segment by using all these considerations and making several passes over the collected line segments.

Output Processing

CalComp Plotting Systems use several media to pass data from the Host Computer to the plotter. The type of media available includes:

- Industry standard magnetic tape
- Punched cards
- Serial communication lines
- Direct parallel interfaces

The type of media and the characteristics of the Host Computer determine the level of output processing that has to be done.

In large scale computing systems using magnetic tape output, most buffering, error processing, and interrupt processing are handled by the operating system. On the other end of the spectrum are small online plotting systems where CalComp Software handles the buffering, error processing and interrupt processing.

There are several types of buffering used by the PLOT subroutines to store and forward the plot commands to the output device. Unbuffered output is used to transmit data, a character at a time, to online plotting systems which are basically dedicated to plotting and do not have direct memory output (DMA, DMC, etc.). Single buffered output is used in most

Output Processing (cont)

tape systems to send a fixed length work area of plotting data to the tape. This is generally used when the operating system does its own buffering, when output cannot be overlapped with processing, or on multiprocessing systems. Double buffering is used when output can be overlapped with processing and one buffer can be filled with plotting data while the other is being output. All buffering techniques are limited by the record size of the output media and the plotting system.

Error processing and interrupt handling are performed by the operating system and only by the CalComp Plotting Software on small computers with online plotting systems.

The output processing section must ensure that all plotting data is sent to the plotter. This is important with buffered data. The PLOT subroutine dumps all buffers after specified calls by either outputting short records or filling records with dummy plot information.

Other PLOT Entry Points

Closely associated with the function of the PLOT subroutine are four other subroutines. Where possible, these are written as other entry points in the PLOT subroutine or as separate subroutines which call the PLOT subroutine with special IPEN values. These subroutines are:

- PLOTS
- FACTOR
- WHERE
- NEWPEN

PLOTS Subroutine

The PLOTS subroutine is the initialization subroutine. It should be the first subroutine called since it performs the following functions:

- "Opens" the output media
- Allocates buffer space
- Initializes variables and buffers
- Writes Search Address Record 1

The exact functions of the various parameters in the calling sequence for the PLOTS subroutine vary depending upon the Host Computer and the plotting system.

FACTOR Subroutine

The FACTOR subroutine magnifies or reduces the effective value of all succeeding plot coordinates. In order to do this, the STEPS constant (the number of increments per unit of measure) is adjusted by the factor passed.

This is not a cumulative factor, but is replaced by each new call.

WHERE Subroutine

The WHERE subroutine returns to the user, the values in the PLOT subroutine for the current pen position in the user's coordinate system together with the current factor. The values may be rounded or the values may be as exactly as passed in the previous call, such that if they were immediately used in a call to PLOT, no movement would result.

The subroutine takes the current pen position in increments and converts it back to user coordinates by using the adjusted STEPS constant.

The result of using the WHERE subroutine following a call to the SYMBOL subroutine depends on where and how the symbols are generated. If the symbols are generated as vectors in the Host Computer, then WHERE returns the last plotted point in the symbol string. However, if the symbol string is passed to the plotting system, then WHERE might return the coordinates of the starting point of the string or the starting point of the next character.

NEWPEN Subroutine

The NEWPEN subroutine selects one of the plotter's pens. The selection of a new pen by the NEWPEN subroutine results in the following events:

- If any pen is down, it is raised
- The newly designated pen is selected
- The new pen is moved to the same physical location occupied by the previous pen

The facts that the newly selected pen must be moved into the position of the previously selected pen, and that the pens are spaced 0.6 inches (1.52 cm) apart, require the user to know where the pen carriage is at the time a selection is made.

EXAMPLE: The pens are mounted in a row along the beam in what is defined as the Y-axis direction. If pen 1 (rightmost pen) is selected and moved near the right limit of the plotting surface, and pen 4 is then selected; the pen carriage must move 1.8 inches in order to bring pen 4 into the position that pen 1 occupied. The pen carriage's proximity to the limit of the plotter means that the move could not be made and a resulting loss of origin would occur.

The PLOTS subroutine initially selects pen 1.

For plotting systems with only one pen, the NEWPEN subroutine is not supplied or calls may be ignored.

The data passed to some plotting systems includes the pen select codes and the associated movement while other systems just pass the pen number.

SYMBOL SUBROUTINE

The SYMBOL subroutine converts a string of alphanumeric and/or special characters into data to direct the plotter to draw that string of characters. All CalComp SYMBOL routines follow the same steps in producing symbols as shown in Figure 5-9, although not all steps are executed in the user's Host Computer. These steps can be divided among the following:

- Host Computer
- Programmable Controller
- Plotting Hardware

Computations for height and angle are done in the Host Computer and generation of vectors from offset tables are done in the controller (or hardware). The following paragraphs describe the design of SYMBOL offsets, the makeup of the offset tables, and the computations to generate symbols.

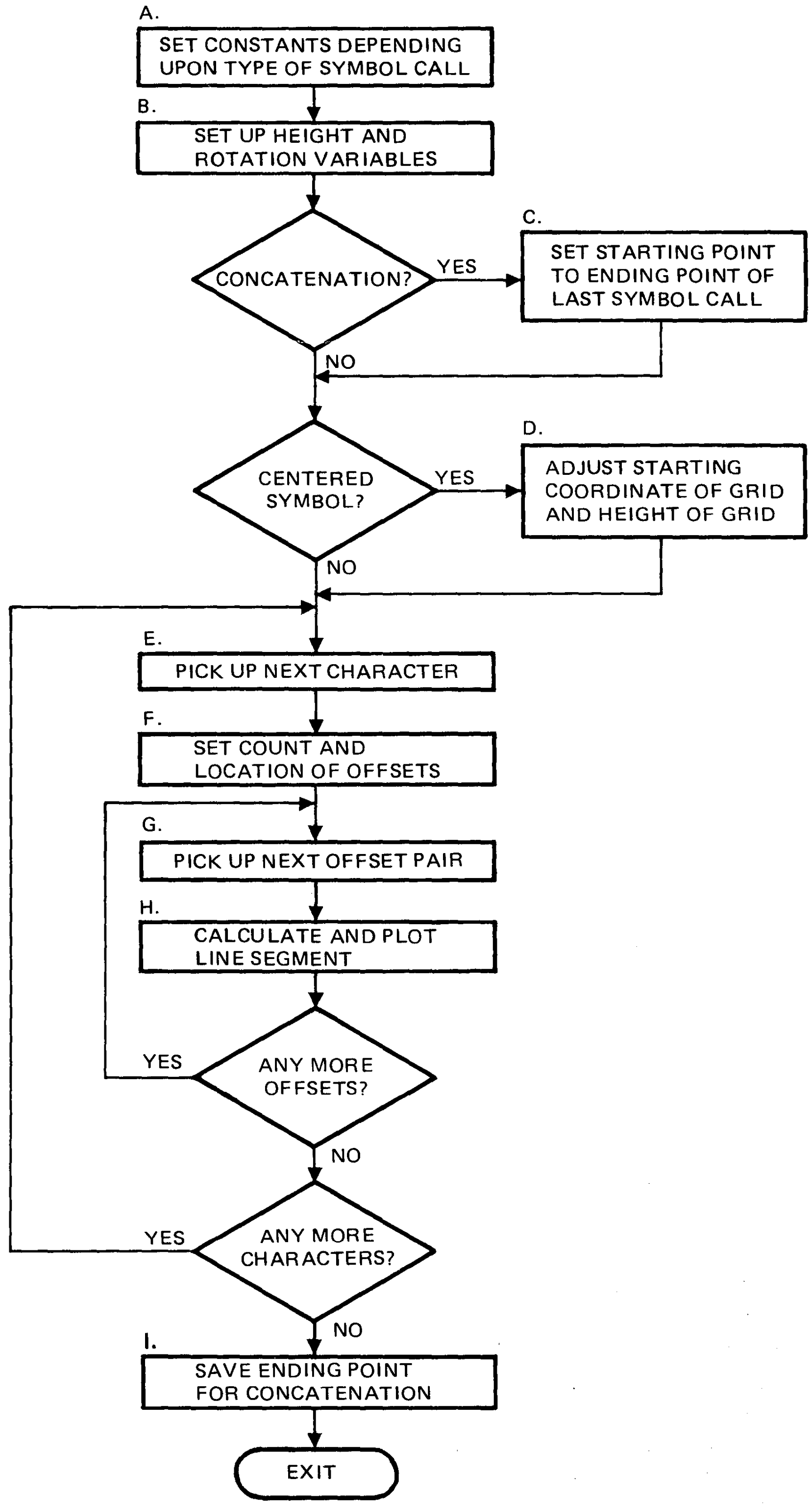


Figure 5-9. SYMBOL Subroutine Flowchart

SYMBOL Design

Symbols generated by the SYMBOL subroutine are designed using a grid coordinate system on which each character of the desired set can be described by straight-line segments connecting grid points (see Figure 5-10). The standard symbol set is designed on a 7 x 7 grid. Special sets are designed on larger grid systems, e.g., the CAL EDIT symbol set is designed on a 15 x 15 grid.

A character is represented by a series of grid coordinates (NX,NY) called offsets. These offsets are integer values representing: end points of the line segments that compose the character; and pen up (or beam off) codes. The ability to plot discontinuous characters (characters whose parts cannot be connected by a continuous line, e.g., the character =) is provided by allowing a special value of NX to be interpreted as follows:

> Lift the pen and position it at the coordinates specified by the next offset pair. Lower the pen and proceed with plotting the character.

An NX value of 7 is used in Figure 5-10. Standard symbols are designed using a 4 x 7-unit grid. The lower left corner of the grid is considered (0,0). Figure 5-10 also shows how the symbol "M" is plotted, illustrating its positioning on the grid by its defining offsets. The grid design and offsets of other standard symbols are further shown in Figure 5-10. Note that the first seven offsets of the symbol R also represent symbol P. Many such relationships exist and can be used to minimize the total number of offsets required to define a character set. When plotting or displaying data points on a curve, it is desirable to have available symbols centered at specified coordinates. These centered symbols should be defined on a square grid. A 4 x 4-unit grid (Figure 5-10) is used in designing centered symbols. The offsets of a centered symbol are defined with respect to an origin at the lower left corner of the symbol. The offsets defining the symbol always start and end at its center (2,2).

Symbols used with electronic (CRT) plotters must not contain any retraced lines, because retracing effectively boosts beam intensity and creates an undesirable effect of spottiness wherever lines are retraced. To avoid retracing lines, the beam is turned off at line intersections. Because of this, more offsets are needed for a symbol generated on a CRT plotter than for a symbol generated on an electromechanical plotter.

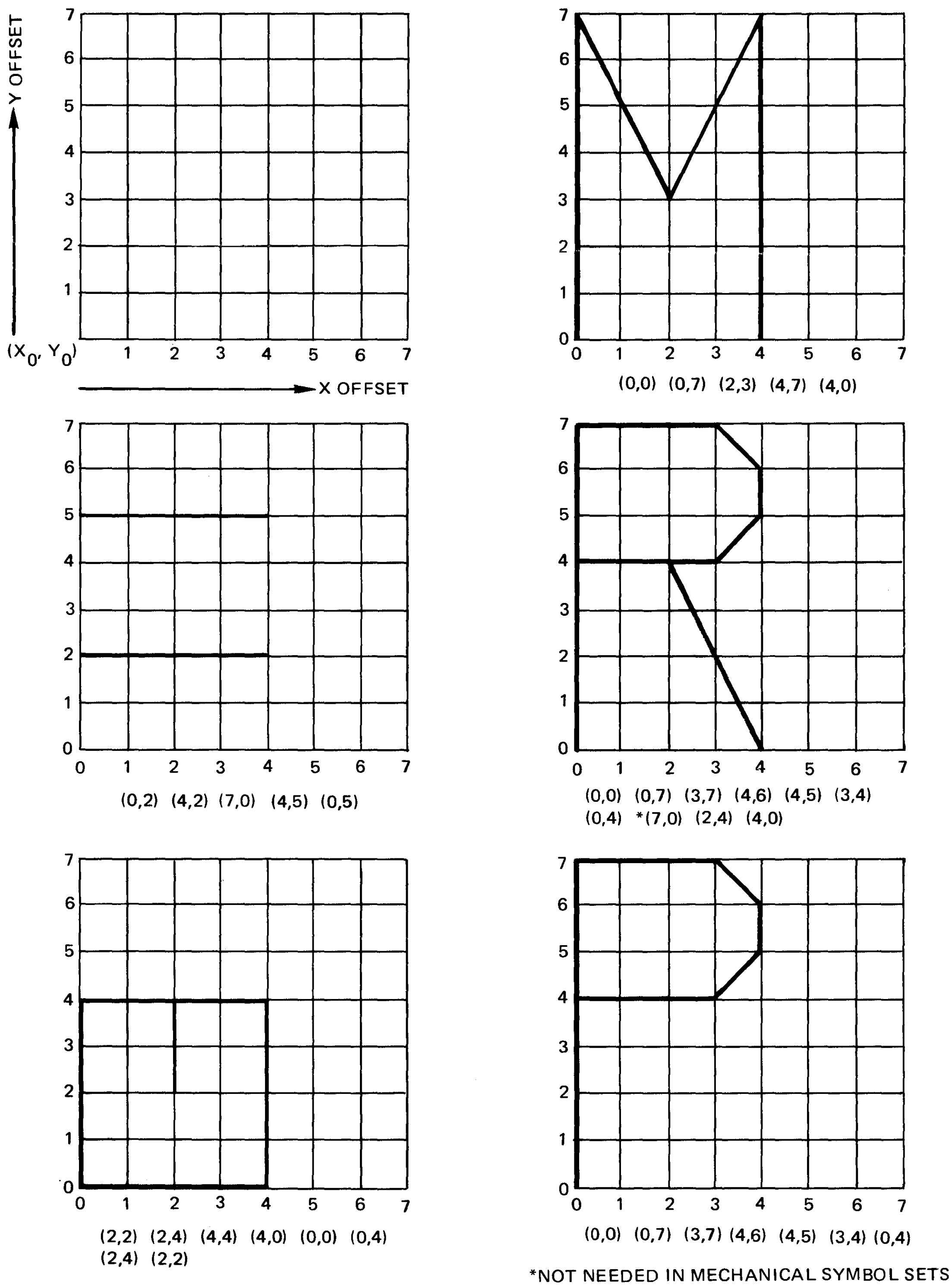


Figure 5-10. SYMBOL Grid System

SYMBOL Selection Tables

Every standard character in a computer has a unique internal numeric representation. The term BCD (binary coded decimal) may be used to describe this type of character representation. An integer equivalent is assigned by the SYMBOL subroutine to every symbol the subroutine can produce. There may be several characters that have integer equivalents but not BCD equivalents, since the character set of the SYMBOL subroutine is larger than the standard computer character set.

If the last parameter in the call to SYMBOL is positive (NCHAR), symbols are generated using the BCD equivalents. NCHAR represents the number of characters to be produced. Each character is isolated. Processing continues sequentially through the string of BCD character until NCHAR symbols have been produced. BCD information must be continuous for all sequential character positions. In computers with word formatting, the first character must be in the leftmost character position of the first word of IBCD.

If the last parameter in the call to SYMBOL is zero, a single symbol is generated using the BCD equivalent. The BCD information is assumed to be in the rightmost character position of the word IBCD.

If the last parameter in the call to SYMBOL is negative, a single symbol is generated, using its integer equivalent (INTEQ). If INTEQ in the call statement is less than 15, a centered symbol is produced, and SYMBOL assumes a translated origin when using the symbol's offsets.

Figure 5-3 shows the symbols available in 925 controller-driven plotting systems. The SYMBOL subroutine uses two tables in generating symbols:

- Address Table
- Offset Table

The Address Table has two sections. The first section is associated with integer equivalents, and the second section is associated with BCD equivalents. The integer-equivalent section extends through and contains the entire second section, giving each symbol a unique integer equivalent.

When a single symbol is generated, its numeric value (either integer or BCD equivalent) is used to determine a position in the Address Table. The Address Table contains the position in the Offset Table where the offset pairs defining the character begin. This table also contains the number of offset pairs needed to draw the character.

SYMBOL Selection Tables (cont)

If the sign of the last parameter in the call to SYMBOL is positive, (or if the parameter's value is zero), the starting location of the Address Table's BCD section is used. If the sign of the last parameter is negative, the starting location of the Address Table's integer-equivalent section is used. This starting address is indexed by the symbol's numeric value to determine the symbol's location in the Address Table to be used.

The BCD section of the Address Table must have a 1:1 correspondence with the BCD values of the computer.

> EXAMPLE: If the character J is represented internally as 33_{10}, the thirty-third location after the start of the BCD section must contain the starting location of the offsets that define J.

Many computers may contain BCD values that have no standard-character counterpart. For these values, the corresponding location in the Address Table may apply to nonstandard characters.

The first locations in the integer-equivalent section of the Address Table are usually used for defining centered symbols. After the centered symbols, any number of nonstandard symbols may be used.

The Offset Table contains the offset pairs that define the symbols in the SYMBOL character set. Overlapping of offset pairs is used whenever possible.

SYMBOL Generation

The SYMBOL subroutine permits the user to draw symbols at various sizes and orientation. To do this the user supplies the height (in inches or centimeters) of the character and the angle (in degrees) of rotation. In some systems an aspect ratio is also allowed. This allows the user to modify the width of a character. Using these factors, we calculate the X and Y components of a horizontal and vertical grid section where:

M — is the number of grid units in the character (7 for standard character and 4 for centered characters).

1) XAX=HEIGHT*COS(ANGLE)*ASPECT/M

2) XAY=HEIGHT*COS(ANGLE)/M

3) YAY=HEIGHT*SIN(ANGLE)/M

4) YAX=HEIGHT*SIN(ANGLE)*ASPECT/M

> NOTE
>
> If ASPECT=1.0 then XAX equals XAY and YAY equals YAX (which is the normal case).

These equations are similar to those in paragraphs <u>Symbol Mode Control Codes</u> and <u>Extended Symbol Mode Control Codes</u> (pages 5.13 and 5.14). The equations are illustrated in Figure 5-11.

The distance from grid point (0,0) to any other grid point (NX,NY) is (XM,YM) calculated as follows:

5) XM=NX*XAX+NY*(-YAY)

6) YM=NX*YAX+NY*(XAY)

If the origin of the grid is at (X0,Y0), the page coordinates of grid point (NX,NY) are (XN,YN) where:

7) XN=X0+XM

8) YN=Y0+YM

or by substituting:

9) XN=X0+NX*XAX-NY*YAY

10) YN=Y0+NX*YAX+NY*XAY

EXAMPLE: Use the R shown in Figure 5-11 and substitute a height of 7.0, an Aspect of 1.25 and an angle of 12° to get:

XAX=7.0*COS(12°)*1.25/7=1.22

XAY=7.0*COS(12°)/7=.98

YAY=7.0*SIN(12°)/7=.21

YAX=7.0*SIN(12°)*1.25/7=.26

SYMBOL Generation (cont)

Using these numbers, the offsets in Figure 5-11 and starting the R at (0.0) results in the following calls to the PLOT subroutine:

```
CALL PLOT(0.0,0.0,3)          XN=0.0+0*1.22-0*.21
                              YN=0.0+0*.26+0*.98

CALL PLOT(-1.47,6.86,2)       XN=0.0+0*1.22-7*.21
                              YN=0.0+0*.26+7*.98

CALL PLOT(2.19,7.64,2)        XN=0.0+3*1.22-7*.21
                              YN=0.0+3*.26+7*.98

CALL PLOT(3.62,6.92,2)        XN=0.0+4*1.22-6*.21
                              YN=0.0+4*.26+6*.98

CALL PLOT(3.83,5.94,2)        XN=0.0+4*1.22-5*.21
                              YN=0.0+4*.26+5*.98

CALL PLOT(2.82,4.70,2)        XN=0.0+3*1.22-4*.21
                              YN=0.0+3*.26+4*.98

CALL PLOT(-.84,3.92,2)        XN=0.0+0*1.22-4*.21
                              YN=0.0+0*.26+4*.98

CALL PLOT(1.6,4.44,3)         XN=0.0+2*1.22-4*.21
                              YN=0.0+2*.26+4*.98

CALL PLOT(4.88,1.04,2)        XN=0.0+4*1.22-0*.21
                              YN=0.0+4*.26+0*.98
```

For centered symbols, the origin of the grid is translated. Since the grid used for centered-symbol design is 4 x 4 units, the coordinates of the lower left corner are (-2,-2) in relation to the center.

Substitution of these values using equations 9 and 10 gives the translated origin (X0,Y0), where:

```
X0=X0+(-2)*XAX-(-2)*YAY
Y0=Y0+(-2)*YAX+(-2)*XAY

          or

X0=X0-2(XAX-YAY)
Y0=Y0-2(YAX-XAY)
```

This translation is performed internally in the SYMBOL subroutine whenever a centered symbol is specified, so that all grid coordinates are defined relative to the lower left corner. Spacing between adjacent characters is based on the height of the characters. The distance from the origin of one character to the origin of the next is (XT,YT) as follows:

11) XT=7*XAX

12) YT=7*YAX

SYMBOL Generation (cont)

After generation of each character, these values of XT and YT are added to the previous origin so that the origin of the next character to be drawn is always maintained. This new origin is used in lieu of the X and/or Y value when a call to SYMBOL with XPAGE and/or YPAGE equal to 999.0 is used. This is called the Concatenation Feature.

> NOTE
>
> Those versions of SYMBOL produced before September 1967, base the spacing on 6/7 of the HEIGHT, so that XT=6*XAX and YT=6*YAX. The concatenation feature is activated in those versions when XPAGE or YAPGE=-0.

AXIS, LINE, NUMBER, and SCALE Subroutines

Flowcharts for the remaining Host Computer Basic Software Subroutines are shown in Figures 5-12 through 5-15. These programs are normally written in FORTRAN and vary little among computers or plotting systems.

Subroutine AXIS calls PLOT, NUMBER, and SYMBOL subroutines. Using the FIRSTV and DELTAV, it prints numbers every inch (2.0 centimeters) along the length of the axis with a centered title and then plots the axis line with tic marks every inch (centimeter). The subroutine needs modifications to change from English to Metric Units and may require modification of the Hollerith Constant for certain computers. The numbers will be 0.1 inches (0.21 cm) high with title of 0.14 inches (0.28 cm).

Subroutine LINE calls upon the PLOT, SYMBOL, and WHERE subroutines. Using the scale factors placed in the array by SCALE, it determines the closest end of the line to the current pen position, moves there, and draws a line through the scaled data points placing centered symbols on the line as requested. The subroutine only needs a change in symbol height (from 0.08 inches to 0.19 cm) to change from English to Metric Units.

Subroutine NUMBER calls only upon the SYMBOL subroutine. This is sometimes written in assembly where necessary to improve the floating point to BCD conversion. The NUMBER subroutine will vary between computers but the plotting system determines whether subroutine SYMBOL is called with each character or is called with a string. The NUMBER subroutine converts a real

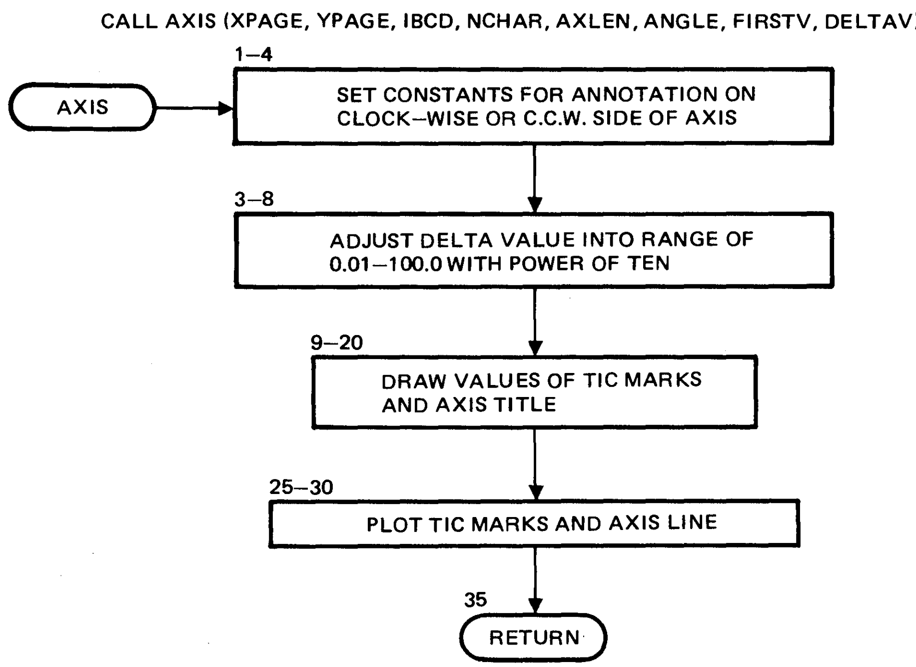


Figure 5-12. AXIS Flowchart

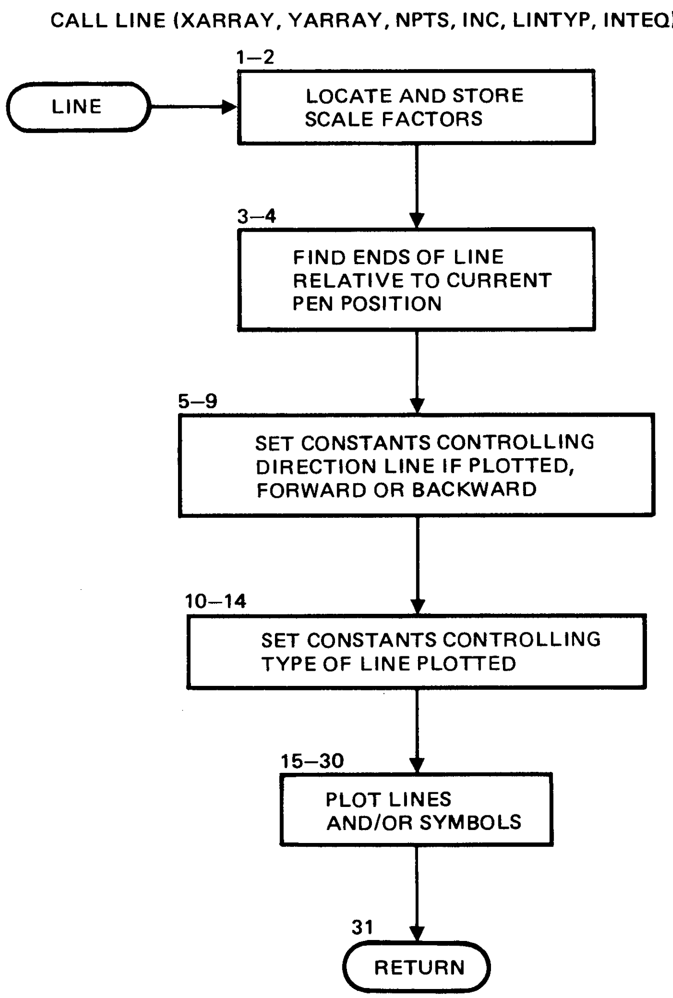


Figure 5-13. LINE Flowchart

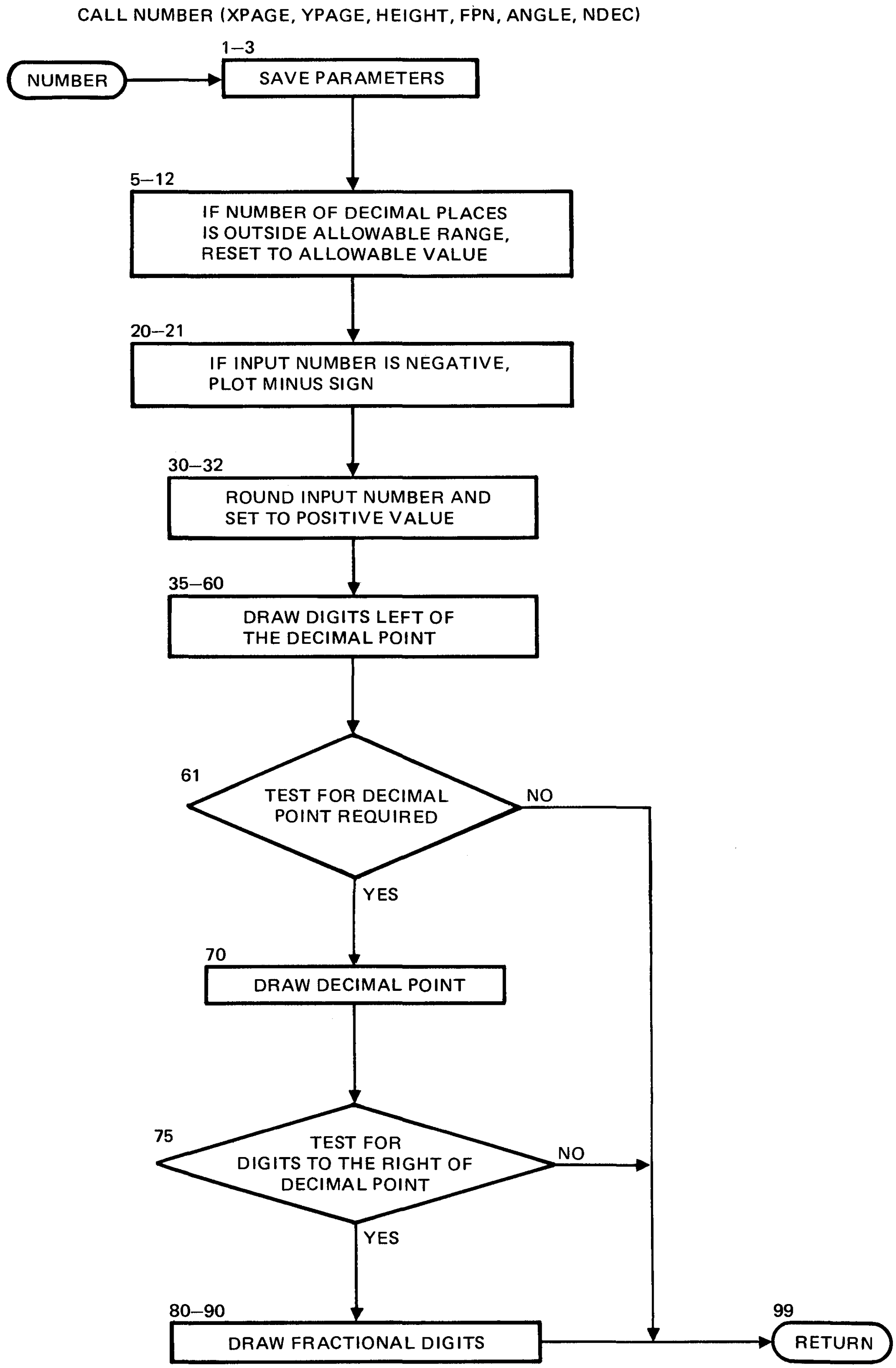


Figure 5-14. NUMBER Flowchart

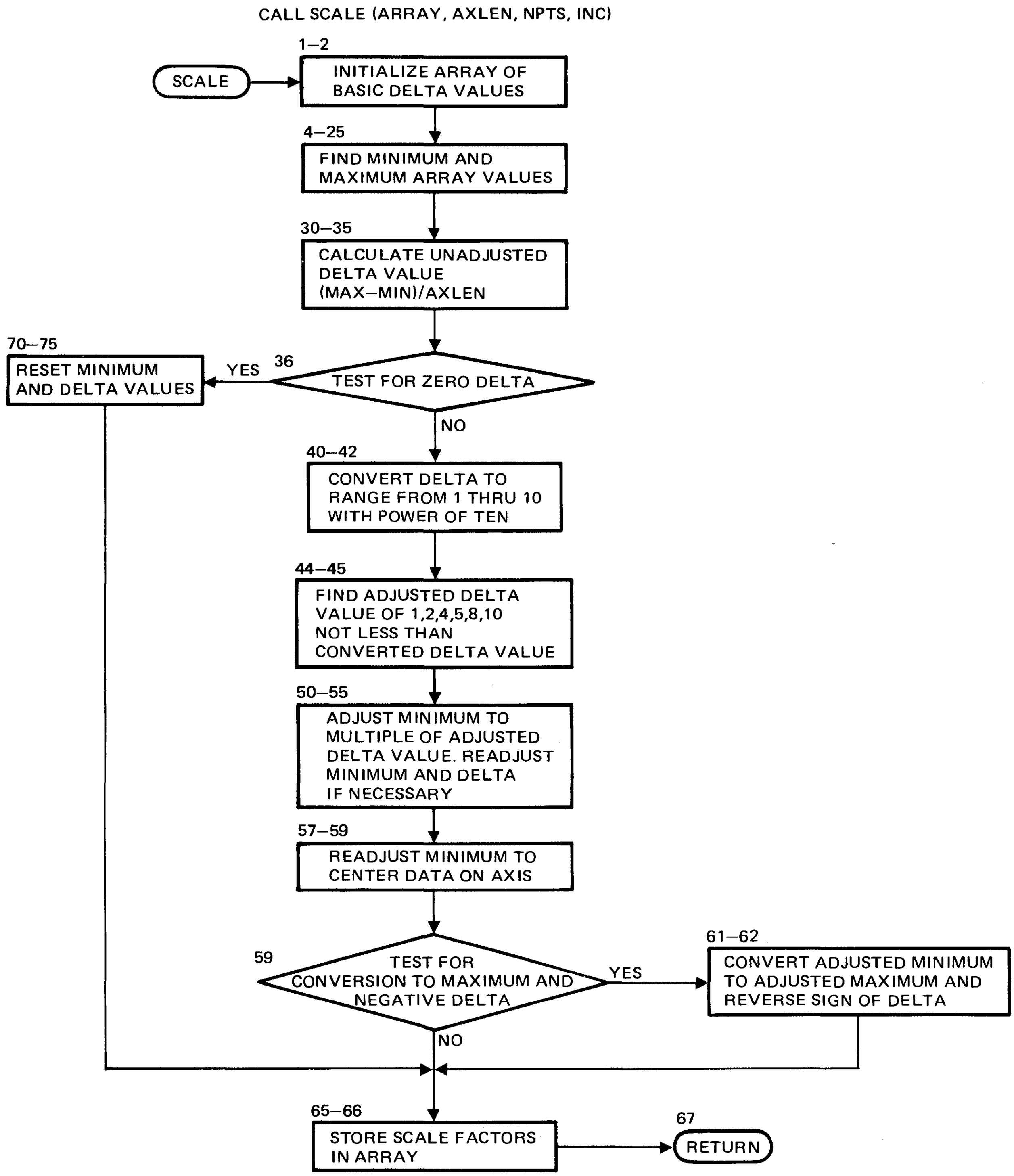


Figure 5-15. SCALE Flowchart

number to BCD representation and plots a minus sign (if negative) and as many digits of the number as requested, including a decimal point (if necessary). Metric versions of the NUMBER subroutine may substitute a comma for the decimal point.

The purpose of the SCALE subroutine is to determine scale factors that will convert arbitrary data to a specified range. Normally, it is used as the first step in drawing a graph of fixed size using arbitrary data. The SCALE subroutine does not use any other plotting subroutines and does not generate any plotting. The calculation used to scale data is:

$$\text{NEW} = \frac{\text{OLD - MINIMUM}}{\text{DELTA}}$$

Where the SCALE subroutine calculates the MINIMUM and DELTA to transform the old data into a range from 0 to A, and where A is the axis length over which the data is to be plotted. The subroutine first calculates the DELTA such that it is a power of 10 times 1, 2, 4, 5, or 8 and is greater than the range of the data divided by the axis length. A MINIMUM is then selected such that it is a multiple of the DELTA and is less than the least data value. The subroutine checks to see that the transformed data does fall into the allowable range; if not, the MINIMUM and/or the DELTA are adjusted so that transformed data does fall into the range. The MINIMUM is then adjusted to center the data in the range unless centering would change the sign of the MINIMUM.

GRAPHIC CONTROLLER SOFTWARE DESIGN

The CalComp Graphic Controller Software (GCS) is a set of programs (written in Assembly Language) that controls the operation of the CalComp 925 Controller. The GCS Software is supplied on a tape cartridge with each 925 Controller. The standard GCS reads 925 format tapes created by the standard CalComp Host Computer Basic Software (HCBS).

Flow of GCS

The standard GCS consists of a set of programs or modules. There are four basic modules (EXEC, READ, PLOT, and SYMBOL) in each system. These modules interact as shown in Figure 5-16. The EXECUTIVE module (EXEC) contains an initialization and a control section. The initialization section initializes:

- The control portion
- The READ and PLOT modules
- Other optional functions

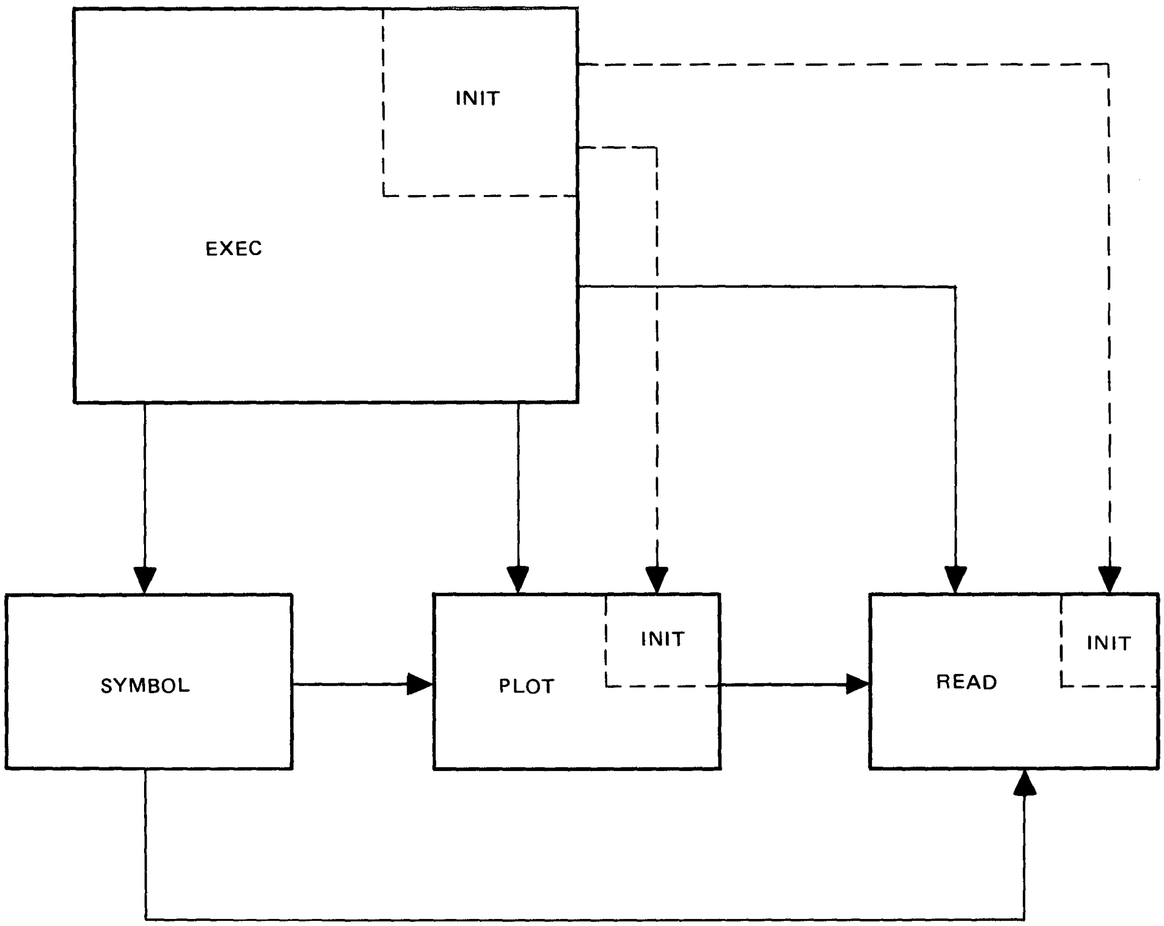


Figure 5-16. GCS Flowchart

Flow of GCS (cont)

The control section:

- Interprets the data
- Interprets the operator commands
- Passes data to the other modules (as necessary)

The controller memory (core) is divided into pages with each page containing 512 words. A standard 925 Controller consists of 8K of core (16 pages). Each module is assigned pages as shown in Table 5-7. The PLOT module does not use all the pages allocated, but will always start with page 10.

Executive Module

The Executive Module (EXEC) contains the main controlling program for the GCS, a number of standard subroutines used by many modules, transfer vectors to other modules, and common constants and variables. The Executive initializes the controller and all other necessary modules. It then interprets the data and calls upon the appropriate modules to handle the various control codes encountered. The EXEC also interprets the various operator inputs and controls searching, single or multiple plotting, and stopping functions.

Some of the subroutines contained in the EXEC module are as follows:

- The interrupt processor to handle interrupts and interprets the type of interrupt.
- A triple precision add function.
- A rounding subroutine.
- A line splitting subroutine.
- A subroutine to read data entry switches.
- A subroutine to display data in the display lights.
- A pause subroutine to give messages to the operator.

Table 5-7. 925 Controller Memory Layout

Page Number	Contents
0	EXEC and transfer vectors
1	READ buffers
2	READ routine
3	EXEC
4	Reserved for options
5	SYMBOL routine
6	SYMBOL tables
7	SYMBOL tables
10	PLOT routine
11	PLOT routine
12	PLOT routine
13	PLOT routine
14	PLOT routine
15	PLOT routine
16	PLOT routine
17	PLOT routine

READ Module

The READ module contains the subroutines to:

- Input data records
- Get characters from the input buffer
- Pack characters into double precision binary numbers

Various READ modules are available to read 7-track or 9-track tape, card readers, online interfaces, etc. The input subroutine reads and checks the data. Error recovery is tried, and if not successful, an error message is displayed. If necessary, the data is converted to usable form and buffer pointers are initialized. The get-and-pack character routines are used by other modules to obtain information from the input data. The get character subroutine obtains the next character from the input data. The pack character subroutine obtains the next 1, 2, or 3 characters from the input data and packs them into a double precision integer extending the sign, if necessary.

PLOT Module

The PLOT module contains all the functions necessary to drive the particular plotter attached to the system. Some of the subroutines included are the following:

- Plotting output
- Pen selection
- Pen up
- Pen down
- Initialization

The plotting output subroutine accepts a delta move and converts it to information necessary to drive the plotter. In some systems the output is buffered and the plotter is started when the buffer is full or other external conditions are met. Otherwise, the plotter is driven as the data is generated.

The pen up and pen down subroutines generate the information to change the pen state and set the flags necessary for other subroutines. The pen selection subroutine selects the appropriate pen and offsets the plotter to compensate for the physical distance between pens. The plotter is checked for being ready and various parameters are set by the initialization subroutine.

SYMBOL Module

The SYMBOL module contains the functions and extensive tables necessary to create characters.

The height and angle tables are generated from the XAX, XAY, YAX, and YAY values in the input data. These tables are generated whenever the values change. The characters making up the string are picked up one at a time from the input data and calls to the PLOT subroutine are made for each vector making up the character. The total movements made for the symbol string are accumulated and used to adjust the next delta in the input data.

Other GCS Modules

Graphic Controller Software (GCS) systems encompass a wide variety of functions and include numerous modules in addition to the standard modules.

Some of the other modules used are:

- Loader modules to create GCS from magnetic tapes and to copy cartridges.
- Simulator modules to interpret other data formats.
- Printing modules to drive COM printers.
- Subroutines to draw circles.
- Overlays to handle plotters with different increment sizes.
- Scaling and Windowing modules.
- Modules to handle optional hardware features, such as, the Remote Positioner Unit, Lightheads, and Cutters.
- Special CAL EDIT Executives.

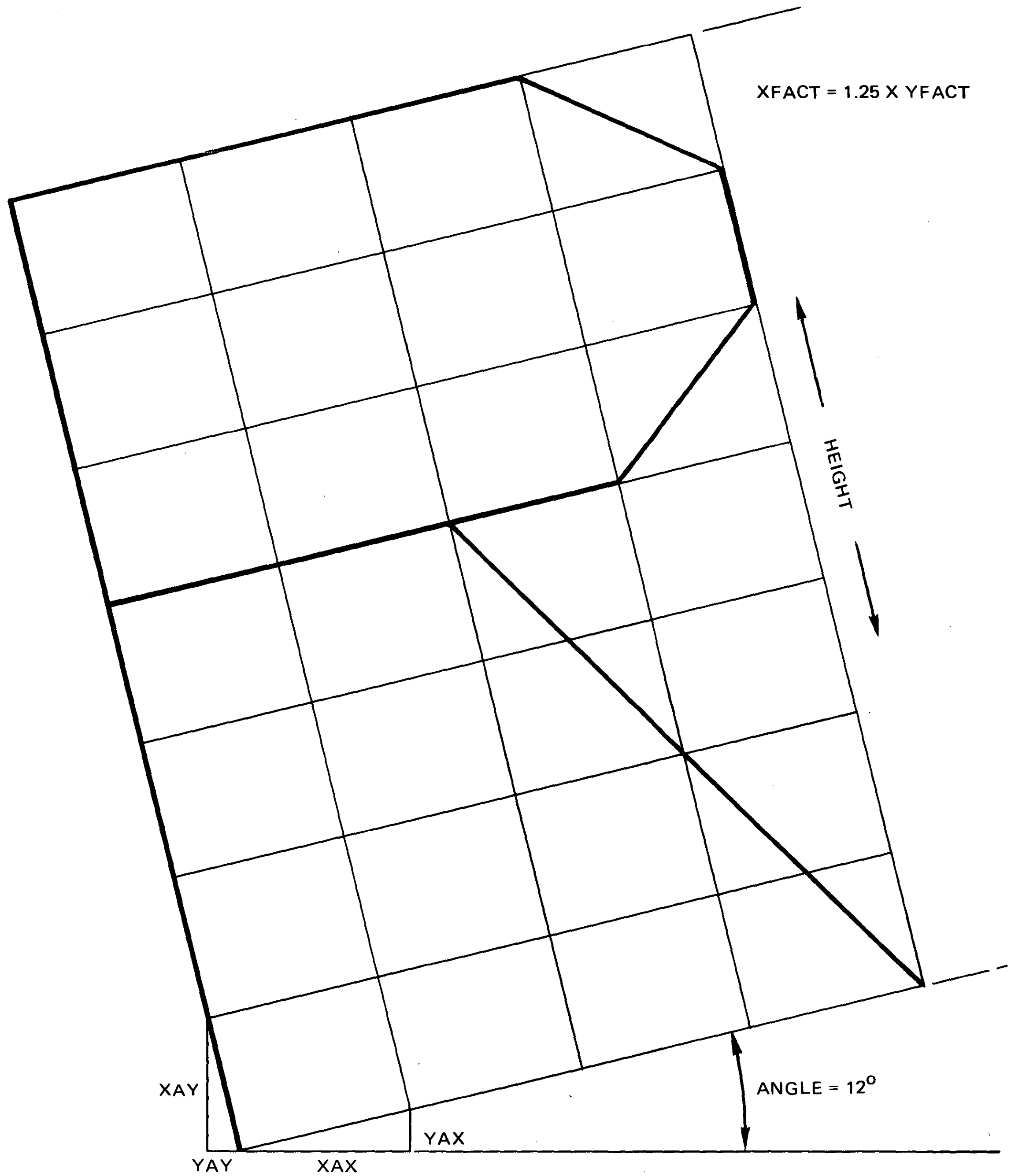


Figure 5-11. SYMBOL Grid

SECTION 6
CONTENTS

FUNCTIONAL SOFTWARE

GENERAL

Host Computer Resident Software

Standard Host Computer Subroutines
Host Computer Symbol Sets
Host Computer Photo Plotter Verifier
Host Computer Assembler for CalComp Controllers
Host Computer Microfiche Management Program
Host Computer Digitizer Support Software
Host Computer Polygon Painting Software
Host Computer World Map Plotting Software

Graphic Controller Resident Software

Graphic Controller Scaling Software
Graphic Controller Windowing Software
Graphic Controller Line Printer Simulators
Graphic Controller Emulators
Graphic Controller Microfiche Management Software
Graphic Controller Digitizer Support Software
Graphic Controller Sizing Software
Graphic Controller Circle Software
Graphic Controller Forms Overlay Software

SECTION 6
FUNCTIONAL SOFTWARE

GENERAL

CalComp offers a wide range of programs and subroutines to perform many of the functions that are widely used in various plotting applications. The CalComp Functional Software is an intermediate level of software which relieves the user of programming many commonly used graphic functions. Functional Software is subdivided into:

- Host Computer Resident Software
- Graphic Controller Resident Software

HOST COMPUTER RESIDENT SOFTWARE

Host Computer Resident Functional Software is usually written in FORTRAN so that it may be easily translated for many computers. Some products consist of subroutine packages so they may be used as graphic utility libraries, while others are complete programs. A brief description of each product is given in one of the following paragraphs.

Standard Host Computer Subroutines

CalComp's Standard Host Computer Functional Subroutines perform many of the functions that are used repeatedly in plotting applications. These functions consist of a set of FORTRAN subroutines offered in packages to simplify programming in appropriate areas. The packages are for scientific, business, drafting, and general applications. Several miscellaneous functional programs are also included.

CalComp's Functional Scientific Package consists of eight FORTRAN subroutines designed to simplify both data reduction and the graphic display of raw data. The Scientific subroutines can be a powerful tool for the presentation of data in research, design analysis, testing, quality control, technical documentation, and similar applications.

Standard Host Computer Subroutines (cont)

Combined with CalComp's Basic Software, these subroutines provide the capability to do every standard kind of scientific data presentation in a simple and efficient way.

The subroutines are for plotting logarithmic data, polynomials, polar coordinates, and for smoothing and refining graphs that have to be plotted with a minimum of data points. The CURVX and CURVY subroutines let you plot polynomials by entering coefficients and exponents. You don't have to identify a series of X and Y coordinates to draw a curve.

The Scientific subroutines are:

CURVX	plots a function of X over a given range
CURVY	plots a function of Y over a given range
FLINE	draws a smooth curve through a set of data points
LGAXS	plots an annotated logarithmic axis
LGLIN	plots log-log or semi-log data
POLAR	plots data points using polar coordinates
SCALG	generates scaling for logarithmic plotting
SMOOT	draws a smooth curve through specified data points

CalComp's Functional Business Package consists of seven FORTRAN subroutines suitable for plotting automatically most of the functions required for management information in business. The Business Package can help make your CalComp plotter a leading tool for coordination, planning and decision-making. Clear and meaningful charts and graphs can be an inexpensive byproduct of your existing computer programs in any significant area-cash flow, accounts receivable, inventory control, sales and marketing statistics, production control, bill of materials or other business areas. Standard or logarithmic line or bar graphs correlating meaningful parameters in any of these areas are drawn simply and economically using the CalComp Business subroutines. This means you will use more and better graphs and get them with less effort. The result can be better information available to everyone in your organization who can use it.

The Business subroutines are:

AXISB	draws an axis with business annotation
AXISC	draws an axis with calendar month annotation
BAR	draws bar graphs
LBAXS	draws a logarithmic axis with business-oriented annotation
LGLIN	plots log-log or semi-log data
SHADE	draws shading between lines
SCALG	generates scaling for logarithmic plotting

CalComp's Functional Drafting Package consists of five FORTRAN subroutines that provide most of the symbols and annotation that appear repeatedly in drafting jobs. Generally, the Drafting subroutines combined with CalComp's Basic Software will let programmers concentrate on the important part of the drawing while the CalComp subroutines take care of housekeeping items like dimension lines and annotation.

The Drafting subroutines are:

AROHD	draws an arrowhead
ARROW	draws a line that ends in an arrow
CNTRL	draws a centerline
DIMEN	draws an annotated dimension line
LABEL	draws annotation between specified points

CalComp's Functional General Package consists of eight FORTRAN subroutines that greatly simplify the job of drawing lines in several standard shapes and figures. The General

Package is a powerful supplement to CalComp's Basic Software for doing all kinds of plotter drawings. The General subroutines let you draw curves, circles, arcs, polygons, dashed lines and grids with a single call to the appropriate subroutine.

For example, to draw an elliptically shaped figure (or part of an ellipse) you just include in your program a call to the subroutine ELIPS and enter the length of axis you want, the angular orientation of the figure, the location where the drawing starts and what segment of the ellipse you want drawn (if the whole ellipse is not desired).

The General subroutines are:

CIRCL	draws a circle, an arc, or spiral
DASHL	draws dashed lines between data points
DASHP	draws a dashed line to a specified point
ELIPS	draws an ellipse or an elliptical arc
FIT	draws a curve through three points
GRID	plots a linear grid
POLY	draws an equilateral polygon
RECT	draws a rectangle of any size

FORGN is a complete FORTRAN program for generating forms. It was designed for data processing forms-keypunch transmittals, card layouts, tape record formats and similar data forms. FORGN simplifies the task of producing standard forms for any purpose. This program prints title information, body copy, annotation, and rules all lines and columns. A standard 8½ X 11-inch form drawn by FORGN can have a title line, 25 data lines and 80 columns per line. The size can be varied to suit your requirements. Several forms can be stacked on a single page.

FLOCT is a complete FORTRAN program that draws fully annotated computer program flowcharts of any size. FLOWCT combines with FORGN to automate a large part of the time-consuming job of program documentation. To use FLOCT you enter parameter cards that:

- Print a title at the location you specify
- Draw the appropriate flowchart symbols where you specify
- Insert the correct comments within the flowchart symbols
- Indicate the end of the chart

The program then takes over and executes the drawing and annotating.

CRVPT is a FORTRAN subroutine for polynomial curve fitting, a data reduction technique valuable to a number of applications where collected data points have small inherent errors. The curve fitting subroutine can simplify the plot of these data points and average out the overall error. CRVPT plots the curve based on the least squares approximation technique.

CRVPT is accompanied by a control program that accepts data to produce your graph from scratch. If, however, you already have a program to plot polynomials, you can add the curve fitting refinement by just calling the CRVPT subroutine in your program.

Host Computer Symbol Sets

CalComp offers several Host Computer Symbol Sets to produce characters of various styles. The actual subroutines are compatible with the Basic Software SYMBOL subroutine, however, additional calls and features are supported.

The Drafting Letter and Symbol Set provides a unique programming/plotting tool for users who want plotted annotation that looks as refined as that produced with manual lettering devices.

The subroutine differs considerably in internal characteristics as well as in sophistication of output from the Basic SYMBOL subroutine. However, very few changes in prior established calls to SYMBOL need be made to a user's program.

An additional "SET MODE" call is provided to permit the user to select:

a) Variable aspect ratio (ratio of character width to height).

b) Italic letters (slanted to the left or right).

c) Variable spacing between characters.

The Block Letter Symbol Set subroutine affords the user a distinctive programming/plotting tool for the generation of block letter annotation. Centered symbols of standard format are provided to facilitate standard curve generation identification. All other symbols generated will be in the block letter style, annotation being in a specialized font.

Standard calls to this subroutine are in agreement with those for the Basic Software SYMBOL subroutine. A special call referred to as a "SET MODE" call is used to permit user initialization of selected options. The options provided are:

a) Variable aspect ratio (ratio of character width to height).

b) Italic letters (characters slanted to the left or right).

c) Variable spacing between characters.

Host Computer Symbol Sets (cont)

The Upper Case/Lower Case Symbol Set subroutine provides a unique programming/plotting tool for users who want plotted annotation that looks as refined as that produced with manual lettering devices. This subroutine incorporates selection capabilities for either Upper Case or Lower Case lettering.

The subroutine differs in internal characteristics as well as in sophistication of output from the Basic SYMBOL subroutine. While existing programs which call SYMBOL will run without change (using the Drafting Symbol subroutine), users can make changes to take advantage of the new features provided. A SET MODE option permits the user to select:

a) Variable aspect ratio (ratio of character width to height).

b) Italic letters (slanted to the left or right).

c) Variable spacing between characters.

d) Upper Case or Lower Case.

The Shaded Block Letter Symbol Set subroutine provides the user with a programming/plotting tool for the generation of shaded block letter annotation. In addition, standard block letters may also be generated.

Standard calls to this subroutine are in agreement with the Basic Software SYMBOL subroutine. A special "SET MODE" call is used to permit initialization of the following options:

a) Generate standard or shaded block letters.

b) Distance between shade lines.

c) Variable aspect ratio (ratio of character width to height).

d) Italic letters (characters slanted to left or right).

e) Variable spacing between characters.

Host Computer Photo Plotter Verifier

The Host Computer Photo Plotter Verifier program accepts input coded in accordance with EIA Standard RS 274 -- such as used to drive photo plotters -- and outputs CalComp compatible code. This makes it possible to verify data prepared for a photo plotter on a CalComp drum plotter. Input is magnetic tape or punched cards. Any of four types of views can be plotted:

- A simple movement diagram using dots to represent flashes and lines to represent paths.
- A similar plot with centered symbols in place of dots.
- A plot with paths outlined as they would appear on film and centered symbols representing flashes.
- A complete reproduction of the intended artwork with paths and flashes outlined.

The first type is for rough checking. The fourth type is for highly accurate verification. The other two types can be used to reduce plot time when the fourth type is not essential. Centered symbols, when used, are chosen by the user and are scaled proportional to the aperture size intended.

A "window" capability makes it possible to check any specified portion of a plot in detail. All plots can be done at any scale. It is possible to rotate a plot 90 degrees to fit on a drum plotter. The program also produces a tally of apertures used for lines and flashes.

Host Computer Assembler for CalComp Controllers

The Host Computer Assembler for CalComp Controllers is a program written in FORTRAN (G) and Assembly Language for the IBM/360 Computer. The IBM/360 Computer assembles programs written in 900 series Assembly Language and produces source listings, object decks, and object tapes. The Assembly Language is fully described in the CalComp Model 910/915 Programmer's Reference Manual and in the supplement concerning the assembler. The assembler enables a programmer to write symbolic Assembly Language programs for CalComp Controllers. These programs are then processed by the assembler, which produces a listing of the source cards and the assembled code, and an object deck (option) and/or an object tape (option). Many useful features are provided in the assembler, including pseudo-operation codes allowing relocation of programs by a fixed amount together with automatic generation of character codes necessary for some input-output operations, and diagnostic error and warning messages. Object decks previously created may be merged selectively with newly generated code to produce an object tape.

A loader/cartridge-write utility program which operates on the Controller is also supplied. This program reads the object tape created by the assembler and loads the program into the Controller memory. Memory may then be written onto a tape cartridge for further use.

Host Computer Microfiche Management Program

The CalComp Host Computer Microfiche Management Program (CALCOM) processes a user's print file and produces microfiche output with user-specified titling and indexing. Output may be online (directly to a CalComp 2131), or a tape may be written for offline printing on a CalComp 2151 or 1670 Microfiche System. CALCOM permits fiche pages to be printed by column, with the eye-readable title placed horizontally along the top of the fiche. The index pages are the last pages printed on the fiche frame, and contain one index entry for each printed page on the fiche. A user-specified heading may be written on the index page.

The CALCOM Program supports either a 24X, 42X, or 48X reduction size and either an 8½ X 11 or a 14 X 11 page size.

Host Computer Digitizer Support Software

Digitizer Functional Software accepts cards produced by a CalComp 942 Digitizer, runs under FORTRAN on a user's Host Computer, and produces plotter-ready output via calls to CalComp Basic Plotting Software and CalComp Digitizer HCBS. Proper operation of this software requires the user to prepare his digitized deck in a specific way, using a specific CalComp menu.

Menu functions supported by this software include:

- Connecting a series of points with straight lines, dashed lines, or smooth curves.
- Plotting circles or arcs.
- Defining closed polygonal figures, shading them if desired, and causing them to be repeated at specified locations.
- Plotting menu-selected symbol strings at specified heights, angles, and positions.
- Accommodating certain utility functions such as "recall last X, Y point," pen selection, block address generation, arrowheads at line ends.
- Allowing user selection of symbols from a 64-character symbol area, including the standard alphabet and digits 0-9.

The Digitizer Functional Software is designed to simplify the job of getting from raw data to a finished graph using the CalComp 942 Digitizer, the user's computer, and a CalComp Plotting System.

Host Computer Polygon Painting Software

CalComp's PAINT routine is designed to fill in i.e., paint areas defined by polygonal boundaries. Complete single polygons, polygons with interior polygonal holes, or polygons within polygons may be painted with the PAINT routine. Up to 100 boundary points with as many as 20 polygons may be specified in a single call.

Host Computer Polygon Painting Software (cont)

The technique used is valid with any plotting accessory where a specific width can be assigned to a specific pen number; PAINT is designed for use particularly with the CalComp 7182 lighthead. Painting is done by first outlining the figure using the smallest available width. The remainder is then filled in, using a larger width if the figure contains no acute angles.

Host Computer World Map Plotting Software

The CalComp Host Computer World Map Plotting Software package consists of a set of subroutines used to plot and annotate various types of projections of the world. In addition, a data base consisting of longitude and latitude coordinate pairs is provided to determine the geographic and political boundaries of the world used in producing a map.

WMAP is a FORTRAN subroutine which the user calls to specify:

1) Which type of projection is to be plotted.

2) What annotation, if any, is required.

3) What portion of the world is desired (window whose interior portion is projected and plotted).

The data base consists of approximately 10,500 coordinate pairs having neighbor spacings of not less than 5 nautical miles and not greater than 150 nautical miles.

The map projection types available are:

1) Nontransformed Rectangular

2) Mercator

3) Cylindrical

4) Quadratic

5) Azimuthal Equidistant

6) Polar Perspective

GRAPHIC CONTROLLER RESIDENT SOFTWARE

Graphic Controller Resident Functional Software products are written in the Assembly Language for the CalComp 925 Graphic Controller. Additional functions for Standard GCS may include scaling or windowing, or there may be independent functions, such as simulators or emulators. A brief description of current products is given in the following paragraphs.

Graphic Controller Scaling Software

Scaling GCS is an optional addition to Standard GCS for CalComp plotting systems. There are two scaling packages available offering different features: General Scaling and Extended Scaling.

General Scaling accepts operator inputs to allow scaling of a plot by a fraction (N/M). N and M can range from 1 to 999, thus giving a wide degree of scaling.

Extended Scaling accepts operator inputs to allow scaling of a plot a fraction (NX/MX) in X, a fraction (NY/MY) in Y, an angle of rotation in tenths of degrees, and a Mirroring Function. NX, MX, NY, and MY can range from 1 to 9999. This gives more flexibility in scaling.

Graphic Controller Windowing Software

Windowing GCS is an optional addition to Standard GCS for CalComp plotting systems. Windowing GCS accepts operator inputs to:

1) Plot only specified rectangular areas.

2) Plot at a specified scale.

3) Plot at a specified location.

Some uses of Windowing GCS include:

- Verifying a small corrected portion of a large plot, in a fraction of the time and materials needed to do the entire plot.

- Exploding (scaling up) a small portion of a very dense plot for detailed verification. This is particularly useful for integrated circuit masks and printed circuit board layouts.

- Segmenting plot data too large for the plotter, while maintaining the original plot integrity. This is particularly useful for large contour maps, integrated circuit masks, etc., when verified on smaller plotters.

Features supported by Windowing GCS include:

Segmenting	- Up to four separate rectangular "windows" may be defined for a single plot, with corners in the range ±99.9 inches.
Plotting	- Plot data enclosed by the windows may be excluded (all data outside the windows are plotted) or included (only data inside the windows are plotted).
Scaling	- Data to be plotted may be scaled from 1:999 to 999:1.
Translating	- Windows which include plot data may be moved relative to plotter limits; thus the upper right corner of a large plot could be moved to the center area of the plotter, for example.

Graphic Controller Line Printer Simulators

The Line Printer Simulator (PRINTSIM) Software permits CalComp COM Plotting Systems and COM Printers to read magnetic tape in standard line printer format and print the data on film.

Standard line printer formatted tapes are required; no changes to the Central Computing System Software are necessary. Each software package is designed to support one specific line printer format. No microfiche formatting, titling, or indexing capability is provided with this software. Titling and indexing capability can be obtained from Host Computers using the CALCOM Software.

Graphic Controller Emulators

Several emulators exist on CalComp Graphic Controllers to accept input formatted for other Graphic Systems and output the information to the user's CalComp Graphic System.

The most common emulator is the CalComp Tape Unit Simulator (TAPESIM). This Graphic Controller Software makes it possible for a CalComp Programmable Controller (925) Plotting System to plot tapes written for any nonprogrammable CalComp System. Input tapes in formats compatible with 470, 750, 760, 770, 780, or 905 tape units are processed by this software. The software includes a General Scaling Feature, allowing plots to be factored by any 3-digit ratio (NNN/DDD).

Graphic Controller Microfiche Management Software

CalComp's Microfiche Management Software (MMS) enables the user to produce titled and indexed microfiche on the CalComp Microfiche Management System from standard computer print tapes. Microfiche produced by the CalComp MMS Software consists of the following:

- Sheets of 105mm film (approximately 4 X 6 inches) with eye-readable titles across the top of each sheet.
- An array of data pages.
- An index to the data pages in the lower right corner of each sheet.

Some of the features of MMS are:

- Horizontal titling and vertical pagination
- Reductions of 24X, 42X or 48X
- Page sizes of 8½ X 11 or 14 X 11
- Multireel file and multifile reel processing
- Up to three lines of title information
- "FROM/TO" title information
- Conditional fiche break: i.e., fiche advance controlled by a data field on the print tape

Graphic Controller Microfiche Management Software (cont)

The MMS Software is delivered on a cartridge. Parameter settings are entered by the user at print time through the console. Frequently used settings may be recorded on spare cartridges. The MMS Software also supports 16mm Microfilm Output.

Graphic Controller Digitizer Support Software

The CalComp Graphic Controller Digitizer Support Software is designed for check plotting of card data produced by the CalComp 942 Digitizer. It operates directly on a CalComp 925 Controller with a CalComp 980 Card Reader and a CalComp plotter, bypassing the user's Host Computer. The user's deck must be prepared in a specific way, using a specific CalComp menu.

This software is compatible with CalComp's Host Computer Digitizer Functional Software, but does not offer as many features. Menu functions supported by this software include:

- Connecting a series of digitized points with straight lines
- Plotting circles or arcs
- Plotting menu-selected symbol strings at specified heights, angles, and positions
- Accommodating certain utility functions, such as, "recall last X, Y point" and pen selection
- Allowing user selection of symbols from a 64-character symbol area
- Optionally plotting card sequence numbers

Graphic Controller Sizing Software

Sizing GCS is an optional addition to Standard GCS for Flatbed Plotters. The feature allows the operator to adjust the size and orientation of the plot to fit preprinted forms or grid paper laid out on the plotter. It is useful when the paper has changed size or is laid down skewed to the plotter. To perform the Sizing Operation, the operator enters the Original X and Y Dimensions of the form or grid. Then using the Remote Positioner, the operator digitizes points representing the actual X and Y Dimensions. The software then compares the actual and original values and computes the necessary scale and rotation values.

The Sizing Feature includes windowing, which may be used to limit the size of the plot so that it fits on the form.

Graphic Controller Circle Software

The Circle Generator GCS is an optional addition to Standard GCS for CalComp Graphic Systems. The Circle Generator allows the user to produce circles or circular arcs 20 inches (unscaled) in diameter with as few as 10 characters on tape. Along with the GCS addition, a FORTRAN subroutine is provided for the user's Host Computer. The FORTRAN Circle subroutine uses the Standard 925 PLOT or CALCMP and BUFF subroutine to put control codes and deltas on tape defining the circle. The subroutine accepts center point, radius, starting angle, ending angle, and dash code to produce dashed (length of dash is about 6 degrees) or solid arcs, or full circles.

Graphic Controller Forms Overlay Software

The Software Forms Overlay GCS is an optional addition to various GCS products for the CalComp 1670 COM Plotting System. This option may be added to Standard GCS, PRINTSIMS, and various emulators.

The Software Forms are created using Standard 1670 HCBS for the user's Host Computer. The created forms are then loaded in the 925 Controller using the Software Forms Overlay GCS. The number and complexity of forms which may be entered on any single cartridge is a function of the available memory in the 925 Controller. The forms are then written on the cartridge.

Graphic Controller Forms Overlay Software (cont)

The forms are accessed and plotted during the normal operation of the overlayed GCS under user control.

SECTION 7
CONTENTS

APPLICATIONS SOFTWARE

GENERAL

THREE-D - A Perspective Drawing Program
GPCP - A General Purpose Contouring Program
FLOWGEN/F - A FORTRAN Flowchart Generator
AUTONET - An Automatic Network Display Program
SAMPS - A Subdivision and Map Plotting System
DATAGRAPH - An Automatic Graph Generator
PROVE-I - Programs for Numerical Control Verification
SYMTRAN - A Symbol Generating and Diagramming Language
A.S.A.P. - Automated Symbolic Artwork Program
CONTOUR - A Basic Contouring Program
AUTOGANT - An Automatic Gantt Bar Chart Display Program
FLOWGEN/C - COBOL Flowchart Generator

SECTION 7
APPLICATIONS SOFTWARE

GENERAL

Applications software from CalComp solves graphics problems immediately.

For several applications, CalComp has done the entire programming job. The resulting programs are fully operational, fully checked out, and guaranteed to perform as the program documentation says they will. You supply the data, CalComp Application Software does the rest. You save all the time and most of the cost of programming.

Each CalComp package is a programming solution based on broad experience that is generally beyond the capabilities of any single plotting system user. The chances are that a CalComp program is more efficient and easier to implement than any program a user could reasonably develop.

THREE-D –A Perspective Drawing Program

THREE-D automatically draws three-dimensional perspective representations of any data that can be expressed as a single-valued function of two variables. Surfaces may be drawn as transparent (all lines showing, or opaque (hidden lines removed). Typical applications include representation of terrain, pressure or heat distribution over a surface, mathematical formulas, and demographic data. Stereo pairs may also be generated.

All of the above features are incorporated into THREE-D I. In addition, THREE-D II can grid irregularly spaced data, smooth a rough array, and perspectively annotate the X-Y-Z axes.

GPCP – A General Purpose Contouring Program

GPCP is an extremely accurate program used to produce contour maps. The program uses a unique gridding algorithm to construct an analytically smooth surface from random data values, and has been used in a wide variety of applications with seismic, demographic, geological, thermal, and pressure data. The user can arbitrarily delete contour lines in desired areas, plot bold index contours with or without labels, annotate the map with symbols and lines, save the calculated grid mesh point values, and input known gradients to assist the gridding algorithm.

GPCP I requires that all data fit in core to produce a map. GPCP II uses data-partitioning methods to segment the data so that the size of the map is not limited to core. In addition to saving the grid mesh point values, GPCP II permits the user to save the raw data points and the computed surface tangent values.

Both versions of the program are available with extra-cost options. These options include grid-to-grid operations (add, divide, compare, etc.), integration for surface area and volume, profiling (surface cross-sections), trend surface analysis, and additional data posting methods.

FLOWGEN/F – A FORTRAN Flowchart Generator

FLOWGEN/F enables users to input FORTRAN source and obtain a fully annotated flowchart of the routine formatted to fit on 8-1/2 x 11 sheets. The program is designed to simplify and standardize the task of program documentation. FLOWGEN/F eases the burden of program debugging, maintenance, and modification. The program analyzes unfulfilled references, numbers each page, and provides off-page references at both source and destination elements.

FLOWGEN/F I is designed for IBM 1130 FORTRAN. Full ANS FORTRAN IV is supported by FLOWGEN/F II.

AUTONET – An Automatic Network Display Program

CalComp's AUTONET programs automatically produce network analysis charts from the output of PERT, CPM, PMS, and other scheduling programs. There are three versions of AUTONET.

AUTONET I produces single-page CPM charts utilizing numbered event rectangles, annotated activity lines, and a linear time scale having calendar dates and elapsed time noted at the top and bottom of the chart.

AUTONET II plots event-oriented as well as activity-oriented charts using a linear time scale. When a chart has more levels than can fit on a plotter page, the overflow will be placed onto another page. Time scales between pages will be consistent.

AUTONET III can handle activity-oriented, event-oriented, or combined activity-event-oriented networks. The number of activities is not limited by core. Sufficient peripheral file space can make the number essentially unlimited.

AUTONET III can plot using a linear time scale or flexible time scale with time-phased (date oriented) or nontime phased (logical relationship) placement of nodes. Networks can be plotted on expected completion date, or latest allowable completion date with the critical path determined by the program or specified by the user.

Activities can be grouped via any user-specified activity/data item such as department or function, and events can be flagged as scheduled or completed. The program has complete X and Y paging capabilities including off-page referencing. Portions of a network may be shown via a user-specified time interval or range within the activity grouping item.

SAMPS – A Subdivision and Map Plotting System

SAMPS assists engineers in subdivision design, mapping, and related tasks by automatically drawing the results of surveying computations. SAMPS drawn maps have been accepted as legal documents in every jurisdiction where the system has been installed.

SAMPS I produces subdivision maps with bearings, distances, and other related information. SAMPS II is an augmented version capable of calculating and annotating lot areas in thousandths of an acre, providing variable line segments, and annotating arcs and curves with radius, length, and delta angle.

DATAGRAPH – An Automatic Graph Generator

DATAGRAPH enables the user to produce management information charts and graphs in a wide variety including line charts, bar charts, histograms, pie charts, and scatter diagrams. These graphs can be presented on one of many standard chart styles provided with the program, or on a custom chart designed by the user. A least squares or polynomial fit trend line can also be shown on the charts.

DATAGRAPH automatically scales the data for each chart, calculates chart dimensions, annotates, and draws the chart from user-supplied data files. A COBOL file conversion program is available to prepare the input data files.

PROVE-I – Programs for Numerical Control Verification

PROVE-I is a program that enables users of IBM APT/ADAPT systems to produce verification plots in any planar view of the X-Y-Z axes and also any transparent perspective view (hidden lines not removed).

PROVE-I is analogous to other post processors used with APT/ADAPT. As such, it is appended to the system as an overlay and utilizes standard APT/ADAPT language statements to control its functions.

SYMTRAN — A Symbol Generating and Diagramming Language

SYMTRAN is a programming language that translates input derived from a rough sketch into a finished, formal diagram. Typical diagrams that can be produced include:

1. Schematic diagrams such as electrical, electronic, and hydraulic.

2. Pictorial diagrams that show the interconnections between various systems and between their components.

3. Logic diagrams that show system flow.

SYMTRAN consists of an Input Processor, a Graphic Symbol Library, and a Drawing Processor. The Input Processor enables the user to input the coded data necessary to produce a diagram. The Graphic Symbol Library contains all of the symbols necessary for the diagram. The Drawing Processor obtains a symbol from the library, places it on the diagram as required, and issues the plotter commands necessary to draw the symbol.

A.S.A.P. – Automated Symbolic Artwork Program

A.S.A.P. processes digitized data to produce finished drawings involving placement and connection of repetitive symbols. Examples are electrical and electronic schematics, building layouts, logic diagrams, and hydraulic schematics.

Drawings are produced by using the digitizer to select symbols from a menu, placing them onto the drawing area, and providing the necessary descriptive annotation.

Features of A.S.A.P. include extensive editing capabilities, menu modification while digitizing, flexible text annotation, macro (sub-drawing) creating and modification, and a wide variety of symbol selection, placement, and labeling techniques.

Optional features include automatic numbering and connection of symbols, filling-in (painting) of polygons, and an interconnection report indicating connection points which lie on a common path.

CONTOUR – A Basic Contouring Program

CONTOUR enables users of small computers to display single-valued functions of two variables (X and Y) by means of lines connecting points of equal values (contour lines). The program uses a least squares gridding algorithm to construct a smooth surface from a set of irregularly positioned values, or the surface may be input as a set of values at the intersection points of a gridded rectangular array. CONTOUR allows the user to post data points, plot the grid intersection points (with or without posting), and save the grid for the next contour map in the run stream. In addition, a trend surface map can be produced.

AUTOGANT – An Automatic Gantt Bar Chart Display Program

AUTOGANT enables the user to draw Gantt Bar Charts. The program provides the user with the ability to process unlimited activities, to show float and critical paths, to 'snapshot' a network, and to select from eight different sorting sequences. The user can also show a 'time now' line and plot during activities.

FLOWGEN/C – COBOL Flowchart Generator

FLOWGEN/C enables the user to obtain flowcharts of COBOL source programs in a form that is usable for finalized documentation. The COBOL source input accepted is compatible with some features of later COBOL versions incorporated. Output is in the form of three pages of COBOL flowcharts per each 11 x 14-inch binder page. A listing of the source input can also be obtained on the Line Printer.

DOCUMENT REVIEW SHEET

TITLE: CalComp Software Reference Manual

MANUAL: 1005

FROM

NAME: ______________________________

POSITION: ______________________________

ADDRESS: ______________________________

Comments concerning this publication are solicited for use in improving future editions. Please provide any recommended additions, deletions, corrections, or other information you deem necessary for improving this manual. The following space is provided for your comments.

COMMENTS: ______________________________

Forward all comments to:

CALCOMP
California Computer Products, Inc.
2411 W. La Palma, Anaheim, CA 92801
Attention: Program Development Department

CALCOMP
California Computer Products, Inc.
2411 W. La Palma, Anaheim, CA 92801
Tel (714) 821-2011 Twx 910-591-1154

1005C 1.5K 876 PRINTED IN U.S.A.