AN ABSTRACT OF THE THESIS OF

LARRY ROBERT HUBBLE for the MASTER OF SCIENCE
(Name) (Degree)

in COMPUTER SCIENCE presented on May 3, 1973
(Major) (Date)

Title: COMPlot: COMBINED PLOT DRIVERS FOR TEkTRONIX

GRAPHIC TERMINALS, CALCOMP AND HEWLETT

PACKARD PLOTTERS

Abstract approved:

Joel Davis

COMPlot is a versatile set of plot drivers designed to be used on a Tekterminal, Calcomp, or Hewlett Packard plotter in a time-sharing environment. Although the hardware aspects of each plotting device are quite different, COMPlot allows the plotting devices to be treated as if they were the same.
COMPLIT: COMbined PLOT Drivers for Tektronix Graphic Terminals, Calcomp and Hewlett Packard Plotters

by

Larry Robert Hubble

A THESIS

submitted to

Oregon State University

in partial fulfillment of the requirements for the degree of

Master of Science

June 1973
ACKNOWLEDGMENTS

I wish to express my appreciation to Dr. Joel Davis for his assistance and helpful suggestions during the preparation of this thesis.

I also wish to thank Mr. Jeff Ballance of the computer center for his suggestions and careful review of a draft of this thesis.

Finally, I would like to thank my wife for typing several rough drafts of this thesis.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.  INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>What are Plot Drivers</td>
<td>1</td>
</tr>
<tr>
<td>Environment</td>
<td>1</td>
</tr>
<tr>
<td>II. THE HARDWARE DEVICES</td>
<td>3</td>
</tr>
<tr>
<td>Tektronix 4002 Graphics Terminal</td>
<td>3</td>
</tr>
<tr>
<td>Tektronix 4010 Graphics Terminal</td>
<td>6</td>
</tr>
<tr>
<td>Calcomp Drum Incremental X-Y Plotter</td>
<td>7</td>
</tr>
<tr>
<td>Hewlett Packard 7200A Graphic Plotter</td>
<td>9</td>
</tr>
<tr>
<td>Combinations Allowed</td>
<td>10</td>
</tr>
<tr>
<td>Similarities and Differences of the Plotting Devices</td>
<td>10</td>
</tr>
<tr>
<td>III. IMPLEMENTATION</td>
<td>14</td>
</tr>
<tr>
<td>Initialization</td>
<td>14</td>
</tr>
<tr>
<td>Suppression of Off-Screen Plotting</td>
<td>15</td>
</tr>
<tr>
<td>Software Character Generator</td>
<td>17</td>
</tr>
<tr>
<td>Scaling and Rotation of Data</td>
<td>19</td>
</tr>
<tr>
<td>IV. DESCRIPTION OF COMPLOT SUBROUTINES</td>
<td>22</td>
</tr>
<tr>
<td>Mechanics of Plotting</td>
<td>22</td>
</tr>
<tr>
<td>Languages</td>
<td>23</td>
</tr>
<tr>
<td>ALPHAS</td>
<td>24</td>
</tr>
<tr>
<td>AXIS(XLOW, XHIGH, XORG, YLOW, YHIGH, YORG, XTI C, YTIC, NSMTICX, NSMTICY)</td>
<td>25</td>
</tr>
<tr>
<td>BAUD(IRATE)</td>
<td>27</td>
</tr>
<tr>
<td>DOUBLE</td>
<td>28</td>
</tr>
<tr>
<td>ERASE</td>
<td>29</td>
</tr>
<tr>
<td>IGRINPUT(X, Y)</td>
<td>29</td>
</tr>
<tr>
<td>ITALICS</td>
<td>31</td>
</tr>
<tr>
<td>NORMAL</td>
<td>31</td>
</tr>
<tr>
<td>PAGE</td>
<td>31</td>
</tr>
<tr>
<td>PLOT(X, Y, IPEN, MARK)</td>
<td>32</td>
</tr>
<tr>
<td>PLOTEND</td>
<td>34</td>
</tr>
<tr>
<td>PLOTLUN(LUN)</td>
<td>34</td>
</tr>
<tr>
<td>PLOTTYPE(ICODE)</td>
<td>35</td>
</tr>
<tr>
<td>POINTS</td>
<td>36</td>
</tr>
<tr>
<td>POSITION(X, Y)</td>
<td>36</td>
</tr>
<tr>
<td>ROTATE(DEGREES)</td>
<td>37</td>
</tr>
<tr>
<td>SCALE(XFACT, YFACT, XBIAS, YBIAS, XLOW, YLOW)</td>
<td>37</td>
</tr>
<tr>
<td>SIZE(WIDTH, HEIGHT)</td>
<td>39</td>
</tr>
</tbody>
</table>
SYMBOL(X, Y, DEGREES, HEIGHT, NCHAR, BCDARRAY) 40
TEKPAUSE 41
TK4010 42
VECTORS 42
WINDOW(XLOW, YLOW, XHIGH, YHIGH) 43

V. EXAMPLES 45
   GRAFIT 45
   FORTRAN Subroutine 49

VI. CONCLUSIONS 54

BIBLIOGRAPHY 55

APPENDICES 56
   APPENDIX I 56
      ASCII Codes for Control Characters 56
      ASCII Codes for Printing Characters 57
   APPENDIX II 58

INDEX OF FLOWCHARTS 83
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Flowchart of the plotting portion of GRAFIT.</td>
<td>47</td>
</tr>
<tr>
<td>2. Graph produced by GRAFIT of the solution to a second order differential equation.</td>
<td>48</td>
</tr>
<tr>
<td>3. Graph produced by subroutine PLOTAR.</td>
<td>53</td>
</tr>
</tbody>
</table>
I. INTRODUCTION

What are Plot Drivers

A plot driver and specifically COMPlot is a set of subroutines intended to provide a basis for easily programming graphics applications. These subroutines expand relatively simple instructions specified by the programmer to include all of the necessary details for the plotting device. With a plot driver, a programmer does not have to worry about the specific hardware details of the plotting device being used, which are generally not easily available to him.

All plot drivers are expected to perform at least the following tasks: 1) plot to a coordinate, 2) draw a character string, 3) use units that are meaningful and convenient, and 4) define the size and position of the display with respect to the physical limitations of the plotting device. COMPLOT is more versatile than most plot drivers, since besides providing these four general capabilities, there are other capabilities to facilitate programming a graphics application.

Environment

Most computation at the Oregon State University computer center
is performed on a Control Data Corporation (CDC) 3300 computer under the supervision of the OS-3 time-sharing operating system (6, p. 1-50). Users of OS-3 commonly write graphics programs for the following devices:

1) Tektronix 4002 graphics terminal
2) Tektronix 4010 graphics terminal
3) Calcomp drum incremental X-Y plotter
4) Hewlett Packard 7200A graphic plotter

Therefore, COMPLOT was designed to be utilized in a time-sharing environment with any of the above plotting devices.

The following subroutines contained in COMPLOT perform functions similar to or are extensions of subroutines found in George Rose's TEKPLOT/CALTEK (5, p. 1-33): AXIS, DOUBLE, IGRINPUT, ITALICS, PAGE, POINTS, ROTATE, TEKPAUSE, and WINDOW. These subroutines are present in order to allow all hardware features of the plotting devices to be utilized and were given the same names in an attempt to avoid confusion to users of both packages.
II. THE HARDWARE DEVICES

First, we consider each of the plotting devices individually. Then, a summary of the similarities and differences is given afterwards.

Tektronix 4002 Graphics Terminal

This Tekterminal is equipped with a cathode ray storage tube with a display area 8.18 inches wide and 6.08 inches high. In this area one can display points, line segments, and alphanumeric and special characters. In addition (in alphas mode), the Tekterminal can operate as a Teletype.

The Tekterminal can operate in one of four distinct modes:

1) alphas mode
2) linear interpolate (vectors) mode
3) point plot mode
4) incremental plot mode

These modes can be changed by keys on the Tekterminal or by program control (see Appendix I).

In alphas mode the display area contains 39 lines of 85 characters each. If more than 39 lines are requested, a page full indicator on the Tekterminal is set, processing is interrupted and the user is placed
in OS-3 Control Mode. Hardware generated characters may be produced in two sizes, 0.08 inches high by 0.08 inches wide or 0.16 inches high by 0.08 inches wide, each of which may be displayed in two type fonts: Roman or italics. Information exchange between the computer and Tekterminal is provided through a keyboard and the display area. The keyboard includes the full ASCII character set (see Appendix I).

In vectors and point plot mode the display area has 1,024 by 1,024 addressable coordinates of which only 1,024 by 761 are viewable. The horizontal axis has addresses 0 through 1,023, inclusive and the visual vertical axis 0 through 760, inclusive. The coordinate (0, 0) is located at the lower left corner of the visual display. In point plot mode, a single point is intensified when addressed. In vectors mode, a line segment is drawn between consecutively addressed coordinates. For uniform density, line segments should be restricted to approximately two inches. Addressing any coordinate takes from two to four ASCII characters depending on its position in relation to the last coordinate addressed. Each ten bit coordinate is split in half and transmitted as two ASCII characters \((b^8 b^7 b^6 b^5 b^4 b^3 b^2 b^1)\). The normal order of transmission to specify a point is the following: 1) high order five bits of Y, 2) low order five bits of Y, 3) high order five bits of X, and 4) low order five bits of X. Bits \(b_1^1\) through \(b_5^1\) contain either the low order or high
order bits of the coordinate address. Bits \( b_6, b_7, \) and \( b_8 \) are used as indicator flags as follows:

<table>
<thead>
<tr>
<th></th>
<th>( b_8 )</th>
<th>( b_7 )</th>
<th>( b_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>High order Y</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Low order Y</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>High order X</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Low order X</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

If the high order X or Y bits have not changed from their previous values, they need not be sent.

In incremental plot mode, the Tekterminal performs as an incremental plotter. Each plotted increment is produced in one of eight directions and is approximately 0.008 inches in length. Since no useful application was found for this mode, none of the COMPLOT subroutines utilize incremental plot mode.

The Tekterminal can not only display graphics information, but it can also input graphics information. Graphics input allows a user to select an \((x, y)\) coordinate by indicating its position with a cross-hair cursor. The crosshair is positioned by manipulating a mechanical device, the joystick, which is adjacent to the terminal. Under program control, when the crosshair cursor is in position, striking
any key on the keyboard\(^1\) causes the ASCII code for the key struck and the coordinate of the crosshair cursor to be sent to the computer.

Since the Tekterminal does not provide a permanent record of what transpires between itself and the computer, a hardcopy unit is available by special order. This unit quickly and easily reproduces all information on the Tekterminal's screen at any particular time.

Detailed information about the hardware specifications of the Tekterminal is available from Tektronix, Inc. (8, p. 1-1, 2-13)

**Tektronix 4010 Graphics Terminal**

The 4010 is virtually the same as the 4002 with the following exceptions:

1) In alphas mode, there are 35 lines of 72 characters each.
   There is one size character and normally only one character font supplied.

2) In vectors mode there are 1,024 by 1,024 addressable coordinates, of which only 1,024 by 781 are viewable.
   The 4010 does not have a point plot mode; it is simulated using the same technique as on the Calcomp plotter.

3) The keyboard does not have the full ASCII character set.

\(^1\)Avoid certain keys: SOH, US, and carriage return. See caution in description of IGRINPUT, p. 29.
It prints the same characters as the Teletype (see Appendix I).

4) The crosshair cursor is positioned by thumb dials located on the keyboard.

5) The visual display area is 7.5 inches wide and 5.625 inches high.

Detailed information about the hardware specifications of the 4010 Tekterminal is also available from Tektronix, Inc. (9, p. 1-1, 2-13) 

**Calcomp Drum Incremental X-Y Plotter**

Virtually any graphics display may be created on a Calcomp plotter by moving a pen over a roll of paper held on a drum. The pen may be raised and lowered onto the paper, it may be moved parallel to the axis of the drum and the drum itself may be rotated in small increments.

The Calcomp plotter may be instructed to move one hundredth of an inch in one of eight directions (3, p. 3.1-3.6). A six bit code (see Table I) is used to specify the direction or to control the position of the pen — either lifted from the paper (no line is drawn) or placed on the paper (a line will be produced). In COMPLOT, four six bit
### TABLE I. CODES FOR CALCOMP PLOTTER

<table>
<thead>
<tr>
<th>Direction or pen position</th>
<th>Six bit code in octal</th>
</tr>
</thead>
<tbody>
<tr>
<td>+Y</td>
<td>4</td>
</tr>
<tr>
<td>+X, +Y</td>
<td>5</td>
</tr>
<tr>
<td>+X</td>
<td>1</td>
</tr>
<tr>
<td>+X, -Y</td>
<td>11</td>
</tr>
<tr>
<td>-Y</td>
<td>10</td>
</tr>
<tr>
<td>-Y, -X</td>
<td>12</td>
</tr>
<tr>
<td>-X</td>
<td>2</td>
</tr>
<tr>
<td>-X, +Y</td>
<td>6</td>
</tr>
<tr>
<td>Pen lifted</td>
<td>40</td>
</tr>
<tr>
<td>Pen dropped</td>
<td>20</td>
</tr>
</tbody>
</table>

codes are stored per CDC 3300 word\(^2\) and 34 words are accumulated before a binary record is written on the plot unit (see PLOTLUN, p. 34).

The drum of the Calcomp plotter is 29 inches wide. The height of a graphics display is essentially limited by the amount of plotting paper remaining on the plotter. COMPLOT restricts the height to be less than or equal to 200 inches.

A line drawn between two coordinates on the Calcomp plotter is achieved by using an algorithm to approximate a straight line by a series of incremental moves. An assembly language version of a published ALGOL algorithm (7, 162-P1-0) was used.

\(^2\) Word size on the CDC 3300 is 24 bits.
On a Calcomp plotter mode differences similar to those on a Tekterminal are simulated by proper pen control. Point plotting is accomplished by moving to the specified coordinate with the pen lifted, and then dropping and lifting the pen in place. Point plotting on the Calcomp only works well if a special Rapidograph pen is requested for the Calcomp plotter.

Hewlett Packard 7200A Graphic Plotter

The Hewlett Packard 7200A graphic plotter (HP plotter) is a Teletype-compatible x-y plotter and receives the information necessary for its control over the same line which connects the Teletype to the computer. The display area is coordinate-addressable with the horizontal and vertical axes having addresses 0 to 9,999, inclusive.

The actual size of the display area is controlled by the user with dials located on the plotter. The maximum size being 15 inches wide and 11 inches high. The distance between limits 0 and 9,999 can be adjusted from 50% to 100% of each axis length. This allows the user to adjust graph size to most pre-printed grid systems or to place the graph in any quarter of the useable plotting area.

There are two hardware selectable plotting modes:

1) point plot
2) vectors
In point plot mode, a point is produced at the addressed coordinate. In vectors mode, a straight line segment is constructed between consecutively addressed coordinates. For best results, no line segment should be longer than three inches. To address a coordinate on the display area, the following information is supplied coded as ASCII characters: the decimal value of the x-coordinate, one space, the decimal value of the y-coordinate, carriage return and line feed (2, p. 1-4).

Combinations Allowed

COMPLOT allows concurrent plotting on two devices, in some cases. The allowable combinations are:

1) 4002 Tekterminal and Calcomp plotter
2) 4010 Tekterminal and Calcomp plotter
3) HP plotter and Calcomp plotter

The other combinations are not feasible from a hardware point of view. For concurrent plotting these combinations would involve more than one line to the computer or a multiplexer.

Similarities and Differences of the Plotting Devices

One of the major aims behind the design of COMPLOT was to allow a graphics program to utilize any of the plotting devices.
Before the availability of COMPLIT, programs had to be designed specifically for each individual plotting device. COMPLIT allows the same graphics program to use any of the plotting devices by supplying a subroutine with the appropriate parameter (see PLOTTYPE, p. 35). In fact, plotting can occur on two different plotting devices concurrently if desired (see p. 10), with one of the devices always being the Calcomp plotter. The Calcomp is normally used to generate a specific size graphics display, but the Tekterminals and HP plotters allow the user to see his display as it is created. By plotting on two devices, we get the advantages of both devices.

George Rose's TEKPLOT/CALTEK (4, p. 2) was designed to operate on both the Tekterminal and Calcomp plotter; however, the graphics program must contain duplicate calls to each subroutine if plotting is desired on both the Tekterminal and Calcomp plotter.

The Tektronix terminals are capable of very high speed graphics; however, the user has no permanent record of the graphics produced. The Calcomp plotter, unlike the Tekterminals, does produce a permanent copy but the results are not available until some time after the user has signed off the time-sharing system. The Hewlett Packard plotter produces a very high quality graph (at a relatively slow rate), but is limited to a maximum size of 11 inches by 15 inches.

Unfortunately, there are only a few similarities found in the available plotting devices. In the sense that all plotting breaks down
to the drawing of a line between two coordinates, all of the devices are similar. Both Tekterminals are similar, differing only in the control information characters. Both the Tekterminals and the HP plotter have a display area that is coordinate addressable.

One graphic mode of a 4002 Tekterminal is point plot mode, which allows it to intensify a single point on its screen. The other plotting devices do not have a point plot mode. Since point plot is commonly used, it must be simulated on the other plotting devices.

The Calcomp plotter is an incremental device, whereas the other devices are all coordinate-addressable. A line drawn between two points on the Calcomp plotter will be a sequence of short line segments that approximate a straight line. On the other devices, a straight line segment will be produced. Therefore, the information required to instruct a device to move between two coordinates will be different for each.

One method of annotating a graphics display on the Tekterminal, when it is in alphas mode, is simply to write on its screen as if it were a Teletype. However, the Calcomp and HP plotters will not support this method since it is impossible to write on the display area. To facilitate annotation of a graphics display on all plotting devices, a software character generator is provided. Since characters are actually drawn, they can be produced on any of the plotting devices.
One other major difference between the plotting devices arises when a graphics display has been completed. On a Tekterminal, a graphics display is generally terminated by erasing the screen and then beginning a new display. The Calcomp plotter must be advanced to a new area and the pen positioned in the lower left corner of the new display area. Finally, the paper must be physically removed from the HP plotter and replaced by new paper.

There are still a few minor differences that have not been discussed. These are covered in Chapter IV. Obviously, the differences outnumber the similarities among the plotting devices. However, the COMPLTOE subroutines automatically handle the differences. A programmer using COMPLTOE need not concern himself with the hardware details of the plotting device or devices being used.
III. IMPLEMENTATION

COMPLLOT allows the user to treat all of the plotting devices in the same way. We will discuss the difficulties which must be resolved in order to do that.

Initialization

Prior to the plotting of the first point, an initialization process is required for all plotting devices. Initialization for the Tekterminal or HP plotter is quite simple: the current positions are set to zero and the status area is set to alphas mode. The same information is set for the Calcomp; however, the Calcomp is an incremental device, thus, all movements are relative to the current position. This poses a problem initially since the position of the pen is unknown. Therefore, before plotting can begin, COMPLLOT must force the pen into a known position.

One feature of the Calcomp plotter allows the pen to be forced to a known position: once the pen has been positioned to the left boundary of the device, the Calcomp plotter essentially ignores any instructions to move further to the left. Since the computer operator leaves the pen near the left boundary before the plotter is readied, COMPLLOT assumes the pen is not more than seven inches from the left boundary. The initialization process entails moving the pen at
most seven inches to the left and up three inches, which should leave the pen at the extreme left boundary and above the last user's plot. The pen is then moved one inch to the right to insure a small margin.

Suppression of Off-Screen Plotting

All plotting devices have physical boundaries. Since plotting may not occur outside of these boundaries, plotting outside of the boundaries must be suppressed by plot drivers. As far as the user is concerned, the best solution to this problem is to compute an intercept on the boundary and plot to the point calculated. In some cases it is convenient to suppress plotting of coordinates outside of any arbitrary rectangular boundary controlled by the user. This convenience is facilitated by the use of subroutine WINDOW (see p. 43). The algorithm used to suppress off-screen plotting was adapted from that presented by George Rose (4, p. 15-19).

If the user is in point plot mode and the current coordinate is outside of the boundaries, plotting is suppressed and no further computations are required. In vectors mode, the problem of suppression is more complex. Three different cases arise: 1) both the last coordinate and the current coordinate are on-screen, 2) either the last coordinate was on-screen and the current coordinate is off-screen or the last coordinate was off-screen and the current coordinate is on-screen, and 3) both the last coordinate and the current
coordinate are off-screen.

Obviously, the first information that must be determined is whether a coordinate is on-screen or off-screen. A four bit off-screen flag \((b_1 b_2 b_3 b_4)\) is calculated for each coordinate \((x, y)\) in the following fashion:

\[
\begin{align*}
  b_1 &= 0 \text{ if } y \text{ is greater than or equal to lower bound for } y \\
  b_1 &= 1 \text{ if } y \text{ is less than lower bound for } y \\
  b_2 &= 0 \text{ if } y \text{ is less than or equal to upper bound for } y \\
  b_2 &= 1 \text{ if } y \text{ is greater than upper bound for } y \\
  b_3 &= 0 \text{ if } x \text{ is greater than or equal to lower bound for } x \\
  b_3 &= 1 \text{ if } x \text{ is less than lower bound for } x \\
  b_4 &= 0 \text{ if } x \text{ is less than or equal to upper bound for } x \\
  b_4 &= 1 \text{ if } x \text{ is greater than upper bound for } x
\end{align*}
\]

If the off-screen flag is zero, the coordinate is on-screen and if the off-screen flag is non-zero, the coordinate is off-screen.

Case one can now be solved. If both the last off-screen flag and the current off-screen flag are zero, plot to the current coordinate and no further calculations are required.

In case two, an intercept must be computed of the line drawn between the last and current coordinates and the appropriate boundary. If the last coordinate was off-screen, plot to the computed intercept with the pen raised and then to the current coordinate with
the pen down. If the last coordinate was on-screen, simply plot to
the computed intercept.

In the third case, it is possible that a line constructed between
the last and current coordinates will intersect that portion of the
display area where plotting is not being suppressed. If the last off-
screen flag logically ANDed with the current off-screen flag is zero,
we do indeed have a possible intersection. Otherwise, plotting is
suppressed and no further computations are necessary. If an inter-
section occurs, intercepts are computed with the boundaries and the
line segment computed is then plotted, unless the segment falls in
the region where plotting is being suppressed.

Once plotting of the current coordinate has been completed, the
last off-screen flag is replaced with the current off-screen flag. In
addition if the user calls either the WINDOW or SIZE subroutines,
the last off-screen flag must be recomputed using the current position.

Software Character Generator

Each character in the COMPlot character set is defined as a
variable number of coordinates. The coordinates are based on a
rectangular grid system which has seven horizontal grid lines and
nine vertical grid lines. Each grid point is assigned a number or
coordinate according to the following scheme:
In all cases characters are drawn starting at some specified reference point, number zero, in the lower left corner of the character. If we assume one unit distance between grid lines in both the horizontal and vertical directions, the distance from any numbered grid point to the reference point is computed by dividing the grid number by seven. The quotient is the number of units to move in the vertical direction, and the remainder is the number of units to move in the horizontal direction. An additional parameter in the description allows the character to be translated down several units.

The first coordinate of a character is always plotted to with the
pen up, and the remaining coordinates are connected with line segments in the order specified. For some characters it is necessary to lift the pen at some time while drawing the character. To allow this, one other coordinate number is allowed: 63, which indicates that the following coordinate is to be plotted with the pen raised and the pen is dropped once that coordinate is reached.

Therefore, the coordinates necessary to describe any character are numbered from 0 to 63, and are represented directly by six bits, four coordinates per CDC 3300 word. Also stored are the number of coordinates necessary to describe each character and the width of each character, since all characters are not of the same width.

The reference point for the first character is supplied by the user. Once a character has been drawn, COMPLOT uses the width of the character just drawn to compute the reference point for the next character to be drawn. When the character string has been completed, the pen is moved to the last reference point computed.

Scaling and Rotation of Data

Most programmers expect to be able to plot in units that are meaningful or convenient to them, not in the units of the plotting device being used. Therefore, every coordinate that is plotted is internally scaled and translated by COMPLOT. The programmer may also specify a rotation of the Cartesian coordinate system if he
desires. Since not all the plotting devices handled by COMPLOT use the same plotting units, this compatibility problem must be solved.

In order to do so, all coordinates are first converted to units of one hundredth of an inch (which corresponds to one Calcomp increment).

If the programmer has not specified a rotation or the angle is zero, the following formulas are used to perform the conversion to units:

\[
\begin{align*}
\text{xp} &= x \times \text{xsc} + \text{xorigin} \\
\text{yp} &= y \times \text{ysc} + \text{yorigin}
\end{align*}
\]

Where

\[
\begin{align*}
x &= \text{x-value of coordinate specified by user} \\
y &= \text{y-value of coordinate specified by user} \\
\text{xp} &= \text{number of units in horizontal direction from physical lower left boundary} \\
\text{yp} &= \text{number of units in vertical direction from physical lower left boundary} \\
\text{xsc} &= 100 \times \text{XFACT} \\
\text{ysc} &= 100 \times \text{YFACT} \\
\text{xorigin} &= 100 \times \text{XFACT} \times (-\text{XLOW}) + 100 \times \text{XBIAS}
\end{align*}
\]

---

3 The six variables in upper case are provided by the user. For a description of these variables, see description of parameters for SCALE, p. 38.
yorigin = 100 * YFACT * (-YLOW) + 100 * YBIAS

The formulas used when the rotation is non-zero:

\[
\begin{align*}
xp &= x * xsc * \cos(th) - y * ysc * \sin(th) + xorigin \\
yp &= x * xsc * \sin(th) + y * ysc * \cos(th) + yorigin
\end{align*}
\]

Where variables are the same as those used above except:

\[
\begin{align*}
th &= \text{angle of rotation in radians} \\
\cos &= \text{cosine function} \\
\sin &= \text{sine function}
\end{align*}
\]

If plotting is desired on a Tekterminal or HP plotter, a multiplicative factor is used to convert the units to the physical units of either the Tekterminal or HP plotter. This has the effect of reducing or enlarging any size Calcomp display to the physical size of the Tekterminal or HP plotter. The same multiplicative factor is used for both the x and y values of the coordinate to insure that there is no distortion involved.
IV. DESCRIPTION OF COMPLOT SUBROUTINES

Mechanics of Plotting

The procedure to use COMPLOT for any graphics application can be broken up into four basic operations:

1) defining the plotting device or devices
2) defining size of display area and setting up acceptable units
3) producing the graphics display
4) terminating the display

These operations are generally carried out in the order listed. There are several COMPLOT subroutines applicable in each of the four basic operations:

1) BAUD, PLOTLUN, PLOTTYPE, TK4010
2) ROTATE, SCALE, SIZE
3) ALPHAS, AXIS, DOUBLE, IGRINPUT, ITALICS, NORMAL, PLOT, POINTS, POSITION, SYMBOL, VECTORS, WINDOW
4) ERASE, PAGE, PLOTEND, TEKPAUSE

Generally, calls to subroutines in the first group would be located in the beginning of a graphics program. Next would be calls to those in the second group, etc. However, the user is not restricted to use the subroutines in the order listed.
Languages

COMPLOT is intended to facilitate any graphics application written in either the FORTRAN language or COMPASS (assembly language of the CDC 3300). Information is furnished to COMPLOT in the form of parameters passed to the subroutines. A certain number of parameters must be passed and they must agree in type (real or integer) to that assumed by COMPLOT. The type is indicated in this chapter by the first character of the name. If the first character is either I, J, K, L, M, or N, the parameter must be a one-word integer constant or variable. All other letters indicate a two-word floating point constant or variable.

An example should clarify the method required to call one of the subroutines. Assume a line is desired from the present position to the coordinate (X, Y). IPEN is an integer variable containing one and MARK is another integer variable containing zero. Then the necessary call would be:

In FORTRAN,

CALL PLOT(X, Y, IPEN, MARK)

There is one exception; the AXIS subroutine has four optional parameters. See its description on p. 25.
One other example will be given to illustrate a call to `IGRINPUT`, which is a function. Assume the coordinate of the crosshair cursor is to be put into the variables `X` and `Y` and the variable `ICHAR` is to contain the ASCII code for the key depressed by the user.

In **FORTRAN**,

\[
\text{ICHAR} = \text{IGRINPUT}(X, Y)
\]

In **COMPASS**,

<table>
<thead>
<tr>
<th>RTJ</th>
<th>PLOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>77</td>
<td>X</td>
</tr>
<tr>
<td>77</td>
<td>Y</td>
</tr>
<tr>
<td>77</td>
<td>IPEN</td>
</tr>
<tr>
<td>77</td>
<td>MARK</td>
</tr>
</tbody>
</table>

The remainder of this chapter will be devoted to the description of the actual subroutines available in the **COMPLOT** package. Relocatable binary decks of the subroutines are saved on the loader library file named `*COMPLOT`.

**ALPHAS**

In case it is necessary to input or output character information while in the process of creating a display, **COMPLOT** must be
informed. A call to ALPHAS indicates that alphanumerical information is to follow. The Tekterminal must be placed in alphas mode by a call to ALPHAS before ordinary FORTRAN or COMPASS statements can be used for input or output. On the HP plotter, ALPHAS causes the current plot to be terminated, and further output is reverted to the Teletype. Since the Calcomp plotter is independent of the terminal, ALPHAS has no effect.

There are no parameters.

AXIS(XLOW, XHIGH, XORG, YLOW, YHIGH, YORG, XTIC, YTIC, NSMTICX, NSMTICY)

If a visible axis is required by the user, the AXIS subroutine may be used to draw X and Y axes together with tic marks. After a call to AXIS, the pen will be left at the point (XORG, YORG) and the mode is set to vectors. Since AXIS uses data marks to draw tic marks (data marks are never rotated), axes with tic marks may not be rotated. If ROTATE has been called with a non-zero quantity, the user will have to supply an axis subroutine which plots the tic marks rather than using data marks.

The parameters are:

XLOW = Minimum value of the x-axis in the user’s units
XHIGH = Maximum value of the x-axis in the user’s units
XORG = The value of x at which the y-axis crosses the
x-axis

YLOW = Minimum value of the y-axis in the user's units
YHIGH = Maximum value of the y-axis in the user's units
YORG = The value of y at which the x-axis crosses the y-axis

XTIC = The distance between large tic marks on the x-axis in the user's units. Set XTIC = 0.0 if no tic marks are desired.

YTIC = The distance between large tic marks on the y-axis in the user's units. Set YTIC = 0.0 if no tic marks are desired.

NSMTICX = The number of small tic marks to be drawn between each of the large tic marks on the x-axis. If no small tic marks are desired, set NSMTICX = 0. If XTIC is zero, this parameter is ignored.

NSMTICY = The number of small tic marks to be drawn between each of the large tic marks on the y-axis. If no small tic marks are desired, set NSMTICY = 0. If YTIC is zero, this parameter is ignored.

The first eight parameters are expressed in the user's units.
The last four parameters are all optional (from left to right). If
they are omitted, they will be set to zero.

**AXIS** checks all parameters to make sure they are valid. The following checks are performed on both the X and Y parameters:

1. If HIGH < LOW, they are interchanged
2. If ORG > HIGH, ORG = HIGH
3. If ORG < LOW, ORG = LOW
4. If TIC < 0, TIC = -TIC
5. If TIC = 0.0, NSMTIC = 0

Any of the above corrections are done internally and will not actually change the value of the parameters passed to **AXIS**.

**BAUD(IRATE)**

If the plotting device being used is a Tekterminal, each time the screen is erased a time delay must be supplied to allow the phosphors of the screen to cool down. This delay, approximately one-half second, is supplied by outputting the appropriate number of characters to produce the required delay. **COMPLOT** assumes the Tekterminal is connected to the computer at 2400 baud (number of bits transmitted per second). If 110, 300, or 1200 baud is actually being used, the time delay is much too long. **BAUD** when called will set up the appropriate number of characters.
The parameter is:

\[
\text{IRATE} = \text{Baud rate at which the Tekterminal operates.}
\]

If the plotting device is not a Tekterminal, BAUD causes internal status information to be changed, but there will be no difference apparent to the user.

DOUBLE

Some displays on a Tekterminal have certain areas that are of interest. These areas could be plotted in double intensity to differentiate them from the rest of the display. Double intensity is accomplished by calling DOUBLE. After a call to DOUBLE, all lines will be broader, points will be bigger, and characters will be twice their normal size. One call to DOUBLE will suffice for all graphics; however, any time a carriage return is used, double intensity is reset to normal intensity. Therefore, each line of output sent to the terminal, desired to be double intensity, must be preceded by a call to DOUBLE. One word of caution: double intensity characters are unreadable unless there is one blank between each character and two blanks between words. When normal intensity is again desired, call NORMAL. DOUBLE affects only the Tekterminal; it will produce no difference in displays on the Calcomp or HP plotters.
There are no parameters.

**ERASE**

Once a display has been completed, it is necessary to indicate that a new display is about to begin. Also, it is sometimes necessary to clear the display area before beginning a display. On the Tekterminal, the screen is erased, and the alpha cursor is positioned to the upper left-hand corner of the screen. On the Calcomp plotter the pen is lifted and moved back to the left margin and above the current plot. If no plotting has been done since the last call to ERASE, the plotter will not be moved. If plotting has been done on the HP plotter prior to the call to ERASE, the plot is terminated and control is returned to the Teletype. In all cases, the mode is set to alphas.

There are no parameters.

**IGRINPUT(X, Y)**

IGRINPUT allows the user to input graphics information from the Tekterminal to his program. When IGRINPUT is called, a full screen crosshair cursor appears. The crosshair may be positioned by using the joystick adjacent to the terminal. After selecting the desired point, the user can cause the coordinate of that point to be sent to IGRINPUT by pressing any key. Carriage return, SOH, and
US should be avoided since they change graphics input mode. US will leave the terminal in alphas mode. Carriage return is intercepted by IGRINPUT and causes the crosshairs to be turned on again. SOH puts the user in OS-3 Control Mode, and essentially interrupts the program. When the coordinates have been sent to IGRINPUT, the crosshair will disappear. Occasionally the Tekterminal will not digitize the crosshair coordinate properly. Should this happen, the crosshair will reappear and the key must be pressed again. Before IGRINPUT may be used, the button marked Keyboard/Aux must be set to both Keyboard and Aux. Also, the intensity knob on the joystick must not be turned completely off.

IGRINPUT is a function. Its value is the ASCII code for the character that was typed in response to the crosshair cursor.

The parameters are:

\[ X = \text{The x-value (in the user's units) of the coordinate selected by the crosshair.} \]

\[ Y = \text{The y-value (in the user's units) of the coordinate selected by the crosshair.} \]

IGRINPUT has no effect on either the Calcomp or HP plotters since neither allow graphics input. The variables X and Y will remain unchanged, and the value of IGRINPUT will have no meaning.
ITALICS

A call to ITALICS causes the 4002 Tekterminal to be placed in alphas mode and the italics character font to be selected. To return to the normal character style, a call to ALPHAS must be used. ITALICS has no effect on either the Calcomp or HP plotters.

There are no parameters.

NORMAL

Once double intensity mode has been specified by a call to DOUBLE, a call to NORMAL will return the Tekterminal to normal intensity mode. NORMAL has no effect on the Calcomp or HP plotters.

There are no parameters.

PAGE

If you view the Tekterminal screen as a sheet on which to write alphanumeric information, then the "home" position or starting position is the upper left-hand corner of the screen. Therefore, PAGE may be used to bring the alpha cursor to the "home" position. The Tekterminal is left in alphas mode after a call to PAGE. On the Calcomp and HP plotters, PAGE sets the mode to alphas, but there is no change in the pen position.
There are no parameters.

\textbf{PLOT}(X, Y, \text{IPEN}, \text{MARK})

\textbf{PLOT} is used to move the pen from the current position to the coordinate \((X, Y)\). Before doing so, if the mode is alphas, the mode will set to vectors. There are two manners in which plotting is done: 1) vectors mode -- straight line segments are used to connect the coordinates, and 2) point plot mode -- a single dot will be drawn at the coordinate specified (see \textsc{VECTORS, p.42} and \textsc{POINTS, p.36}). In some cases it is simply necessary to position the pen without causing any line or dot to be drawn. This can be accomplished by setting \text{IPEN} = 0.

The parameters are:

\begin{itemize}
  \item \text{X} = x\text{-value of the coordinate in the user's units}
  \item \text{Y} = y\text{-value of the coordinate in the user's units}
  \item \text{IPEN} = \text{pen position to the coordinate (X, Y). Since the plotting mode affects the pen, there are four possibilities:}
  \begin{itemize}
    \item a) \text{IPEN} = 0, vectors mode -- the pen is moved to the coordinate \((X, Y)\), but no line is drawn.
    \item b) \text{IPEN} = 0, point plot mode -- the pen is moved to the coordinate \((X, Y)\), but no dot is drawn.
  \end{itemize}
\end{itemize}
c) IPEN ≠ 0, vectors mode -- a line is drawn from the current position to the coordinate (X, Y).

d) IPEN ≠ 0, point plot mode -- the pen is moved to the coordinate (X, Y), and a single dot is drawn.

If IPEN = 0, the MARK parameter is ignored; therefore, no data mark will be drawn.

\[
\text{MARK} = \text{The number from the table below of the data mark to be drawn at the coordinate (X, Y). Data marks are always drawn in vectors mode. If the user is in point plot mode, he will be returned to point plot mode after the data mark has been completed. The following table lists the numbers (odd numbers are small data marks, 0.04 inch, and even numbers are large data marks, 0.08 inch):}
\]

\[
0 = \text{no data mark}
\]

\[
1\text{-} 2 = \text{x}
\]

\[
3\text{-} 4 = \text{plus (+)}
\]

\[
5\text{-} 6 = \text{dash (-)}
\]

\[
7\text{-} 8 = \text{vertical bar (|)}
\]

\[
9\text{-} 10 = \text{up arrow (↑)}
\]

\[
11\text{-} 12 = \text{down arrow (↓)}
\]

\[
13\text{-} 14 = \text{right arrow (→)}
\]

\[
15\text{-} 16 = \text{left arrow (←)}
\]
17-18 = box (□)
19-20 = triangle (Δ)
21-22 = asterisk (*)
23-24 = x with bars (x)
25-26 = x in a box (∞)
27-28 = hexagon (○)

Any number not in the above table will be equivalent to setting MARK = 0.

**PLOTEND**

Once a display has been completed, COMPLOT must be notified of this fact and PLOTEND can be used to accomplish this. PLOTEND simply calls TEKPAUSE and then ERASE. Before returning to OS-3 Control Mode, PLOTEND should be called to make sure that all plotting has been terminated properly.

There are no parameters.

**PLOTLUN(LUN)**

PLOTLUN is used to change the logical unit number on which Calcomp plots are produced. If PLOTLUN is not called, plots for the Calcomp plotter will be produced on logical unit ten. PLOTLUN has no effect on the Tekterminal or HP plotter. If the plotting device is the Calcomp plotter and PLOTLUN is called with a logical
unit number that is unequipped, the logical unit number will be equipped to the plotter and information will be requested to label the plotter,

The parameter is:

\[ \text{LUN} = \text{Logical unit number on which Calcomp plots are produced. } 0 \leq \text{LUN} \leq 99. \]

**PLOTTYYPE(ICODE)**

PLOTTYYPE is used to indicate on what device or devices plotting is to be performed. Until PLOTTYYPE is called to indicate otherwise, plotting will be on the Calcomp plotter only (ICODE = 0). Also, if an illegal ICODE is specified, zero will be assumed.

The parameter is:

- ICODE = 2, Plot on both Calcomp plotter and Tekterminal
- ICODE = 1, Plot on the Tekterminal only
- ICODE = 0, Plot on the Calcomp plotter only
- ICODE = -1, Plot on the Hewlett Packard only
- ICODE = -2, Plot on both Calcomp and Hewlett Packard plotters
POINTS

Calling POINTS indicates that further plotting is to be done in point plot mode. All subsequent plotting will cause a dot to be drawn at each specified coordinate. Point plot mode is simulated on the Calcomp plotter, where it does not work well unless a special Rapidograph pen is requested for the Calcomp plotter.

There are no parameters.

POSITION(X, Y)

There is always a current position associated with any plotting device being used. If the Tekterminal is in vectors or point plot mode or on the Calcomp and HP plotters, POSITION will return the (x, y) coordinate of the current graphics position in the user's units. On the Tekterminal, in alphas mode there is always an alpha cursor present on the screen. The alpha cursor indicates where the first character will be printed and is moved to the next position each time a character is printed. POSITION allows the user to obtain the coordinate of the position of the alpha cursor. Before POSITION may be used on a Tekterminal, the button marked Keyboard/Aux must be set to both Keyboard and Aux.

The parameters are:

X = The x-value (in the user's units) of the current position
or the position of the alpha cursor.

\[ Y = \text{the y-value (in the user's units) of the current position} \]
or the position of the alpha cursor.

**ROTATE(DEGREES)**

In most cases, plotting will be done in the Cartesian system of coordinates with the x-axis horizontal and the y-axis vertical. Occasionally the need might arise to plot in a Cartesian coordinate system with the x and y-axes rotated from their horizontal and vertical positions, respectively. By calling ROTATE, the user declares himself to be plotting in a Cartesian coordinate system in which the x and y-axes are rotated from their horizontal and vertical positions the specified number of degrees. The center of rotation is the user's point \((0, 0)\). Positive rotation is assumed to be in the counterclockwise direction.

The parameter is:

\[ \text{DEGREES} = \text{The number of degrees to rotate the coordinate system.} \]

**SCALE(XFACT, YFACT, XBIAS, YBIAS, XLOW, YLOW)**

A call to SCALE defines scale and bias factors for any subsequent plotting. The bias factors may be used to provide space at the
left and bottom of the plot for annotation. SCALE should be called before the axes are drawn or any plotting is done. If SCALE is not called, then the indicated default values are assumed.

The parameters are:

\[ XFACT = \text{Scale factor to convert from the user's units to inches.} \]

\[ XFACT = \frac{\text{Length of x-axis in inches}}{\text{Length of x-axis in user's units}} \]

Default value = 1.0

\[ YFACT = \text{Scale factor to convert from the user's units to inches.} \]

\[ YFACT = \frac{\text{Length of y-axis in inches}}{\text{Length of y-axis in user's units}} \]

Default value = 1.0

\[ XBIAS = \text{Number of inches to translate the graphics display to the right of the physical left boundary of the plotting device.} \]

Default value = 0.5 inches.

\[ YBIAS = \text{Number of inches to translate the graphics display upward from the lower physical boundary of the plotting device.} \]

Default value = 0.75 inches.

\[ XLOW = \text{Minimum x-value of the data to be plotted in the user's units.} \]

XLOW will generally be the same for both SCALE and AXIS. Default value = 0.0
YLOW = Minimum y-value of the data to be plotted in the user's units. YLOW will generally be the same for both SCALE and AXIS. Default value = 0.0

SIZE(WIDTH, HEIGHT)

On the Calcomp plotter, SIZE is called to indicate the width and height (in inches) of the complete display. The size must also include any bias factors to be specified in the SCALE subroutine. The width is restricted to 29.0 inches or less (the width of Calcomp plotter). SIZE makes the Tekterminal and HP plotter appear as if they were WIDTH inches wide and HEIGHT inches high. After SIZE has been called, plotting is allowed any place within the desired size (i.e., SIZE defines the window to be the complete display; see WINDOW, p. 43). SIZE also sets the minimum and maximum limits that can be used by WINDOW. Therefore, WINDOW cannot be called to define a window larger than the size of the display area requested.

The parameters are:

WIDTH = Total width of the display area in inches. Default value = 8.18 (width of Tekterminal screen in inches)

HEIGHT = Total height of the display area in inches.

Default value = 6.08 (height of Tekterminal
SYMBOL(X, Y, DEGREES, HEIGHT, NCHAR, BCDARRAY)

SYMBOL is a software character generator. That is, the requested characters are actually drawn. In this manner, characters can be produced on any of the plotting devices, including the Calcomp plotter and HP plotter.

The parameters are:

X, Y = X and Y give the coordinate of the lower left point of the first character in the user's units.

DEGREES = Angle in degrees at which the character string and all characters are to be drawn. This is added to the rotation implied by ROTATE.

HEIGHT = Height of the characters in inches. For best results, this should be a multiple of 0.08 inches, since this is the basic grid size used by COMPLOT in drawing all characters.

NCHAR = Number of characters to be drawn.

BCDARRAY = Array containing the BCD codes for the characters to be drawn. If this is a real array, there must be eight characters per element. If it is an integer array, there
should be four characters per element.

The SYMBOL character set includes:

A through Z
0 through 9
space
:\=*%<.>";,-$/*+/\:/?

Character width varies from character to character, and spacing between characters is automatically adjusted according to the particular pair. As a result, characters are spaced to be pleasing to the eye in a manner similar to that practiced by a professional draftsman. In general, the average distance between the lower left of one character and the next is HEIGHT inches.

TEKPAUSE

TEKPAUSE is designed to allow a user to stop at some point in his program to inspect his plotted output, and then resume execution by simply typing any character. If plotting on the Calcomp plotter only, TEKPAUSE will have no effect. In the other cases, TEKPAUSE sets the mode to alphas and waits for the user to respond by typing
any character. TEKPAUSE discards the character that it read and then exits. On the Tekterminal, TEKPAUSE positions the alpha cursor to the upper left-hand corner of the screen.

There are no parameters.

TK4010

COMPLOT assumes that any Tekterminal being used is a Tektronix 4002 graphics terminal. If a 4010 Tekterminal is to be used instead, TK4010 must be called as one of the first subroutines in the COMPLOT package. It sets up the necessary status information to properly control the 4010 Tekterminal. If the plotting device is not a Tekterminal, internal status information is changed, but there will be no apparent difference to the user.

There are no parameters.

VECTORS

Calling VECTORS causes all subsequent plotting to be in the form of smooth lines connecting each specified coordinate.

There are no parameters.

---

Again, SOH should be avoided since it will interrupt the current program and place the user in OS-3 Control Mode.
43

WINDOW(XLOW, YLOW, XHIGH, YHIGH)

In all cases COMPLOT suppresses the plotting of points that fall outside of certain bounds. Normally the bounds used will be the size of the display area specified in the SIZE subroutine (or its default values). However, in some cases it is convenient for the user to control the bounds used by COMPLOT. Therefore, WINDOW may be used to alter the bounds from the complete display area. Each point to be plotted is checked to see if its x-value is between XLOW and XHIGH, and its y-value is between YLOW and YHIGH. Then, the following cases are checked:

1) If the current point and the last point were in bounds, the point is plotted in the current mode.

2) If the current point is in bounds and the last point was not, an intercept is computed and plotted; then the current point is plotted in the current mode.

3) If the current point is out of bounds and the last point was in bounds, an intercept is computed and plotted.

4) If both the current and last point were out of bounds and the line between them passes through the area of plotting, two intercepts are computed and plotted.

5) If both points are in bounds, the current point is plotted.
The parameters passed to WINDOW must define a window that is not larger than the display area set up by SIZE. If the window is too large WINDOW will truncate it at the boundary of the display area.

The parameters are:

\[ \begin{align*}
XLOW &= \text{Low x-value in the user's units} \\
YLOW &= \text{Low y-value in the user's units} \\
XHIGH &= \text{High x-value in the user's units} \\
YHIGH &= \text{High y-value in the user's units}
\end{align*} \]

WINDOW checks to make sure that \( XLOW < XHIGH \) and \( YLOW < YHIGH \). If they are not, they are switched internally.
V. EXAMPLES

COMPLOT is currently being used successfully in several graphics programs and large interactive graphics systems: ORTEP, 3-d plotting programs, a program to flowchart systems, GRAFIT, GROPE, and DRAFT to name a few. In order to illustrate how COMPLOT is used in a graphics application, first consider how the GRAFIT system uses COMPLOT to produce displays. As a last example, a FORTRAN subroutine will be written to plot any two arrays of data passed to it.

GRAFIT

Although GRAFIT uses subroutines from COMPLOT in many different places the major use is in the plotting section. Figure 1 contains a flowchart of the plotting portion of GRAFIT.

If the following sequence of commands are supplied to GRAFIT (1, p. 1-43), the graph in Figure 2 is produced. The underlined items were typed by the computer:

\[ >X''(T) = -LR \times X' - LC \times X \]

\[ X = 1 \]
\[ X' = -0.25 \]
\[ LR = 0.1 \]
\[ LC = 2.5 \]
>DOMAIN, 0, 10, 101

> PLOT, X, X', DASHES, X'', POINTS, MARK(1)

> EXIT
Figure 1. Flowchart of the plotting portion of GRAFIT.
Figure 2. Graph produced by GRAFIT of the solution to a second order differential equation.
Subroutine PLOTAR draws an axis, labels the tic marks, and then plots the data passed to it. The first argument, X, is an array containing the x-values of the coordinates to be plotted and the second argument, Y, is another array containing the corresponding y-values. The third argument, N, indicates the number of coordinates to be plotted. The dimensions of the graph produced by PLOTAR are acceptable for a thesis or technical journal.

In order to illustrate the use of PLOTAR, a main program, PLOTFILE, was written. PLOTFILE reads the data on logical unit one and then calls PLOTAR to plot the data read. Listings of both the routines are shown below:

SUBROUTINE PLOTAR(X, Y, N)
DIMENSION X(1), Y(1)
C SCAN BOTH DATA ARRAYS FOR THEIR MAXIMUM
C AND MINIMUM VALUES.
XLOW=XHIGH=X(1)
YLOW=YHIGH=Y(1)
DO 10 I=2, N
IF (X(I).LT.XLOW) XLOW=X(I)
IF (X(I).GT.XHIGH) XHIGH=X(I)
IF (Y(I).LT.YLOW) YLOW=Y(I)
10 CONTINUE
RETURN
END
IF (Y(I).GT.YHIGH) YHIGH=Y(I)

CONTINUE

C  SET UP ORIGINS FOR THE AXIS.  PICK MINIMUM
C  VALUE UNLESS ZERO IS IN THE RANGE.

XORG=YORG=0.0

IF (XLOW.GT.0.0 .OR. XHIGH.LT.0.0) XORG=XLOW

IF (YLOW.GT.0.0 .OR. YHIGH.LT.0.0) YORG=YLOW

C  SINCE PLOTTYPE IS NOT CALLED CALCOMP PLOTTER
C  ASSUMED.  DEFINE SIZE AND SCALE FACTORS.

CALL SIZE(8.0, 5.0)

XSCL=7.0/(XHIGH-XLOW)

YSCL=4.0/(YHIGH-YLOW)

CALL SCALE(XSCL, YSCL, 0.85, 0.85, XLOW, YLOW)

C  DRAW THE AXIS AND LABEL IT.

XTIC=(XHIGH-XLOW)/10.0

YTIC=(YHIGH-YLOW)/10.0

CALL AXIS(XLOW, XHIGH, XORG, YLOW, YHIGH, YORG, *XTIC, YTIC)

YP=YORG-0.75/YSCL

XP=0.04/XSCL

XX=FLOAT(FIXF(XLOW/XTIC))*XTIC

IF (XLOW.EQ.XORG) XX=XLOW+XTIC

IF (ABS(XX-XORG).LT.XTIC/1.0E6) GO TO 40
ENCODE (8, 30, XL) XX

30 FORMAT(F8.2)
CALL SYMBOL(XX+XP, YP, 90.0, 0.08, 8, XL)

40 XX=XX+XTIC
IF (XX. LE. XHIGH) GO TO 20
XP=XORG-0.75/XSCL
YP=0.04/YSCL
XX=FLOAT(FIXF(YLOW/YTIC))*YTIC
IF (YLOW.EQ.YORG) XX=YLOW+YTIC

50 IF (ABS(XX-YORG).LT.YTIC/1.0E6) GO TO 60
ENCODE (8, 30, XL) XX
CALL SYMBOL(XP, XX-YP, 0.0, 0.08, 8, XL)

60 XX=XX+YTIC
IF (XX. LE. YHIGH) GO TO 50

C NOW PLOT THE DATA POINTS USING A DATA MARK
DO 70 I=1,N

70 CALL PLOT(X(I), Y(I), I-1, 3)

C TERMINATE THE PLOT.
CALL PLOTEND
RETURN
END
PROGRAM PLOTFILE
DIMENSION X(300), Y(300)

C READ THE DATA ON LOGICAL UNIT 1.
N=1

10  X(N)=FFIN(1)
    IF (EOF(1)) GO TO 20
    Y(N)=FFIN(1)
    N=N+1
    IF (N.LE.300) GO TO 10

20  N=N-1

C PLOT THE DATA JUST READ.

CALL PLOTAR(X,Y,N)

END

Figure 3 shows the output produced by PLOTAR when PLOTFILE was used to read some available data and call PLOTAR.
Figure 3. Graph produced by subroutine PLOTAR.
VI. CONCLUSIONS

A comprehensive package of graphics subroutines has been presented. Although COMPLOT is a very versatile set of plot drivers, its core requirements are quite small, approximately 1,800 (3,4008) CDC 3300 words. At the Oregon State University computer center, the only available plot drivers that can come close to offering the same flexibility as COMPLOT is TEKPLOT/CALTEK (5, p. 1-33); therefore, TEKPLOT/CALTEK will be used for a brief comparison. The TEKPLOT package, which plots only on a Tekterminal, occupies approximately 500 more storage locations than the COMPLOT package. If plotting is desired on both the Tekterminal and Calcomp plotters, both CALTEK and TEKPLOT must be used; together, they occupy approximately 4,600 (11,0008) CDC 3300 words.

If the same graphics application is coded using both TEKPLOT and COMPLOT, the execution time using COMPLOT is slightly faster. Comparing COMPLOT and CALTEK on the Calcomp plotter, we find that COMPLOT is between two and three times faster than CALTEK.

COMPLOT is the only plot driver available at the Oregon State University computer center that supports the full 64 BCD characters in its software character generator.
BIBLIOGRAPHY


7. Stockton, F. XYMOVE Plotting. Collected Algorithms from the Communications of the Association for Computing Machinery. Algorithm 162, 162-P-1-0.


APPENDICES
## APPENDIX I

### ASCII Codes for Control Characters

<table>
<thead>
<tr>
<th>7-Bit Octal Code</th>
<th>Character</th>
<th>Action taken on Tekterminal. Applies to both 4002 and 4010 unless stated otherwise</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>NUL</td>
<td></td>
</tr>
<tr>
<td>001</td>
<td>SOH</td>
<td>(4002) Alpha cursor to home position&lt;sup&gt;6&lt;/sup&gt;</td>
</tr>
<tr>
<td>002</td>
<td>STX</td>
<td></td>
</tr>
<tr>
<td>003</td>
<td>ETX</td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>EOT</td>
<td></td>
</tr>
<tr>
<td>005</td>
<td>ENQ</td>
<td>Request current position (preceded by ESC on 4010)</td>
</tr>
<tr>
<td>006</td>
<td>ACK</td>
<td>Rings the bell</td>
</tr>
<tr>
<td>007</td>
<td>BEL</td>
<td>Backspace one character</td>
</tr>
<tr>
<td>010</td>
<td>BS</td>
<td>Line feed</td>
</tr>
<tr>
<td>011</td>
<td>HT</td>
<td></td>
</tr>
<tr>
<td>012</td>
<td>LF</td>
<td></td>
</tr>
<tr>
<td>013</td>
<td>VT</td>
<td></td>
</tr>
<tr>
<td>014</td>
<td>FF</td>
<td>If preceded by ESC, erase screen</td>
</tr>
<tr>
<td>015</td>
<td>CR</td>
<td>Carriage return</td>
</tr>
<tr>
<td>016</td>
<td>SO</td>
<td>(4002) Select italics mode</td>
</tr>
<tr>
<td>017</td>
<td>SI</td>
<td></td>
</tr>
<tr>
<td>020</td>
<td>DLE</td>
<td></td>
</tr>
<tr>
<td>021</td>
<td>DC1</td>
<td>(4002) Turn on crosshairs</td>
</tr>
<tr>
<td>022</td>
<td>DC2</td>
<td></td>
</tr>
<tr>
<td>023</td>
<td>DC3</td>
<td></td>
</tr>
<tr>
<td>024</td>
<td>DC4</td>
<td></td>
</tr>
<tr>
<td>025</td>
<td>NAK</td>
<td></td>
</tr>
<tr>
<td>026</td>
<td>SYN</td>
<td></td>
</tr>
<tr>
<td>027</td>
<td>ETB</td>
<td>End of File</td>
</tr>
<tr>
<td>030</td>
<td>CAN</td>
<td>(4002) Erase screen</td>
</tr>
<tr>
<td>031</td>
<td>EM</td>
<td>Select double intensity</td>
</tr>
<tr>
<td>032</td>
<td>SUB</td>
<td>(4010) If preceded by ESC, turn on crosshairs</td>
</tr>
<tr>
<td>033</td>
<td>ESC</td>
<td>Select alternate character set</td>
</tr>
<tr>
<td>034</td>
<td>FS</td>
<td>(4002) Select point plot mode</td>
</tr>
<tr>
<td>035</td>
<td>GS</td>
<td>Select vectors mode (FS may be used on 4010)</td>
</tr>
<tr>
<td>036</td>
<td>RS</td>
<td>Select incremental plot mode</td>
</tr>
<tr>
<td>037</td>
<td>US</td>
<td>Select alphas mode</td>
</tr>
</tbody>
</table>

<sup>6</sup> SOH cannot be input to a user's program since it places the user in OS-3 Control Mode.
<table>
<thead>
<tr>
<th>7-Bit Octal Code</th>
<th>Character Printed on 4002</th>
<th>Character Printed on 4010</th>
</tr>
</thead>
<tbody>
<tr>
<td>040</td>
<td>space</td>
<td>space</td>
</tr>
<tr>
<td>041</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>042</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>043</td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>044</td>
<td>$</td>
<td>$</td>
</tr>
<tr>
<td>045</td>
<td>%</td>
<td>%</td>
</tr>
<tr>
<td>046</td>
<td>&amp;</td>
<td>&amp;</td>
</tr>
<tr>
<td>047</td>
<td>'</td>
<td>'</td>
</tr>
<tr>
<td>050</td>
<td>(</td>
<td>(</td>
</tr>
<tr>
<td>051</td>
<td>)</td>
<td>)</td>
</tr>
<tr>
<td>052</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>053</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>054</td>
<td>,</td>
<td>,</td>
</tr>
<tr>
<td>055</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>056</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>057</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>060-071</td>
<td>0-9</td>
<td>0-9</td>
</tr>
<tr>
<td>072</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>073</td>
<td>;</td>
<td>;</td>
</tr>
<tr>
<td>074</td>
<td>&lt;</td>
<td>&lt;</td>
</tr>
<tr>
<td>075</td>
<td>=</td>
<td>=</td>
</tr>
<tr>
<td>076</td>
<td>&gt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>077</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>100</td>
<td>@</td>
<td>@</td>
</tr>
<tr>
<td>101-132</td>
<td>A-Z</td>
<td>A-Z</td>
</tr>
<tr>
<td>133</td>
<td>[</td>
<td>[</td>
</tr>
<tr>
<td>134</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>135</td>
<td>]</td>
<td>]</td>
</tr>
<tr>
<td>136</td>
<td>^</td>
<td>^</td>
</tr>
<tr>
<td>137</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>140</td>
<td>@</td>
<td>@</td>
</tr>
<tr>
<td>141-172</td>
<td>a-z</td>
<td>A-Z</td>
</tr>
<tr>
<td>173</td>
<td>{</td>
<td>[</td>
</tr>
<tr>
<td>174</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>175</td>
<td>}</td>
<td>]</td>
</tr>
<tr>
<td>176</td>
<td>~</td>
<td>~</td>
</tr>
<tr>
<td>177</td>
<td>DEL</td>
<td>DEL</td>
</tr>
</tbody>
</table>
APPENDIX II

The following pages contain flowcharts for all of the subroutines in the COMPLOT package. COMPLOT uses a labeled data area, CDA. OZ, which has a few items preset initially:

1) Current pen position is set to the physical lower left corner of all plotting devices
2) Pen is lifted
3) Both the last and current off-screen flags are set to zero
4) Mode is set to alphas
5) Scale factors are set so default values are in inches
6) Rotation is set to zero
7) Display size is set to the height and width of the 4002 Tekterminal (8.18 inches wide and 6.08 inches high)
8) Control information is set to operate a 4002 Tekterminal

An index of the subroutines flowcharted is given on p. 83.
ALPHAS

Set status area to alphas mode

Plotting on Tekterminal

Yes

Output control character to put Tekterminal in alphas mode

No

Plotting on HP plotter

No

Return

Yes

Has any plotting been done

Yes

Output characters to terminate plot on HP plotter

No

Return
AXIS

Call VECTORS

Initialize all optional parameters to zero and then pick up parameters passed to AXIS

Are parameters correct?

No

Make any interchanges or corrections necessary

Yes

Draw the axis using subroutine PLOT

Position pen to the user's origin

Return
Is rate specified > 110?

Yes

Compute number of characters needed to provide correct erase-time delay

Return

DOUBLE

Set flag to indicate double intensity

Plotting on Tekterminal

No

Return

Yes

Output control character to put Tekterminal in double intensity mode

Return
ERASE

Plotting on Tekterminal

No

Yes

Output characters to erase screen

Output enough characters to delay while phosphors cool

Call PAGE

Plotting on Calcomp

No

Return

Yes

Output information to move pen back to left boundary and above current plot

Return
IGRINPUT

Initialize code to turn on crosshair

Plotting on Tekterminal

Output character to perform operation code was initialized for

Input and save the first character

Input the next four characters to determine the coordinate

Convert the coordinate to the user's units

Store the coordinate in the addresses passed to the subroutine

Set function value equal to ASCII code for key pressed

Return
Set status area to italics mode

Plotting on Tekterminal

No
Return

Yes
Output characters to put Tekterminal in alphas mode and then italics mode

Return
Reset flag set by DOUBLE subroutine

Plotting on Tekterminal

Yes

Output character to put Tekterminal in mode specified in status area

Return

No

Return
Call ALPHAS

Plotting on Tekterminal

Yes

Is it a 4002

Yes

Output control character to bring 4002 to home position

No

Plot to home position on the 4010

Return

Return
Update last off-screen flag, last point, and last pen position to the current values

Is pen down

Yes

Is there a data mark to draw

Yes

Draw the data mark

No

Return

Scale and rotate the coordinate

Compute the off-screen flag

Is point on-screen

No → 4

Yes → To p. 69

Was the last point on-screen

No → 4

Yes → To p. 69

Plot to the point on all devices

A

B

3
Plotting on Calcomp only

Yes

No

Plotting on HP plotter

Yes

No

Scale point for the Tekterminal

If necessary, put terminal in right mode and handle pen up or down

Output the characters necessary to address the coordinate

Is it a 4010 and is it in point plot

Yes

No

Lift the pen if it was not already

Use algorithm 162 to move pen to new coordinate

Drop the pen if a point is to be produced

Store information to lift or drop the pen, whichever is required

Use algorithm 162 to move pen to new coordinate

From p. 69

To p. 69

A

B
From p. 67

Point plot mode

Yes

Is current point on screen

No

No

Compute intersection with viewable area

Plot to the point or points computed

To p. 67

Yes

Is point in same sector as last one

No

5 From p. 68

Scale point for the HP plotter

Convert the coordinate to decimal

Convert to ASCII and then output characters

Is the pen to be used

No

Output a "A"

To p. 68

Yes

To p. 68
Empty the Calcomp plot buffer

Is \( 0 < \text{LUN} < 99 \)

Yes

Save LUN in the status area

Plotting on Calcomp

Yes

Input information to label the plotter

Equip LUN to the plotter and label with information inputted

Return

Assume 10 for the LUN number

No

Return
Is code specified valid

Yes

Set status area to indicate device or devices being used

Put impossible value in last pen position

Plotting on Tekterminal

No

Return

Yes

Output character to put Tekterminal in mode specified by status area

Return

Assume Cal-comp only for plotting device
POINTS

Set flag to initialize the HP plotter

Set status area to point plot mode

Plotting on Tekterminal

Is it a 4010

Yes

Set flag to simulate point plot

No

Output character to put Tekterminal in point plot mode

Return

Return
Initialize code to get position of alpha cursor

Plotting on Tekterminal

Yes

1

To p. 63

No

Get the current pen position for the coordinate

2

To p. 63
75

ROTATE

Is angle zero

Yes

Set flag to indicate no rotation

No

Set flag to indicate a non-zero rotation

Convert angle to radians. Compute sine and cosine of the angle. Save results in status area

Return
SCALE
Multiply scale factors specified by 100 and save them

Compute translations and save them

Return

SIZE
Multiply size specified by 100

Is size too large

Yes: Assume largest size possible

No: Initialize boundaries to full screen and calculate multiplicative factor to scale for Tekterminal

Compute off-screen flag using last point

Return
SYMBOLO

Plot to the lower left corner of the 1st character

Save the current scales, origins, and rotation

Define scale factor to give height specified. Define new origin to be current position

Call ROTATE and VECTORS

C

Draw a character

D

Yes More characters to be drawn No 7

Plot to last point of character string

Restore original scales, origin, and rotation

Return
Get pointer to table of coordinates for this character

Have all of the coordinates been used

Yes

Update reference point to lower left corner of next character

No

Set flag for pen up to next point

Is coordinate = 63

Yes

No

Use coordinate to compute distance from reference point

Plot to the point computed

Set flag for pen down to next point
TEXPAUSE

Call PAGE

Plotting on Tekterminal or HP plotter

No → Return

Yes

Input one character and discard it

Return
TK4010

Set flag to indicate a 4010

Change control character used for graphics input and to erase the screen

Change length of vertical axis to 780

Call BAUD

Return
VECTORS

Set flag to initialize the HP plotter

Set status area to vectors mode

Plotting on Tekterminal

No → Return

Yes → Output control characters to put Tekterminal in vectors mode

Plot to last point with the pen up

Turn off flag set by POINTS to simulate point plot mode

Return
WINDOW

Scale and rotate both coordinates specified

Are lower bounds < upper bounds?

Yes

Do bounds exceed display area specified?

Yes

Truncate at edge of display area

No

Compute off-screen flag using last point

Return

Interchange them
# INDEX OF FLOWCHARTS

<table>
<thead>
<tr>
<th>ALPHAS</th>
<th>59</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXIS</td>
<td>60</td>
</tr>
<tr>
<td>BAUD</td>
<td>61</td>
</tr>
<tr>
<td>DOUBLE</td>
<td>61</td>
</tr>
<tr>
<td>ERASE</td>
<td>62</td>
</tr>
<tr>
<td>IGRINPUT</td>
<td>63</td>
</tr>
<tr>
<td>ITALICS</td>
<td>64</td>
</tr>
<tr>
<td>NORMAL</td>
<td>65</td>
</tr>
<tr>
<td>PAGE</td>
<td>66</td>
</tr>
<tr>
<td>PLOT</td>
<td>67</td>
</tr>
<tr>
<td>PLOTEND</td>
<td>70</td>
</tr>
<tr>
<td>PLOTLUN</td>
<td>71</td>
</tr>
<tr>
<td>PLOTTYPE</td>
<td>72</td>
</tr>
<tr>
<td>POINTS</td>
<td>73</td>
</tr>
<tr>
<td>POSITION</td>
<td>74</td>
</tr>
<tr>
<td>ROTATE</td>
<td>75</td>
</tr>
<tr>
<td>SCALE</td>
<td>76</td>
</tr>
<tr>
<td>SIZE</td>
<td>76</td>
</tr>
<tr>
<td>SYMBOL</td>
<td>77</td>
</tr>
<tr>
<td>TEKPAUSE</td>
<td>79</td>
</tr>
<tr>
<td>TK4010</td>
<td>80</td>
</tr>
<tr>
<td>VECTORS</td>
<td>81</td>
</tr>
<tr>
<td>WINDOW</td>
<td>82</td>
</tr>
</tbody>
</table>