

The following is a preliminary report of specifications for the plotter attached to the X1 computer of the Mathematisch Centrum, Amsterdam, and of a program package developed for it. The authors reserve the right to change the specification of any of these programs.

From this date, the procedures PLOTFRAME and PLOT are available as machine-code procedures; PLOTFRAME PLOT, PLOT CURVE, PLOT AXIS, and PLOT TEXT are available in ALGOL 60; FIXPLOT, ABSFIXPLOT, and FLOPLOT are not yet available.

Amsterdam, 25th May 1964

J.A.Th.M. van Berckel

B.J. Mailloux

## X1 Plotter System

Since mid March 1964, a Calcomp 507 digital incremental plotter has been operational on the Electrologica X1 computer at the Mathematisch Centrum in Amsterdam.

The plotter consists of a drum over which paper passes from a supply to a take-up roll. A carriage, carrying a pen, runs parallel to the axis of the drum, a few millimeters from the paper. By means of special machine instructions in the X1, the drum may be rotated forward or backward by one tenth of a millimeter; similarly, the carriage, with the pen, may be moved the same distance to left or right; finally, the pen may be lowered to the surface of the paper, or lifted from it.

We may define a system of co-ordinates by supposing that we view the plotter from the front. Then, the +y direction is to the left, and -y to the right; +x is from front to back, and -x from back to front; +z is pen up, and -z is pen down.

The detailed specifications of the plotter follow

drum speed	300 increments/sec.
carriage speed	300 increments/sec.
pen up/down	10 operations/sec.
step size	.1 mm in x or y
step size	.1 $\times \sqrt{2}$ mm for x and y simultaneously
maximum length of paper	ca. 3650 cm. (120 ft.)
width of paper	ca. 30.5 cm. (12 in.)
width available for plotting	ca. 27.94 cm. (11 in.)

## X1 Plotter Instructions

The six least significant bits of (A) or (S) can be sent to the plotter (PL), with or without inversion.

6Y 8 XP	+ (A) $\Rightarrow$ (PL)
7Y 8 XP	- (A) $\Rightarrow$ (PL)
6Z 8 XP	+ (S) $\Rightarrow$ (PL)
7Z 8 XP	- (S) $\Rightarrow$ (PL)

An action is defined only for the following values of the six bits reaching the plotter:

1	+x	(drum forwards)
2	-x	(drum backwards)
4	+y	(carriage left)
8	-y	(carriage right)
16	+z	(pen up)
32	-z	(pen down)
5	+x and +y	simultaneously
9	+x and -y	simultaneously
6	-x and +y	simultaneously
10	-x and -y	

If a plotter order is given within 3 msec. after another plotter order (100 msec. after pen up or down), then the X1 delays the execution of the given instruction for the appropriate length of time. No interruption facilities are provided.

A special condition setting P version of the above four orders exists to determine whether the carriage is at the border of the available plotting area. If the carriage stands at the right-hand edge and an instruction to move it further right is given or at the left-hand edge and an instruction to move it further left is given, the condition register is set affirmative and no motion of the carriage takes place; in all other cases, the condition is set negative.

#### Manual Controls on the Plotter

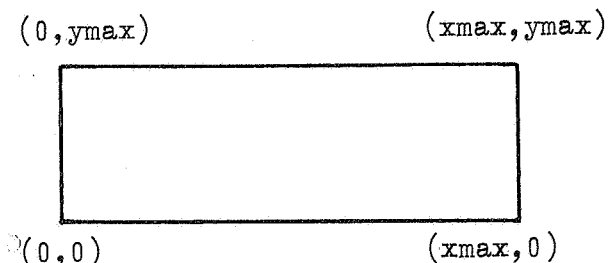
There are several knobs on the plotter to facilitate setting-up:

1. power on/off
2. drum one step forward or backward
3. carriage one step left or right
4. pen up/down
5. drum forward or backward at 120 steps/sec.
6. carriage left or right at 120 steps/sec.
7. chart drive on/off.

### Scaling

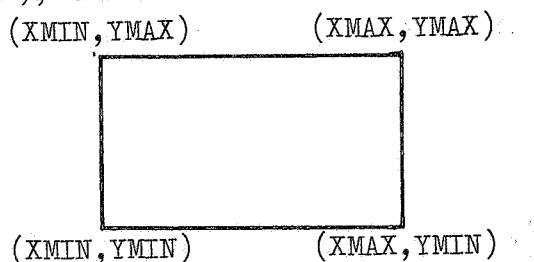
In the sequel, we will make use of certain conventions concerning scaling. For any plotting job, it is necessary to map from the units of the quantities used in the program (hereafter called "data units" or "dits") to some sort of physical "plotter units" ("plits"). We will usually use capital letters to represent variables expressed in data units, and small letters for those in plotter units.

We imagine a rectangle on the paper (the "plotter space") extending from the point  $(0,0)$  at lower left to  $(x_{\max}, y_{\max})$  at upper right, thus:



The unit in this plotter space is 0.1 mm in both x and y directions.

The corresponding rectangle in the "data space" extends from  $(X_{\min}, Y_{\min})$  corresponding to  $(0,0)$  in plotter space to  $(X_{\max}, Y_{\max})$  corresponding to  $(x_{\max}, y_{\max})$ , thus:



Defining the scaling factors

$$Scx = (X_{\max} - X_{\min}) / x_{\max}$$

and

$$Scy = (Y_{\max} - Y_{\min}) / y_{\max},$$

then arbitrary points  $(X, Y)$  in the data space and  $(x, y)$  in the plotter space are mapped onto one another as follows:

$$(X, Y) \Leftrightarrow (x \times Scx + X_{\min}, y \times Scy + Y_{\min})$$

and

$$(x, y) \Leftrightarrow ((X - X_{\min}) / Scx, (Y - Y_{\min}) / Scy).$$

We now proceed to the description of a number of standard procedures developed by the authors.

### The Procedure PLOTFRAME

The purpose of this procedure is to assign values to the quantities XMIN, YMIN, xmax, ymax and to calculate Scx and Scy (these are stored within the procedure PLOT, q.v.). The frame is centered in the y direction on the paper, and the pen is left standing above the point (0,0) in the plotter space ((XMIN, YMIN) in data space).

We here give the heading of the procedure, to show the input parameters

```
procedure PLOTFRAME(XMIN,YMIN,XMAX,YMAX,xmax,ymax);
      value XMIN,YMIN,XMAX,YMAX,xmax,ymax;
      real XMIN,YMIN,XMAX,YMAX; integer xmax, ymax;
```

xmax and ymax are given in plits, of course. If ymax is specified greater than 2750, an error is signalled, and, if the program is continued, ymax is set to 2750.

### The Procedure PLOT

PLOT constitutes the nucleus of the entire set of plotter procedures (including PLOTFRAME); all the other procedures call upon it, either to accomplish some pen movement, etc., or to obtain information about the current pen position, scaling quantities, etc..

Within PLOT are stored the quantities: XMIN, YMIN, xmax, ymax, Scx, and Scy described above; xlast and ylast, representing the current x and y co-ordinates of the pen in plits; and the Boolean pen recording whether the pen is up (true) or down (false).

We now give the heading of the procedure

```
real procedure PLOT(X,Y,IPEN); value X,Y,IPEN;  
                real X,Y; integer IPEN;
```

X and Y usually represent the co-ordinates of a point and may then be in either plits or dits. IPEN usually represents the desired vertical pen position according to a code described below.

#### (a) Calls to Cause Motion of the Pen ( $0 < \text{abs}(\text{IPEN}) \leq 4$ )

For all of these calls, the pen is moved from its current position (recorded in xlast and ylast in plits) in as straight a line as possible to the point represented by X and Y; according as IPEN is positive or negative, X and Y are in dits or plits, respectively. If the point given by X and Y does not lie within the frame, an error is signalled; if the program is then continued, the line will nevertheless be drawn to the prescribed point, unless it lies outside the dimensions of the paper.

The pen is up or down during the drawing of the line depending on the value of  $\text{abs}(\text{IPEN})$ :

```
if  $\text{abs}(\text{IPEN}) = 1$ , the pen is down;  
    2, the pen is up;  
    3, the vertical pen position is unaltered;  
    4, the pen is up, and the prescribed point may lie  
        outside the frame in the +x direction.
```

In this last case ( $\text{abs}(\text{IPEN})=4$ ), xlast and ylast are reset to zero, thus defining the origin of a new frame. If  $\text{IPEN}=-4$  and  $X < 0$ , then the pen moves to the position  $(\text{xmax}+1500,0)$ — i.e. 15 cm. beyond the frame.

After each of these calls, PLOT has the value zero.

(b) Calls to Obtain Information ( $5 \leq \text{IPEN} \leq 19$ )

After each of these calls, PLOT has some value which may be used by the calling program.

IPEN=5	PLOT=the value of X(in dits) converted to plits
6	PLOT=the value of Y(in dits) converted to plits
7	PLOT=the value of X(in plits) converted to dits
8	PLOT=the value of Y(in plits) converted to dits

For the above four calls, the relevant quantity X or Y must lie within the frame or an error condition will be signalled; if the program is continued, the conversion is carried out nevertheless.

IPEN=9	PLOT=the value of XMIN(in dits)
10	PLOT=the value of YMIN(in dits)
11	PLOT=the value of Scx
12	PLOT=the value of Scy
13	PLOT=the value of xmax(in plits)
14	PLOT=the value of ymax(in plits)
15	PLOT=the value of xlast(in plits)
16	PLOT=the value of ylast(in plits)
17	PLOT=the value of xlast converted to dits
18	PLOT=the value of ylast converted to dits
19	PLOT=+1 if the pen is up, -1 if the pen is down.

For the above 11 calls, the values of the parameters X and Y are irrelevant.

(c) Initialization Calls, etc. ( $20 \leq \text{abs}(\text{IPEN}) \leq 25$ )

IPEN=20

This call is used to cause a single movement of the pen according to the value of X; the value of Y is irrelevant. The actions corresponding to the various values of X have already been given under the heading X1 Plotter Instructions. Note that the quantities xlast, ylast, and pen are not updated by this call.

abs(IPEN)=21

This call causes XMIN to receive the value of X, and YMIN the value of Y; if IPEN=+21, the pen is raised, and the quantity pen is initialized.

IPEN=22

This call causes Scx to receive the value of X, and Scy the value of Y.

abs(IPEN)=23

This call causes xmax to receive the value of X, and ymax the value of Y; if IPEN=+23, then xlast and ylast are initialized, and the pen is moved to the point (0,0) in plits ((XMIN,YMIN) in dits). The frame is thereby centered in the middle of the paper in the y dimension.

IPEN=24

This call causes xlast to receive the value of X, and ylast the value of Y.

IPEN=25

This call causes pen to be set true or false according as X is positive or not; the value of Y is irrelevant.

NOTE 1: Certain interlocks are built into the procedure to prevent the use of any quantities which are currently undefined. These initialization calls may be used at any time, but, for all other calls, ( $0 < \text{abs}(\text{IPEN}) \leq 19$ ), it is necessary that calls to initialize all of the quantities XMIN, YMIN, Scx, Scy, xmax, ymax, xlast, ylast, and pen have been given previously. The easiest way to accomplish this is to use PLOTFRAME.

NOTE 2: Notice carefully that after a call with IPEN=20, the quantities xlast, ylast, and pen become undefined, and must be redefined appropriately before calls with  $0 < \text{abs}(\text{IPEN}) \leq 19$  may be employed.

NOTE 3: Failure to observe the above points will result in an error condition.

NOTE 4: A call of PLOT with an invalid value of IPEN will result in an error signal.



### The procedure PLOT CURVE

This procedure is used to draw continuous curves through a set of given points. A special third degree interpolation formula is used.

We here give the procedure heading

```
real procedure PLOT CURVE (X,Y,I); value X,Y,I;
                        real X,Y; integer I;
```

X and Y usually represent the co-ordinates (in dits) of one of the points through which the curve is to pass. The quantity I is used to specify whether we are beginning, continuing, or ending the curve, as follows.

I=0

The desired initial value of the slope of the curve is given by X (in dits); the value of Y is irrelevant. This call serves to initialize the procedure to begin the interpolation process.

I=1

This call is used to initialize the procedure when no initial slope is known; the values of X and Y are irrelevant.

I=2

This call supplies the co-ordinates of one of the points through which the curve is to pass, in X and Y (in dits).

I=3

This call is used to terminate the current curve; the values of X and Y are irrelevant.

I=4

This call is used to terminate the current curve and to specify the desired value of the final slope at the last point given. X supplies the value of the slope (in dits) and the value of Y is irrelevant.

NOTE 1: The procedure is intended to be used by first giving a call with I=0 or 1, followed by a number (at least 3) of calls with I=2, followed by a call with I=3 or 4. The pen position is not altered until the third call with I=2; during this call, the pen moves to the first point given (pen up, of course!), and then draws the segment of the curve to the second point (pen down). During successive calls with I=2, 3 or 4, the

procedure first moves the pen to the last point drawn on the curve (pen up), if necessary, and then continues the curve to the point given in the previous call. Thus, it is possible to use other plotter procedures during the drawing of a curve (e.g. PLOT, PLOTTEXT, etc.).

NOTE 2: If a call with I=2,3 or 4 is given before a call with I=0 or 1, or if a call with I=3 or 4 is given before 3 calls with I=2 have been given, or if a call with I $\neq$ 0,1,2,3 or 4 is given, an error is signalled.

NOTE 3: By slope, we mean  $dY/dX$ .

NOTE 4: After a call with I=0 or 1 and after the first two calls with I=2, PLOT CURVE has the value zero and the pen position is completely unaltered. After any other call, PLOT CURVE has the value of the slope of the curve at the previously given point and the pen is down at the previously given point.

NOTE 5: If any part of the curve falls outside the frame, an error is signalled. If the program is continued, the curve will be drawn nevertheless.

### The procedure PLOTAXIS

The purpose of this procedure is to facilitate the drawing of co-ordinate axes. We here give the procedure heading.

```
procedure PLOTAXIS(X,Y,angle,LENGTH,DL);
      value X,Y,angle,LENGTH,DL;
      real X,Y,angle,LENGTH,DL;
```

The procedure draws a line beginning at the point given in X and Y, in the direction given by angle, for a distance given by LENGTH. Tick marks are made along this line at intervals given by DL.

X,Y,LENGTH, and DL are in dits if LENGTH>0, and otherwise in plits; angle is given in degrees on the paper.

If DL<0, the tick marks are made 1.5 mm. to each side of the line. If DL>0 (but DL<abs(LENGTH)) then the ticks are made to one side of the line only (either above, or to the right, whichever is appropriate). If DL>abs(LENGTH), no tick marks whatsoever are given.

If angle is other than a multiple of  $90^\circ$  and the scaling factors, Scx and Scy, are unequal, then no simple interpretation can be given to the quantities LENGTH and DL.

At exit from the procedure, the pen is always down, at the point (X+LENGTH\*cos(angle), Y+LENGTH\*sin(angle)).

### The Procedure PLOTTEXT

The purpose of this procedure is to plot letters, digits, punctuation marks, and, in general, all valid characters in an ALGOL 60 <string> except for those <delimiters> written in bold face (or underlined) in the ALGOL 60 report. We here give the procedure heading.

```

procedure PLOTTEXT(X,Y,angle,height,italicity,first,i,text);
    value X,Y,first,i;
    real X,Y,angle,height,italicity;
    Boolean first;
    integer i;
    string text;

```

The procedure draws the characters given in text, one after another, beginning at the point X,Y and continuing in the direction given by angle.

X and Y are given in dits or plits according as height is positive or negative, respectively.

angle is given in degrees relative to the plotter frame.

abs(height) is the desired height in plits of the characters. (i.e. the height of a capital letter).

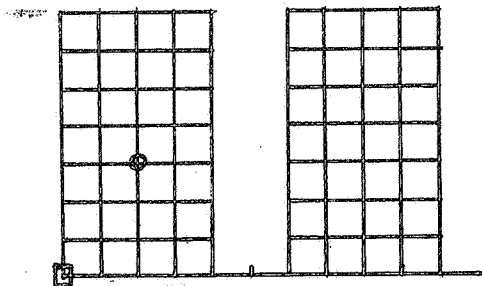
italicity gives the slant of the characters. The top of each character is shifted to the right (i.e. along the direction of the string of characters) by an amount proportional to its height; italicity is the constant of proportionality. For ordinary characters, italicity=0; for italics, italicity=ca. 0.2; negative values are also possible.

when first is true, the drawing of characters begins at the point X,Y; when first is false, it begins immediately after the last drawn character and the values of X and Y are ignored. At least one call with first = true must be given before any calls with first = false; otherwise, an error will be signalled.

If abs(i)=0 or 1, only the characters given in text are drawn; for  $2 \leq \text{abs}(i) \leq 12$ , the characters given in text are preceded by a special character according to the following code:

abs(i)	char.	abs(i)	char.
2	+	8	X
3	X	9	□
4	X	10	◊
5	*	11	° (degrees)
6	Y	12	π
7	⊠		

All characters are described within 4 by 7 boxes, thus:



When  $i \geq 0$ , the point X,Y represents the lower, left hand corner of the box (squared in the diagram) of the first character. When  $i < 0$ , the point X,Y represents the point circled in the diagram. The latter point is the "center" of the box (for  $i < 0$ ); furthermore, the drawing of these characters begins and ends at that point for  $-10 \leq i \leq -2$ . Thus, the characters produced for  $-10 \leq i \leq -2$  are well suited to mark specific points of lines or curves.

text is a string containing the characters to be plotted; it may be empty, if desired.

The pen is always down at exit from this procedure.

If  $i$  is not in the range  $-12 \leq i \leq 12$ , an error will be signalled.

Note that angle,height and italicity are called by name, not by value. These quantities are evaluated exactly once for each character drawn. Thus, it is possible to draw convergent or divergent lines, or curves, of characters, etc.

### The procedures FIXPLOT, ABSFIXPLOT, FLOPLOT

All three of these procedures may be used to draw numbers on the paper. The three procedures differ only in the format in which the number is drawn.

We here give a typical procedure heading.

```
procedure FIXPLOT(X,Y,angle,height,italicity,first,n,m,number);
      value X,Y,first,n,m,number;
      real X,Y,angle,height,italicity,number;
      integer n,m;
      Boolean first;
```

The interpretation of X,Y,angle,height,italicity,and first is the same as for PLOTTEXT. In all cases, number is the number to be drawn; the interpretation of n and m depends on which of the three procedures is used.

#### FIXPLOT

The quantity number is drawn in fixed-point representation with n digits before the decimal point and m after. More exactly:

the sign of number (+ or -);  
the integer part of number, in n digits;  
a decimal point;  
m digits of the fraction part of number;  
a space.

If m=0, the decimal point is suppressed.

Nonsignificant leading zeroes in the integer part are replaced by spaces, except in the units position when m=0.

The number is correctly rounded in the last decimal place. If  $\text{abs}(\text{number}) \geq 10^n$ , then "+inf" or "-inf" will be drawn, space permitting; otherwise, a shortened version, "+in", "-in", "+i" or "-i" will be drawn.

The total number of characters (including spaces) drawn is

$m+n+3$  if  $m \geq 1$   
or  $n+2$  if  $m=0$

The actual parameters for n and m must be of type integer; furthermore they may not be negative, and the case  $n=m=0$  is forbidden.

## ABSFIXPLOT

This procedure is identical in function with FIXPLOT except that the sign of the number is replaced by a blank space.

## FLOPLOT

The quantity number is drawn in floating-point format. The parameter  $n$  specifies the total number of digits in the mantissa, and  $m$  the number in the exponent. More precisely:

the procedure calculates a mantissa  $xi$  and an exponent  $d$ , such that

$$\text{number} = xi \times 10^{\uparrow d},$$

where

$$.1 \leq \text{abs}(xi) < 1;$$

then, the following is drawn

the sign of  $xi$ ;  
 a decimal point;  
 the first  $n$  digits of  $xi$ , correctly rounded in the last place;  
 10;  
 the sign of  $d$ ;  
 the exponent  $d$ , in  $m$  (1,2, or 3) digits, with leading zeroes replaced by spaces;  
 a space.

If  $d \geq 10^{\uparrow m}$ , then "+inf" or "-inf" is drawn depending on the sign of number; the sign appears in the usual place, followed by  $n+m$  spaces, followed by "inf", followed by a space.

If  $d \leq -10^{\uparrow m}$ , then "+0" or "-0" is drawn, depending on the sign of number; the sign appears in the usual place, and the "0" appears at the place of the  $n$ 'th digit.

$n+m+5$  characters are always drawn by the procedure.

The actual parameters for  $n$  and  $m$  must be of type integer; furthermore  $n \geq 0$  and  $m=1,2,\text{ or }3$ .

General

When the parameter first is true, X and Y specify the position of the lower left-hand corner of the box containing the sign of number. When first = false, X and Y are ignored, and drawing continues in the next position following the last output of any of the procedures PLOTTEXT, FIXPLOT, ABSFIXPLOT, or FLOPLOT.