

United Kingdom Atomic Energy Authority
RESEARCH GROUP
Report

USERS GUIDE TO THE CULHAM COMPUTER
GRAPHICAL OUTPUT SYSTEM

F. M. LARKIN

Culham Laboratory
Abingdon Berkshire

1967

Available from H. M. Stationery Office
FOUR SHILLINGS AND SIXPENCE NET

USERS' GUIDE TO THE
CULHAM COMPUTER GRAPHICAL OUTPUT SYSTEM

by

F.M. LARKIN

A B S T R A C T

Basic facilities available in the Culham Laboratory standard graphical computer output system (GHOST) are described, and their practical use is explained. This report supersedes Cos Note 16/66.

UKAEA Research Group,
Culham Laboratory,
Abingdon,
Berks.

December, 1967

C O N T E N T S

	<u>Page</u>
1. OUTLINE OF THE SYSTEM	1
(Main features and design criteria of the graphical output software)	
2. CONTROL OF GRAPHICAL OUTPUT HARDWARE	2
(The B-L 120 printer/plotter, incremental plotter, DEC type 338 display. Frame advance, storage of pictures on magnetic tape and retrieval therefrom)	
3. CORRESPONDENCE BETWEEN MATHEMATICAL SPACE AND PLOTTER SPACES	4
(Mapping regions, space selection, definition of units, limiting rectangles)	
4. STRAIGHT LINE AND CURVE DRAWING	9
(Points and straight lines, rotationally invariant curves, single-valued graphs, full exploitation of an EXTERNAL FUNCTION)	
5. OUTPUT OF CHARACTERS AND POINT PLOTTING SYMBOLS	11
(The seven standard character sets, plotting and typing modes, input and output of character strings, modifications to size, position and shape of characters)	
6. OUTPUT OF NUMBERS	16
(Floating point, fixed point and integer styles, numbers in plotting and typing modes)	
7. MISCELLANEOUS FACILITIES	16
(Border, broken lines, rotation, annotation, contouring etc.)	
8. PROCEDURAL ARRANGEMENTS	20
(The standard GHOST processing run, control cards, QUICK and SLOW modes, streamlined version of the GHOST software for use of the DEC type 338 display, incorporation of 'non-GHOST' graphical output)	
9. ACKNOWLEDGEMENTS	23
10. REFERENCES	23

APPENDICES

APPENDIX A: SUMMARY OF STANDARD GHOST FACILITIES	24
APPENDIX B: THE STANDARD GHOST CHARACTER SETS	28
APPENDIX C: EXAMPLE OF PROGRAM AND OUTPUT	32

1. OUTLINE OF THE SYSTEM

Most computer programs are executed with the object of producing informative output. Usually this output consists of numbers and/or alphabetic information, which is routed through paper, or magnetic tape channels and finally appears as printed symbols on ordinary paper. However, for many purposes, a graph, picture or ciné film will provide a much more convenient representation of the program results than does the usual printed output. Several numerically controlled plotting devices have been installed at Culham Laboratory and a special graphical output language (GHOUL) has been devised to facilitate their use. GHOUL embodies fairly general concepts (Larkin, 1967) which could be implemented in any high level programming language, for a wide variety of graphical output hardware. This report explains how to use GHOST, the Culham implementation of GHOUL in the form of a graphical output system.

Standard GHOST instructions allow the computer user to generate and manipulate graphical information, and then transmit it to any, or all, of the available plotters. As far as the user is concerned these instructions constitute a self contained, indefinitely extendable set of FORTRAN routines, which is independent of the computer and largely independent of the plotters themselves. Furthermore, they are readily interpretable in terms of simple manual, or mental, operations which a human would naturally employ when writing, plotting a graph or drawing a picture. As far as possible the FORTRAN subroutine names are mnemonic references to these operations.

The standard graphical output routines do not use any COMMON or PUBLIC variables. However, in normal usage a minimum of one extra magnetic tape is required; this still leaves three other tapes under independent control of the user's program.

The GHOST software has been designed with the emphasis on flexibility and ease of use. Also, its modular structure, in a high-level programming language, makes its adaptation to other computers and other graphical output hardware a relatively simple task. However, it is worth noting that the adoption of these design criteria has necessarily resulted in a reduction in the speed with which specialised pictures in simple format may be produced. Thus, if computing efficiency is of over-riding importance it might be advisable to forego the use of the facilities described here and to use a machine language program in order to produce the desired graphical output in a form directly acceptable to the required plotters. For most applications, especially in a research and development environment, the efficiency attained using GHOST should be acceptable. Also if required, the basic B-L 120 graphical output commands (c.f. the Programming Manual, ref.3) can be used instead of, or together with, normal GHOST instructions, (see end of section 8).

An example of the use of this graphical output system is given in Appendix C, and the resulting picture is shown in Fig.3. Examples of the use of ciné computer output, using GHOST, may be seen in the documentary film 'Worth a Thousand Words', which was produced by the Computing and Applied Mathematics Group of Culham Laboratory.

2. CONTROL OF GRAPHICAL OUTPUT HARDWARE

The GHOST software is currently capable of handling the following equipment:

(1) A Benson-Lehner model 120 microfilm printer/plotter. This provides output in the form of:

- (a) '½ frame' 35 mm film, nominal frame size 0.7 × 0.68 inches, (17.75 × 17.25 mm)
- (b) 16 mm film, nominal frame size 0.275 × 0.275 inches, (7.0 × 7.0 mm)
- (c) 'Hardcopy', nominal frame size 7.0 × 6.8 inches, (17.75 × 17.25 cm).

The hardcopy, each frame of which is effectively a photographic enlargement of the corresponding microfilm frame, can be obtained in two forms. The so-called 'instant' hardcopy is produced, soon after the originating job has run on the computer, on a special type of photographic paper which is developed and dried, but not properly fixed. Such a record fades within a few days, especially if exposed to strong sunlight, but it enables a usable form of the graphical output to be provided quickly. If required, 'permanent' hardcopy can be provided. This consists of photographic paper records which have been processed by a standard (time consuming) technique, and are not normally produced until the evening shift of the day on which the originating job is run on the computer. Choice of hardcopy type is under program control (see section 8, below).

It should be noted that 35 mm film and 16 mm film cannot be obtained concurrently, but either may be obtained concurrently with hardcopy (subject to the conditions outlined in section 8). The B-L 120 can be operated in either of two modes, the (normal) "B-L 120" mode and the (abnormal) "S-C 4020" mode. Choice of mode is controlled by the operators at the request of users. As far as GHOST users are concerned, the only points to note are that the REPEAT, SIDWYS and LARGE C instructions are inoperative in "S-C 4020" mode.

(2) An incremental pen plotter.

This is not available at the time of writing, but it is expected to have two (mutually exclusive) pens and a paper width of up to 30".

(3) A D.E.C. type 338 graphical input/output device.

This comprises a light-pen and cathode ray tube, controlled via a PDP 8 computer peripheral to the KDF 9. Nominal frame size 9.375 × 9.1 inches.

At the time of publication this may be used as a rapid turn-round output device, but not yet as an input device. Furthermore, special streamlined software will be necessary in order to fully exploit the man/machine interactive capabilities of the display. This is explained more fully in section 8.

From the point of view of the user graphical output channels (1a) or (1b), (1c), (2) and (3) may be regarded as physically separate devices which may be controlled, independently or jointly, by his computer program. On entry to the user's program these devices will be in the "switched-off" state; they must be "switched on" before use and may be "switched off" again when no longer required. Instructions for affecting this symbolic switching are as follows:

CALL FILM35(ONOFF)
CALL FILM16(ONOFF)
CALL HRDCPY(ONOFF)
CALL INCPLT(ONOFF)
CALL DISPLY(ONOFF)

The output medium to which each routine refers should be obvious from its mnemonic name. The INTEGER variable ONOFF, must be set equal to zero in order to switch off the corresponding graphical output channel, and different from zero in order to switch on.

The command CALL FRAME will cause one frame advance on all graphical output channels currently in the switched-on state. The commands

CALL FRAM35
CALL FRAM16
CALL FRAMHC
CALL FRAMIP
CALL FRAMDP

will each cause a single frame advance on its corresponding graphical output channel (provided of course, that the channel is switched-on).

NOTE: If the $\frac{1}{2}$ -frame 35 mm pictures are required to be mounted in standard mounts, it is wise to provide blank frames between the pictures by means of extra frame advance commands. The same advice should be followed if a scene of ciné film is to be made by optically "holding" a single still picture.

Graphical output information is normally stored on a standard system tape 98: however it may also be copied onto one or more extra magnetic tapes designated by the user's program. This information may be retrieved during the same run or at a later date. The command

CALL STORE(NTAPE)

where NTAPE signifies a literal, or symbolic, integer satisfying $1 \leq \text{NTAPE} \leq 97$, will cause subsequent graphical output instructions to be recorded on tape NTAPE. This storage process may be temporarily suspended by means of the command

CALL SUSPND(NTAPE)

Storage may be resumed by repeating the STORE command. When no more storage is required each separate storage tape must be cancelled by means of an instruction of the form

CALL CANCEL(NTAPE)

This has the same effect as the SUSPND instruction, followed by a REWIND NTAPE command.

Notice that graphical output information may be diverted (not merely copied) to tape NTAPE from the standard storage tape 98 by preceding the STORE instruction with the command

CALL SUSPND(98)

Standard storage must be resumed by writing

CALL STORE(98)

before the end of the job.

The CANCEL command has the same effect as SUSPND, when applied to tape 98. Thus, for example, when a large quantity of graphical output is to be produced, most of it can

be diverted to a storage tape while certain key pictures are allowed through the standard process for checking purposes. When the user is satisfied as to the correctness of the stored bulk of the pictures these may be plotted, in a separate job, by means of the RETRVE command described below. Notice, however, that the RETRVE command has no effect on tape 98.

The retrieval of graphical output information stored in this fashion on storage tape NTAPE, and its routing to all currently switched-on plotting channels, is achieved by an instruction of the form

CALL RETRVE(NTAPE, M, N)

This will search tape NTAPE for $(N-M + 1)$ frames of graphical information, starting with the M^{th} frame and ending with the N^{th} . Any frames (or parts of frames) having numbers in this range will be retrieved and plotted in the usual fashion. The INTEGER variables M and N must satisfy the conditions $0 < M \leq N$. Notice that for this purpose, frame numbers are allocated as the graphical information is generated and each frame retains this 'absolute' number regardless of how, or when, it is recorded onto the storage tape.

Any one of the six permissible frame advance commands also acts as a frame separator. Thus frame 0 comprises any graphical output instructions generated before the first frame advance command. Frame 1 starts with the first frame advance command and extends up to, but does not include, the second frame advance command, etc. It should be clear that, in order for stored graphical instructions to be retrieved, they must include at least one frame advance command. If a user wishes to store, on magnetic tape, the complete graphical output from his job he will probably find it convenient to use the technique described in section 8.

It is hoped that the incremental plotter will have two plotting pens, using different coloured inks; these will be selected and altered by means of the instructions

CALL REDPEN and CALL BLKPEN .

The instruction CALL REPEAT(NTIMES)

where NTIMES is a positive INTEGER quantity not exceeding 15, only affects the micro-film and/or hardcopy output channels. It causes the preceding frame to be repeated NTIMES times on any of these channels which is currently switched-on. It is principally useful when producing ciné films. Note that this instruction is used instead of a frame advance command; i.e. if immediately preceded by a frame advance command the result will be NTIMES blank frames!

3. CORRESPONDENCE BETWEEN MATHEMATICAL SPACE & PLOTTER SPACE

When drawing a graph, a human being, consciously or otherwise, employs two distinguishable frames of reference. One such frame of reference is in plotter space (P-space), i.e. the real, physical space of the plotting paper, in which titles, annotation, etc., are arranged in convenient positions, with relative distances usually measured in inches or centimetres. The human operator establishes a correspondence between this P-space and the mathematical space (M-space) in which his graph originates. He mentally identifies points in P-space, which he can see, with corresponding points in M-space, which he usually has to imagine. Obviously this mapping between M-space and P-space must be defined before any plotting of M-space quantities can be performed.

It is assumed that the user will wish to map rectangular regions in M-space into rectangles in any selected number of the available P-spaces. In all cases the sides of these rectangles will be taken as parallel with corresponding rectangular, cartesian, co-ordinate axes, although their shapes, sizes and off-sets from their respective origins are under program control.

Figures 1 and 2 illustrate regions of interest in M-space and P-space, respectively. Notice that a natural, physical origin of co-ordinates exists in P-space, and two rectangular regions are specified relative to this. The rectangle ABCD in M-space is mapped linearly into the rectangle A'B'C'D' in P-space, i.e. XP depends linearly upon XM and YP depends linearly upon YM, although the constants in the two linear relations need not agree.

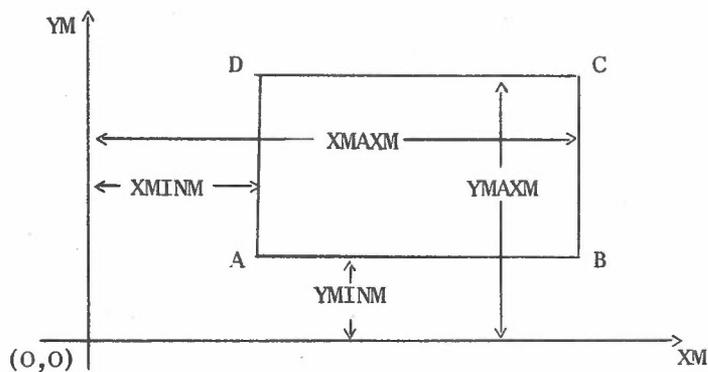


Fig.1

The Rectangular Region of Interest in M-space

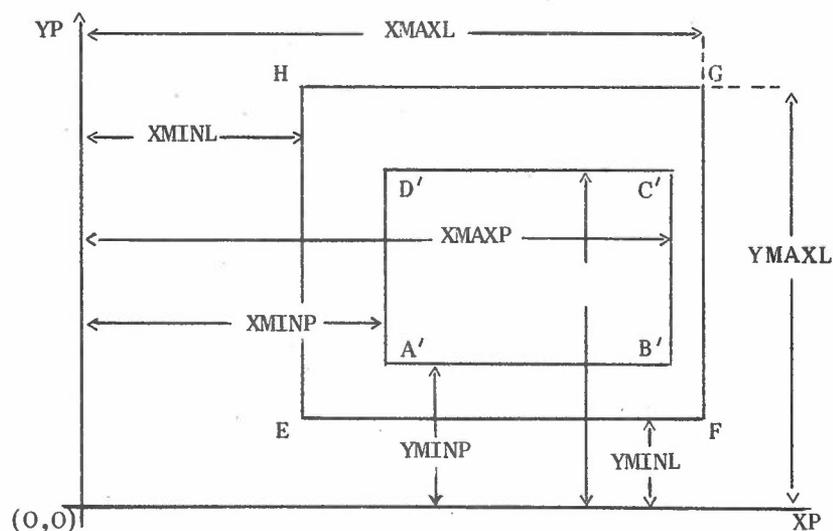


Fig.2

Rectangular Regions of Interest in P-space

The rectangle EFGH in P-space encloses the region to which any form of plotting initiated by the user will be limited. By suitably positioning EFGH relative to A'B'C'D', room may be allowed for annotation of axes, titles and/or other descriptive material. The effect is that M-space information may be plotted only if it lies within ABCD (thereby appearing within A'B'C'D') and P-space information may be plotted only if it lies within EFGH. Notice that the dimensions of rectangles ABCD, A'B'C'D' and EFGH may be altered by the user at any time during the course of his program. Thus, for example, it is possible to plot consecutively the images of several M-spaces side by side on a single P-space frame.

It should be realised that, whereas the mapping regions are necessary in order to establish a correspondence between M-space and the various P-spaces, the limiting rectangles have been introduced in order to simplify the location of textual output. A limiting rectangle is analogous to the smallest rectangle which just contains all the text on a page in a typical book (excluding page number and footnotes). It also serves, incidentally, to delineate that part of a P-space outside of which no plotting may occur.

Since a human operator is constantly shifting his mental frame of reference from M-space to P-space and back, while plotting a graph, special GHOUL commands are provided in order to simulate the process. This shift of attention is referred to below as selection of the corresponding space, a process which should not be confused with the operation of switching on an output channel. In GHOST the two operations are performed by the same routine, but this is not necessary and, logically, perhaps even rather undesirable.

Once any particular space is selected (regardless of whether or not it is still switched on) all subsequent dimensioned arguments will be interpreted as being measured in the units (inches, centimetres, mathematical units, etc.) which currently apply in the selected space. A P-space cannot be deselected by switching it off - only by explicitly selecting a different space.

Thus, one might switch on several output channels, select M-space, cause a curve to be drawn (i.e. output on all currently switched-on channels), next select a particular plotter space and finally write a title, all on the same frame. Note that any information plotted when a particular P-space (or M-space) is selected will also appear, suitably scaled, on all currently switched on output channels, since correspondences between their mapping regions will have been established either explicitly, or by default. It should be clear that, whereas any number of the available output channels (each with its associated P-space, but not M-space, of course) may be switched on simultaneously, only one P-space (or M-space) may be selected at any given instant. Furthermore, it is even possible to switch-off a selected output channel without deselecting it.

In order to define rectangle ABCD, M-space is first selected by means of the command:

```
CALL MSPACE
```

Any references to lengths or co-ordinates, made by any of the graphical output instructions following this command, will be interpreted as referring to mathematical space. Thus, the region enclosed by rectangle ABCD may now be defined by means of the instruction:

```
CALL REGION(XMINM, XMAXM, YMINM, YMAXM)
```

where the arguments must be either literal or symbolic REAL quantities.

In order to complete the mapping, rectangle A'B'C'D' must be defined by another REGION command, after selecting the required plotter space and defining units of inches, centimetres or millimetres. The hardware switching instructions, FILM35, etc. are used for the purpose of P-space selection as well as for switching the corresponding channels on and off. Thus, in order to define the rectangle A'B'C'D' in units of inches in hardcopy space the appropriate instructions would be

```
CALL HRDCPY(1)
CALL INCHES
CALL REGION(XMINP, MXAXP, YMINP, YMAXP)
```

While hardcopy space is still selected, rectangle EFGH could be defined by means of the command

```
CALL LIMITS(XMINL, XMAXL, YMINL, YMAXL)
```

where, as before, the arguments of REGION and LIMITS must all be literal or symbolic REAL quantities. The units of measurement in any of the plotter-spaces may be set or changed at any position in the user's program by means of any of the commands

```
CALL INCHES
CALL CMS
CALL MMS
```

Units applying in any particular space will normally be defined immediately after selecting that space for the first time. Errors will occur if the dimensions set by REGION or LIMITS in P-space exceed the space available on the corresponding plotter.

Notice that whereas, for example, the instruction

```
CALL HRDCPY(1)
```

has the two effects

- (i) 'switch on hardcopy channel, if not already on',
- (ii) 'select hardcopy-space',

the command

```
CALL HRDCPY(0)
```

only has the effect

'switch off hardcopy channel, if not already off'.

It does not 'deselect' hardcopy space in any way, so that subsequent graphical output instructions will be interpreted as referring to that space, although actual output will, of course, only occur on those channels which are currently switched-on. Thus any graphical output directed to any currently selected channel will appear on every channel which is currently switched-on, regardless of whether or not the currently selected channel is itself still switched-on.

Since microfilm and hardcopy outputs are each photographs of the same physical cathode ray tube, the convention is adopted that the channel whose REGION was last specified takes precedence over the other. The incremental plotter and on-line display are quite independent.

In order to illustrate the mapping process, consider the following example. Suppose we wish to map a rectangle in M-space, defined by:

XMINM = -5.0, XMAXM = 2.0

YMINM = 0.0, YMAXM = 6.0

into a rectangle in hardcopy-space defined by

XMINP = 1.0 cm., XMAXP = 16.0 cm.

YMINP = 1.0 cm., YMAXP = 16.0 cm.

and also into a rectangle in incremental-plotter-space defined by

XMINP = 1.0 inches, XMAXP = 7.0 inches

YMINP = 1.0 inches, YMAXP = 6.0 inches

In hardcopy-space all annotations initiated by the user is to be restricted to a rectangle defined by

XMINL = 0.0 cm., XMAXL = 16.0 cm.

YMINL = 0.0 cm., YMAXL = 16.0 cm.

and in incremental-plotter-space the limiting rectangle is to be defined by

XMINL = 0.0 inches, XMAXL = 7.5 inches

YMINL = 0.0 inches, YMAXL = 6.5 inches.

The required mappings and associated plotting limits could be set up by means of the following sequence of commands:

CALL MSPACE

CALL REGION(-5.0, 2.0, 0.0, 6.0)

CALL HRDCPY(1)

CALL CMS

CALL REGION(1.0, 16.0, 1.0, 16.0)

CALL LIMITS(0.0, 16.0, 0.0, 16.0)

CALL INCPLT(1)

CALL INCHES

CALL REGION(1.0, 7.0, 1.0, 6.0)

CALL LIMITS(0.0, 7.5, 0.0, 6.5)

The two mappings are now completely established, and will remain so until explicitly altered by the user's program. At this stage all subsequent plotting instructions will be interpreted as referring to incremental-plotter-space, in units of inches, until another space is explicitly selected by the user's program. Notice that, as illustrated in the above example, the REGION, LIMITS and appropriate units should be specified for each separate plotting channel before any output on that channel is initiated. Once established, these properties remain associated with their corresponding channels until explicitly changed (apart from the previously noted REGION interdependence of microfilm and hardcopy).

If, while M-space is selected, the REGION command is given with the two y limits both set equal to 0.0, these will later be chosen automatically so as to accommodate conveniently the curve drawn by the first succeeding curve-drawing command. That is, the mapping will not be properly established until one of the commands, CURVEO, CURVEC, GRAPHN, GRAPHP or GRAPHF, is executed (see next section).

4. STRAIGHT LINE AND CURVE DRAWING

In this section, and those following, any reference to a position or length on a graph should be interpreted as applying to the currently selected space, in the units currently relating to that space.

A straight line joining the points (X1, Y1) and (X2, Y2) may be drawn by first specifying the starting point, either by the command

```
CALL POINT (X1, Y1)
```

or by the command

```
CALL POSITN(X1, Y1)
```

and then following this, either by the command

```
CALL JOIN(X2, Y2)
```

or by the command

```
CALL LINE(DELTAX, DELTAY)
```

where, of course,

```
DELTAX = X2 - X1
```

```
DELTAY = Y2 - Y1
```

A polygonal arc may be continued to the points (X3, Y3), (X4, Y4), etc., either by the instructions

```
CALL JOIN(X3, Y3)
```

```
CALL JOIN(X4, Y4)    etc. etc.
```

or by the corresponding LINE instructions.

In terms of normal manual plotting operations the POSITN command lifts the pen from the paper and moves it to the position specified in the argument list; no mark is made on the paper, but the 'current character typing position' has been altered. The POINT command lifts the pen from the paper and sets it down again at the specified point, marking a dot on the paper in the process; it does not alter the current character typing position. It is recommended that the POINT command be used for starting a line, in preference to the POSITN command, which should be reserved for specifying the current character typing position (see section 5).

Four basic commands are provided for the purpose of drawing smooth curves which interpolate a finite number of given points. These cover the following four most frequently occurring situations:-

- (i) Non-periodic and single-valued graphs;
- (ii) Periodic and single-valued graphs;
- (iii) Open-ended, and possibly multi-valued, curves;
- (iv) Closed and multi-valued curves.

The statements

```
CALL CURVEO(XV, M, N)
```

and

```
CALL CURVEC(XV, YV, M, N)
```

will each result in the drawing of a smooth curve through the (N-M+1) points ((XV(I), YV(I)), I = M, N). In both cases the shape of the interpolated curve will depend only upon the relative point positions and not upon the position or orientation of the co-ordinate

axes; the order of the points along the curve will agree with their order as implied by their positions in the vectors XV and YV. In the case of CURVEC the interpolated curve will be closed; in the case of CURVEO the curve will be open-ended, starting from the point (XV(M), YV(M)) and ending at the point (XV(N), YV(N)). Of course, these two commands will be used only when a metrical relationship is known to exist between the two co-ordinate axes, as, for example, when spot positions on a contour are known and it is required to approximate the contour itself.

The instruction

```
CALL GRAPHN(XV, M, N)
```

will result in the drawing of a smooth graph through the points ((XV(I), YV(I)), I = M,N). The abscissae are assumed to satisfy the conditions

$$XV(M) < XV(M+1) < \dots < XV(N-1) < XV(N).$$

The curve Y(X) will be single-valued and its shape will be invariant under a translation, but not a rotation, of the co-ordinate axes.

The instruction

```
CALL GRAPHP(XV, YV, M, N, PERIOD)
```

performs a function similar to that of GRAPHN, except that the interpolating curve produced will be periodic with period PERIOD. Note that, if $N-M \geq 100$ in any of the above four curve drawing routines the given points will be connected by straight lines, to form a polygonal arc, i.e. the effect will be as if the PTPLOT command had been used (see section 7).

The instruction

```
CALL GRAPHF(FUNXN)
```

where FUNXN is the name of any EXTERNAL function subprogram, e.g.

```
FUNCTION FUNXN(X),
```

defined over that part of the range of its argument X which lies within the allowed plotting region of the currently selected space, will result in the graph of FUNXN(X) being plotted. For example, if M-space happens to be selected the range of the independent variable X would be (XMINM, XMAXM), (see Fig.1). If FUNXN(X) violates either of the y limits YMINM and YMAXM the corresponding part of the graph will be chopped off, unless of course the call of GRAPHF is also being used for the purpose of defining YMINM and YMAXM by the device of setting them both to 0.0 in the REGION call, (see previous section).

The apparent restriction of this particular EXTERNAL function format is illusory; the following example illustrates how FUNCTIONS or SUBROUTINES with any desired number of arguments may be used here:

Suppose SUBROUTINE GUY(X,Y,A,B,C,J) is to evaluate a Y(X) which depends upon the parameters A,B and C. These parameters are to be changed during the calculation so as to alter the form of Y(X). Adopt the convention that when GUY is entered with the index J equal to 1 the routine will use existing, stored values of A,B and C, and the input value X, in order to evaluate Y. However, if GUY is entered with J = 2 the routine will merely store the input values of A, B and C, e.g.

```

          SUBROUTINE GUY(X,Y,A,B,C,J)
C
      GO TO (1,2),J
C
          STORE PARAMETER VALUES
C
      2   AS = A
          BS = B
          CS = C
          RETURN
C
      =====
C
          EVALUATE Y(X), USING STORED PARAMETERS
C
      1   Y = AS*X**2 + BS*X + CS
          RETURN
C
      =====
C
          END

```

Now, let us write the EXTERNAL function as

```

          FUNCTION FAWKES(X)
C
      CALL GUY(X,Y,DUMMY,DUMMY,DUMMY,1)
      FAWKES = Y
      RETURN
C
      =====
C
          END

```

It should be clear that the following sequence of commands

```

      CALL GUY(X,Y,1.0,- 2.0, 0.5, 2)
      CALL GRAPHF(FAWKES)
      CALL GUY(X,Y,0.0, - 0.2, 3.0, 2)
      CALL GRAPHF(FAWKES)
      CALL GUY(X,Y, - 5.0, 0.0, 2.0, 2)
      CALL GRAPHF(FAWKES)

```

will result in the plotting of three different curves, Y(X). Clearly, the number and type of parameters A,B,C, ... etc., is not limited by this device.

5. OUTPUT OF CHARACTERS AND POINT PLOTTING SYMBOLS

For the purposes of graph annotation and marking of points by plotting symbols, seven character sets are provided. These sets have been agreed in collaboration with N.P.L. (Heap and Nott, 1967). The sets are numbered 0, 1, 2, 3, 4, 5 and 6, and may be selected by means of the command

```
CALL CTRSET(NOSET)
```

where NOSET is a literal or symbolic INTEGER quantity which may assume any of the above values. If this command is omitted character set 0 is automatically selected. Characters output when set 0 is selected will take the form of 'hardware' characters (i.e. fixed size and orientation) in the case of the microfilm recorder and on-line display, and these will be simulated by 'pseudo-hardware' characters (i.e. drawn by the pen) in the case of the incremental plotter. Each character set consists of 64 characters, some of which are duplicated among the sets. These are illustrated in Appendix B, but it will be convenient here also to tabulate their principal distinguishing features:

Character Set

0	B-L120 Hardware or Pseudo-hardware (contains Hollerith set)
1	Upper case Roman (contains Hollerith and Teletype character sets)
2	Lower case Roman and miscellaneous mathematical symbols
3	Upper case Greek and miscellaneous mathematical symbols
4	Lower case Greek and point plotting symbols
5	Flexowriter case 'normal' and KDF 9 line printer characters
6	Flexowriter case 'shifted' characters.

Other character sets may be introduced later, those numbered 10, 20, 30, etc. being reserved for hardware characters. While the on-line C.R.T. display is being used only character set 1 will be available.

Within each set every character is distinguished by a serial number running from 0 to 63. In the case of the integers 0, 1, 2, ... 9, the serial number is equal to the character itself. Where applicable, similar characters in different sets have the same serial number; e.g. characters A, a and α are in different sets, but they all have the same serial number, 11.

In the case of the six 'software' character sets (1, 2, 3, 4, 5 and 6), the nominal size of characters may be specified by means of an instruction of the form

```
CALL CRSIZE(HEIGHT)
```

where HEIGHT is a literal or symbolic quantity specifying the nominal character height. The nominal width of a character is 0.55 of its nominal height (except for hardware characters, when the ratio is $\frac{1}{2}$). Thus, in a page 8.0 inches square it will be possible to plot 40 lines of characters with HEIGHT equal to 0.2 inches, with 72 characters on each line. Applied to the hardware characters, this command can only affect the typing space made available to them. However, refer to the LARGE and SMALL commands described later in this section for a limited degree of control over the size of hardware characters.

If this command is to be executed while M-space is selected the argument HEIGHT should be set to some reasonable fraction of (YMAX-YMIN); the character width will then be chosen by automatic reference to the previously assigned value of (XMAX-XMIN).

Once a 'reasonable' character size has been established in any one of the spaces used, it should not be necessary to repeat the process for any of the other spaces which might be selected. Regardless of how mapping regions are altered, spaces selected and units changed, the local proportions of characters should remain constant, as will their height relative to the maximum height available. Of course, character size may be altered, if desired, by means of a further call of subroutine CRSIZE, in any required space.

A single character may be located on a page, either by giving the Cartesian co-ordinates of its centre, or by specifying the number of lines (measured downwards from the upper edge of the limiting rectangle) and the number of spaces (measured to the right from the left-hand edge of the limiting rectangle) on a (space, line) co-ordinate grid. Characters may be output individually, or in strings.

For the purpose of character output two modes, plotting mode and typing mode, are distinguished. All plotting mode commands must supply the (x,y) co-ordinates of the position at which a character is to be plotted.

Whenever a character is output by means of a 'TYPE $\alpha\beta$ ' command a current typing position is established, immediately to the right of the position of the last character. If another typing mode command then follows, the character next referred to is output at the current typing position. Of course, no positional information is supplied with a typing mode command.

After defining a character set and size, a character may be plotted centred on position (X,Y) by means of the command

```
CALL PLOTNC(X, Y, NOCHAR)
```

where NOCHAR is the serial number of the required character within the currently selected set. Of course, subsequent PLOTNC commands may be given without repeating the CRSIZE and CTRSET instructions.

For some purposes, for example in printing names or titles, the position of the next character will be uniquely determined as one space to the right of the character plotted immediately previously. The first character in a string of such related characters could be plotted, either by means of the PLOTNC command, or by means of the two instructions

```
CALL POSITN(X, Y)
CALL TYPENC(NOCHAR)
```

Subsequent characters in the string could then be plotted by means of an appropriate sequence of TYPENC commands. Note that the TYPENC instruction automatically produces a character space after the numbered character has been plotted, just as on a normal typewriter; no such spacing occurs with the PLOTNC instruction.

In addition to the facility for plotting characters by referring to their serial numbers, Hollerith characters may be plotted by referring either explicitly to them, or to a vector in which a string of Hollerith characters (≤ 80) is stored. Thus the instruction

```
CALL PLOTCS(X, Y, Phrase, NC)
```

where 'Phrase' is either a vector containing not more than 80 Hollerith characters, or a string of such characters in FORTRAN H-format, will result in the plotting of a line (or lines) of text comprising the first NC characters in the string, with the first character centred on the point (X, Y). Similarly, the command

```
CALL TYPECS(Phrase, NC)
```

will result in the first NC of the Hollerith characters in 'Phrase' being plotted, with the first character being centred on the current character typing position. However, notice that true Hollerith characters will only appear if character set 0 or 1, is currently selected; otherwise characters in the selected set which correspond to those in set 1 will be plotted.

Hollerith text may be read from a card, into a vector PHRASE, by means of the instruction

```
CALL READCS(PHRASE, NC)
```

where NC is the number of characters, starting from column 1, required to be read from the card. Note that the vector PHRASE must appear in a DIMENSION statement within the routine in which the instruction appears; also, in order to allow space for the full 80 characters on the card, the dimension of PHRASE must be at least 10.

For some purposes, especially when using TYPE $\alpha\beta$ instructions it is more convenient to locate character typing positions on a (space, line), rather than an (X, Y), co-ordinate system. In this scheme, the top, left-hand typing position has co-ordinates (1, 1), the height of a line being taken as the current nominal character height and the width of a space being the current nominal character width. Thus, instead of using the POSITN instruction, the current character typing position may be located by means of the command

```
CALL PLACE(NSPACE, NLINE)
```

where NSPACE is the number of character widths measured from the left-hand edge of the rectangle specified by the LIMITS instruction, and NLINE is the number of character heights measured downwards from the upper edge of this rectangle. Note that, although the LIMITS instruction is not normally used in M-space, the limiting rectangle is automatically set equal to the rectangle specified by the M-space REGION command, and character output may be effected in M-space just as easily as in any of the plotter-spaces. Of course, if required the M-space limiting rectangle can be altered by means of the LIMITS command.

The command

```
CALL SPACE(NSPACE)
```

has the effect of shifting the current character typing position |NSPACE| character widths to the right if NSPACE \geq 0 and |NSPACE| character widths to the left if NSPACE \leq 0.

Similarly, the command

```
CALL LINEFD(NLINES)
```

performs the typewriter operation 'line-feed' NLINES times. If NLINES \geq 0 the current character typing position moves down the page and if NLINES \leq 0 the position moves up the page. If line feeding moves the current character typing position off the bottom of a page it reappears at the top left-hand corner. Similarly, the current character typing position is automatically transported to the bottom right-hand corner after too many upward line shifts. The command

```
CALL CRLNFD
```

performs the familiar typewriter operation 'carriage-return, line-feed'. Note that 'carriage-return, line-feed' is automatically provided if, for example, a TYPECS instruction inadvertently overflows the edge of the page set by the LIMITS command.

As an example of some of the above facilities, the following sequence of commands might be used to write a title on a frame of graphical output.

```
CALL CTRSET(1)
CALL CMS
CALL CRSIZE(0.5)
CALL PLACE(3, 1)
CALL TYPECS(8HG FAWKES, 8)
CALL CRLNFD
CALL READCS(PHRASE, 7)
CALL SPACE(10)
CALL TYPECS(9HPLOT CODE, 9)
CALL SPACE(2)
CALL TYPECS(PHRASE, 7)
```

Characters may be italicised or de-italicised by means of the command

```
CALL ITALIC(ONOFF)
```

where the argument is either a literal or symbolic INTEGER quantity.

ONOFF \neq 0 signifies 'italics required'

ONOFF = 0 signifies 'revert to non-italic mode'

Three commands, SUFFIX, SUPFIX and NORMAL, are provided whose function is conveniently illustrated by remarking that graphical output of the equation

$$A_{BC}^D = E_F G$$

may be achieved by means of the sequence of commands

```
CALL TYPECS(1HA, 1)
CALL SUPFIX
CALL TYPECS(1HB, 1)
CALL SUFFIX
CALL TYPECS(1HC, 1)
CALL NORMAL
CALL TYPECS(1HD, 1)
CALL NORMAL
CALL TYPECS(2H = E, 2)
CALL SUFFIX
CALL TYPECS(1HF,1)
CALL NORMAL
CALL TYPECS(1HG, 1)
```

A maximum total of five (mixed) SUFFIX and SUPFIX levels is allowed, excluding the starting level. Thus, on C the example attains the second level. Note the the NORMAL command reverts one level towards the starting level.

The following six commands refer only to the microfilm or hard-copy output channels.

Two sizes of hardware characters are available; they may be selected by means of the instructions

```
CALL LARGE C
```

and

```
CALL SMALL C
```

The small characters are about 2/3 the height of the large characters, although the typing space allowed is the same in both cases. If these commands are omitted any hardware characters will be plotted in the larger style.

Two characters plotting densities are available; these may be selected by means of the commands

```
CALL LIGHT
```

```
CALL HEAVY
```

If these commands are omitted the lighter plotting mode will be assumed. These commands are available in addition to the DENSITY instruction described in section 7, but they apply only to the microfilm and hardcopy channels.

Two orientations of hardware characters are allowed - standard, and rotated anti-clockwise through 90°. These orientations may be selected by means of the commands

```
CALL STNDR
```

```
CALL SIDWYS
```

If these commands are omitted the standard orientation is assumed. These commands are available in addition to the ROTATE command described in section 7, which does not affect the orientation of hardware characters.

6. OUTPUT OF NUMBERS

It is very often necessary to annotate graphical output with printed numbers; this is facilitated by the following instructions.

The command `CALL PLOTNE(X, Y, Z, NDP)` causes the REAL quantity Z to be plotted in FORTRAN E-format (Floating point), with NDP decimal places and the decimal point character centred on the point (X, Y). Note that, since the decimal point does not lie at the centre of the space allowed for a character, it will not actually coincide with the point (X, Y), but will lie a little below it. Similarly, the command

`CALL PLOTNF(X, Y, Z, NDP)`

causes the REAL quantity Z to be plotted in FORTRAN F-format (fixed point), with NDP decimal places and the decimal point character centred on the point (X, Y). The command

`CALL PLOTNI(X, Y, I)`

causes the INTEGER quantity I to be plotted, with the position of its implied decimal point character centred on the point (X, Y).

The commands `CALL TYPENE(Z, NDP)`

`CALL TYPENF(Z, NDP)`

and `CALL TYPENI(I)`

correspond, in an obvious fashion, with the above three PLOTN α instructions. However, in the case of the TYPEN α commands the position of the decimal point is irrelevant, the numbers being located so that their signs (first preceding blank in the case of positive numbers) are centred on the current character typing position. In all cases above, the allowed number of decimal places is limited to a maximum of ten.

As with other character output commands, character set 0 is automatically selected if the CTRSET instruction is omitted altogether.

7. MISCELLANEOUS FACILITIES

The instruction `CALL BOX(XMIN, XMAX, YMIN, YMAX)` will result in the drawing of a rectangular box, with edges parallel with corresponding co-ordinate axes unless modified by the ROTATE instruction (see below). The rectangle is defined by the arguments (XMIN, XMAX, YMIN, YMAX) in the same fashion as rectangles are defined by the REGION and LIMITS commands (see Figs.1 and 2).

The instruction `CALL BORDER` draws a box around the rectangular mapping region.

Automatic rotation of a selected part of a graph may be achieved by preceding the commands which are required to be modified by rotation with a command of the form

`CALL ROTATE(XC, YC, ANGLE) ,`

expansion about (XC, YC) by RATIO is achieved by means of the instruction

`CALL EXPAND(XC, YC, RATIO),`

while the command `CALL EXPROT(XC, YC, RATIO, ANGLE)`

combines these two operations.

Here (XC,YC) denotes the centre of rotation and ANGLE denotes the anti-clockwise algebraic rotation, in units of radians. This modification will have no effect on the REGION and LIMITS instructions or on the orientation of hardware characters (refer to the STNDRD and SIDWYS commands). When rotation is no longer required the command should be repeated with ANGLE set to 0.0.

Some degree of control over the line density of graphical output is possible, by means of the command

```
CALL DENSTY(LTORDK)
```

where LTORDK is a non-negative, literal or symbolic INTEGER. This has the effect of causing subsequent graphical output to be drawn LTORDK times. Thus, no output occurs if LTORDK=0, standard density is obtained if LTORDK = 1, twice standard density is obtained if LTORDK = 2, etc. Refer also to the LIGHT and HEAVY commands described in section 5.

Straight or curved lines, drawn by other standard routines in the graphical output system, can be modified to appear as broken lines by preceding the appropriate commands by the instruction

```
CALL BROKEN(N1, N2, N3, N4)
```

where the arguments are all positive INTEGERS. The effect of this will be to split up any subsequent straight line or curve into a number of 'periods' along its length. The length of each 'period' will be $(N1 + N2 + N3 + N4) * DS$, where DS is a small standard length in plotter space, and this will be further subdivided into four sections, of lengths $N1 * DS$, $N2 * DS$, $N3 * DS$ and $N4 * DS$ respectively. Within each 'period' the first and third subsections will be drawn, but the second and fourth will be blank, giving a broken effect to the line. Thus, for example, broken lines with a unit mark-space ratio would be drawn following a command of the form

```
CALL BROKEN(N, N, N, N)
```

where N is a positive INTEGER, and chain dotted lines would follow the command

```
CALL BROKEN(4, 1, 1, 1)
```

The broken line facility will not be available for use with the on-line C.R.T. display.

Reversion to full lines is obtained by means of the command

```
CALL FULL
```

Note that due caution should be exercised when using the DENSITY and/or BROKEN facilities, since either can drastically increase the amount of computer and plotter time necessary for the job.

A certain amount of checking is done by the graphical output system and, in the event of inconsistency on the part of the commands given by the user, diagnostic information may be printed on the line-printer. The statement

```
CALL DGNSTC(ONOFF)
```

may be used for the purpose of suppressing or re-establishing this diagnostic output. The output is suppressed if the INTEGER ONOFF is set to zero and is re-established if it is set not equal to zero.

In the normal course of events every frame of graphical output will contain some standard annotation along the lower edge, which will not, however, conflict with the user's

output (i.e. it will be outside LIMITS). This will record the frame number (and corresponding frame numbers on all the other devices), and the date and time at which the decoding of FORTRAN instructions, into plotter commands for that frame, was started. This standard frame annotation can be suppressed or re-established by means of the command

CALL ANNOTE(ONOFF)

where, as before, suppression occurs if ONOFF = 0 and re-establishment if ONOFF ≠ 0, ONOFF being a literal or symbolic INTEGER. Note that frame counting continues even though the standard annotation may be temporarily suppressed.

The instruction

CALL GRATIC

will cause a graticule to be drawn, spanning the mapping region in the currently selected space. The positions of the component lines, which are equispaced in both the X and Y directions, will be chosen so as to emphasise convenient values along the co-ordinate axes, as a human operator might do naturally.

The instruction

CALL SCALES

will cause the numerical annotation of convenient positions along the border of the mapping region in the currently selected space. These positions will agree with those chosen by GRATIC, so that the two commands may be used in conjunction, if required.

The command

CALL AXES

will cause rectangular Cartesian co-ordinate axes to be drawn, spanning the mapping region. These axes will be annotated at points corresponding to the annotation of SCALES.

Three commands, similar to SCALES, GRATIC and AXIS, which allow the user to specify the instruments DX and DY at which annotation is to be performed, are also provided. The command

CALL SCALSI(DX,DY)

annotates scales round the mapping region at intervals DX and DY in the currently specified units applying to the currently selected space. These intervals are measured from the origin of the coordinates, which will therefore always be an annotated point if it is within the mapping region. If more than 16 scale annotations are implied along either edge, the given interval is ignored and is replaced by a standard interval. The command

CALL AXESSI(DX,DY)

annotates scales along the axes at intervals DX and DY in the currently specified units applying to the currently selected space. These intervals are measured from the origin of the co-ordinates. If one axis does not cut the mapping region that scale is ignored; however, if neither axis does not cut the mapping region, the subroutine SCALSI is invoked. If more than 16 scale annotations are implied on an edge, the given interval is ignored and is replaced by a standard interval. The command

CALL GRATSI(DX,DY)

draws a graticule over the mapping region at intervals DX and DY in the currently specified units applying to the currently selected space. These intervals are measured from the origin of the co-ordinates, which will therefore always be a point of intersection if it is within the mapping region. Not more than 100 lines may be drawn in either direction.

A set of contours may be drawn by means of the single command

```
CALL CONTRA(A, JXS, JXF, NX, JYS, JYF, NY, H, M, N)
```

where A(NX, NY) is a rectangular array satisfying $NX \leq 96$ and $NY \leq 96$, and the surface in which contours are required to be drawn is regarded as being constructed by spanning the mapping region in the currently selected space with a sub-array of A, namely ((A(I, J), I = JXS, JXF), J = JYS, JYF), and defining the spot height of the surface at each grid point (I, J) as A(I, J).

The indices I and J are regarded as increasing in the positive x directions, and the positive y-direction respectively. Thus, the spot height A(JXS, JYS) will lie at the bottom left hand corner of the mapping region, and A(JXF, JYF) will lie at the top right hand corner.

H(N) is a vector of required contour heights; only contours of heights (H(K), K = M, N) will be plotted. Every branch of each contour will be numbered by the integer indicating its position in the vector H. Of course, arrays of dimensions exceeding 96 in either direction may be contoured by segmenting them and separately contouring the segments onto juxtaposed regions in plotter space.

For reasons of core store economy, the contouring facility will not be automatically available during interactive use of the on-line C.R.T. display. However, the appropriate routine could be explicitly loaded by the user, provided core space is available, and thereafter called upon as per normal GHOST practice.

The instruction

```
CALL PTPLOT(X,Y,M,N,NC)
```

will cause character number NC, in the currently selected character set, to be plotted at positions ((X(J), Y(J)), J = M, N), provided $0 \leq NC \leq 63$. However, if NC is set equal to -1 the points are connected by an open-ended polygonal arc.

For reasons of economy, users are normally restricted to a maximum of 20 frames of B-L 120 hardcopy output per run. However, if more than 20 are required they can be specially requested by means of the HCSTOP command. For example, the instruction

```
CALL HCSTOP(30)
```

will increase the limit to 30 frames of hardcopy. This command should be used with discretion. It is inadvisable to request a large number of hardcopy frames as a matter of course at the start of every run - programs have been known to fail in such a way as to produce thousands of blank hardcopy frames, causing considerable expense and inconvenience. At the time of writing there is no software limit to the production of frames on media other than B-L 120 hardcopy. Hardcopy output will be suppressed if this command is used with a zero argument.

As an aid to the debugging of programs using GHOST commands the GARGS command has been introduced. The instruction

```
CALL GARGS(1)
```

causes automatic printing of arguments of those GHOST routines which are unlikely to appear inside long loops. For example, the arguments of REGION and LIMITS will be printed, but those of POINT and JOIN will not. This GHOST argument printing can be suppressed at any time by means of the instruction `CALL GARGS(0)`.

The last executable instruction in any CHAIN of the user's program which contains GHOST commands should be

CALL GREND

This routine ensures that any graphical information left in storage buffers is properly dealt with, and then returns to the calling routine. If this instruction is not executed by the user's job the result will probably be the loss of the last part of the required graphical output.

8. PROCEDURAL ARRANGEMENTS

In order to minimise demands on core storage during execution of the user's program, and to simplify the control cards which the user must supply in order to activate the graphical output system, the following procedure is adopted. During execution of the user's program, the graphical output commands merely write their arguments onto KDF 9 tape 98. At some convenient later time tape 98, possibly containing graphical output from several users' jobs, is processed by a special program in order to produce instructions suitable for controlling the various plotters. Thus, normally the user need only include the tape IOD card

*IOD TAPE 98/CARRY 9/CARRY 9

in order to make use of the graphical output system.

If, however, the STORE and RETRVE facilities are to be used IOD cards appropriate to any other storage tapes nominated must also be included.

In order to provide as rapid a turn-round as possible for daytime graphical output, two different GHOST processing modes have been introduced. During working hours the B-L 120 will normally be loaded with the 35 mm film camera, and with 'instant' hardcopy. If output on either or both of these media is required the user should ensure that the GHOST command first executed by his program is

CALL QUICK

This output will then be returned as soon as the machine operators can manage.

On the other hand, if either 16 mm film or permanent hardcopy output is required, the GHOST command first executed by the program should be

CALL SLOW

In this case the KDF 9 GHOST processing run will normally take place during the evening, and the graphical output should be available by next morning.

It may be that, for example, a user will require both permanent hardcopy and 35 mm film from the same job. If so, his first two GHOST commands should be

CALL QUICK

CALL SLOW

in either order. In that case the 35 mm film will result from the rapid, daytime processing run, and the permanent hardcopy from the later evening run. In fact, unless he explicitly requests otherwise, the user will then receive both types of film and both types of hardcopy.

If the first executed GHOST command is neither

CALL QUICK

nor CALL SLOW

the run will be treated as though the latter were required.

If, after viewing the results from a QUICK processing run, a user wishes to amend his pre-programmed requirements for the subsequent SLOW run he should, as soon as possible, inform the machine operators who may then be able to make appropriate modifications to the evening processing run.

Of course, the QUICK and SLOW commands will not apply to the on-line C.R.T. display.

Whenever a job containing the RETRVE command is run a separate I.O.D. card, appropriate to the storage tape, should be attached to the outside of the job deck. The operators will then insert this I.O.D. card into the GHOST processing job. Naturally, the user must indicate his requirements in the comments section of his job slip, bearing in mind that this will be a relatively unusual thing for the operators.

If a user intends to store the complete graphical output, rather than just selected portions, he may find it more convenient to divert it all to a private storage tape. This is done by omitting the standard I.O.D. card (above) and replacing it by

*IOD TAPE 98/COMMON/SAVE

The output from this job will contain the tape number *n* of the COMMON tape used. Retrieval of the graphs can be affected by the operators at a later time, using the IOD card.

*IOD TAPE 99/*n*/SAVE

which must, of course, be supplied to them by the user. Experienced users will appreciate the legitimate variations of this procedure; users with less experience should seek the advice of a member of the computing section before making use of any of the graphical output storage facilities.

For use of the on-line CRT display a special operating mode exists, which necessitates the loading of a streamlined version of the GHOST software into the user's job area of the KDF 9's core store. This occupies about 4000 words, leaving about 4000 more for the rest of the user's program. Unfortunately, this restriction of core store is a necessary penalty, if the main GHOST facilities and the full on-line interactive features of the 338 display are to be preserved. Even so, certain GHOST facilities, notably broken lines and choice of character sets, have had to be curtailed. These restrictions are dealt with elsewhere in the text.

It is intended that another report, describing use of the on-line CRT in detail, will be issued in due course. However, as far as graphical output is concerned, its use may be summarised as follows:

- (a) The user is able to initiate computation and graphical output from his on-line console;
- (b) At the time of publication, such output appears on the C.R.T. screen, during the course of computation. However, it is hoped that, in the future, the graphical output will be stored on the magnetic disc in the form of a display file, which can then be accessed by the on-line user and displayed and modified, as required, on the 338 display.

Having observed this graphical output on the CRT screen, the on-line user may modify his program as necessary by means of the standard COTAN system (Lang, 1967). Of course, no intermediate storage on tape 98 will take place.

Having developed his program by interactive use of the on-line console and the 338 display, the user may, if he desires, make the on-line modifications necessary for reverting to the standard GHOST software and output channels, and then initiate remote job entry into the background job stream. His results will then eventually appear on hardcopy and/or microfilm and/or incremental pen plotter paper. Alternatively, it should be possible to photograph the on-line C.R.T. screen in order to obtain permanent records.

The streamlined GHOST software necessary for interactive use of the on-line C.R.T. display is obtained by inserting the on-line file equivalent of a

*SUBSTITUTE OLDISP

card immediately following the *CHAIN card of any chain in which graphical output is required.

When completing the KDF 9 job slip the estimated number of frames of graphical output on each separate channel (distinguishing between "instant" and "permanent" hardcopy) should be indicated.

Users are not encouraged to introduce any non-GHOST graphical output routines into new programs, since this can interfere with standard operating procedure. However, any existing programs which use such (older) routines can, and should, be run under general GHOST control. This is achieved by removing the existing control card

*IOD TAPE 101/IBM-SC/SAVE

and replacing it by the card

*IOD TAPE 98/CARRY 9/CARRY 9

In addition, a card

*SUBSTITUTE GOSBQ

must be inserted immediately following the *CHAIN card of any chain in which graphical output is effected by any non-GHOST routines. This substitution can occasionally cause core store difficulty, but this can usually be resolved satisfactorily.

It will be helpful (though not essential) both to user and operators, if the first executable statements in the main routine of such a program be as follows:

CALL QUICK (or SLOW, or both)

CALL HRDCPY (1)

CALL HRDCPY (0)

This will automatically provide the operators with "housekeeping" information about the graphs.

Under no circumstances should a user effect graphical output from a non-GHOST routine, independently from GHOST control as outlined above, without making sure that the machine operators are fully aware that non-standard operation is required.

9. ACKNOWLEDGEMENTS

During the course of development of the graphical output software a number of people, both permanent staff and visitors to the Laboratory, have been involved. In particular B. Brunning, Mrs G.M. Leslie, J. McDowell and A. Sykes have all made invaluable contributions to the project.

The author is grateful to the National Physical Laboratory for permission to use the information defining the software characters (1 to 6) designed by Miss M. Laws.

10. REFERENCES

1. LARKIN, F.M. 'A Graphical Output Language and its Implementation'. Culham Laboratory, June, 1967. CLM -P 139.
2. HEAP, B.R. and NOTT, C., Private Communication.
3. Users' Manual for the Culham Laboratory KDF 9 Computer. Culham Laboratory, Computing and Applied Mathematics Group. 1 April, 1965.
4. A Users' Guide to COTAN. Culham Laboratory, Computing and Applied Mathematics Group. August, 1967.

APPENDIX A

SUMMARY OF STANDARD GHOST FACILITIES

Operations available in the UKAEA Culham GHOST implementation of GHOUL

(i) Control of Graphical Output Hardware

Operation	FORTRAN Subroutine	Page
Switch 35 mm camera on or off; select if ONOFF = 1	FILM35(ONOFF) *	3
Switch 16 mm camera on or off; select if ONOFF = 1	FILM16(ONOFF) *	3
Switch Hardcopy camera on or off; select if ONOFF = 1	HRDCPY(ONOFF) *	3
Switch Incremental Plotter on or off; select if ONOFF = 1	INCPLT(ONOFF) *	3
Switch On-line Display on or off; select if ONOFF = 1	DISPLY(ONOFF) / *	3
Advance frame on 35 mm camera	FRAM35 *	3
Advance frame on 16 mm camera	FRAM16 *	3
Advance frame on Hardcopy camera	FRAMHC *	3
Advance frame on Incremental Plotter	FRAMIP / *	3
Advance frame on On-line Display	FRAMDP *	3
Advance frame on all "on" devices	FRAME	3
Select red pen on Incremental Plotter	REDPEN } mutually	4
Select black pen on Incremental Plotter	BLKPEN } exclusive	4
Store subsequent graphical output commands on tape NTAPE	STORE(NTAPE) *	3
Suspend storage on tape NTAPE	SUSPND(NTAPE) *	3
Cancel storage on tape NTAPE	CANCEL(NTAPE) *	3
Retrieve frames numbered M, M+1, ... N, from storage tape NTAPE	RETRVE(NTAPE,M,N) *	4
Repeat preceeding BL-120 frame NTIMES times	REPEAT(NTIMES) *	4

* Indicates that the operation refers to specific hardware or operational mode. Other implementations might alter or delete these operations, since they are not strictly part of the GHOUL graphical output language.

/ These commands will only be obeyed if the on-line-display version of the GHOST software is loaded (see section 8).

(ii) Mapping From Mathematical Space to Plotter Space

Operation	FORTRAN Subroutine	Page
Select mathematical space	MSPACE	6
Define mapping region	REGION(XMIN, XMAX, YMIN, YMAX)	7
Define limiting rectangle	LIMITS(XMIN, XMAX, YMIN, YMAX)	7
Define plotter space units to be inches	INCHES	7
Define plotter space units to be centimetres	CMS	7
Define plotter space units to be millimetres	MMS	7

(iii) Straight Line and Curve Drawing

Operation	FORTRAN Subroutine	Page
Lower plotting pen onto point (X,Y)	POINT (X,Y)	9
Join current plotting point to point (X,Y) by a straight line	JOIN (X,Y)	9
Draw straight line with vector components (DELTA X, DELTA Y) from current point	LINE(DELTA X, DELTA Y)	9
Draw smooth, open-ended, rotationally invariant curve through given points	CURVEO(XV, YV, M, N)	9
Draw smooth, closed, rotationally invariant curve through given points	CURVEC(XV, M, N,)	9
Draw non-periodic graph through given points	GRAPHN(XV, YV, M, N,)	10
Draw periodic graph through given points	GRAPHP(XV, YV, M, N, PERIOD)	10
Draw graph of given, EXTERNAL function	GRAPHF(FUNXN)	10

(iv) Output of Characters and Point Plotting Symbols

Operation	FORTRAN Subroutines	Page
Select character set NOSET	CTRSET(NOSET), CRSET(NOSET)	11
Define character size	CRSIZE(HEIGHT)	12
Define character typing position in (x, y) co-ordinate system	POSITN(X, Y)	13
Define character typing position in (space, line) co-ordinate system	PLACE(NSPACE, NLINE)	14
Space NSPACE algebraic character positions to the right	SPACE(NSPACE)	14
Effect NLINEs algebraic line-feeds	LINEFD (NLINEs)	14
Perform the combined operation "carriage-return, line feed"	CRLNFD	14
Plot numbered character at point (X, Y)	PLOTNC(X, Y, NOCHAR)	13
Type numbered character at current position	TYPENC(NOCHAR)	13
Read first NC characters of string into vector PHRASE	READCS(PHRASE, NC)	13
Plot first NC characters of string Phrase starting at point (X, Y)	PLOTCS(X, Y, Phrase, NC)	✓ 13
Type first NC characters of string Phrase starting at current typing position	TYPECS(Phrase, NC)	✓ 13
Plot number Z in E-format, NDP decimal places, decimal point at point (X, Y)	PLOTNE(X, Y, Z, NDP)	16
Plot number Z in F-format, otherwise as for PLOTNE	PLOTNF(X, Y, Z, NDP)	16
Plot number I in I-format, otherwise as for PLOTNE	PLOTNI(X, Y, I)	16
Type number Z in E-format, with sign position located at current typing position	TYPENE(Z, NDP)	16
Type number Z in F-format, otherwise as for TYPENE	TYPENF(Z, NDP)	16
Type number I in I-format, otherwise as for TYPENE	TYPENI(I)	16
Select or deslect italic characters	ITALIC(ONOFF)	15
Select suffix character plotting mode	SUFFIX	15
Select superfix character plotting mode	SUFFIX	15
Revert one level towards normal character plotting mode	NORMAL	15
Select large hardware characters on B-L120	LARGE	* 15
Select small hardware characters on B-L120	SMALL	* 15
Select light plotting density on B-L120	LIGHT	* 15
Select heavy plotting density on B-L120	HEAVY	* 15
Select standard hardware character orientation on B-L120	STNDRD	* 15
Select sideways hardware character orientation on B-L120	SIDWYS	* 15

✓ Note that "Phrase" may be either a vector or a literal character string.

(v) Miscellaneous Commands

Operation	FORTRAN Subroutines	Page
Draw rectangular box of prescribed dimensions	BOX(XMIN, XMAX, YMIN, YMAX)	16
Rotate subsequent graphical output through ANGLE, about the point (XC, YC)	ROTATE (XC, YC, ANGLE)	16
Select line density of subsequent output	DENSTY(LTORDK)	17
Nominate subsequent lines to be drawn in a broken style	BROKEN(N1, N2, N3, N4)	17
Revert to full lines	FULL	17
Select or deselect diagnostic printing	DGNSTC(ONOFF)	17
Select or deselect standard frame annotation	ANNOTE(ONOFF)	18
Draw graticule spanning mapping region	GRATIC	18
Annotate scales on mapping region	SCALES	18
Draw and annotate co-ordinate axes	AXES	18
As for GRATIC, SCALES and AXES, except that the specified increments, for purposes of annotation, are given as arguments	GRATSI(SPINCX, SPINCY) SCALSI(SPINCX, SPINCY) AXESSI(SPINCX, SPINCY)	18 18 18
Connect an ordered sequence of points by straight lines, or plot characters	PTPLOT(X, Y, M, N, NC)	19
Select, or deselect, informative printing of arguments of certain routines	GARGS(ONOFF)	19
Draw specified contours in surface defined by spot heights	CONTRA(A, JXS, JXF, NX, JYS, JYF, NY, H, M, N)	19
Draw rectangular border around mapping region	BORDER	16
Expand subsequent dimensions, about (XC, YC) by given RATIO	EXPAND(XC, YC, RATIO)	16
Combined operation of EXPAND and ROTATE	EXPROT(XC, YC, RATIO, ANGLE)	16
Stop hardcopy output after NFRAME frames	HCSTOP(NFRAME), GPSTOP(NFRAME)*	19
Request quick processing mode	QUICK	*
Request slow processing mode	SLOW	*
Terminate graphical output (empty storage buffers)	GREND	20

APPENDIX B

The Standard GHOST Character Sets

In this table, and the succeeding ones, the index number of a character whose co-ordinates are (α, β) is given by $10\alpha + \beta$; e.g. the index number of M is $(10 \times 2) + 3 = 23$.

$\alpha \backslash \beta$	0	1	2	3	4	5	6	7	8	9
0	O	1	2	3	4	5	6	7	8	9
1	space	A	B	C	D	E	F	G	H	I
2	J	K	L	M	N	O	P	Q	R	S
3	T	U	V	W	X	Y	Z	.	,	'
4	π	?	~	+	-	*	/	=	()
5	α	β	γ	δ	\pm	'	"	Σ	g	.
6	∂	d	<input type="checkbox"/>	\int						

Character Set 0 : B-L hardware characters

Note that the above characters are only represented symbolically; the small size of the actual hardware characters makes it difficult to reproduce them legibly.

$\alpha \backslash \beta$	0	1	2	3	4	5	6	7	8	9	
0	0	1	2	3	4	5	6	7	8	9	
1		A	B	C	D	E	F	G	H	I	
2		J	K	L	M	N	O	P	Q	R	S
3		T	U	V	W	X	Y	Z	.	'	;
4		:	?	!	+	-	*	/	=	()
5		[]	<	>	↑	'	"	#	\$	&
6		@	%	←	\						

Character Set 1: Contains Hollerith and Teletype characters

$\alpha \backslash \beta$	0	1	2	3	4	5	6	7	8	9	
0	0	1	2	3	4	5	6	7	8	9	
1		a	b	c	d	e	f	g	h	i	
2		j	k	l	m	n	o	p	q	r	s
3		t	u	v	w	x	y	z	.	'	;
4		:	?	!	+	-	*	/	=	()
5		.	.	≤	≥	±	≡	≈	~	√	∧
6		∂	∇	□	∫						

Character Set 2: Lower case Roman characters and miscellaneous mathematical symbols

$\alpha \backslash \beta$	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1		A	B	Γ	Δ	E	Φ	Γ	H	I
2	Θ	K	Λ	M	N	O	Π	Υ	P	Σ
3	T	Y	Ψ	Ω	X	Y	Z	.	/	i
4	:	?	!	+	-	*	/	=	()
5	\Rightarrow	\Leftarrow	U	U	C	C		L	E	V
6	<	>		T						

Character Set 3: Upper case Greek characters and miscellaneous mathematical symbols

$\alpha \backslash \beta$	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1		α	β	γ	δ	ϵ	ϕ	γ	η	ι
2	θ	κ	λ	μ	ν	\omicron	π	ξ	ρ	σ
3	τ	υ	ψ	ω	χ	υ	ζ	.	/	i
4	:	?	!	+	-	*	/	=	()
5	Δ	∇	\square	\diamond	\circ	\boxplus	\boxtimes	\boxdot	\boxminus	\boxtimes
6	\star	\star	\star	\star						

Character Set 4: Lower case Greek characters and point plotting symbols

$\alpha \backslash \beta$	0	1	2	3	4	5	6	7	8	9	
0	0	1	2	3	4	5	6	7	8	9	
1		A	B	C	D	E	F	G	H	I	
2		J	K	L	M	N	O	P	Q	R	S
3		T	U	V	W	X	Y	Z	°	°	°
4	°	?	!	+	-	*	/	=	()	
5	[]	<	>	↑	'	"	10	£	_	
6	→	%									

Character Set 5: Contains Flexowriter case "normal" and KDF 9 line printer characters

$\alpha \backslash \beta$	0	1	2	3	4	5	6	7	8	9	
0	0	1	2	3	4	5	6	7	8	9	
1		a	b	c	d	e	f	g	h	i	
2		j	k	l	m	n	o	p	q	r	s
3		t	u	v	w	x	y	z	°	°	°
4	°	?	!	+	-	*	/	=	()	
5	[]	<	>	↑	x	÷	≠	£	_	
6	→										

Character Set 6: Contains Flexowriter case "shifted" characters

APPENDIX C

Examples of Program and Output

```
C      EXAMPLE OF THE USE OF GHOST.  GRAPHS OF BESSEL FUNCTIONS
C
C      EXTERNAL BESSJO, BESSJ1
C
C      SWITCH ON AND SELECT HARDCOPY.  ADVANCE TO A NEW FRAME.
C      CALL HRDCPY (1)
C      CALL FRAME
C
C      DEFINE UNITS, MAPPING REGION, LIMITING RECTANGLE, CHARACTER
C      SET AND SIZE
C      CALL CMS
C      CALL REGION (2.0,14.0,2.0,12.0)
C      CALL LIMITS (0.0,15.0,0.0,15.0)
C      CALL CTRSET (1)
C      CALL CRSIZE (0.5)
C
C      WRITE TITLE AT TOP OF GRAPH
C      CALL PLACE (10,3)
C      CALL TYPECS (26HGRAPHS OF BESSEL FUNCTIONS,26)
C
C      SELECT MATHEMATICAL SPACE AND DEFINE MAPPING REGION
C      CALL MSPACE
C      CALL REGION (0.0,12.0,-0.5,1.0)
C
C      DRAW THE CURVES OF JO(X) AND J1 (X)
C      CALL GRAPHF (BESSJO)
C      CALL GRAPHF (BESSJ1)
C
C      ANNOTATE THE CURVES, AFTER DEFINING CHARACTER SIZE IN M-SPACE
C      CALL CRSIZE (0.075)
C      CALL PLOTCS (1.0,0.9,1HJ,1)
C      CALL SUFFIX
C      CALL TYPECS (1HO,1)
C      CALL NORMAL
C      CALL PLOTCS (3.3,0.4,1HJ,1)
C      CALL SUFFIX
C      CALL TYPECS (1H1,1)
C      CALL NORMAL
C
C      DRAW BORDER AROUND MAPPING REGION.  DRAW AND ANNOTATE THE AXES
C      CALL BORDER
C      CALL AXES
C
C      TERMINATE GRAPHICAL OUTPUT
C      CALL GREND
C
C      CALL EXIT
C      =====
C
END
```

GRAPHS OF BESSEL FUNCTIONS

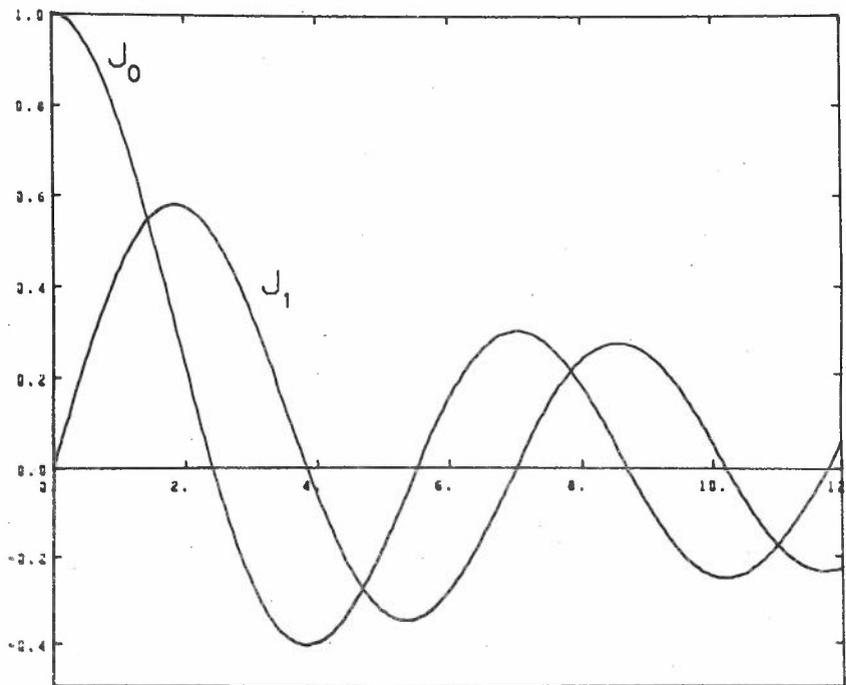


Fig.3
Graphical Output from Example Program