ATLAS
COMPUTER
LABORATORY

ALGOL GROATS MANUAL

(REVISED AUGUST 1969)

SCIENCE
RESEARCH
COUNCIL

CHILTON, DIDCOT, BERKS

G R O A T S

A Graphic Output package for Atlas using the SC 4020

by

F. R. A. Hopgood

(revised August 1969)

CONTENTS

1. INTRODUCTION

The Stromberg Carlson SC 4020 plotter provides the Atlas Laboratory with a very powerful graphical utility. To help the Algol user, a set of basic procedures, called GROATS, has been written to interface the user and the SC 4020. It was originally intended to use the same procedures defined for the Benson-Lehner plotter but this put unnecessary restrictions on the system. The new package has little in common with the old Benson-Lehner package and will mean that existing users will have to rewrite the graphical parts of their programs. The more powerful facilities available in the GROATS package should make this attractive.

The facilities defined in this manual represent about 4 man-months of work and is the basis of a system that can be extended by adding library procedures based on the GROATS package.

The manual starts with several Chapters outlining the facilities available. It is hoped that these Chapters will act as a primer for the system. A more rigorous definition of the package and the SC 4020 is given in Chapters 8 and 9. The user is well advised to read the description of the SC 4020 in Chapter 9 at least once as it will help to put the system in perspective. However, at this stage, we need only concern ourselves with the basic operations of the plotter and a brief description will suffice.

The SC 4020 can produce graphical output on a continuous roll of plain paper (not graph paper). The plotting area is divided up into pages (or frames) which are $7\frac{1}{2}$" square and it is possible to advance the paper automatically by an SC 4020 order, ADVANCE FILM. As the name of this operation implies, it is also possible to produce the output on microfilm but more of this later. The possible plotting positions on the page are not continuous. The square page is divided up into a 1024 x 1024 raster and all plotting positions must be approximated to by the nearest raster position. As these positions are about .007" apart, this should not cause any problems. It is also possible to plot one of the 64 hardware characters at any coordinate of the raster. In addition straight lines can be drawn between any two raster positions.

From these simple operations, it is possible to produce a set of procedures for curve drawing, generating software characters from short vectors, producing a graticule and so on. It is important to remember that so far only the simplest form of operation of the SC 4020 has been described. It is not until Chapter 5 that we need to concern ourselves with the more sophisticated use of the SC 4020. Until then we shall assume that the user is producing graphical output on continuous paper in much the same way as he did on the Benson Lehner plotter. However, as stated earlier, it is possible to produce output on 35 mm or 16 mm microfilm. This means that it is possible to produce cine films on the SC 4020. The GROATS system has, therefore, been designed to aid the user in film making as well as the more conventional graphical output.

Little of this work is original and the author would like to refer the reader to the Acknowledgements and References.

2. THE DEFINITION OF REGIONS

The user will, in general, prefer to define the plotting area in terms of his own units rather than have to convert them into inches of paper, millimetres of microfilm, or raster positions. Almost always he will have some coordinate system and units of his own that he will prefer to use. Consequently, as soon as the graphical output has been identified (see Chapter 6), the user should call:

```
limits (xmin, ymin, xmax, ymax) ;
```

This defines the minimum and maximum values of x and y which coincide with the corners of the $7\frac{1}{2}$" square page in the user's own units. All references to positions on the page will now be defined in terms of this scaling defined by the user. Apart from being more convenient for the user, it also means that the system, as far as the user is concerned, is independent of the type of graphical device being used. The scaling in both the x and y directions is assumed to be linear. For example the x-value halfway between 'xmin' and 'xmax' will be equivalent to the centre point of the page.

To draw the function $y = \sin(x)$ for $x = 0$ to 2π so that it covers the complete plotting area, we could write:

```
limits  (0, -1, 2*pi, 1) ;
for     x := 0 step dx until 2*pi - dx do                (2.1)
vector  (x, sin(x), x + dx, sin(x + dx) ) ;
```

The procedure 'vector(x_1, y_1, x_2, y_2)' is assumed to draw a straight line between the points (x_1, y_1) and (x_2, y_2). If 'dx' is sufficiently small then the graph produced will approximate to the smooth sine curve. By changing the call of the procedure 'limits' in the example (2.1) to:

```
limits  (-pi, -2, 3*pi, 2) ;                             (2.2)
```

the graph would have been drawn within a $3\frac{3}{4}$" square area in the centre of the page. The procedure 'limits' may be called many times so that the user can have several different sets of units if he so desires. This constant changing of units could be inconvenient and the GROATS system is defined so that the plotting area can be divided into separate regions each having its own coordinate system.

2.1 SUB-REGIONS

Normally, apart from the desired graph, additional information such as titles will be on the plotting area. Although the user will appreciate being able to plot points in terms of his own units, it would be unreasonable to expect the user to define the position of his titles in the same units. It is therefore quite useful to be able to define sub-regions of the total plotting area with their own particular scaling.

A sub-region is defined by:

```
region(xmn, ymn, xmx, ymx, I) ;
```

This defines sub-region I as that part of the currently selected region which has minimum and maximum values (xmn,ymn) and (xmx,ymx). Initially the region selected is the whole plotting area and this is region 0. The scaling for this region is defined by the initial call of 'limits'. The arguments of the procedure 'region' are in the units of the selected region. For example:

```
          limits  (0, 0, 3, 3);
          region  (1, 1, 2, 2, 1);
```

defines the units to be used for the whole plotting area. The sub-region 1 is then defined as the $2\frac{1}{2}$" square in the centre of the plotting area. This is shown in Fig. 2.1.

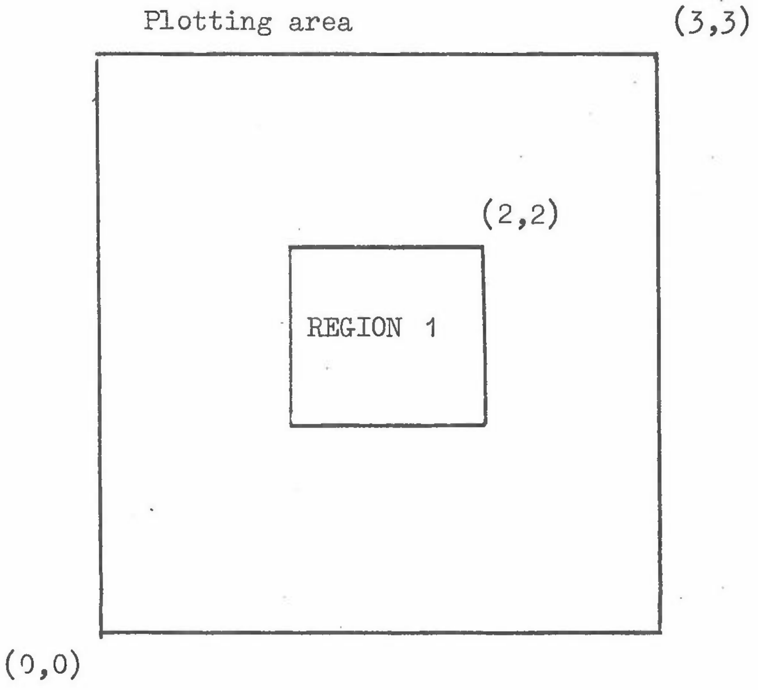


Fig. 2.1

Once more than one region has been defined, it is necessary to define which region is being referred to. Initially region 0 is selected. A sub-region I can be selected by:

```
          select region (I);
```

All subsequent commands now refer to region I and not region 0 which was selected originally. However no units and scaling have been defined for region I and, before generating any graphical output, this must be defined by a call of procedure 'limits'. The maximum and minimum values defined by the call of 'limits' always refer to the currently selected region. For example, the 'sine' curve plotted in the centre of the plotting area in example (2.2) could have been produced by:

```
limits  (0, 0, 3, 3);
region  (1, 1, 2, 2, 1);
select region (1);                                      (2.3)
limits  (0, -1, 2*pi, 1);
for  x := 0 step dx until 2*pi - dx do
vector (x, sin(x), x + dx, sin(x + dx) );
```

Region 1 is defined as in Fig. 2.1. The region is then selected and the bottom left-hand corner is defined as the point (0, -1) and the top right-hand corner as $(2\pi, 1)$. The sine curve is then drawn across region 1 as shown in Fig. 2.2.

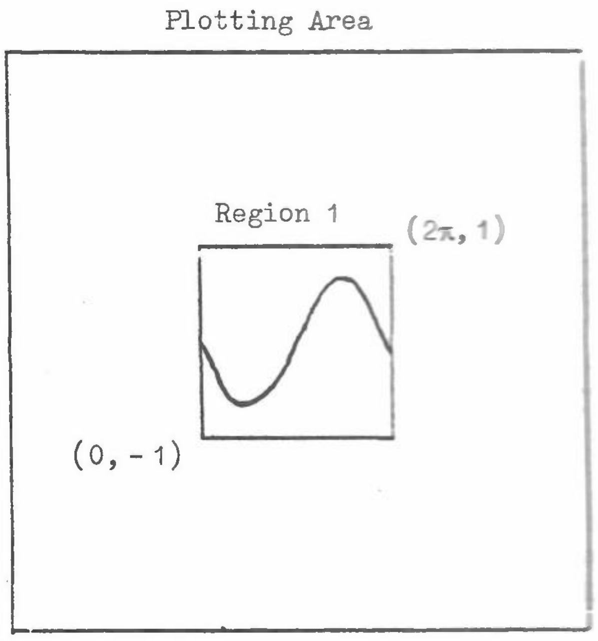


Fig. 2.2

In this example not too much has been gained by the definition of region 1. However, if four separate graphs were required in the four quarters of the plotting area and each required a different set of units then this would best be done by defining four separate sub-regions rather than redefining the units to be used for the complete region.

The procedures 'limits' and 'region' may be called more than once for each sub-region. This greatly increases the flexibility of the system. For example, the same graph can be plotted at different positions on the plotting area by redefining the position of the sub-region by calling 'region' again. Also different magnitudes of the same graph in the same region can be produced by redefining the scaling using 'limits'. These facilities, which are useful in normal graphical work, become essential when making a film. The redefinition of the position of a sub-region over several frames of film can be used to simulate camera movement or alternatively the movement of objects in the region. Similarly by redefinition of the scaling by calls of 'limits' it is possible to simulate camera zoom.

It is important to remember that all GROATS procedure will have their positional arguments defined in terms of the units of the currently selected region and a conversion is made automatically to give the value in terms of the plotting area.

Additional points of interest which might be overlooked are:-

1. Regions can be defined which may be partially or completely outside the plotting area. This can be quite useful when producing films. Gradually moving the position of a region until it appears on the plotting area simulates the movement of a scene onto the visible area of a film.

2. Regions can be redefined in terms of their own units.

3. Sub-regions can be defined in terms of units of other sub-regions. It will probably be usual to define the sub-regions in terms of the main region but in some recursive situation it may be quite useful to have a nested set of region definitions.

The terms 'region' and 'sub-region' have been used rather loosely in the above section. All regions, other than region 0, are thought of as sub-regions. However once more than one region or sub-region has been defined they all have equal standing so that the terms then become interchangeable. The superiority of region 0 is only that initially it is defined as the whole plotting area and that the initial call of 'limits' refers to it. Currently the allowed regions are numbered 0 to 10.

2.2 WINDOWS

A region can be thought of as defining a window through which part of an infinite scene can be viewed. All regions are initially defined so that they act as windows which stop the appearance of any information plotted outside the boundary of the currently selected region. In the example (2.1), the sine curve was only drawn between 0 and 2π so that the limits of the sine curve coincided with the boundary of the region. If instead we had:

```
limits  (0, -1, 2*pi, 1);
for     x := -50*pi step dx until 50*pi do                (2.4)
vector  (x, sin(x), x+dx, sin(x+dx) );
```

then many periods of the sine curve would attempt to be drawn. However, as the selected region is a window containing the viewable plotting area within the boundaries of the region, the curve actually plotted in (2.1) and (2.4) would be the same. This automatic 'scissoring' can often be very helpful to the user as it saves him from worrying too much about what happens at the edges of the region. This is taken care of by the system.

A region need not always be defined as 'containing' the graphical output. There is a second mode of operation where any drawing on the plotting area will be displayed even if it is outside the boundaries of the selected region. This is achieved by calling the procedure:

```
                extend;
```

From this point onwards the currently selected region's boundary and limits act only as a means for defining the units to be used. For example:

```
        limits  (0, 0, 3, 3);
        region  (1, 1, 2, 2, 1);
        select region (1);                                       (2.5)
        limits  (0, 1, 2*pi, 1);
        for  x  :=  -2*pi  step  dx  until  4*pi-dx  do
        vector  (x, sin(x), x+dx, sin(x+dx) );
```

would define region 1 in the centre of the plotting area and, as the region 1 is initially defined as contained, only the middle cycle of the sine curve (that is the one inside the boundaries of region 1) would be displayed. This is shown in Fig. 2.2. However, if we had:

```
        limits  (0, 0, 3, 3);
        region  (1, 1, 2, 2, 1);
                                                                 (2.6)
        select region (1);
        limits (0, -1,  2*pi, 1);

        extend;

        for  x  :=  -2*pi  step  dx  until  4*pi-dx  do
        vector  (x, sin(x), x+dx, sin(x+dx) );
```

then all three cycles would be drawn stretching from one side of the plotting region to the other. A region, which has been defined as 'extended', can later be redefined as a containing window by calling:

```
        contain;
```

when the region is selected.

2.3 PASSIVE REGION

In example (2.6) there does not appear to be any mechanism for stopping a user attempting to plot information off the allowable SC 4020 plotting area. For example:

```
        limits  (0, 0, 3, 3);
        region  (1, 1, 2, 2, 1) ;
        select region (1);
        limits  (0, -1,  2*pi, 1);                               (2.7)
        extend;
        for  x  := -50*pi  step  dx  until  50*pi  do
        vector  (x, sin(x), x+dx, sin(x+dx) );
```

This program looks as though it will attempt to draw cycles of the sine curve several feet away from the 7½" square plotting area of the SC 4020 hardcopy! There must be some mechanism ensuring that, in extended mode, a user does not plot outside the bounds of the SC 4020 plotting area. This could have been done by having the GROATS package insert a constant test just before plotting. Instead a slightly more flexible approach was adopted. In the GROATS system, one region is always defined as the <u>passive</u> region. Initially region 0 is defined as the passive region. <u>The passive region always acts as a window which contains all plotting within its boundary independent of which region is selected.</u> Region 0 is initially passive and, as its boundary is initialised to coincide with the boundary of the SC 4020 plotting area, it will always ensure that no illegal plotting takes place. There is no reason why a different region should not be defined as the passive region. However, under no circumstances should a larger region be defined as the passive region. Unless familiar with the system, it is probably wise not to alter the region defined as passive or its boundaries. A region smaller than the complete SC 4020 plotting area can be defined as the passive region. This would ensure that all plotting was contained within a smaller region than the 7½" square SC 4020 hardcopy page. A region may be defined as the passive region by calling:

```
          passive;
```

which defines the currently selected region as the passive region. In example (2.7), the original setting of region 0 as the passive region ensures that only the three cycles of the sine curve that appear on the SC 4020 plotting area will be plotted. However if, for example, we had:

```
          limits  (0, 0, 3, 3);
          region  (0.5, 1, 2.5, 2, 2);
          select region (2);
          passive;
          select region (0);
          region  (1, 1, 2, 2, 1)
          select region (1);
          limits  (0, -1, 2*pi, 1);
          extend;
          for  x  :=  -50*pi  step  dx  until  50*pi  do
          vector  (x, sin(x), x+dx, sin(x+dx) );
```

then, although the extend mode of region 1 would normally allow the three cycles of the sine curve to be displayed, the new passive region 2 restricts the curve plotted to the two cycles between $x = -\pi$ and 3π.

2.4 REFLECTION, ROTATION, and EXPANSION of REGIONS

So far it has been assumed that the mapping of points and lines defined in a selected region onto the plotting area is linear and is done in a straightforward manner. That is, if the limits of the plotting area and sub-region are defined by:

```
          limits  (XMIN,YMIN,XMAX,YMAX);
          region  (XMN,YMN,XMX,YMX,1);
          select region (1);
          limits  (xmin,ymin,xmax,ymax);
```

then the point (x,y) in units of region 1 is equivalent to the point (X,Y) on the plotting area in units of region 0 where:

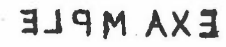
$$X = XMN + \frac{(x - xmin)}{(xmax - xmin)} (XMX - XMN).$$

$$Y = YMN + \frac{(y - ymin)}{(ymax - ymin)} (YMX - YMN)$$

That is the standard conversion that is done and it will produce the normal results expected. If, for example, in region 1 a procedure is called which produces the letters EXAMPLE by vectors then the word will appear in this form on the plotting area. If instead the declaration of region 1 had been:

```
          region  (XMX,YMN,XMN,YMX,1);
```

then



would have been plotted. Similarly by interchanging the maximum and minimum y values in the region declaration, the characters could be turned upside down.

It is also possible to allow a rotation or expansion of the material defined in the selected region before doing the conversion to a position on the plotting area. For example, before the conversion given above, the point (x,y) could first be mapped into the point (x',y') where:

$$x' = xorig + e \times((x - xorig) \cos\alpha - (y - yorig) \sin\alpha)$$

$$y' = yorig + e \times((y - yorig) \sin\alpha + (y - yorig) \cos\alpha)$$

The point (xorig,yorig) is defined as the centre of rotation and expansion. The conversion, shown above, consists of a rotation of α radians anti-clockwise about this centre together with an expansion by a factor 'e' about the same centre. The GROATS package allows either expansion or rotation of this form to be specified. The only limitation is that, if both are defined, then the same centre of rotation must be used for both the expansion and rotation. Rotation is achieved by the call:

```
          rotate  (xorig,yorig,alpha);
```

while expansion is achieved by:

```
          expand  (xorig,yorig,e);
```

It is important to remember that basically all regions are defined in terms of the plotting area even though they may have been defined in terms of other regions. This means that, in attempting to define the position of the point (x,y) on the plotting area, only the rotation and expansion defined for the selected region will be applied. The rotations and expansions of other regions, in terms of which the selected region may have been defined, have no effect on the conversion. Similarly, if a region is defined in terms of a rotated region, its definition will be as if no rotation had been defined on this region. This ensures that all regions are defined as rectangular with boundaries parallel to the x and y axes. The rotation and expansion are local to the plotting of points in the selected region.

In subsequent Chapters, procedures will be described having x and y values as arguments. These arguments are always in the units of the currently selected region.

2.5 SHIELDING

In section 2.2 regions were defined as windows which contained the plotting that could be seen. An alternative is to define a region as a shield which only allows plotting outside the bounds of the selected region. The currently selected region can be defined as a shield by calling:-

```
shield;
```

The region can be redefined as a window by calling:-

```
window;
```

Together with the contain/extend switch this allows the four possibilities:-

window	contain	Only plotting inside region displayed
window	extend	All plotting displayed
shield	extend	Only plotting outside region displayed
shield	contain	Only plotting on the region boundary is displayed

The 'shield' facility is particularly useful for doing wipes starting from the centre of the picture in cine films.

2.6 ACTIVE REGION

One drawback with the local shielding is that the shielding or masking will only effect plotting in the selected region. Consequently if plotting is being done in several different regions it is not possible to define a mask that will effect plotting in all the regions. The active region is similar to the passive region in that its action is global to all regions. The passive region only has meaning if it is a window and contains all plotting within it. Conversely the active region only has effect when it is also defined as a shield. In this case only plotting outside the active region will be displayed. The currently selected region is defined as the active region by calling

```
active;
```

For example

```
          limits (0, 0, 10, 10);
          region (1, 1, 9, 9, 1);
          region (3, 3, 7, 7, 2);
          select region (1);
          passive; window;
          select region (2);
          active; shield;
          select region (0);
          scene;
```

will allow only that part of the scene defined between the limits of regions 1 and 2 to be displayed.

3. LINE DRAWING

All line drawing in the GROATS package is eventually accomplished by calling the procedure:

```
vector ($x_1$,$y_1$,$x_2$,$y_2$);
```

which defines a line joining the points (x_1,y_1) and (x_2,y_2). If the selected region is 'contained' (see Section 2.2) then only the part of the line appearing inside the selected region is drawn. If the selected region is 'extended' then the complete line is drawn assuming that it is within the boundaries of the passive region (see Section 2.3). It is important to remember that all output on the SC 4020, apart from the generation of the hardware characters, is achieved by line drawing so that this section applies equally well to curve drawing and software character generation as well as line drawing.

The standard settings controlling the procedure 'vector' will mean that the line is drawn once unbroken. Unless the line is horizontal or vertical, this will be achieved by a set of calls of the SC 4020 DRAW VECTOR operation (see Section 9.8.3). Horizontal and vertical lines are produced more efficiently on the SC 4020 by the special axes generating instructions and these will be used wherever possible. The SC 4020 at the Atlas Laboratory has been set up so that the axes generating operations produce lines slightly fainter than the lines produced by the DRAW VECTOR operation. This may be useful when producing vertical and horizontal grid lines but, in standard line drawing, it may cause uneveness in plotting intensity. To force all lines in the selected region to be produced by DRAW VECTOR operations, the procedure:

```
use vectors;
```

should be called. This will increase the number of instructions generated for the SC 4020 (for a line across the complete plotting area, 16 instructions would be required instead of 1) and should therefore be used sparingly. To return the selected region to using the axes generating operations, call:

```
use axes;
```

Darker lines on the SC 4020 can be obtained by repeating the drawing of the line several times. The standard setting is to draw the line once only. By calling the procedure:

```
dark (I);
```

all subsequent line drawing in the selected region will consist of each line drawn I times. Lines tend to become too dark if I is greater than 5. If I = 0 then the line will not be drawn at all.

Thicker lines can be produced by calling the procedure:

```
thick (I);
```

After calling this procedure, all lines drawn in the selected region will consist of the original line together with I - 1 lines on either side of the original line. These additional lines are drawn parallel to the original line and touching it. For example, if I = 2, 3 lines will be drawn touching each other. If I is less than 1 no lines will be drawn. This facility can be quite useful in film work. Single lines tend to appear rather thin on microfilm and it is desirable to have I set to 2 or more.

Broken lines can be produced by calling the procedure:

```
broken (I);
```

The standard setting is I = 0 which will draw the line unbroken. The parameter I can be set between 1 and 63. It causes the line to be broken up into visible and invisible segments approximately .007" x I in length on the hard copy output. This distance does vary depending on the angle of the line with the horizontal. This is not ideal as far as the user is concerned but it does help to produce SC 4020 output as efficiently as possible. Short broken lines will cause the SC 4020 to work very inefficiently and they should be used sparingly.

Small reproductions of large drawings having considerable detail can become indistinct due to a large number of lines being truncated to a vector of zero length. Such vectors appear on the SC 4020 as a dot. The drawing of these dots can be inhibited by calling:

```
zero invisible;
```

All zero length lines will not be drawn in the selected region once this procedure has been called. The drawing of zero length lines can be reinstated by calling:

```
zero seen;
```

It is hoped that this suppression of zero length lines may be useful when zooming in on a complex scene. Features will remain invisible until they become at least one raster unit in length.

When constructing line drawings, it quite often happens that several lines are required in succession with the end-point of one line being the starting point of the next. To avoid unnecessary repetition of the common point, the user may call:

vector (x_1, y_1, x_2, y_2);

to point (x_3, y_3);

instead of:-

vector (x_1, y_1, x_2, y_2);

vector (x_2, y_2, x_3, y_3);

The procedure 'to point' takes its initial point from the end-point of the previous line drawn. The end-point saved after each line is drawn is not local to a particular region. Consequently it is wise to always start a set of line drawing commands by a call of 'vector'.

When drawing curves, it is sometimes useful to shade the area above or below the curve in order to make the difference between the two regions stand out. It is possible to do this in the GROATS package by calling the procedure:

```
shadow (I);
```

before drawing the curve. The area which is shaded depends on the value of I.

I = 1.	The area to the left of the curve is shaded
I = 2.	The area to the right of the curve is shaded
I = 3.	The area below the curve is shaded
I = 4.	The area above the curve is shaded

The shading is produced by drawing an axis, originating from the start of each vector drawn, to the edge of the selected region.

To delimit the plotting areas being produced, it is useful to surround them with a border. Two procedures are provided for this purpose. A call of:

```
border;
```

will draw a straight line border around the edge of the selected region. A call of:

```
box (xmin,ymin,xmax,ymax);
```

will draw a rectangle where the minimum and maximum points are (xmin,ymin) and (xmax,ymax) respectively.

4. CHARACTER TYPING AND PLOTTING

4.1 THE GROATS CHARACTER SET

There is a set of 64 hardware characters available on the SC 4020 (see Appendix 1) which may be plotted at any raster position of the plotting area. In addition to plotting a character at a specified position, it is possible to type strings of characters on the plotting area. This could, of course, be done by several calls of the plotting operation. However, as it is likely to be required frequently, the SC 4020 has special hardware for simulating the action of a typewriter. Once in this typewriting mode, the hardware will automatically increment the typing position after each character is typed so that there is the correct spacing between it and the following character. In addition, the hardware automatically moves to a newline when the right hand margin is reached and the top of the page when the last line has been typed. This is the most efficient method of generating textual information on the SC 4020 but does not fit easily into the GROATS system. For example, the GROATS system is designed around the concept of regions and any automatic newline facility should return to the left of the selected region rather than the complete plotting area. Consequently the GROATS system does not make use of the automatic new-line or top-of-page hardware. Instead checks are inserted to ensure that all typing appears on the plotting area. If typing goes outside the boundaries of the selected region then it is ignored assuming the region is contained. The rules for typing being visible are as defined in Chapter 2. However the hardware typing orders can cause a large compression of the information required by the SC 4020. Therefore the GROATS package uses these special orders when typing hardware characters away from the edge of the plotting area. In the typing mode, two of the 64 hardware characters have special significance as they can be used to change the SC 4020 from typing mode to plotting mode (these are the characters having SC 4020 internal number in octal: 12 and 56). To avoid confusion the GROATS package has decided not to allow these two characters in either the typewriting or plotting mode. A third character, the plotting dot (octal 52) is also not available in typing mode. However, as it is probably the most important plotting character, it has not been eliminated from the set of GROATS characters. Instead the user must take care not to use the plotting dot in typing operations.

The basic set of hardware characters available on the SC 4020 is too restrictive for most requirements and so it has been augmented by a set of software characters made up of short vectors (see Appendix 2). The main objections to the hardware set are that only a fixed size of about 0.1" in height is available and the lower case alphabet is not provided. The internal numbering of characters is also not very useful as it bears no resemblance to the standard Atlas internal code which Algol strings will normally be converted into. Consequently the GROATS package has defined a set of 256 characters with its own numbering system. This coincides with the Atlas internal code numbering wherever possible so that characters in Algol strings or data will have their expected interpretation. Therefore, as far as the user is concerned, he has a 256-character set. He need not concern himself with the difference between hardware and software characters. The GROATS package will attempt to use hardware characters wherever possible.

Character size and spacing are defined for each region independently so that it is possible to have different character sizes for each of the regions defined. When the units to be used in a region are defined by a call of the procedure 'limits', several typing parameters for that region are automatically initialised. In particular the size of characters defined for that region are set to the standard hardware size. Consequently, unless one of the character procedures is called, the user will automatically use hardware characters wherever possible. The hardware character size is taken as one of the standard sizes of characters in the system. It is possible to redefine character size by defining the height and width required for the characters. This is achieved by calling:

```
character size (HEIGHT, WIDTH) ;
```

The height and width are defined in terms of units of the selected region. As the limits of the selected region are changed, the size of characters will also change on the plotting area. In order that the user may keep the size of characters constant, he may call:

```
hardwaresize;
```

All regions have character settings initialised to this and characters will be printed this size even though the limits of the region are altered.

A second method of defining a character size independent of the region is by calling:

```
character factor (N) ;
```

This will cause the characters to be printed N times as large as the standard size for the character font. (See Section 4.4.)

Each character is automatically surrounded by an area which will separate it from neighbouring characters in both the horizontal and vertical directions. The character size is defined as HEIGHT units above and below the line upon which the character sits and WIDTH defines the width of the character with sufficient space between it and its neighbour. It is possible in text editing that the amount of space allowed between characters may need to be decreased or increased. This can be arranged by calling:

```
character space (X) ;
```

This inserts an extra space of X units between characters in the horizontal direction.

4.2 PLOTTING CHARACTERS

To plot the GROATS character numbered N at the point (x,y), the user calls:

```
plot (x,y,N) ;
```

The character is plotted so that the centre of the character is approximately at the point (x,y). For example:

```
limits (0,0,2,2);
plot   (1,1,95);
```

will plot the 'plotting dot' at the centre of the selected region. The plotting dot is carefully positioned so that it is positioned accurately at the point requested. Most of the hardware and software characters take up several raster positions in height and width so that it is not too obvious where the centre of the character is positioned. In the case of the hardware characters, these tend to be positioned so that the centre of the character corresponds with the plotting position. The main exception is the 'degree' character (GROATS number 94) which is considerably offset). Hardware symbols which appear to be centred accurately are +, -, 0, *. These therefore make good plotting characters. Some of the software characters (especially the lower case alphabet) are considerably offset and should therefore not be used if accurate plotting is required.

If the same character is to be plotted at several different points then the procedure call:

```
plotn (x,y);
```

can be used. This plots at the point (x,y) the same character used in the last call of the procedure:

```
plot (x,y,N);
```

Alternatively the character to be plotted in calls of 'plotn' can be defined by:

```
set plot char (N);
```

For example the three programs:

```
plot (1,1,95);                         (4.1)
plot (1,2,95);

plot (1,1,95);                         (4.2)
plotn(1,2);

set plot char (95);
plotn (1,1);                           (4.3)
plotn (1,2);
```

are all equivalent.

The definition of 'last plotted character' is not linked to the selected region but is global to the region selection commands.

4.3 TYPING

Before any typing may be done in a region, it is necessary to set the current typing position. Once this has been done, any typing command will cause typing to take place starting from this position. The simplest method of setting the current typing position is to call:

```
move type to (x,y);
```

where the position (x,y) is defined in the units of the currently selected region. The typing position corresponds to the centre of the next character to be typed. In order to start typing from the top left hand corner of the selected region, call:

```
move type to top;
```

The typing position is set so that the first character typed remains within the boundaries of the selected region.

To adjust the typing position by moving on one position call:

```
increment type position;
```

This is equivalent to typing a space. Several spaces can be left between characters typed by calling:

```
type blanks (I);
```

To reset the typing position to a newline call:

```
type newlines (I);
```

If I = 0 the typing position is reset to the left hand edge of the selected region but the Y-coordinate of the current typing position is not altered. In both the procedures with argument I, the user is allowed to use negative values as argument with the obvious meaning.

Once the typing position has been set, single characters may be typed by the call:

```
type (N);
```

which types the GROATS character having position N at the current typing position and moves the typing position one position to the right. The amount that the typing position is incremented obviously depends on the current settings of the various parameters associated with size of characters and distance between them. The procedure 'type' may be useful in some circumstances but it is more likely that strings of characters will require typing in which case the procedure 'type string' (to be described later) will be used.

To output a number starting at the current typing position, a procedure 'type number' is provided. This is based on the ICT 'print' output procedure. Calling:

```
type number (A, I, J);
```

will output the number A with I places before the decimal point and J places after. The special cases when J = 0 or I = 0 are used to define integer and floating point output formats. A complete description of this procedure is given in Chapter 8.

In order to obtain suffices and superfices, procedures are provided which automatically adjust the current character size and typing position. For example:-

```
type (33);
superfix;
type (18);
super return;
type (34);
suffix
type (19);
suf return;
type (35);
```

will type:-

A^2B_3C

Only a single level of using these procedures has been given in the example. However there is no reason why a superfix of a superfix or a suffix of a suffix or any other combination should not be typed. The standard typing for typing strings of characters is:

```
type string (I) ;
```

This types the string of characters whose position is defined by the integer I. This procedure could have been defined with the string of characters as the argument but it is more flexible in Algol to access the string indirectly. The position of a string of characters is obtained by

```
I:= pointer ('ABCD') ;
```

The procedure 'pointer' sets I to the address of the string 'ABCD'. Typing the string can then be done by calling

```
type string (I) ;
```

or alternatively

```
type string (pointer ('ABCD')) ;
```

In order to allow strings of characters to be read as data and accessed by the type string procedure, we have:

```
read string (A, IMIN, IMAX, STR) ;
```

The fourth argument, STR, of this procedure is a string consisting of two characters. All characters on the currently selected input are read and ignored until the first of these two characters appears. The input string is then read into the one-dimensional array A starting at A[IMIN] until the second of the two characters appears on the input stream. The two characters themselves are ignored and, after the procedure call, the input pointer is to the character after the terminating character for the read. On exit from the procedure, IMAX is set to the next available location in the array A. This enables several input strings to be stored next to each other in the array. The array elements each contain two characters and an extra word of the array is used to contain the length of the string. For example if the input stream is:

```
rubbish C this is the required string X more ..
```

and the procedure call is:

```
j:= 200 ;
read string (A, 1, j, 'CX') ;
```

then 'this is the required string' is stored in A[1] onwards.

To use this stored string in the procedure 'type string' it is necessary to be able to define its position and this can be done by:

```
I:= addr (A, IMIN) ;
```

which sets I to the position of the string stored in A[IMIN] onwards. For example the string read in above could be output by:

```
type string (addr(A, 1)) ;
```

One problem with the procedure 'read string', as defined, is that the only characters accessible to it as arguments are those which can be read in from data streams or stored as strings in the body of the program. The Algol system only allows those characters which have

Atlas Internal Code values to be accessed in this way. Consequently some method must be provided for typing the remaining characters of the GROATS character set. This is done by calling the procedure:-

```
                    displacement is (M);
```

After the call of this procedure, any typing instruction referring to character N will be assumed to refer to character numbered M+N.

For example:

```
                    displacement is (-64);
                    type string (pointer ('abc'));
```

will cause the characters ABC to be typed instead of 'abc' as the GROATS numbering of the two alphabets differs by 64. The characters '$\alpha\beta\gamma$' could be typed by:

```
                    displacement is (128);
                    type string (pointer ('abc'));
```

The procedure 'displacement is' acts as a generalised case shift and therefore allows all the GROATS symbols to be accessed.

Even so, this global setting of the displacement can be too inflexible and it would be more convenient to be able to change the displacement in the middle of the string in some cases.

This is achieved in the GROATS system by the ability to redefine any character in a string as a setting procedure. Once this has been done the corresponding setting procedure will be called each time this character is encountered in a string instead of typing the character. The possible procedures that may be called in this manner are:

```
               1    displacement is (I);
               2    displacement is (-I);
               3    superfix;
               4    super return;
               5    suffix;
               6    suf return;
               7    type blanks (I);
               8    type blanks (-I);
               9    type newlines (I);
               10   type newlines (-I);
               11   move type to top;
               12   thick (I);
               13   dark (I);
```

A character is replaced by a setting procedure by:

```
               define char as special (N, J, I);
```

which replaces the GROATS character N appearing in a string to be typed by a call of the J^{th} setting procedure in the above list. For example:

```
               define char as special  (33,1,128);
               define char as special  (34,3,0)  ;
               define char as special  (35,4,0)  ;
```

will replace A by a call of the procedure:

```
               displacement is (128);
```

while B and C will act as the commands superfix and super return. Calling the procedure:

```
                    type string (pointer ('DB2CAD'));
```

would produce:

$D^2\delta$

It is possible that other setting procedures will be added to the above list as the need arises. This use of spare characters for defining 'carriage operations' of a typewriter, it is hoped, will greatly increase the flexibility of the typing commands.

4.4 CHARACTER FONTS

The standard software character font is called FONT 0. This font is provided automatically with the GROATS package and has a basic size 16 x 10 raster units (this is approximately hardware size). It is made up of single straight lines and over expansion of its simple character forms make them appear thin and inelegant. Three additional character fonts are available and can be added to the GROATS package by calling the relevant library procedure. These fonts correspond to those defined at Bell Telephone Laboratories and described in a paper 'Three fonts of Computer drawn letters' by Mathews, Lochbaum, and Moss in Communications of the ACM, 10, 10 (Oct. 1967) 627-630. These fonts are much more elegant than the standard font 0 and can be used for high quality textual output. Complete details are given in the library procedures SC9 - SC11. Each region is initially associated with FONT 0. The selected region will have all characters plotted defined as font I if we first call:-

```
                    font (I);
```

5. SC 4020 CONTROL PROCEDURES

In the previous chapters, we have described the procedures required for plotting, drawing or typing information on the SC 4020 without concerning ourselves with the destination of this information. The SC 4020 has both hard copy and microfilm cameras available and it is possible to direct the graphical output to one or other of these (or to both). The details of the SC 4020 method of operation is described in Chapter 9. As far as the user is concerned, it behaves rather like a normal pen-plotter having a roll of continuous paper divided up into plotting areas 7½" square together with a roll of microfilm divided up into frames. In the following description 'frame' will be used to denote either the microfilm frame or the 7½" square plotting area. Any call of a graphical output procedure in the GROATS package will cause the correct plotting to occur on the frames that are selected. Once the user's identification has been placed at the head of his graphical output (see Chapter 6) both outputs are advanced to a new frame and the hard copy output is selected. If no control procedures are called by the user, all output will appear on the current frame of the hard copy. In order to get output on microfilm, call:

```
select camera (I) ;
```

If I = 1 then the microfilm output is selected and the hard copy becomes unselected. All output commands after this call of 'select camera' will then appear on the microfilm frame only. If I = 3 then both hard copy and microfilm will be selected. To return to plotting only on the hard copy, the procedure 'select camera' must be called with I = 2.

In order to produce output on a new frame, it is possible to advance the output to the next frame by calling

```
advance film ;
```

This will advance the selected outputs to the next frame position ready for a new frame of output to be produced. The ability to advance the frames on microfilm and hard copy separately and to output graphical information on only one or other of the two output forms means that two distinct types of output can be produced on a single run. For example:

```
select camera (3) ;
draw sine curve ;
select camera (2) ;
advance film ;
draw cosine curve ;
advance film ;
select camera (3) ;
draw tangent curve ;
advance film ;
```

would cause the sine, cosine and tangent curves to be drawn on three separate frames of hardy copy while the sine and tangent curves are output on the same frame of microfilm. This switching of output streams on and off can be quite useful in microfilm work. A selection of the frames produced on microfilm can be output on hard copy for debugging and monitoring purposes.

The fact that both output streams can be turned on and off independently does mean that the user could get confused as to the order in which frames had been produced. For example if alternate frames were being output on hard copy and microfilm then it might be difficult to merge the two. To help the user, a count is kept of the number of times 'advance film' has been called and this is printed at the top left hand corner of the selected frames after the 'advance film' procedure has moved to a new frame. The count is initialised so that:

```
FRAME 1
```

appears on the initial frame of hard copy that is selected.

This frame count may be a nuisance in microfilm work and it can be removed by calling:

```
frame (I);
```

If I is negative then this stops the frame count being output. A positive value of I will reset the count to I so that the next call of 'advance film' will cause the new frame or frames to be numbered I.

The SC 4020 hard copy output will be returned to the user in a roll. Unfortunately, this does mean that the output is difficult to cut and flatten if it is left rolled for any length of time. The Laboratory is looking into the possibility of getting a machine to cut the hard copy roll as soon as it is processed. To aid the cutting process, a cut mark is drawn for each frame outward from the bottom right hand corner. The user can stop this cut mark appearing by calling:

```
set cut mark (false);
```

The standard setting of the SC 4020 plotting areas is called REDUCE IMAGE mode and produces hard copy frames $7\frac{1}{2}$" square and microfilm frames with the standard spacing between each. A second mode called EXPAND IMAGE mode is available and is intended for 35 mm microfilm users who require to abut several frames of microfilm to allow plots to extend continuously through several adjacent frames. The exact size of the frame in EXPAND IMAGE made is adjusted for 35 mm film. The hard copy plotting area is also increased in size and in EXPAND IMAGE is approximately 8" square. The centre of the plotting area remains in the same position and expression is uniform in both directions. The size of the hardware characters, however, does not change so that, in EXPAND IMAGE mode, the spacing between hardware characters increases and the characters become rather smaller than the equivalent software characters. To change to EXPAND IMAGE mode the user should call:

```
expand image;
```

A call of the procedure:

```
reduce image;
```

returns the user to REDUCE IMAGE mode. The procedures have no effect if the correct mode is already selected.

In Chapter 3, the procedure 'dark' was described which enabled vectors and software characters to be produced darker than the standard plotting intensity. The procedure 'dark' does not however alter the intensity of the hardware characters. It is possible to define two different intensities of these using the SC 4020 hardware. Hardware characters are either produced in HEAVY or LIGHT mode. The standard setting used by the GROATS system is the HEAVY mode. If the user wants to change to the LIGHT mode he should call:

```
expose light;
```

The HEAVY mode can be reset by calling:-

```
                    expose heavy;
```

As stated above the hardware character modes, HEAVY and LIGHT, are produced by hardware and are therefore global and not defined just for the selected region. This is, of course, different from the 'dark' setting of vectors which is local to the region.

An interesting facility available on the SC 4020 is the ability to produce a 'form slide' which can be used as a standard background to a set of plotting frames. The 'form slide' has to be specially produced and loaded into the SC 4020. To superimpose the 'form slide' on the output frames currently selected the user calls:-

```
                    project form;
```

Examples of possible 'form slides' are maps and business forms. Details of how to get the operator to load your own form slide are given in Chapter 6. A special Atlas Laboratory form slide is displayed on the identification frame and it is necessary to inform the operator that it is to be replaced by the user's own slide.

In the production of movies, it is often necessary to repeat the same or a set of frames many times. This could be achieved by repeated generation of the same frame many times. However the SC 4020 does provide a more efficient method of doing this by hardware. There is a hardware facility for marking a position on the IBM tape and later rewinding to this position and repeating the intervening plotting information several times. In the GROAT's system, the information to be plotted several times must be preceded by the call:

```
                    advance repeat (0);
```

and is terminated by the call:

```
                    advance repeat (I);
```

The initial call advances the selected output to a new frame. All the frames between this frame and the current frame are repeated I times. The number of times the frames are repeated must be less than or equal to 31. There is no simple limit on the number of frames which may be repeated. It largely depends on the amount of information plotted on each frame. The total number of IBM records repeated must be less than 511. If the user attempts to repeat more than 511 records then a diagnostic will be given.

6. HOW TO USE GROATS ON THE CHILTON ATLAS

The GROATS package of graphical output procedures has been added to the standard I/O packages available. The set of possible I/O packages is now:-

ICT, KDF9, ELLIOTT, GROATSICT, GROATSKF9, GROATSELLIOTT

These can be added to a user program by the relevant processor command. For example:

```
INPUT ATLAS CARDS WITH GROATSICT I/O PROCEDURES;
```

would indicate a program to follow in the ATLAS CARDS dialect using the GROATS and ICT I/O procedures. Complete details of the Algol system are given in the relevant Algol manual. In what follows GROATSICT will be used as the example. Everywhere this could be replaced by GROATSKDF9 or GROATSELLIOTT. The size of the GROATSKDF9 and GROATSELLIOTT packages is about 40 blocks. This does mean that the minimum amount of store required at compilation and execution time will be 165 and 55.

Unlike the standard I/O packages, the GROATSICT package does not fit easily into the Atlas Algol system and it has been necessary to bend the system a little to accommodate it. The GROATSICT package has global declarations of scalars and arrays as well as procedure declarations. The Algol compiler was not designed to accept precompiled I/O packages of this form and it has been found necessary to redeclare these global variables in the outer block of the user's program. This is inconvenient to the user but they have been grouped together as library item SC1 so that all that is necessary is to insert this library item at the head of the program after the initial <u>begin</u>. There is unfortunately one further restriction. The declarations defined for scalars and arrays in the user's outer block must be identical to those at the head of the program. This means that the user may not add his own declarations to those declared in SC1 but must declare them in an inner block. As Algol <u>own</u> <u>arrays</u> appear to be declared in the outer block in the Atlas implementation, they cannot be used in programs that use the GROATS package. As far as the user is concerned, his Algol program should therefore be contained between:-

```
                    begin
                    library  SC1 ;
                    begin
                    ......

and                 ......

                    end
                    end
```

instead of between <u>begin</u> and <u>end</u>.

A typical program with its Job Description would look like:

```
JOB
10000  BLOGGS  GROATS OUTPUT OF SINE
COMPUTING 30 SECONDS
STORE 50/150 BLOCKS
OUTPUT   0  LINEPRINTER  300  LINES
OUTPUT  12  LINEPRINTER   50  LINES
OUTPUT  13  LINEPRINTER   50  LINES
TAPE IBM 14 COMMON/SC/USER'S NAME AND ADDRESS
DISC COMMON 126/180
DISC 99 R502 ALGOLIB
COMPILER ALGOL
INPUT INTERNAL ICT WITH GROATSICT I/O PROCEDURES;
begin
   library  SC1;
   begin
      real  x,dx,pi;
      pi := 3.14159;
      dx :=  .05;
      identification;
      limits (0, -1, 2*pi, 1);
      for  x := 0  step  dx  until  2*pi - dx  do
      vector (x, sin(x), x + dx, sin(x + dx) );
      end plotting;
   end
end
```

The program for plotting the sine curve is the same as given in program (2.1). It is necessary now to explain the additions that have been made to the Job Description and the program.

6.1 EXTENSIONS TO JOB DESCRIPTION

Although it is possible to run the SC 4020 on-line to a computer, the method adopted at the Atlas Laboratory is for the GROATS package to generate on Atlas a ½" IBM magnetic tape containing the necessary SC 4020 orders. The tape is unloaded from Atlas and later mounted on the SC 4020's IBM tape deck. The SC 4020 orders can then be read from the magnetic tape by the SC 4020 and the relevant graphical output produced. This accounts for the IBM tape declaration in the Job Description. The IBM magnetic tape must always be numbered 14 and the user should make sure that his name and address appear in the tape declaration.

The two output streams 12 and 13 must be declared in the Job Description. They are used for printing messages to the SC 4020 operator and also giving details of the graphical output produced by the job. The number of frames of hardcopy and microfilm produced by the job will be printed and these figures will be used for charging purposes and also giving the SC 4020 operator some idea of the size of the job. Although several frames of identification always appear before and after each job, the user will only have one frame of hardcopy and microfilm added to his totals. The information on streams 12 and 13 are basically the same. Stream 13 is returned to the user with his program. Stream 12 is kept with the magnetic tape when it is passed to the SC 4020 operator from Atlas. It is retained by the SC 4020 operator for record purposes. It will normally be possible to have a job processed again if the Laboratory is informed within five days of the program being run on Atlas (telephone

Ext. 6640). After this period the SC COMMON IBM tape is returned to store and will be used again. The number of the COMMON tape used will be marked on stream 13 when it is returned to the user. Quoting this SC number will help the operator in locating the correct tape for the rerun. If a user requires to keep the graphical information on the IBM tape for a longer period than five days, he should obtain his own private tape to be used in place of the COMMON tape.

The DISC directive in the Job Description is needed so that the standard Algol Library on the disc is associated with the program. This contains the library item SC1 which needs to be inserted at the head of the program. The Algol Library contained on the disc is basically a subset of the standard library stored on the magnetic tape ALGOLIB (It is hoped to put the more frequently used of the Algol library procedures on the DISC version of the library tape.).

6.2 IDENTIFICATION OF GRAPHICAL OUTPUT

When the SC 4020 is fully loaded, it will be producing graphical output at a far greater rate than the Benson-Lehner Model J. This does mean that much greater care must be taken in ensuring that all output produced by the SC 4020 is correctly identified and terminated. This identification will be automatically produced for the user by the GROATS package if he calls the procedure:

```
identification;
```

before attempting to plot any graphical output at all and calling the procedure:

```
end plotting;
```

after he has completed his graphical output. The GROATS package will attempt to call the procedure 'end plotting' if the program comes to some unexpected error stop. The user should ensure that the procedure is called when the program terminates correctly. In the case of error stops, it may not always be possible to enter the procedure 'end plotting' in which case the user may get the last frame of his output overwritten. The main purpose of the 'end plotting' procedure is to add some file marks at the end of the graphical output. If these are not present then the records on the tape from a previous job using that tape could get reproduced. If the user is in any doubt as to whether 'end plotting' had been called, he can examine output stream 13 which will have the comment:

```
FILEMARKS ADDED AT JOB TERMINATION
```

output if 'end plotting' has been called.

The procedure 'identification', apart from identifying the graphical output, also initialises buffers and tables so that a failure to call 'identification' before any plotting procedure will result in a complete breakdown of the system. The successive frames of output produced by 'identification' are as follows:

1. The word CUT is produced in large type with a dotted line at the bottom of the frame. The SC 4020 operator will cut the output along the dotted line so that this record will go with the previous user's output. Similarly the word CUT appearing at the front of the next user's output will appear on the end of this output.

2. An alignment frame is produced consisting of a square along the edges of the plotting area with diagonals and lines joining the centres of opposite sides. The symbol + is plotted at the centre of the square and at two of the vertices while ◘ is plotted at the other two vertices of the square.

3. An identification frame marked FRAME 0 and having a cut mark at the bottom is produced next. This frame is overprinted with the standard Atlas Laboratory form slide. Near the bottom of the frame, the word ALGOL is printed. On another line will appear the Atlas Job Title followed by the time and date when the frame was produced. This will enable the user to match up his graphical output with Atlas output. As the time on his Atlas output streams is taken at a different time from the call of the procedure 'identification', the two times will not match exactly although they should be quite close to each other. This frame will also have the job number printed in large type so that it can be read on the microfilm.

After the identification frame, the frame is advanced and will be headed FRAME 1 and a cut mark will appear at the bottom. On leaving the procedure 'identification', the system will be initialised as follows:-

1. The SC 4020 will be set to REDUCE IMAGE mode and the hardware characters will be produced in HEAVY mode.

2. The user's graphical orders on the IBM tape will start at a new record.

3. The frame count will be set to 2 (the next frame to be output after this one.

4. Cut marks will be produced at the end of each frame.

5. The plotting regions 0 to 10 will all be defined as the complete plotting area with scaling defined by:

   limits (0,1023,1023,0);

   The region 0 will be selected. Character size in all regions will be defined as hardware size. All vector drawing will be single lines (dark(1); thick(1); broken(0);); axes will be used wherever possible and zero length vectors will not be plotted. All regions will be defined as contained.

6. The heading

```
* * * * * * * * * * * * * * * * * * * * * *
IBM TAPE TO SC4020 PLOTTER
DONT RETURN TO COMMON
* * * * * * * * * * * * * * * * * * * * * *
```

   is produced on Output Stream 13.

7. The hardcopy camera is selected by itself.

The 'end plotting' procedure repeats the 'identification' frame which appeared at the start of the job and outputs on streams 12 and 13 the number of frames of hardcopy and microfilm produced.

6.3 ADDITIONAL FACILITIES

In the debugging stage of writing a program it may be useful to stop the generation of graphical records on the IBM tape or alternatively get a more detailed knowledge of the type of orders being produced on the IBM tape. Several debugging aids have been included in the GROATS package for this purpose. They were originally intended for debugging the GROATS package itself but may be of use to the general user.

If the IBM TAPE directive in the Job Description is missed out then the GROATS system will carry on as usual producing records of output ready to go to the IBM tape but will not attempt to write to the non-existent tape. This is therefore recommended for the first few attempts at running the program. Abortive runs due to compilation errors or logical errors will then not require the loading of the IBM tape. This is useful both to the user and the operating system. It will mean that a job, which probably would not have used the IBM tape deck, will not reserve it unnecessarily. Also, in many cases, the job will not now go into the Atlas tape queue which could improve the user's turn round considerably.

A facility is provided for turning on and off the sending of records to the IBM tape. This can be useful when testing a run which is intended to produce many thousands of frames on the SC 4020. The user can do a preliminary run which allows only a small proportion of the complete number of frames to actually go to the IBM tape. Once this has been shown to be producing the correct output then the complete production run could be done. Passing of records to the IBM tape is turned off by calling:

```
                    pass to ibm tape (false);
```

and turned on again by:

```
                    pass to ibm tape (true);
```

In each case the setting refers to the next IBM tape record. A new IBM tape record can be forced at any point by calling:

```
                    ignore rest of record;
```

For example:

```
                    draw sine curve;
                    pass to ibm tape (false);
                    draw cosine curve;
                    pass to ibm tape (true);
```

would not necessarily cause the cosine curve to be stopped. Part of the sine curve is probably in an uncompleted buffer of SC 4020 orders when the passing to IBM tape is turned off. Consequently when this buffer is filled with part of the cosine curve, the completed buffer would be ignored. Similarly part of the cosine curve might be in an uncompleted buffer when passing was turned on again so that this would be output. To stop the production of the cosine curve requires:

```
draw sine curve;
ignore rest of record;
pass to ibm tape (false);
draw cosine curve;
ignore rest of record;
pass to ibm tape (true);
```

It is possible to print out the SC 4020 orders being put onto IBM tape by calling:

```
print SC 4020 records (true);
```

Each record to be passed to the IBM tape will be printed on the currently selected output stream. Each 36-bit SC 4020 instruction is printed on a new line as 12 octal digits (The format for the SC 4020 orders is given in Appendix 4.). The printing can be inhibited by calling:

```
print SC 4020 records (false);
```

As described above, the orders that will be printed correspond to the last complete record and, in order to print only the relevant orders, the procedure 'ignore rest of record' can be used. The ability to print records or inhibit the passing of the records to the IBM tape act independently so that, for example, passing could be suppressed while still getting the records printed.

6.4 OPERATOR REQUESTS

In general there is a standard setting-up of the SC 4020 which cannot be altered by the user. However there are two ways in which this standard setting-up can be altered by the user.

Normally the SC 4020 will have the 16 mm microfilm camera loaded and any microfilm output will be produced on this. It is hoped that the Laboratory will also obtain a 35 mm camera fairly soon. If a user requires to use this camera in place of the 16 mm camera then he must call:

```
mount 35 mm camera;
```

before calling the procedure 'identification'. Calling after the call of 'identification' would, of course, mean that the identification would have been produced on 16 mm film. A message will be output on output streams 12 and 13 so that the operator will load the correct camera.

The standard form slide used will be the Atlas identification slide. To replace this slide by the user's own at some point in the run, call:

```
insert user form slide;
```

This will produce a message on the output streams 12 and 13 warning the operator that the user's form slide is required. An order stopping the SC 4020 will be generated at this point so that the operator has time to change the form slide. If more than one user form slide is required then special arrangements should be made with the SC 4020 operators.

6.5 SIMPLIFIED LAYOUT

If the procedure 'identification' is not called before any plotting orders are obeyed then the program will come to an illegal stop. A layout which ensures that this will not happen is as follows:-

```
begin
   library SC1 ;
   identification ;
   begin


      { program }


   end;
   endplotting ;
end
```

This positioning of the calls of 'identification' and 'endplotting' is to be recommended.

7. EXTENDED FACILITIES

In this section we shall describe some of the system routines of less general use. Also we shall describe some procedures which, although not as basic to the GROATS package as the procedures already described, should be used sufficiently often that it is more reasonable to include them as part of the GROATS system rather than as library items on the standard library tape. It is possible that library items, which are found to be used frequently, will be added to this set.

7.1 GRATICULE PRODUCTION

Unlike the standard graphical pen-plotters, the SC 4020 hardcopy output uses specially treated paper. A consequence of this is that, unlike the pen-plotters which use graph paper, the SC 4020 sheet is initially blank and if a graticule of any kind is required then this must be produced by the user.

A procedure in the GROATS package is available for covering the selected region by a graticule. For example:

```
graticule at (dx, dy) ;
```

will draw horizontal lines across the region at a separation of 'dy' and vertical lines at a separation of 'dx' in the units of the selected region. The program:

```
limits  (0, 0, 7.5, 7.5) ;
graticule at  (0.1, 0.1) ;
graticule at  (1, 1) ;
```

would define the whole region as having inches as units and graph paper would be produced at intervals of 0.1". The second call of 'graticule at' would darken the markings every 10th line. The lines are drawn so that lines in both directions would pass through the point (0 , 0) if that was in the bounds of the selected region. It is perhaps worth emphasizing that graticules will be produced much more efficiently in the 'use axes' rather than 'use vectors' mode. (See Chapter 3)
A second procedure is:

```
axes at (x, y) ;
```

which draws axes through the point (x , y) reaching to the edges of the selected region.

Scales may be printed in the horizontal and vertical directions by calling:

```
scales at (dx, dy, xo, yo, mx, nx, my, ny) ;
```

This takes (x , y) as the origin and prints the values at intervals of 'dx' and 'dy' in the two directions. In the x-direction, the printing will occur just below the line y = yo and the form of printing will be equivalent to:

```
typenumber  (x, mx, nx) ;
```

For integers, the least significant digit will be positioned at the point to which the printing refers. Otherwise the decimal point will be at this position. In the y-direction, the printing will be to the left of the line x = xo and the centre of the characters will be positioned at the point to which they refer. For example:

```
                    limits  (0, 0, 10, 10);
                    region  (2, 2, 8, 8, 1);
                    select region (1);
                    limits  (0, 0, 5, 5);
                    graticule at  (0.1, 0.1);
                    graticule at  (1, 1);
                    scales at  (1, .5, 0, 0, 1, 0, 1, 1);
```

would define a region in the centre of the plotting area having units ranging from 0 to 5 in both directions. The graticule is again at intervals of 0.1 with the unit positions darker. Around the edge of the region the scales are marked every integer position in the x-direction and every 0.5 in the y-direction. In the y-direction one place before and after the decimal point is printed.

7.2 IDENTIFICATION SUBROUTINES

In the production of the identification frame, two procedures have been written for obtaining information about the job. The first is:

```
                    job title  (A);
```

which stores as a string in the array A, from A[1] onwards, the title ot the job. This could then be output on a frame by:

```
                    type string  (addr(A,1)) ;
```

The array A must be one-dimensional and have a range from A[1] to A[41] at least.

The second procedure will read into the array A either the current time or date:

```
                    job time date (A,I);
```

If I = 1 the current time is put in the array A, while I = 2 will put the current date. Once more the string is stored from A [1] onwards (the range must be from A[1] to A[5] at least).

7.3 LINEPRINTER SIMULATION

In the description of the SC 4020, we have been considering its more normal mode of acting as a graphical output device. As stated previously, SC 4020 orders are stored on an IBM tape by the GROATS package. The IBM tape is produced in odd-parity mode and the SC 4020 will translate these orders into vector drawing, character generation or control procedures.

It is possible to run the SC 4020 as though it resembled a line-printer. If lineprinter output is produced on microfilm, it is much more compact and therefore it is a more reasonable output medium when very large amounts of information are required. The number of lines produced per minute by the SC 4020 is also much higher than the conventional lineprinter. The SC 4020 switches to lineprinter simulation mode when it encounters an even-parity record on the IBM tape. In this mode it is possible to define page throws or alternatively a page throw is inserted after every 64th line.

The character set of the SC 4020 agrees fairly closely with the lineprinter used at the Laboratory. Appendix 6 lists the form that lineprinter characters will take on the SC 4020 in the cases where there is no direct conversion.

The lineprinter simulation mode is entered by calling:

```
lineprinter on;
```

This advances the selected cameras so that the lineprinter output starts at the head of a new frame. After this call, all output, which was to go to the more conventional output devices, will go to the SC 4020. If more than one output stream is defined then the output streams will be merged on the SC 4020 in the order of production.

The SC 4020 graphical mode can be returned to by calling:

```
lineprinter off;
```

Again the frame will be advanced so that the graphical output will appear on a new frame and the framecount that is printed will be one after the count at the last graphical frame output.

7.4 REPETITION OF SC 4020 ORDERS

It is possible that some part of a frame may behave as a static background which is required to appear on several frames. Rather than having to recompute this information, it is possible to save the orders in an array. For example:

```
background;
scene (1);
background;
scene (2);
background;
scene (3);
```

might define a program which produced 3 frames called scenes 1 to 3 with a static background. This could have been written:

```
start saving (A, 1, 1000);
background;
finish saving (A, MAX);
scene (1);
use saved (A, 1, MAX);
scene (2);
use saved (A, 1, MAX);
scene (3);
```

The one-dimensional array integer A is provided by the user to store the SC 4020 orders. The procedure 'start saving' defines the limits of the array A to be used. In the example, the SC 4020 orders will be stored from A[1] onwards. The procedure 'finish saving' defines the end of the set of orders to be saved. The variable MAX is set to the first free location not used by the SC 4020 orders. The procedure 'use saved' causes the SC 4020 orders to be reoutput at this point. In the example, the array elements containing the SC 4020 orders are stored in A[1] to A[MAX-1]. Three more array elements are required than the number of SC 4020 orders.

7.5 GRAPHIC LIBRARY

It is hoped that the GROATS package defines a fundamental graphic system which can be extended by library routines to cover most user applications. A new class of library items having the name SC has been added to the standard library on the disc and to the tape version ALGOLIB. Unless the library item is unusually long the item will be available on the disc library R502 ALGOLIB. In the list following of items currently available, those marked with an asterisk are on the disc as well as ALGOLIB. Additional library items will be given in Algol Papers as they become available.

SC1 (*)

This contains the standard set of declarations required at the head of every GROATS job.

SC2

An uncompiled listing of the GROATS package.

SC3 (*)

```
        procedure use saved and restart (A,IST,J,IMX);
        value IST,IMX,J;
        integer IST,IMX,J;
        integer array A;
```

Three procedures exist in the GROATS package for saving graphical output so that it does not need recomputing if it is required for a later frame. These procedures are:

```
        start saving (A,IST,IMX);
        finish saving (A,J);
        use saved (A,IST,J);
```

The first procedure defines the start of graphical output to be stored in A[IST] onwards with a maximum position to use at A[IMX]. The second procedure defines the end of the graphical information to be saved. The information is stored between A[IST] and A[J]. The third procedure defines a subsequent use of this information.

It frequently happens that the amount of information that is to be saved gradually increases. On one frame a certain amount of graphical output is presented; on a subsequent frame this together with some additional output is to be displayed and so on.

SC3 is intended for such a use. The procedure 'use saved and restart' will cause the equivalent action of:

```
        use saved (A,IST,J);
```

and then will continue storing information in A[J] onwards until a call of 'finish saving' will define the end of the augmented set of saved information. The program below indicates how it could be used:-

```
        start saving (A,1,10000);
        scene;
        finish saving (A,J);
        advance film;
        for i:= 0 step 1 until 10 do
        begin
            use saved and restart (A,1,J,10000);
            output addition to scene(i);
            finish saving (A,J);
            advance film;
        end;
```

SC4 (*)

```
        procedure texture(xmn,ymn,xmx,ymx,type,deg);
        value xmn,ymn,xmx,ymx,type,deg;
        integer type,deg;
        real xmn,ymn,xmx,ymx;
```

The rectangular area in the selected region having opposite corners (xmn,ymn) and (xmx,ymx) will be 'textured' by one of several possible patterns.

The periodicity of the pattern varies according to the setting of the parameter 'deg'. If deg = 0 then the periodicity is 20 raster positions, deg = 1 then 10 raster positions, deg = 2 then 5 and so on. On the hardcopy output, 20 raster positions is about .15 inches.

There are 9 different designs available in the procedure and the one used is determined by the parameter 'type'. These are as follows:-

type	design
0	No design at all.
1	Plotting dots aligned by rows.
2	Horizontal lines.
3	Vertical lines.
4	Diagonal lines (bottom left to upper right).
5	Diagonal lines (top left to bottom right).
6	Horizontal and vertical lines.
7	Diamonds formed from the two sets of diagonal lines.
8	The equivalent of 1 together with itself displaced by .5 of period in each direction.

The periodicity of the pattern is, in fact, $10 \times 2^{1-deg}$. Values of 'deg' outside the range 0 to 2 will probably have little value as they produce textures either too coarse or too fine.

SC5 (*)

```
        procedure histogram (X,Y,DX,YV,M,N,TYPE,DEG);
        value X,Y,DX,M,N;
        integer M,N;
        real X,Y,DX;
        integer array TYPE,DEG;
        real array YV;
```

The procedure draws axes across the selected region through the point (X,Y). In addition a histogram will be drawn. The interval in the x-direction will be DX. The histogram drawn will have y-values, YV[i] in the interval $(X+i \times DX,\ X+(i+1)\times DX)$. The histogram is defined for x-range, $(X+M\times DX,\ X+(N+1)\times DX)$. That is the function values defined are YV[M] to YV[N]. The Ith interval will be shaded with texture of type TYPE[I] and degree DEG[I] as defined in SC4.

The elements of the histogram may extend either above or below the base line y = Y.

<u>SC6</u> (*)

```
        procedure  conic (X,Y,A,B);
        value  X,Y,A,B;
        real   X,Y,A,B;
```

The procedure will draw the conic

$$\frac{(x-X)^2}{A^2} + \frac{(y-Y)^2}{B^2} = 1 \qquad \text{if } A > 0,\ B > 0$$

$$\frac{(x-X)^2}{A^2} - \frac{(y-Y)^2}{B^2} = 1 \qquad \text{if } A > 0,\ B < 0$$

$$(y-Y)^2 = 4A(x-X) \qquad \text{if } B = 0$$

The units are those of the selected region. For $B \geq 0$, the curve is cut off at the bounds of the selected region. The procedure uses library item SC7.

<u>SC7</u> (*)

```
        procedure  curvfunc (F,XMN,XMX)
        value  XMN,XMX;
        real   XMN,XMX;
        real  procedure  F;
```

This procedure draws the curve y = F(x) between XMN and XMX. The function F(x) must be single valued between XMN and XMX (XMN < XMX).

<u>SC8</u> (*)

```
        procedure  curve (C,N,XV,YV);
        value    C,N;
        integer  C,N;
        real  array  XV,YV;
```

The set of points XV[I], YV[I] for i = 0(1)N are joined to form a <u>closed</u> curve if C = 0 and an <u>open</u> curve if C = 1. The points I and I+1 are joined by the arc of the circle which also goes through the point I+2. Points can be repeated but no two consecutive points can be identical. (In the case of <u>closed</u> curve, points N and 0 are treated as consecutive.) The curve produced will appear smooth if sufficient points are used.

SC9, SC10 and SC11

Three additional character fonts have been added to the standard one used in the GROATS output package. These fonts correspond to those defined at Bell Telephone Laboratories and described in the paper 'Three fonts of Computer drawn letters' by Mathews, Lochbaum, and Moss in Communications of the ACM, 10, 10 (Oct. 1967) 627-630. These fonts are much more elegant than the standard font 0 defined in the GROATS package and can be used for high quality textual output. They do, of course, require more computing time for each character generated.

The library items contain:-

SC9 <u>procedure</u> initialise font 1;

SC10 <u>procedure</u> initialise font 2;

SC11 <u>procedure</u> initialise font 3;

Before any of the three fonts can be used, the necessary initialising procedure must be added to the program and called. Typing in any one of the fonts can be started by calling:

font (I);

where I is the desired font number.

The standard size of each of the character fonts (the one obtained after calling character factor (1);) is:-

	FONT 1	FONT 2	FONT 3
width of character	18	28	36
height of character	12	18	25
x coordinate of centre	9	14	18
y coordinate of centre (from top)	6	9	12

The measurements are in raster positions.

The complete GROATS character set is not currently available in fonts 1, 2, 3. The characters available are those with GROATS numbering as follows:-

FONT 1

2, 3, 6, 8-59, 66, 69, 70, 75, 78-82, 84, 85, 86, 90, 91, 96-122, 128, 139, 159, 161-186, 189, 225-250.

FONT 2

2, 3, 5, 6, 8-59, 62, 66, 69, 70, 75, 76, 78-87, 90, 91, 96-122, 128, 130, 131, 135, 136, 139, 141-143, 158, 159, 161-186, 189, 225-250.

FONT 3

2, 8-10, 12, 15-25, 31-58, 66, 78, 81, 82, 97-122, 128, 142.

SC12 (*)

```
        procedure  fast vector;
```

The vector drawing procedures in the GROATS package allow vectors to be thickened, darkened or broken by the setting of certain parameters. This flexibility in the form of output produced does mean that vector drawing is not as efficient as it might be. The above procedure 'fast vector', when called, will replace the vector drawing routines by a more efficient version which does not allow lines to be thickened, darkened or broken. It will still use axis-drawing wherever possible and also ignore zero-length vectors if required. It is estimated that, by using 'fast vector', vector drawing will be speeded up by a factor of between 2 and 3. The procedure 'fast vector' can only be used with the GROATSICT package.

A second call of 'fast vector' will restore the original vector routine. By switching between the two vector routines, a user can get the best of both worlds and have generality as well as efficiency.

SC13 (*)

```
        procedure composite histogram (X,Y,DX,YV,M,N,NU,TYPE,DEG);
        value  X,Y,DX,M,N;
        integer  M,N;
        real  X,Y,DX;
        integer array  TYPE,DEG,NV;
        real array  YV
```

This procedure is similar to SC5. The procedure draws axes across the selected region through the point (X, Y). In addition a histogram will be drawn. The interval in the X-direction will be DX and the histogram is defined for intervals (X + I * DX, X + (I + 1) * DX) where I ranges from M to N.

For each interval I, instead of a single Y value to be plotted, NV[I] different Y values may be given and each sub-part of the histogram may be textured differently. For each interval I, a set of Y-values, YV[I,J] must be given for J = 0 to NV[I] and YV[I,J] ⩾ YV[I,J - 1] and YV[I,J] will be shaded with texture TYPE [I,J] and degree DEG[I,J]. (See SC5.)

The procedure uses the texture procedure defined as SC4.

SC14 (*)

```
        procedure  shade area (XV,YV,N,XMN,YMN,XMX,YMX,TYPE,DEG);
        value  N,TYPE,DEG,XMN,YMN,XMX,YMX;
        integer  N,TYPE,DEG;
        real  XMN,YMN,MXM,YMX;
        real array  XV,YV;
```

A closed area is defined by the set of points (XV[I], YV[I]) where I = 0(1)N. The point (XV[0], YV[0]) is joined to the point (XV[N], YV[N]). The consecutive points are assumed to be joined by straight lines. The area so defined is textured by a pattern defined by TYPE,DEG. These parameters have the same meaning as in the library item 'texture' defined as SC4.

An even number of vectors that are coincident has the same effect as no vectors so that it is possible to shade figures with holes in or alternatively two distinct figures with the same texture simply by joining the figures with a pair of coincedent vectors from any two points. For example a square defined by the points 1, 2, 3, 4 having a triangular hole defined by 5, 6, 7 can be shaded by defining the area to be shaded as 123456754. The procedure only produces the shading. It does not join the edge points of the area by straight lines. The arguments XMN, YMN, MXM, YMX define the limits of the currently selected region.

SC15 (*)

```
        procedure illustrate (N,XV,YV,XMN,YMN,XMX,YMX,NAREAS,
                  TYPE,DEG,NVEC,NV,NLINES,NPOINTS,NP);
        value N,NAREAS,NLINES,XMN,YMN,XMX,YMX;
        integer N,NAREAS,NLINES:
        real XMN,YMN,XMX,YMX;
        real array XV,YV;
        integer array TYP,DEG,NVEC,NV,NPOINTS,NP;
```

This is an extension to SC14 which will allow a whole scene or picture to be produced which consists of several areas to be shaded and several lines to be drawn.

The arrays XV,YV[1 : N] contain the coordinates of all the points defining the areas and lines.

The number of areas in NAREAS. The Ith area has texture defined by TYP[I], DEG[I] and is defined by the sequence of points XV,YV[K] where K = NV[I,J] and J = O(1) NVEC[I]. In addition NLINES lines will be drawn. The Ith line (I = 1(1)N) is drawn between the points (IXV,YV[K]) where K = NPOINTS[J] and J = 1(1) NPOINTS[I].

This indirect way of addressing the points which define the object does mean that texturing of a scene with certain characteristics but changing in detail can be achieved by simply changing the set of data values in XV,YV.

The shading is done by using SC14 and this must also be added to the program. The values XMN,YMN,XMX,YMX are the limits of the selected region.

SC16 (*)

```
        procedure contour (XV,M,YV,N,ZV,CV,R);
        value M,N,R;
        integer M,N,R;
        real array XV,YV,ZV,CV;
```

A mesh is defined by the set of X and Y values, XV[0 : M] and YV[0 : N]. At the intersections on the mesh, a function is defined having values ZV[0 : M, 0 : N]. This procedure will attempt to draw the set of contour lines at the function values CV[1 : R]. The mesh must be sufficiently fine so that a contour line must not enter and leave a square on the same side.

The contour lines are drawn by using the procedure 'curve' defined as SC8. This joins points together on grid lines which have function values equal to the contour value. These points are found by linear interpolation over each of the grid squares in turn.

The procedure does not draw axes, graticule or scaling. These must be produced by the user.

SC17 (*)

```
procedure  curve (F,P,A,B,INC,PHI,EPS);
value  A,B,INC,PHI,EPS;
real   A,B,INC,PHI,EPS;
procedure  F;
real array  P;
```

This procedure draws the function F in the range A $\leq$ T $\leq$ B where T is a parameter from which both the x and y coordinates are found by F. P is an array of constants which may be used in the definition of F. The maximum increment of T is INC and this is subdivided recursively until chords in adjacent subdivisions differ in direction by less than PHI radians or the chords are shorter than EPS. The maximum chord length satisfying these conditions is found at all points and drawn. Finite discontinuities of the function and its derivatives will be plotted correctly. F has arugments F(X,Y,T,P).

The procedure was provided by P.J. le Riche.

SC18(*)

```
integer array  savearray [1:4000], hasharray [1:256],
         savepointer [1:2,1:256];
integer  savepos, savehigh, fram, framemax;
integer procedure  hash (st,pos);
value  stpos;
string  st;
integer  stpos;

procedure  def (PROC,ST);
procedure  PROC;
string  st;

procedure  release (ST);
string  ST;

procedure  draw (ST,NUM);
value  NUM;
string  ST;
integer  NUM;

procedure  drawvar (ST,NUM,PROC);
value  NUM;
integer  NUM;
string  ST;
procedure  PROC;

procedure  change display (ST1,ST2,N);
value  N;
integer  N;
string  ST1,ST2;
```

The aim of this set of procedures is to ease the generation of cine films by making it much easier to make alterations to a film sequence. The array 'savearray' is used to store sub-pictures. The sub-pictures are given names and procedures are provided for recalling the sub-pictures and displaying them as required.

The procedure 'def' stores away the picture that would have been generated if a call of the procedure PROC had taken place at his point. PROC must have no arguments. The string ST is the name assigned to this pictorial information and it must consist of exactly 6 characters.

For example:-

```
        def (border, 'BORDER');
```

defines the border surrounding the current region and gives it the name BORDER.

Many sub-pictures can be defined by calling 'def' with different arguments. The plotting information is saved in the array 'savearray'. If the array becomes full then subsequent picture definitions are ignored. Space in the array 'savearray' can be reused by deleting information already stored. This is done by:-

```
        release ('BORDER');
```

This would, for example, release the storage used for the sub-picture BORDER <u>and all pictures defined after the definition of BORDER</u>.

To display several subpictures for NUM frames, the names of the sub-pictures should be put in the argument ST of 'draw' separated by commas.

For example if sub-pictures named BORDER, CIRCLE, ELLIPS have been defined then:-

```
        draw ('BORDER', 10);
        draw ('BORDER, CIRCLE', 15);
        draw ('BORDER, CIRCLE, ELLIPS', 10);
```

would produce a film sequence of 10 frames with just BORDER displayed followed by 15 frames with BORDER and CIRCLE displayed and then 10 frames with BORDER, CIRCLE and ELLIPS displayed.

Frequently a constant background is required together with some information whose position varies linearly over a set of frames. The procedure 'drawvar' will generate NUM + 1 frames consisting of the background ST defined as for 'draw' with, in addition, a variable picture defined by calling PROC. The global variable 'framemax' is set to NUM and the global variable takes the value 0,1,2 .. NUM on the NUM + 1 calls of PROC. The procedure PROC can therefore,be given 'fram' and 'framemax', define the variable plotting required.
For example:-

```
procedure  movingbox;
begin
     real  r, a;
     r: = 1/framemax;
     a: = r*(5*(framemax - fram) + 7*fram);
     box (a,a,a + 3,a + 3);
end;
drawvar ('BORDER,CIRCLE', 20, movingbox);
```

would produce 21 frames with the background BORDER and CIRCLE and a box which starts having coordinates (5,5,8,8) and finishes with coordinates (7,7,10,10);

The final procedure 'change display' is used to change the display from ST_1 to ST_2 in N + 1 frames. At each frame, 1/N lines of the display ST_1 are replaced by 1/N lines of ST_2. A pleasing effect of the picture dissolving from one scene into another is achieved. The pictures defined by ST_1 and ST_2 must consist only of lines and must not contain any hardware character plotting. Also it is advisable that the frame count and cut mark are not being produced. For example:-

```
change display ('BORDER, CIRCLE', 'ELLIPS', 10);
```

will change the BORDER and CIRCLE displayed into ELLIPS in 11 moves.

If insufficient space is given by the bounds of 'savearray', the upper limit can be increased as long as the value stored at label 15 of procedure 'hash' is also increased to the same value.

8. COMPLETE DEFINITION OF GROATS PROCEDURES

This section lists all the GROATS procedures available to the user together with a description of what they do.

active

```
procedure active;
```

The currently selected region is defined as the active region. The active region has no effect unless that region is also defined as a shield. If the active region is a shield then any subsequent plotting cannot appear within the bounds of the active region even though the region selected at the time is attempting to output there. The active region as a shield therefore acts as a global mask independent of the region selected. Together with the passive region it can be used for producing unusual wipes during cine production.

addr

```
integer procedure addr (ARY,ELEMENT);
value ELEMENT;
integer ELEMENT;
integer array ARY;
```

This is a code procedure which sets the function 'addr' equal to the address of ARY[ELEMENT]. If a string has been stored in ARY by the 'read string' procedure then 'addr' will obtain the position of the string in the correct form for the procedure 'type string' (see Section 4.3).

advance film

```
procedure advance film;
```

This procedure advances the film on the selected cameras and increments the frame counts to be output in the Job Statistics on Output streams 12 and 13. If a cut mark is required then this is output on the new frame. If the frames are being numbered, then the word 'FRAME' followed by the frame number is output in the top left hand corner and the frame count incremented. The typing position is set to the centre of the character that would just fit into the top left hand corner of the new frame.

advance repeat

```
procedure advance repeat (N);
value N;
integer N;
```

The procedure acts differently depending on the value of N. If N = 0 then this defines the start of a set of SC 4020 orders to be output repeatedly. The first of the orders to be repeated will be placed at the start of a record on the IBM tape. If N > 0, then the intervening SC 4020 orders between the call of the procedure with N = 0 and the current call of 'advance repeat' will be output N times. As the 'advance repeat'order also acts as a frame advance, an integral number of frames will be repeated.

The number of records on the IBM tape that can be repeated in this fashion must be less than 511. Above this number there is likely to be an error on the SC 4020. If more than 510 records are repeated, a diagnostic is issued on stream 13 warning the user.

It is assumed that this procedure will be used for repeated frames of microfilm. Consequently no effort is made to correct the frame count and it is advisable only to use the procedure when neither a frame count nor a cut mark is required.

The hardware for this SC 4020 order is currently unavailable.

axes at

```
procedure axes at (X,Y);
value X,Y;
real  X,Y;
```

Axes are drawn across the selected region passing through the point (X,Y) defined in units of the selected region.

border

```
procedure border;
```

This draws a border around the selected region consisting of four lines along the sides of the region.

box

procedure box(X_1, Y_1, X_2, Y_2);
value X_1, Y_1, X_2, Y_2;
real X_1, Y_1, X_2, Y_2;

This draws lines along the sides of the rectangle which has (X_1, Y_1) and (X_2, Y_2) as its bottom left and top right hand corners. It is equivalent to:

vector (X_1, Y_1, X_2, Y_1);
vector (X_2, Y_1, X_2, Y_2);
vector (X_2, Y_2, X_1, Y_2);
vector (X_1, Y_2, X_1, Y_1);

broken

```
procedure broken (I);
value I;
integer I;
```

This procedure defines whether all vectors drawn in the currently selected region will be broken or not. If $I < 1$ or $I > 60$ then the vectors drawn will be unbroken. Otherwise all lines drawn will broken with the size of each broken segment being approximately I raster units in either the X or Y directions. The I raster units refer to the coordinate which is changing most rapidly. Therefore the length of segments increases as I increases but may vary slightly depending on the angle of the line with an axis. The status of line drawing in the selected region can be changed at any time by another call of the procedure 'broken'.

character factor

```
procedure character factor (N);
value N;
real  N;
```

The size of characters in the currently selected region are set to be N times as large as the font size. The only font available at the moment is font 0. This has font size equal to 8 raster units in height above and below the line and a width of 10 raster units.

character size

```
procedure  character size (HGHT, WDTH);
value  HGHT, WDTH;
real   HGHT, WDTH;
```

The height of characters in the currently selected region is set to HGHT units above and below the line on which the character sits. The width of the characters are set to WDTH.

character initialisation

```
procedure  character initialisation;
```

This is a code procedure which initialise the form of the GROATS software character set by loading the array 'character'. If non-standard meanings for characters have been introduced by calling 'setting char' or by redefinition, then the standard GROATS set can be reinitialised by calling this procedure.

character space

```
procedure  character space (LENGTH);
value  LENGTH;
real   LENGTH;
```

This procedure sets the spacing between characters in the currently selected region to be LENGTH units in the X-direction.

contain

```
procedure  contain;
```

The currently selected region is defined as contained. All subsequent plotting will be restricted to inside the region.

dark

```
procedure  dark (N);
value  N;
integer  N;
```

Lines in the currently selected region are drawn N times so that they will appear darker the larger the value of N used. If N ≤ 0 then the lines will be invisible.

define char as special

```
procedure  define char as special (N,J,I);
value  N,J,I;
integer  N,J,I;
```

The character having GROATS number N is redefined as a setting procedure and any appearance of this character in the string for the procedure 'type string' will cause the equivalent procedure to be executed. This action is independent of the currently selected region. The possible setting procedure are as follows:

J	SETTING PROCEDURE
1	displacement is (I);
2	displacement is (-I);
3	superfix

4	super return
5	suffix
6	suf return
7	type blanks (I)
8	type blanks (-I)
9	newlines (I)
10	newlines (-I)
11	move type to top
12	thick (I)
13	dark (I)

I must be positive or zero in the situations where it is used. In positions where I is not used, its value should be set to zero.

displacement is

```
procedure displacement is (N);
value N;
integer N;
```

The character displacement in the currently selected region is set to N. Any attempt to type or plot a character having GROATS number M will cause the typing or plotting of the character having GROATS number $N + M$. Care should be taken that $0 < N + M \leqslant 255$ for all characters used.

end plotting

```
procedure end plotting;
```

This procedure must be called by all jobs using the GROATS package once the graphical output is complete. It should not be called more than once for each run. Any plotting after the first call will be ignored. The procedure prints on output streams 12 and 13 the number of IBM records, frames of hardcopy, and frames of microfilm produced by the job. It outputs an identification frame at the end of the graphical output and inserts file marks on the IBM job. In the case of a job coming to an abnormal halt, an attempt is made to call 'end plotting' after the backward trace has been completed.

expand

```
procedure expand (X,Y,RATIO);
value X,Y,RATIO;
real X,Y,RATIO;
```

All subsequent graphical output calls for the selected region will be expanded by a factor RATIO about the point (X,Y). A reference to a point (x_1,y_1) will appear to be a reference to:

$$x = X + \text{RATION} \times (x_1 - X)$$

$$y = Y + \text{RATION} \times (y_1 - Y)$$

This expansion does not influence the definition of one region in terms of an expanded region. (The definition is as though no expansion had been defined.)

expand and rotate

```
procedure expand and rotate (X,Y,RATIO,ANGLE);
value X,Y,RATIO,ANGLE;
real X,Y,RATIO,ANGLE;
```

All subsequent graphical output calls for the selected region will be expanded by a factor RATIO about the point (X,Y) and also rotated ANGLE radians in an anti-clockwise direction about this point. A reference to a point (x_1, y_1) will appear to be a reference to:

$$x := X + \text{RATIO} \times \{ (x_1 - X) \cos(\text{ANGLE}) - (y_1 - Y) \sin(\text{ANGLE}) \}$$

$$y := Y + \text{RATIO} \times \{ (x_1 - X) \sin(\text{ANGLE}) + (y_1 - Y) \cos(\text{ANGLE}) \}$$

The expansion and rotation do not influence the definition of one region in terms of another. The definition takes place as though no expansion or rotation had been defined.

expand image

procedure expand image;

This procedure sets the SC 4020 in EXPAND IMAGE mode. The size of the hardcopy plotting area increases from $7\frac{1}{2}$" square to approximately 8" square. Its main use is for 35 mm microfilm. In EXPAND IMAGE mode, the SC 4020 is set up so that frames of 35 mm microfilm abut. This allows plots to extend continuously over several frames. If the SC 4020 is already in EXPAND IMAGE mode the procedure has no effect. The size of the hardware characters in EXPAND IMAGE mode are the same as in the normal REDUCE IMAGE mode. However the spacing between the hardware characters does increase.

expose heavy

procedure expose heavy;

The SC 4020 hardware characters can be defined in one of two intensities. The HEAVY mode is the standard and a fainter LIGHT mode can be set by calling 'expose light'. The HEAVY mode is reset by calling 'expose heavy'. These settings are global to the GROATS system and not defined for a particular region. The procedure does not effect the intensity of vectors.

expose light

procedure expose light;

The SC 4020 hardware characters can be defined in one of two intensities. The HEAVY mode is standard and a fainter LIGHT mode can be set by calling this procedure. The mode setting is global to the GROATS system and not defined for a particular region. The procedure does not effect the intensity of vectors.

extend

procedure extend;

This procedure sets the currently selected region to be in 'extend' rather than 'contain' mode. In the 'extend' mode, the region behaves only as a means of defining a coordinate system. It does not cut off line drawing when it reaches the edge of the region. Instead line drawing is only contained by the edges of the passive region.

finish saving

```
procedure finish saving (A,MAX);
integer MAX;
integer array A;
```

This procedure defines the end of a set of SC 4020 orders to be stored in the array A. The start of the set of orders is defined by a call of:

```
                    start saving  (A,MIN,M);
```

The SC 4020 orders are stored in A[MIN] to A[MAX-1]. The argument M defines the maximum array element of A allowed for storing the orders.

font

```
                    procedure  font (I);
                    value I;
                    integer I;
```

The character font selected is changed to font I.

frame

```
                    procedure  frame (I);
                    value I;
                    integer I;
```

If I > 0, this procedure defines the frame count to be output on the next frame in the top left hand corner. If I < 0 then no frame count will be output.

graticule at

```
                    procedure  graticule at  (DELTAX,DELTAY);
                    value  DELTAX,DELTAY;
                    real   DELTAX,DELTAY;
```

The currently selected region has lines drawn across it in the horizontal direction at intervals of DELTAY units and in the vertical direction at intervals of DELTAX units. The lines are positioned so that both horizontal and vertical lines pass through the origin (0,0).

hardware size

```
                    procedure  hardwaresize;
```

The software characters in the selected region are set to hardwaresize.

identification

```
                    procedure  identification;
```

The procedure 'identification' must be called before any graphical output is generated. Apart from producing identification consists of the following:

(1) The SC 4020 will be set to REDUCE IMAGE mode and the hardware characters will be produced in HEAVY mode.

(2) The user's graphical orders on the IBM tape will start at a new record.

(3) The frame count will be set to 2.

(4) Cut marks will be set to appear at the end of each frame.

(5) The plotting regions 0 to 10 will all be defined as the complete plotting area with scaling defined by:

limits (0,1023,1023,0);

The region 0 will be selected. Character size in all regions is set to hardware size. All vector drawing will be single unbroken lines. Axes will be used wherever possible and zero length vectors will not be plotted. All regions will be defined as contained.

(6) The heading:

```
* * * * * * * * * * * * * * * * * * * * * *
IBM TAPE TO SC 4020 PLOTTER
DONT RETURN TO COMMON
* * * * * * * * * * * * * * * * * * * * * *
```

is produced on output streams 12 and 13.

(7) The hardcopy camera is selected by itself.

(8) The region 0 is defined as the passive region.

The frames of identification output on both cameras consists of:

(1) The word CUT in large type with a dotted line at the bottom of the frame.

(2) An alignment frame consisting of a frame with diagonals and characters plotted at the intersections.

(3) An identification frame marked FRAME 0 with a cut mark. This frame is overprinted with the standard Atlas Laboratory form slide. Near the bottom of the frame is printed the word ALGOL and the Job Title followed by the time and date when the frame was produced. It also contains the user's job number in large type so that it can be read on microfilm output.

(4) Output is advanced to a new frame marked FRAME 1.

ignore rest of record

<u>procedure</u> ignore rest of record;

This procedure outputs to IBM tape the set of SC 4020 orders already waiting in the partially filled buffer. The next graphical output procedure will produce SC 4020 orders at the start of a new record.

increment type position

<u>procedure</u> increment type position;

The current typing position in the selected region is moved one position to the right. No checks are made to ensure that the typing position remains within the bounds of the selected region.

insert user form slide

<u>procedure</u> insert user form slide;

This procedure must be called by the user if he wishes to use his own form slide during a run. The procedure should be called after the call of 'identification' but before any calls of 'project form' requiring the user's own form slide. The message:

```
                    *************************************
                    USE SLIDE PROVIDED.  TAPE WILL STOP
                    *************************************
```

is output on Streams 12 and 13. An SC 4020 order is output at this point which will stop the SC 4020 to allow the operator to insert the user's form slide.

```
    is selected region

                    integer procedure  is selected region;
```

This procedure has as its value the number of the selected region.

```
    job time date
                    procedure  job time date (A,I);
                    value  I;
                    integer  I;
                    integer array  A;
```

This code procedure will store the current time (I = 1) or date (I = 2) in the array A from A[1] to A[5] as a string which could then be output by:

```
                    type string  (addr (A,1));
```

The format of the time and date is:

```
                    ab . cd . ef

where:      Date :-  ab = day    ;  cd = month    ;  ef = year
            Time :-  ab = hours  ;  cd = minutes  ;  ef = seconds
```

```
    job title
                    procedure  job title (A);
                    integer array  A;
```

The Job title (the line following the word 'JOB' in the Job Description) is stored as a string in A[1] onwards. The number of arrangements used depends on the length of the title. It will not exceed 41. The string could be output by calling:

```
                    type string (addr (A,1));
```

```
    limits
                    procedure  limits (XMN,YMN,XMX,YMX);
                    value  XMN,YMN,XMX,YMX;
                    real  XMN,YMN,XMX,YMX;
```

The procedure defines the user's units for the currently selected region. If the region has been defined by:

```
                    region (xmn,ymn,xmx,ymx,i)
```

where i is the selected region, then the point (XMN,YMN) in the selected region corresponds to the point (xmn,ymn) defined on the plotting area and, similarly, (XMX,YMX) corresponds to the point (xmx,ymx). It is assumed that the range is uniform so that the point (X,Y) corresponds to (x,y) on the plotting area where:

$$x = xmn + (X - XMN)\,\frac{(xmx - xmn)}{(XMX - XMN)}$$

$$y = ymn + (Y - YMN)\,\frac{(ymx - ymn)}{(YMX - YMN)}$$

The correspondence holds unless rotation or expansion is defined for the region. The limits for a region may be redefined at any time. There is no need for $XMN < XMX$, $YMN < YMX$, $xmn < xmx$, or $ymn < ymx$ so that reflections can be obtained when one or more of these relations are not satisfied.

lineprinter off

<u>procedure</u> lineprinter off;

The SC 4020 is reset back to the standard graphical mode after being in lineprinter simulation mode. The frame count and cut mark are reset to the values they had before entering the lineprinter simulation mode. The frame is advanced to a new frame. An estimate is made of the number of frames of hardcopy and microfilm that have been produced in lineprinter simulation mode and these totals are added to the counts.

lineprinter on

<u>procedure</u> lineprinter on;

The SC 4020 is set to lineprinter simulation mode after the frame has been advanced. All output commands destined for normal output streams will now be merged on the graphical output. Once in lineprinter simulation mode, the user should not under any circumstances call any GROATS procedures except 'lineprinter off' which returns him to standard graphical mode.

math region

<u>procedure</u> math region;

This procedure is equivalent to:

select region (1);

The procedure was added with the procedure 'screen region' for the user who did not require a large number of different regions. The procedure 'sceen region is equivalent to:

select region (0);

The user would define the units for the SC 4020 plotting area when screen region was selected and the units and position of his graphical output would be defined in math region.

mount 35 mm camera

<u>procedure</u> mount 35 mm camera;

The standard microfilm camera available at the Laboratory is a 16 mm camera. This tends to be more economical for most uses and it is hoped that this will be used wherever possible. However there will be occasions when users would prefer their output on 35 mm film. Textual information can be seen more clearly on 35 mm microfilm and also the ability to abut frames in 'expand image' mode may be required. Output on 35 mm microfilm is obtained by a different camera which must be inserted by the SC 4020 operator in place of the standard 16 mm camera. The procedure 'mount 35 mm camera' therefore prints on output streams 12 and 13 the message:

```
PLEASE USE 35 MM CAMERA
REMOUNT 16 MM CAMERA WHEN JOB IS FINISHED
```

and also inserts an instruction to stop the SC 4020 so that the camera can be changed. It is important that this procedure is called before the call of 'identification' so that the identification frames appear on the 35 mm film. To ensure this, if 'identification' has already been called the above procedure is ignored.

move type to

```
procedure move type to (x,y);
value x,y;
real x,y;
```

This procedure moves the typing position in the currently selected region to the point (x,y). The next character to be typed will be centred on this point.

move type to top

```
procedure move type to top;
```

This moves the typing position to the top left hand corner of the currently selected region so that the next character to be typed just resides within the region.

passive

```
procedure passive;
```

The currently selected region is defined as the passive region. All plotting will be scissored to be within the area of the passive region. It is essential that the limits of the passive region are completely within the plotting area.

pass to ibm tape

```
procedure pass to ibm tape (YES);
value YES;
Boolean YES;
```

Normally, if an IBM tape numbered 14 is defined in the Job Description, any graphical output produced by the program will be passed to the IBM tape as records to be output by the SC 4020. In debugging a program, it may be convenient to run pieces of program without actually requiring the graphical output. For example, if parts of a program are debugged, it may be that the graphical output from these sections can be ignored. The only output really required is from the untested parts of the program. The procedure 'pass to ibm tape' will stop graphical output being passed to the IBM tape if YES = <u>false</u> and will allow it to pass again if the procedure is called with YES = <u>true</u>.

As the graphical output is passed to the IBM tape in records of about 190 SC 4020 orders each, a buffer is used to contain a partially completed output record. The switch set above will be checked when the buffer is about to be passed to tape. Consequently it is possible to get the incomplete record output when it was not required and vice versa. This can always be avoided if the procedure:

```
ignore rest of record;
```

is called immediately before 'pass to ibm tape'. This will ensure that any uncomplete record is dealt with before the 'pass to ibm tape' switch is reset (see Section 6.3).

plot

```
procedure plot (x,y,N);
value x,y,N;
real  x,y;
integer N;
```

The GROATS character N is plotted at the point (x,y). The value of N is remembered so that subsequent plots of this character can be done by calling 'plotn' (see Section 4.2).

plotn

```
procedure plotn (x,y);
value x,y;
real  x,y;
```

This plots at the point (x,y) the character last plotted by a call of 'plot' or 'plotn' in any region unless 'set plot char' has been called in the meantime. In this case, the character plotted is the one defined by 'set plot char'.

pointer

```
integer procedure  pointer (STR);
string STR;
```

The result of this procedure is the address of the string STR stored as an integer. This value can be used as an argument to the procedure 'type string'.

print integer

```
procedure print integer (I, STREAM);
value I, STREAM;
integer I, STREAM;
```

The output produced on streams 12 and 13 is independent of the I/O package used. Consequently the same diagnostic information can be produced with each of the I/O packages. This is achieved by having the two I/O procedures 'print integer' and 'print tx' independent of the standard I/O. The procedure 'print integer' outputs a newline followed by the number I on stream STREAM. The original stream is reselected before returning. This procedure should be used for diagnostics in library programs.

print sc 4020 records

```
procedure print sc 4020 records (YES);
value YES;
Boolean YES;
```

As a debug aid, it is possible to print in octal the 36-bit SC 4020 instructions comprising a record before it is passed to the IBM tape. This is achieved by a call of 'print SC 4020 records' with YES = true. Printing can be turned off by calling it with YES = false. The format of the SC 4020 orders is given in Appendix 4.

print tx

```
procedure print tx (I, STREAM);
value I, STREAM;
integer I, STREAM;
```

This procedure together with 'print integer' should be used for diagnostics in library programs. The procedure is independent of the standard I/O packages and outputs on stream STREAM the string pointed at by I. The value of I is set by a call of 'pointer'. A newline is output after the string and the original output stream is reselected.

project form

```
                    procedure  project form;
```

This procedure will cause the form slide currently loaded on the SC 4020 to be superimposed on the current frame for the selected cameras. This allows background information for output to be produced very efficiently. A typical example is the background map on which information is plotted. The user's own form slide can be loaded by calling 'insert user form slide'. For details of how to obtain form slides, the user should contact P. Nelson (Ext. 406) at the Laboratory.

raster length

```
                    procedure  raster length (X,Y);
                    real  x,y;
```

The GROATS package has been designed so that the user need know as little about the characteristics of the SC 4020 as possible. In particular the fact that the SC 4020 has a raster 1024 x 1024 and that all line drawing occurs between raster coordinates is usually not known to the user. However, in certain applications, it will be necessary to know what the smallest finite increment in each direction is. The procedure 'raster length' returns to the user the length of a single raster position in each direction in his own units for the selected region. This would be necessary in, for example, a library routine attempting to draw smooth curves. The values of X and Y would determine the length of lines in the user's coordinates to achieve smoothness.

The procedure 'raster length' gives the values of X and Y assuming that no rotation or expansion of the region has been defined.

read string

```
                    procedure  read string (ARY,IMIN,IMAX,STR);
                    value  IMIN;
                    integer  IMIN,IMAX;
                    integer array  ARY;
                    string  STR;
```

This procedure is to allow strings to be read in from a data tape. By using the procedure 'addr', the strings can be output using 'type string' or 'print tx'. The string STR consists of two characters. All input before the appearance of the first of these is ignored. The characters after this character are inserted, two per word, into the one dimensional array ARY starting at ARY[IMIN]. The copying is terminated by the appearance of the second character of the string STR. The element ARY[IMAX] is the first unused element after the copied characters. When estimating the number of array positions occupied, the user should add one extra position which is used to contain the length of the string. The input buffer is left pointing at the character following the second character of STR on exit.

reduce image

```
                    procedure reduce image;
```

This procedure resets the SC 4020 to the standard REDUCE IMAGE mode. The size of the hardcopy plotting area is approximately 7½" square rather than the 8" square in EXPAND IMAGE mode.

region

```
                    procedure  region (XMN,YMN,XMX,YMX,I);
                    value  XMN,YMN,XMX,YMX,I;
                    real   XMN,YMN,XMX,YMX;
                    integer  I;
```

This procedure defines the position and size of the region I on the plotting area. The points (XMN,YMN) and (XMX,YMX) define the limits of the region. All points (X,Y) in the region have the value of X between XMN and XMX and the value of Y between YMN and YMX. The size and position of the region is defined in terms of the units of the currently selected region. The definition is independent of any rotation or expansion defined on the selected region and can be thought of as though these had been reset to their initial values before the definition. The position and size of the region can be redefined in terms of its original units. Currently regions 0 to 10 are allowed.

region and select

```
                    procedure  region and select (XMN,YMN,XMX,YMX,I);
                    value  XMN,YMN,XMX,YMX,I;
                    real   XMN,YMN,XMX,YMX;
                    integer  I;
```

In many cases, once the region has been defined, it is necessary to select the region so that the units for the region can be defined. The procedure 'region and select' amalgamates these two operations and is directly equivalent to:

```
                    region (XMN,YMN,XMX,YMX,I);
                    select region (I);
```

rotate

```
                    procedure  rotate (X,Y,ANGLE);
                    value  X,Y,ANGLE;
                    real   X,Y,ANGLE;
```

This procedure defines or redefines the rotation to be applied on all operations in the currently selected region. The rotation is about the point (X,Y) and in an anti-clockwise direction of ANGLE radians.

A region defined by:

```
                    region (XPMN,YPMN,XPMX,YPMX,I);
                    select region (I);
                    limits  (XMN,YMN,XMX,YMX);
                    rotate  (RX,RY,ANGLE);
```

would have the point (X,Y) in region I equivalent to (x,y) in the region in terms of which it was defined where:-

$$x = XPMN + (X' - XMIN)\,\frac{(XPMX - XPMN)}{(XMX - XMN)}$$

$$y = YPMN + (Y' - YMN)\,\frac{(YPMX - YPMN)}{(YMX - YMN)}$$

and

$$X' = RX + (X - RX)\cos(ANGLE) - (Y - RY)\sin(ANGLE)$$

$$Y' = RY + (X - RX)\sin(ANGLE) + (Y - RY)\cos(ANGLE)$$

The conversion is defined as though the rotation is applied first followed by the conversion of units to those of the plotting area. The conversion could have been defined in the opposite order with the rotation applied after the conversion. The two results are not always the same.

scales at

```
procedure  scales at (DELTAX,DELTAY,XORIG,YORIG,MX,NX,MY,NY);
value  DELTAX,DELTAY,XORIG,YORIG,MX,NX,MY,NY;
real   DELTAX,DELTAY,XORIG,YORIG;
integer  MX,NX,MY,NY;
```

The procedure is used to print the scales on a region. It is intended that the procedure will be used with the procedures 'graticule at' and 'axes at'. The point (XORIG,YORIG) can be thought of as defining axes and that the scales in the X direction would be printed just below the X-axis and in the Y direction to the left of the Y-axes. Scales are printed at intervals of DELTAX and DELTAY with the origin taken as (0,0) (see Section 7.1).

The scales are printed by calling

```
type number (A,M,N);
```

where M = MX, N = NX for the horizontal scaling and
M = MY, N = NY in the vertical direction.

screen region

```
procedure  screen region;
```

This is equivalent to:

```
select region (0);
```

and can be used together with procedure 'math region' to define a simpler system when not more than one sub region is required. It is just a shorthand.

select camera

```
procedure  select camera (I);
value  I;
integer  I;
```

This procedure defines the selected cameras from this point onwards until another call of 'select camera' takes place.

If I = 1, the microfilm camera is selected.

If I = 2, the hardcopy camera is selected.

If I = 3, both cameras are selected.

Any output after this command will be plotted on the selected devices.

I < 1 is equivalent to I = 1 while I > 3 is equivalent to I = 3.

select region

```
procedure select region (I);
value I;
integer I;
```

All plotting, typing, or line drawing output after the call of this procedure will be defined in terms of the units defined in region I. In addition the size of characters and typing position will be those defined for region I when it was last selected. Other parameters such as

1. Thickness of lines drawn.
2. Darkness of lines drawn.
3. Whether lines are broken.
4. Whether axes drawing commands are used for lines.
5. Whether shading is defined.
6. Whether zero length lines are drawn.
7. Whether region is contained.
8. Rotation and expansion.

are all reset to the values defined when the region was last selected.

set cutmark

```
procedure set cutmark (YES);
value YES;
Boolean YES;
```

If YES = true, a mark is made on the output at the bottom of the plotting area and out from it. This mark can be used to aid in cutting up hard-copy output. It is hoped eventually to provide some automatic means of cutting the output. Assuming a box drawn around the plotting area, the cutmark extends outward from the bottom right hand corner.

If YES = false, the cutmark is not output. The standard setting is for the output to have the cutmark. It would normally be turned off for microfilm output. The setting is independent of the region selected.

set plot char

```
procedure set plot char (N);
value N;
integer N;
```

All subsequent plotting orders 'plotn' will assume that the character to be plotted has GROATS number N. This setting is independent of the region selected.

shadow

```
procedure shadow (I);
value I;
integer I;
```

This procedure will cause an axes to be drawn from the start of each vector drawn in the selected region. The call of vector (x_1,y_1,x_2,y_2) will draw the line between (x_1,y_1) and (x_2,y_2) and also:

```
if I = 1  then  vector (x₁,y₁, xmin,y₁)
if I = 2  then  vector (x₁,y₁, xmax,y₁)
if I = 3  then  vector (x₁,y₁, x₁,ymin)
if I = 4  then  vector (x₁,y₁, x₁,ymax)
```

where the regions units are defined by:

```
limits (xmin,ymin,xmax,ymax);
```

If vectors are being drawn close together (in curve drawing for example) then the area to one side of the curve will appear shaded. If I = 0 no shading takes place.

start saving

```
procedure start saving (A,IST,IMX);
value IST,IMX;
integer IST,IMX;
integer array A;
```

Production of graphical output on Atlas is expensive. If the same output is to be produced on several frames then it is better to store the computed output rather than recalculate it. The procedure 'start saving' will initialise the output procedures so that, as well as producing graphical output, the graphical output is stored in the array A starting at A[IST]. The section of graphical output is concluded by calling 'finish saving'. It can be reused by calling 'use saved'. SC 4020 instructions are stored one per array element. The value IMX is defined so that graphical output is only stored in the array elements between A[IST] and A[IMX].

suffix
sufreturn
superfix
super return

```
procedure suffix;
procedure sufreturn;
procedure superfix;
procedure super return;
```

These four procedures are used to define superfixes and suffices. The action of 'superfix' is to reduce the size of the characters and move the typing position upwards. The procedure 'super return' resets the typing position and character size to its original value. Similarly 'suffix' reduces the size of the characters and moves the typing position downwards. The procedure 'sufreturn' resets the typing position and character size to its original value. The procedures can be intermingled and used to several depths if desired.

thick

```
procedure thick (N);
value N;
integer N;
```

After a call of the procedure 'thick', all lines drawn in the selected region will consist of the desired line together with N - 1 lines parallel to it and touching it on either side. This is particularly useful on microfilm titling as it enables a much thicker software character to be obtained. The initial setting for all regions is with N = 1.

topoint

```
procedure  topoint ($X_2$,$Y_2$)
value  $X_2$,$Y_2$;
real   $X_2$,$Y_2$;
```

This is equivalent to:

vector (LASTX,LASTY,X_2,Y_2);

where (LASTX,LASTY) are the coordinates of the end of the previous line drawing command. The procedure is particularly useful when drawing a set of connected lines. The position (LASTX,LASTY) is independent of the region selected so that care must be taken that no vector drawing occurs in a different region inbetween calls of 'topoint'.

type

```
procedure  type (N);
value  N;
integer  N;
```

The GROATS character N is typed at the current typing point. This is then incremented.

type blanks

```
procedure  type blanks (I);
value  I;
integer  I;
```

This procedure moves the typing position in the selected region I places to the right.

type newlines

```
procedure  type newlines (I);
value  I;
integer  I;
```

The procedure resets the typing position to the left hand edge of the selected region and advances the typing position I newlines in the y-direction.

type number

```
procedure  type number (X,M,N);
value  X,M,N;
real  X;
integer  M,N;
```

This procedure types the value of the variable X at the current typing position of the selected region. The format of the number depends on M and N:-

If $M = 0$ and $N \neq 0$ a standardised floating number of the form $D_{\alpha} E$ is printed with N decimal places in the mantissa $D(1 \leq D \leq 10)$ and a three digit exponent E, occupying $N+8$ character positions in all.

If $M \neq 0$ and $N \neq 0$, the number is printed in fixed point form with M figures before the decimal point and N figures after the point, occupying $M+N+2$ character positions in all.

If $M \neq 0$ and $N = 0$ the number is printed as an integer occupying $M+1$ character positions in all. The procedure rounds off the decimal form of the number, suppresses non-significant zeros and prints negative signs only, immediately preceding the first printed digit. If a number will not fit the specified field, further character positions will be used to accommodate it.

type string

```
procedure type string (I);
value I;
integer I;
```

This procedure types the string pointed at by I at the current typing position of the selected region. The size and spacing of characters is as defined for the region selected. If 'hardware' size is specified then hardware characters will be used wherever possible. If the selected region is 'contained' then any characters which would have appeared outside the selected region will be ignored. The pointer I to a string can be obtained from either the procedure 'pointer' or 'addr'. If characters have been redefined as setting procedures by 'define char as special' then the relevant procedure will be called instead of typing the character.

use axes

```
procedure use axes;
```

Lines on the SC 4020 can either be drawn by vector or axes drawing commands. The axes drawing commands can produce horizontal or vertical lines across the complete plotting area while vectors, although able to be drawn in any direction between raster positions, cannot be longer than 1/16 of the plotting area's width or height. The line drawing procedures in the GROATS package will attempt to use axes commands for efficiency wherever possible. It is possible that the axes will be fainter than the vectors and the user may request all lines to be drawn as vectors by calling:

```
use vectors;
```

A return to using axes is then achieved by calling:

```
use axes;
```

It is important to remember that vector drawing is much less efficient and 'use vectors' should not be called unless really necessary.

use saved

```
procedure use saved (A,I,J);
value I,J;
integer I,J;
integer array A;
```

SC 4020 output stored in array A between A[I] and A[J] by calling 'start saving' and 'finish saving' can be plotted again by calling 'use saved'. The procedure attempts to update the frame counts correctly. It may however be impossible to do this if the cameras selected are different when the output is used from those set when it was defined. In this case a comment is made on stream 12 for the SC 4020 operator.

use vectors

```
                procedure  use vectors;
```

See procedure 'use axes'.

vector

```
                procedure  vector (X1,Y1,X2,Y2);
                value  X1,Y1,X2,Y2;
                real   X1,Y1,X2,Y2;
```

This is the major line drawing procedure which all other procedures call. It draws a line between (X_1,Y_1) and (X_2,Y_2). If the selected region is contained then only the part of the line interior to the selected region will be drawn. In any case only the part of the line interior to the passive region will be drawn. Such parameters as:

1. Whether to use axes or vectors.
2. How dark to make the line.
3. How thick to make the line.
4. Should zero length lines be plotted.

are all dealt with by vector.

zero invisible

```
                procedure  zero invisible;
```

After this procedure call, all lines in the selected region of zero length will not be drawn. This is the initial setting for all regions.

zero seen

```
                procedure  zero seen;
```

After this procedure call, all lines in the selected region of zero length will be plotted. They will appear as a dot on the graphical output.

9. SC 4020 DESCRIPTION

9.1 INTRODUCTION

The SC 4020 Computer Recorder, designed and manufactured by Stromberg-Carlson Corporation, is a sophisticated microfilm and hard copy plotter which can be used as either on-line or off-line peripheral equipment for a computer. The most obvious use of the SC 4020 is for the rapid production of graphical output. A less obvious application is the use of the SC 4020 as a high-speed printer. Printed records can be obtained on 35mm or 16mm film and this greatly reduces the amount of space required to store the records. This chapter describes the SC 4020's mode of operation at the Laboratory as an off-line peripheral.

9.2 MACHINE DESCRIPTION

The basic components of the SC 4020 are shown in Fig. 9.1.

These are as follows:

1. An IBM 729 IV tape deck capable of reading six channel digital tapes at 200 and 556 bits per inch.

2. A tape adapter which accepts data from magnetic tape at up to 90,000 characters per second. The 6-bit information tracks from the IBM tape are assembled into 36-bit SC 4020 instructions.

3. The Charactron Shaped Beam Tube. This is a special cathode ray tube upon which are displayed vectors, axes and characters. The Charactron tube is the most important part of the SC 4020.

4. Forms Projector. It is possible to illuminate a microfilm slide, called a form slide, by switching on for a short period the light behind it. The light deflection system allows light from both the form slide and the Charactron tube to impinge on the two characters.

5. Microfilm camera. Two cameras are available, a 16mm and a 35mm. Both are sprocket controlled and the operator can manually load one or other of the cameras. The film advance and shutter control of the cameras are controlled by SC 4020 instructions.

6. Hardcopy camera. The hard copy camera is similar in operation to the microfilm camera. The photorecording paper must be developed before viewing.

The operation of an SC 4020 camera is basically different form that of a standard camera in that an SC 4020 camera tends to have its shutter open most of the time. An SC 4020 instruction requesting a line to be drawn on the charactron tube will cause the line to appear for just sufficient time that the film in the camera (with its shutter open) will become sensitised. A complete scene on a microfilm frame is therefore built up by a large number of lines and characters appearing on the charactron tube face for short periods of time. At no time is the complete scene available on the tube face.

9.3 THE CHARACTRON SHAPED BEAM TUBE

A diagram of the tube is shown in Fig. 9.2. As far as line drawing is concerned, the charactron tube behaves very similar to the standard

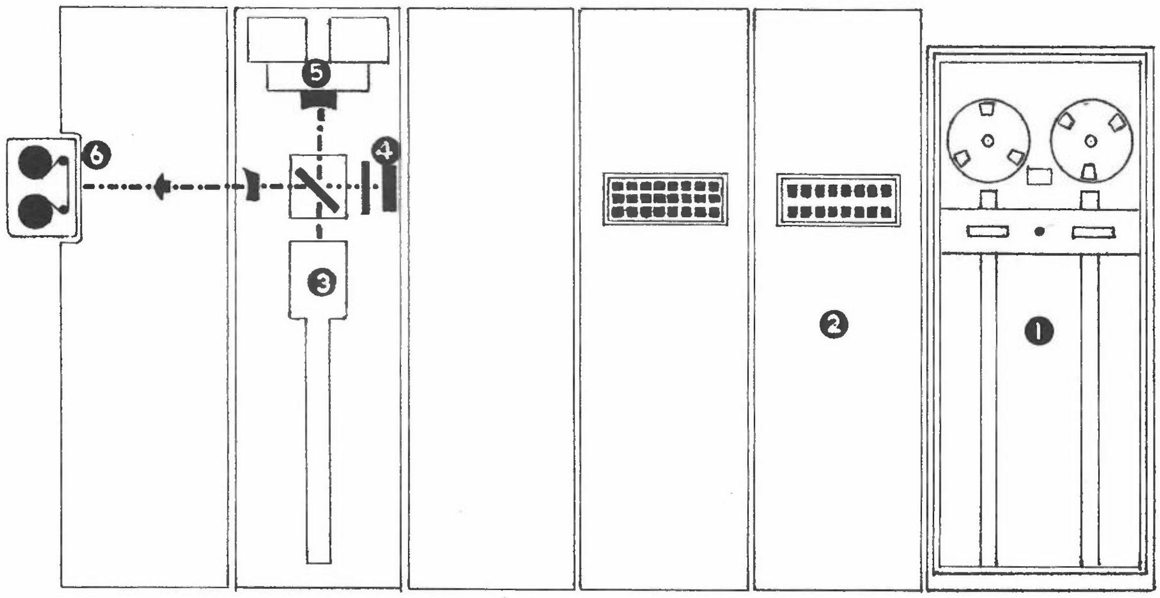


Fig 9 · 1. Diagramatic View of the S C 4020

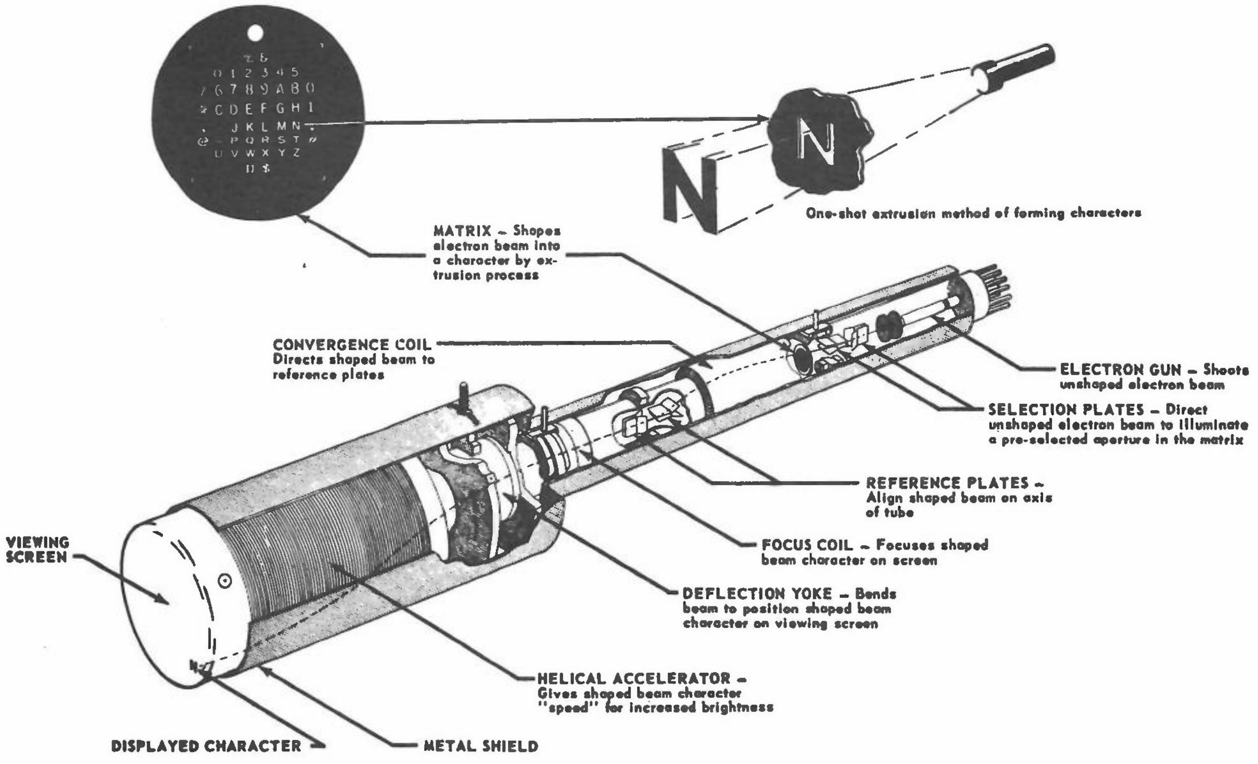

One-shot extrusion method of forming characters
MATRIX – Shapes electron beam into a character by extrusion process
CONVERGENCE COIL Directs shaped beam to reference plates
ELECTRON GUN – Shoots unshaped electron beam
SELECTION PLATES – Direct unshaped electron beam to illuminate a pre-selected aperture in the matrix
REFERENCE PLATES – Align shaped beam on axis of tube
FOCUS COIL – Focuses shaped beam character on screen
DEFLECTION YOKE – Bends beam to position shaped beam character on viewing screen
VIEWING SCREEN
HELICAL ACCELERATOR – Gives shaped beam character "speed" for increased brightness
DISPLAYED CHARACTER
METAL SHIELD

Fig 9 · 2. Charactron Shaped Beam Tube 7 - inch type cathode ray tube. Fig. 9.2 shows how a character is projected on the viewing screen. The electron gun shoots an unshaped electronic beam towards the viewing screen. The beam passes through selection plates which deflect the unshaped electronic beam through the particular matrix aperture for the desired character. As the beam leaves the matrix, it has been shaped into the form of the character aperture through which it has passed. This shaped beam is then accelerated and deflected to the desired position on the tube face. The character matrix is a small thin alloy disc on which 64 character shaped openings are engraved in an 8 x 8 array. The whole matrix is less than $\frac{1}{4}$" square.

As far as line drawing is concerned, the screen is divided into 1024 x 1024 positions and the beam can be deflected so that a line can be produced between any two of the raster positions. Each charactron character occupies approximately 6 horizontal by 9 vertical raster positions on the plotting area of the tube face.

The complete charactron tube assembly can be manually moved through 90°, 180°, 270°. Its standard setting is the CINE position shown in Fig. 9.3. The other positions of the tube are also shown in Fig. 9.3. This manual movement of the tube is not a trivial operation. Consequently plotting at the Laboratory will always be in CINE mode unless another tube position is specially requested.

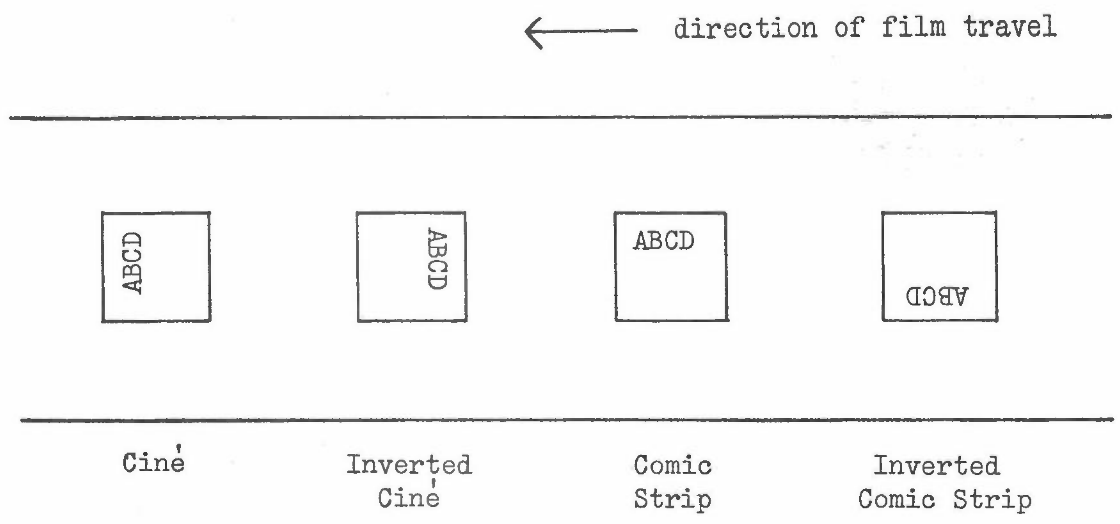


Fig 9·3

9.4 MICROFILM CAMERAS

The 35mm and 16mm ciné cameras are available at the Laboratory. The 16mm camera will normally be mounted. This uses perforated film and is ideal for producing movies direct from the SC 4020. The 35mm camera also uses perforated film. The film can be used in aperture cards or 35mm roll viewers. The image is compatible with standard 35mm movie projectors. The 35mm camera operates at speeds up to 10 frames per second and is supplied with two take-up and two supply magazines. As the Laboratory has no facilities for film developing, this is being done at the Rutherford Laboratory.

9.5 HARD COPY CAMERA

The hard copy camera makes it possible to record the displayed information on photo recording paper. The paper is in a continuous roll approximately $9\frac{3}{8}$" in width. The 4" square display on the charactron tube is recorded as a $7\frac{1}{2}$" square on the paper. The standard pull down between frames is about $9\frac{1}{2}$". The magazine holds about 350 feet of paper. Once a suitable amount of output has been produced on the SC 4020, the hard copy magazine is removed and developed off-line of the SC 4020.

9.6 FORM PROJECTOR

This feature allows the SC 4020 to superimpose fixed data on the variable data generated by the charactron shaped beam tube. The fixed data is produced on a glass slide (for example maps and drawing formats). The glass slide is then mounted in a slide holder assembly which is inserted into the form projector. A strobe light located within the form projector is used to illuminate the glass slide. The strobe light is flashed under program control to cause the slide data to be superimposed on the information coming from the charactron tube. Alternatively, by energising a special AUTO-FORM PROJECT switch, a form may be projected automatically for each frame advance.

9.7 IBM TAPE DECK

The IBM 729 IV tape deck attached to the SC 4020 is similar to the IBM tape decks on Atlas. Graphical output from the GROATS package is stored on IBM magnetic tape as SC 4020 orders. This tape is then removed from Atlas and remounted on the SC 4020. The SC 4020 tape adapter then decodes the information and produces the desired output. Output is normally produced in odd parity mode. However the SC 4020 is also capable of accepting even parity BCD records. Even parity records are assumed to be BCD information to be output as characters and such records are typed with automatic line feeds and page throws inserted between frames.

9.8 SC 4020 INSTRUCTION SET

Appendix 4 defines the SC 4020 order code and the format taken by the 36 bit instructions. The raster positions run from 0 to 1023 in both directions. The origin (0,0) is in the upper left hand corner with X and Y values increasing to the right and down respectively. The SC 4020 instructions refer to these raster positions. The following section will describe each of the SC 4020 operations. The opcode of each instruction is given in brackets behind the instructions name. The names in the next section are those used in Appendix 4.

9.8.1 Control Operations

ADVANCE FILM (46)

The ADVANCE FILM command causes the film in the camera (or cameras) selected to be advanced one frame.

RESET (56)

The RESET command performs simultaneously the function of ADVANCE FILM, STOP TYPE and EXPOSE HEAVY.

SELECT CAMERA (41, 42, 43)

These three instructions open the shutter of the camera or cameras designated and close the shutters of any camera not selected. The instructions are SELECT MICROFILM CAMERA (41), SELECT HARDCOPY CAMERA (42), and SELECT BOTH CAMERAS (43).

EXPAND IMAGE (44)

The EXPAND IMAGE instruction expands the size of the image on the charactron tube face. The exact size is adjusted so that successive frames of 35mm microfilm will abut to allow plots to extend continuously through several adjacent frames. The number of raster positions and the size of the hardware characters remain the same. The expansion therefore causes the space between the hardware characters to be increased in EXPAND IMAGE mode. The image remains expanded until the REDUCE IMAGE command is given.

REDUCE IMAGE (45)

The REDUCE IMAGE command causes the image to be reduced to 4" square on the charactron tube and to remain at 4" until another expand image command is given.

PROJECT FORM (50)

The information on the form slide is superimposed on the current frames of the currently selected cameras. This is done by illuminating the slide for a short period.

9.8.2 Plotting Instructions

PLOT(00)
EXPOSE LIGHT(04)
EXPOSE HEAVY(02)

These 3 instructions are similar in that they plot the hardware character with SC 4020 number C centred on the raster position at (X,Y) (see Appendices 1 and 4). The instruction PLOT operates in the density mode to which the SC 4020 has been set. The other two instructions reset the intensity before plotting the character. All plotting or typing subsequent to the EXPOSE LIGHT or EXPOSE HEAVY instructions is done in the intensity established. Changes to the intensity can only be achieved by another of these instructions or alternatively the RESET instruction.

9.8.3 Line drawing

DRAW VECTOR (6*)

This instruction causes a line to be drawn between the point (X,Y) and the point:

$$(X + (2SX - 1)\ SX,\quad Y - (2SY - 1)\ DY)$$

where SX, SY are single bits taking either the value 0 or 1.

GENERATE X-AXIS (30)

This instruction generates an axis parallel to the X-axis starting at (X,Y) and extending to (I,Y) where I = complement(S). The point (I,Y) must be at least 64 raster positions to the right of the point (X,Y).

GENERATE Y-AXIS (32)

This instruction generates an axis parallel to the Y-axis starting at (X,Y) and extending upwards to (X,S). The two points must be at least 64 raster points apart.

9.8.4 Typing Instructions

For all the operations given above, the SC 4020 accepts 36 bit instructions one at a time, decodes them and produces the required action. This is the normal mode of working. However there is also a typewriter mode of working. Once this mode has been selected, hardware characters are generated more economically as far as both time and space is concerned. The typewriter mode is initiated by either of the two cammands TYPE SPECIFIED POINT or TYPE CURRENT·POINT.

TYPE SPECIFIED POINT (20)

The SC 4020 character, numbered C, is typed at the raster position (X,Y). In addition the 'typing position' is moved to the next character position. In the typewriter mode, the raster grid of 1024 x 1024 positions may be regarded as consisting of 64 lines of 128 typing positions. Once typing mode has been entered by the TYPE SPECIFIED POINT instruction, characters are typed across the line until the 128th space has been filled or a CARRIAGE RETURN character is typed. In either case, typing continues on the next lower line at the extreme left of the grid (X = 0, Y = Y + 16). An ADVANCE FILM instruction must be given after the 64th line has been typed. Otherwise typing will continue on the first line of the same frame at X = 0, Y = 0 and will be superimposed on any information already there.

After the SC 4020 has entered typewriting mode, each subsequent instruction of 36 bits is divided up into six 6-bit fields as shown in Format 7 (Appendix 4). Each C_i denotes an SC 4020 character which is output at the next typing position. The 'typing position' is then incremented. Each 36-bit instruction in typewriting mode therefore types 6 characters. The characters C_i can be any of the characters defined in Appendix 1 except that the character codes 12, 52 and 56 octal do not type characters but act as carriage and mode controlling operations.

C_i = 12 will cause the rest of the instruction to be ignored and returns the SC 4020 from typewriter mode to normal mode.

C_i = 52 will cause the carriage control to move to the next line (X = 0, Y = Y + 16).

C_i = 56 is similar to the RESET instruction. Any remaining characters in the word are ignored.

As the character 12 can also appear in the first character position, this could be considered as a special SC 4020 order for resetting the normal mode of operation. This is given the name STOP TYPE (see RESET).

TYPE CURRENT POINT (22)

This is another instruction which changes the mode of operation to typewriting mode. Instead of defining the next typing position as in TYPE SPECIFIED POINT, the typing position is defined as the position specified by the last point plotted or typed or the origin of the last vector or axis drawn. This instruction has little value in a high-level graphical system. Once the typewriting mode has been entered, the characters C_1 to C_5 are typed.

9.8.5 Lineprinter simulation mode

If at any time the SC 4020 encounters a record on the IBM tape which is even parity instead of the normal odd parity then it enters a completely distinct mode of operation called PRINT MODE. In PRINT MODE, the SC 4020 can be used for producing output equivalent to standard lineprinter output produced from BCD coded even parity files on a computer such as the IBM 1401. Within the PRINT MODE there are two sub-modes called PRINT NORMAL MODE and PRINT LIST MODE. The sub-mode can be selected by the SC 4020 operator. At the Laboratory, the SC 4020 will normally be set to PRINT NORMAL MODE.

PRINT NORMAL MODE

In this mode the first character of each record is interpreted as a control code to allow carriage control commands to be fulfilled. The remaining 6-bit characters in the record are converted as shown in Appendix 7. The second character in the record being printed in the first character position of the line. The possible control characters in the first character position are:-

01	Page Throw
20	Single newline
12	Double newline
52	Carriage return
56	Typewriter reset, advances frame and sets to EXPOSE HEAVY. Stays in Print Mode.

When the 64th line of a page is reached, the SC 4020 executes an automatic advance frame on the next carriage return. In positions other than the first, the BCD code 12 will cause the remainder of the current record to be ignored. Similarly the BCD code 56 will be equivalent to the RESET operation in typewriter mode except that the SC 4020 will stay in PRINT MODE. In PRINT NORMAL MODE records of up to 132 characters can be processed.

PRINT LIST MODE

In this mode, which can be selected by the SC 4020 operator, the first character of the record is not treated as a control character. In PRINT LIST MODE, an automatic carriage return is executed upon reaching the 128th character of the line. The first character of the record is printed in the first character position of the line. Automatic page advance occurs as in PRINT NORMAL MODE.

10. ACKNOWLEDGEMENT AND REFERENCES

I would like to thank Stromberg-Carlson for permission to use diagrams appearing in the Programmer's Reference Manual for the SC 4020.

The main facilities available in the GROATS system were developed after reading the literature describing the following systems:

1. The Atlas Laboratory Graph package in the HARTRAN system.
2. The GHOST system of the Culham Laboratory.
3. The N.P.L. Graphic system.
4. The P.I.B. system of the Polytechnic Institute of Brooklyn (described in the Proceedings of the 6th UAIDE conference).
5. Systems available at Bell Telephone Laboratories.

The author has borrowed ideas from each of these systems. The main influence has been the P.I.B. system. The list of references following are particularly applicable to graphic output systems.

1. Proceedings of the 4th, 5th, 6th, 7th U.A.I.D.E. (Users of Automatic Information Display Equipment) conferences (1965-8)

2. 'User's guide to the Culham Computer Graphical Output System' by F.M. Larkin. 1967. H.M.S.O.

3. 'User's guide to the N.P.L. Graphical Output System' by B.R. Heap and C.W. Nott. 1968. NPL Report No. MA63

4. Atlas Computer Laboratory, SC 4020 Plotter Manual. 1968

APPENDIX 1

SC 4020 HARDWARE CHARACTER SET

SC 4020 internal number for character in octal	Character	GROATS number for character
00	0	16
01	1	17
02	2	18
03	3	19
04	4	20
05	5	21
06	6	22
07	7	23
10	8	24
11	9	25
12	∂	
13	=	28
14	"	2
15	'	32
16	δ	93
17	α	90
20	+	29
21	A	33
22	B	34
23	C	35
24	D	36
25	E	37
26	F	38
27	G	39
30	H	40
31	I	41
32	π	11
33	.(full stop)	31
34	)	9
35	β	91
36	±	3
37	?	12
40	-	30
41	J	42
42	K	43
43	L	44
44	M	45
45	N	46
46	O	47
47	P	48
50	Q	49
51	R	50
52	·(plotting dot)	95
53	$	75
54	*	14
55	γ	92
56	~	
57	d	4
60	Space	1
61	/	15
62	S	51

63	T	52
64	U	53
65	V	54
66	W	55
67	X	56
70	Y	57
71	Z	58
72	o	94
73	,	10
74	(	8
75	∫	5
76	Σ	6
77	□	63

N.B. The two codes 12 and 56 have special meanings on the SC 4020 and are not allowed in the GROATS system. The plotting dot is also not available in typing commands.

The 64 characters are arranged in an 8 x 8 matrix where the first octal digit of the SC 4020 internal number for the character defines the y position and the second octal digit defines the x position in the matrix.

APPENDIX 2

GROATS SOFTWARE CHARACTER SET

GROATS Number	Character	SC 4020 Hardware Number in Octal
0		
1	Space	60
2	"	14
3	±	36
4	$\bar{d}$	57
5	∫	75
6	Σ	76
7		
8	(	74
9	)	34
10	,	73
11	π	32
12	?	37
13	&	
14	*	54
15	/	61
16	o	0
17	1	1
18	2	2
19	3	3
20	4	4
21	5	5
22	6	6
23	7	7
24	8	10
25	9	11
26	<	
27	>	
28	=	13
29	+	20
30	-	40
31	.(full stop)	33
32	'	15
33	A	21
34	B	22
35	C	23
36	D	24
37	E	25
38	F	26
39	G	27
40	H	30
41	I	31
42	J	41
43	K	42
44	L	43
45	M	44
46	N	45
47	O	46
48	P	47
49	Q	50
50	R	51
51	S	62
52	T	63

53	U	64
54	V	65
55	W	66
56	X	67
57	Y	70
58	Z	71
59	≤	
60	∧	
61	∨	
62	≡	
63	□	77
64	⊃	
65	Space	
66	!	
67	£	
68	↑	
69	#	
70	%	
71	←	
72	\	
73	::=	
74	~	
75	$\cancel{\delta}$	53
76	√	
77	∇	
78	;	
79	:	
80	φ	
81	[	
82	]	
83	→	
84	≥	
85	≠	
86	‾	
87	|	
88	$_2$	
89	≃	
90	α	17
91	β	35
92	$\frac{1}{2}$	
93	$_{10}$	
94	$_{11}$	
95	·(Plotting dot)	52
96	@	
97	a	
98	b	
99	c	
100	d	
101	e	
102	f	
103	g	
104	h	
105	i	
106	j	
107	k	
108	l	
109	m	
110	n	
111	o	
112	p	

113	q	
114	r	
115	s	
116	t	
117	u	
118	v	
119	w	
120	x	
121	y	
122	z	
123	=>	
124	<=	
125	⌉	
126	]	
127	□	77
128	(Newline)	
129	(Top of page) Space	
130	⊂	
131	⊃	
132	∀	
133	<	
134	>	
135	‖	
136	⊥	
137	∩	
138	∪	
139	°	72
140	↓	
141	÷	
142	‘	
143	’	
144	◬	
145	⛛	
146	⊡	
147	◈	
148	⊙	
149	⊞	
150	⊠	
151	⟐	
152	❖	
153	✠	
154	⌖	
155	⌑	
156	✛	
157	✳	
158	∠	
159	∞	
160		
161	Å	
162	B	
163	Γ	
164	Δ	
165	E	
166	Φ	
167	Γ	
168	H	
169	I	
170	Θ	
171	K	

172	Λ	
173	M	
174	N	
175	O	
176	Π	
177	Ξ	
178	P	
179	Σ	76
180	T	
181	$\underline{\Upsilon}$	
182	$\underline{\Psi}$	
183	Ω	
184	X	
185	Y	
186	Z	
187	⊳	
188	⊲	
189	≐	
190	⊢	
191	⊣	
192		
193	Space	
194	{	
195	}	
196		
197		
198		
199		
200		
201		
202		
203		
204		
205		
206		
207		
208		
209		
210		
211		
212		
213		
214		
215		
216		
217		
218		
219		
220		
221		
222		
223		
224		
225	α	17
226	β	35
227	γ	55
228	δ	16
229	ε	
230	ϕ	
231	γ	55
232	η	

233	ι	
234	θ	
235	κ	
236	λ	
237	μ	
238	ν	
239	ο	
240	π	32
241	ξ	
242	ρ	
243	σ	
244	τ	
245	ν	
246	ψ	
247	ω	
248	χ	
249	υ	
250	ζ	
251		
252		
253		
254		
255		

APPENDIX 3

VARIABLES DECLARED BY GROATS IN OUTER BLOCK

VARIABLE NAME	TYPE
actve	integer
bparam1	Boolean
bparam2	Boolean
bparam3	Boolean
bparam4	Boolean
bparam 5	Boolean
camera	integer
camera3516	integer
character	integer array [0:3,0:255]
charmax	integer
contan	Boolean
cosrotangle	real
cutmark	Boolean
darkvar	integer
displacement	integer
dotted	integer
draw axes	Boolean
expansion factor	real
factor char	real
factor defined	Boolean
first trap	Boolean
font centre x	integer array [0:3]
font centre y	integer array [0:3]
font height	integer array [0:3]
font var	integer
font width	integer array [0:3]
frame count	integer
hardware	Boolean
hardware space	Boolean
hc frames	integer
height char	real
invisible	Boolean
iparam1	integer
iparam2	integer
iparam3	integer
iparam4	integer
iparam5	integer
last plot	integer
last x	real
last y	real
lpsimon	Boolean

max save addr	integer
mf frames	integer
passve	integer
rotorigx	real
rotorigy	real
rparam1	real
rparam2	real
rparam3	real
rparam4	real
rparam5	real
rparam6	real
rparam7	real
rparam8	real
save addr	integer
save start	integer
sc bf length	integer
sc bf pointer	integer
sc buffer	integer array [1:170]
scontan	Boolean array
scos rotangle	real array
sc records	integer
sdark var	real array
sdisplacement	integer array
sdotted	real array
sdraw axes	Boolean array
selected region	integer
sexpansion factor	real array
sfactor char	real array
sfactor defined	Boolean array
sfont var	integer array
shadowvar	integer
shardware	Boolean array
shardware space	Boolean array
sheight char	real array
shift6	integer
shift8	integer
shift10	integer
shift12	integer
shift14	integer
shift18	integer
shift30	integer

signx	real
signy	real
sin rotangle	real
sinvisible	Boolean array
soft char defn	integer array [1:1000]
soft max	integer
soft pointer	integer
space char	real
srot origx	real array
sshadow var	integer array
ssignx	real array
ssigny	real array
ssin rotangle	real array
sspace char	real array
star string	integer
sthickvar	real array
supsuf	real
swidthchar	real array
swch	integer
swndow	Boolean array
sxmax	real array
sxmin	real array
sxpmax	real array
sxpmin	real array
sxtype	real array
symax	real array
symin	real array
sypmax	real array
sypmin	real array
sytype	real array
thickvar	integer
to be printed	Boolean
to ibm tape	Boolean
widthchar	real
wndow	Boolean
xcharcntr	real
xmax	real
xmin	real
xpmax	real
xpmin	real

xtype	real
ycharcntr	real
ymax	real
ymin	real
ypmax	real
ypmin	real
ytype	real

Array dimensions are [-2:10] when not given.

PROCEDURES DECLARED IN OUTER BLOCK FOR GROATSICT

abs
active
addr
advance
advance film
advance repeat
arctan
axes at
border
box
broken
character factor
character initialisation
character size
character space
contain
convert
cos
dark
define char as special
displacement is
draw vector
end plotting
entier
error trap
exp
expand
expand and rotate
expand image
expose heavy
expose light
extend
file marks
finish saving
font
frame
graticule at
hardware size
identification
ignore rest of record

increment type position
initialise regions
insert user form slide
inview
is selected region
job time date
job title
limits
lineprinter off
lineprinter on
lineprinter sim
ln
math region
mount 35mm camera
move type to
move type to top
newline
out buffer
output paper throw
passive
pass to ibm tape
plot
plot hard
plot n
pointer
print
print integer
print SC.4020 records
print tx
project form
raster length
read
read Boolean
read string
reduce image
region
region and select
rotate
scales at
screen region
see vector

select camera
select input
select output
select region
set cutmark
set plot char
setting char
shadow
shield
sign
sin
soft char defns
space
sqrt
start saving
stop sc 4020
stop typing
suffix
sufreturn
super fix
super return
tape 14
thick
to buffer
to point
trap to
truncate to job no
type
type blanks
type hard
type newlines
type number
type or plot char
type string
use axes
user info
use saved
use vectors
vector
window

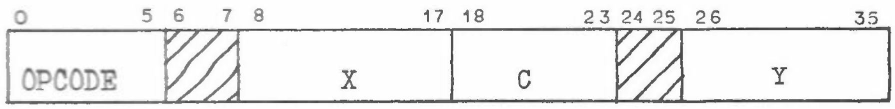
write Boolean
write text
zero invisible
zero seen

APPENDIX 4

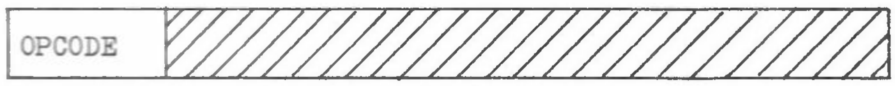
SC 4020 ORDER CODE

Opcode in octal	Format	Name	Operation
00	1	PLOT	Plot C at (X,Y)
02	1	EXPOSE HEAVY	Set 'expose heavy mode', plot C at (X,Y)
04	1	EXPOSE LIGHT	Set 'expose light mode', plot C at (X,Y)
07	2	IGNORE REST OF RECORD	
12	2	STOP TYPE	Return to plotting mode
20	1	TYPE SPECIFIED POINT	Type C at (X,Y), enter typing mode
22	6	TYPE CURRENT POINT	Type C_i from current point, enter typing mode
30	3	GENERATE X AXIS	vector (X, Y, I, Y) where I = complement(s)
32	3	GENERATE Y AXIS	vector (X, Y, X, S)
37	2	STOP SC 4020	
41	2	SELECT MF CAMERA	
42	2	SELECT HC CAMERA	
43	2	SELECT BOTH CAMERAS	
44	2	EXPAND IMAGE	
45	2	REDUCE IMAGE	
46	4	ADVANCE FILM	Advance film if I = 0
46	4	ADVANCE REPEAT	Advance and repeat N times if I = 1
50	2	PROJECT FORM	
56	2	RESET	Equivalent to Advance Film, Stop typing, Expose heavy
6	6	DRAW VECTOR	vector (X, Y, X + SX * DX, Y - SY * DY)

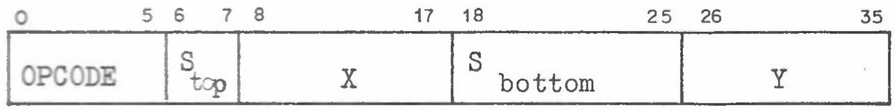
FORMAT TYPE 1



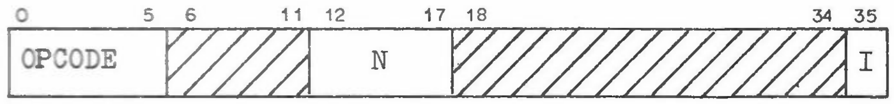
FORMAT TYPE 2



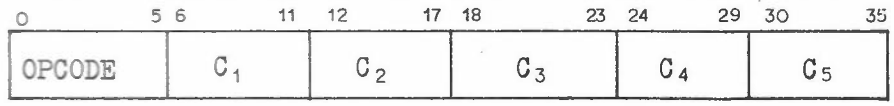
FORMAT TYPE 3



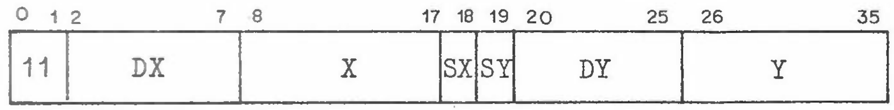
FORMAT TYPE 4



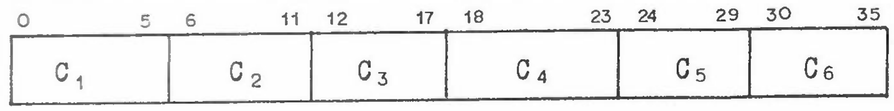
FORMAT TYPE 5



FORMAT TYPE 6



FORMAT TYPE 7



SX = + 1 if bit 18 = 1 else -1

SY = + 1 if bit 19 = 1 else -1

APPENDIX 5

SC 4020 EQUIVALENT OF LINEPRINTER CHARACTERS

Internal Code (Groats Number)	Lineprinter Character	SC 4020 Character
1		Space
2		"
3		±
4		∂
5		∫
6		Σ
7		
8	(	(
9	)	)
10	,	,
11		π
12	?	?
13	&	
14	*	*
15	/	γ
16	0	0
17	'	'
18	2	2
19	3	3
20	4	4
21	5	5
22	6	6
23	7	7
24	8	8
25	9	9
26	<	
27	>	
28	=	=
29	+	+
30	-	-
31	.	.
32	,	,
33	A	A
34	B	B
35	C	C
36	D	D
37	E	E
38	F	F
39	G	G
40	H	H
41	I	I
42	J	J
43	K	K
44	L	L
45	M	M
46	N	N
47	O	O
48	P	P
49	Q	Q
50	R	R
51	S	S
52	T	T

53	U	U
54	V	V
55	W	W
56	X	X
57	Y	Y
58	Z	Z
59		
60		
61		
62		
63		
64		
65		
66		
67	£	
68		
69		
70		
71		
72		
73		
74		
75		$\cancel{S}$
76		
77		
78		
79	:	
80		
81	[	
82	]	
83		
84		
85		
86		
87	⊤	
88	$_2$	
89		
90	α	α
91	β	β
92	½	γ
93	10	δ
94	11	
95		
96		
97	A	A
98	B	B
99	C	C
100	D	D
101	E	E
102	F	F
103	G	G
104	H	H
105	I	I
106	J	J
107	K	K
108	L	L
109	M	M
110	N	N
111	O	O
112	P	P

113	Q	Q
114	R	R
115	S	S
116	T	T
117	U	U
118	V	V
119	W	W
120	X	X
121	Y	Y
122	Z	Z
123		
124		
125		
126		
127		

Where no character is given, the GROATS character will be converted to a space.

APPENDIX 6

BLOCK MAP OF PERMANENT PROCEDURES IN THE GROATSICT I/O PACKAGE

The following numbers are those that will appear in the retroactive trace if the program comes to a fault stop.

```
 1.   tobuffer
 2.        block of 'tobuffer'
 3.        storeword
 4.   outbuffer
 5.        block of 'outbuffer'
 6.        print buffer
 7.             block of 'print buffer'
 8.             print bf
 9.        pack buffer
10.   stoptyping
11.   project form
12.   stop sc 4020
13.   draw vector
14.        block of 'draw vector'
15.   plot hard
16.   type hard
17.   expand image
18.   reduce image
19.   expose light
20.   expose heavy
21.   ignore rest of record
22.   select camera
23.   advance film
24.   advance repeat
25.        block of 'advance repeat'
26.   advanc
27.        block of 'advanc'
28.   printtx
29.   print integer
30.   identification
31.        block of 'identification'
32.   end plotting
33.        block of 'end plotting'
34.   tape 14
35.   trap to
36.   error trap
37.        block of 'error trap'
38.        terminate
39.   userinfo
40.        block of 'user info'
41.   soft char defns
42.   character initialisation
43.   initialise regions
44.        block of 'initialise regions'
45.   job time date
46.   job title
47.   truncate to job no
48.   filemarks
49.   type string
50.        block of 'type string'
51.        string character
```

```
52.   type number
53.       block of 'type number'
54.       x to str
55.   type
56.   plot
57.   plot n
58.   type or plot char
59.       block of 'type or plot char'
60.       decompose
61.   type blanks
62.   type newlines
63.   character factor
64.       block of 'character factor'
65.   character space
66.   character size
67.   hardware size
68.       block of 'hardware size'
69.   displacement is
70.   increment type position
71.   move type to
72.   superfix
73.   super return
74.   suffix
75.   suf return
76.   font
77.   frame
78.   set cutmark
79.   move type to top
80.   vector
81.       block of 'vector'
82.           block of 'vector'
83.           complement
84.           block of 'vector'
85.   see vector
86.       block of 'see vector'
87.   is selected region
88.   convert
89.       block of 'convert'
90.   in view
91.   passive
92.   graticule at
93.       block of 'graticule at'
94.   scales at
95.       block of 'scales at'
96.   axes at
97.       block of 'axes at'
98.   border
99.   region
100.      block of 'region'
101.  limits
102.  select region
103.      block of 'select region'
104.  active
105.  screen region.
106.  math region
107.  window
108.  shield
109.  contain
110.  extend
111.  rotate
112.  expand
```

```
113.   expand and rotate
114.   pointer
115.   addr
116.   read string
117.        block of 'read string'
118.        rd str
119.   mount 35mm camera
120.        block of 'mount 35mm camera'
121.   insert user form slide
122.        block of 'insert user form slide'
123.   start saving
124.   finish saving
125    use saved
126.        block of 'use saved'
127.   lineprinter on
128.   lineprinter off
129.   lineprinter sim
130.        block of 'lineprinter sim'
131.        lp sim on off
132.   shadow
133.   dark
134.   thick
135.   use axes
136.   use vectors
137.   zero invisible
138.   zero seen
139.   box
140.   broken
141.   to point
142.   region and select
143.   raster length
144.   setting char
145.        block of 'setting char'
146.   define char as special
147.   print sc 4020 records
148.   pass to ibm tape
149.   set plot char
150.   abs
151.   sign
152.   sqrt
153.   sin
154.   cos
155.   arctan
156.   ln
157.   exp
158.   entier
159.   select input
160.   select output
161.   read
162.   read Boolean
163.   output
164.   print
165.   write text
166.   space
167.   newline
168.   paperthrow
169.   write Boolean
```

BLOCK MAPS FOR GROATSKDF9 AND GROATSELLIOTT

GROATSELLIOTT

1 - 74 (as Elliott)
75
77
78
80
82
83
84
85
86
87
89
90
91
92
93
94
95
96
97
98
100
102
103
104
106
108
109
110
112
113
115
116
117
119
120
121
122
123
125
126
128
129
130
131
132
134
135
136
137
139
140
141
143
144
146
147

GROATSKDF9

1 - 66 (as KDF9)	
67	tobuffer
69	store word
70	outbuffer
72	print buffer
74	print bf
75	pack buffer
76	stop typing
77	project. form
78	stop sc4020
79	draw vector
81	plot hard
82	type hard
83	expand image
84	reduce image
85	expose light
86	expose heavy
87	ignore rest of record
88	select camera
89	advance film
90	advance repeat
92	advanc
94	print tx
95	print integer
96	indentification
98	end plotting
100	tape 14
101	trap to
102	error trap
104	terminate
105	user info
107	soft char defns
108	character initialisation
109	initialise regions
111	job time date
112	job title
113	truncate job no
114	file marks
115	type string
117	string character
118	type number
120	xto str
121	type
122	plot
123	plot n
124	type or plot char
126	decompose
127	type blanks
128	type newlines
129	character factor
131	character space
132	character size
133	hardware size
135	displacement is
137	move type to
138	super fix
139	super return

148	140	suffix
149	141	suf return
150	142	font
151	143	frame
152	144	set cut mark
153	145	move type to top
154	146	vector
157	149	complement
159	151	see vector
161	153	is selected region
162	154	convert
164	156	inview
165	157	passive
166	158	graticule at
168	160	scales at
170	162	axes at
172	164	border
173	165	region
175	167	limits
176	168	select region
178	170	active
179	171	screen region
180	172	math region
181	173	window
182	174	shield
183	175	contain
184	176	extend
185	177	rotate
186	178	expand
187	179	expand and rotate
188	180	pointer
189	181	addr
190	182	read string
192	184	rdstr
193	185	mount 35mm camera
195	187	insert user form slide
197	189	start saving
198	190	finish saving
199	191	use saved
201	193	lineprinter on
202	194	lineprinter off
203	195	lineprinter sim
205	197	lp sim on off
206	198	shadow
207	199	dark
208	200	thick
209	201	use axes
210	202	use vectors
211	203	zero invisible
212	204	zero seen
213	205	box
214	206	broken
215	207	to point
216	208	region and select
217	209	raster length
218	210	setting char
220	212	define char as special
221	213	print sc4020 records
222	214	pass to ibm tape
223	215	set plot char

APPENDIX 7

BCD TO SC 4020 CHARACTER CONVERSION IN PRINT MODE

BCD Tape Code	SC 4020 Character	BCD Tape Code	SC 4020 Character
00	None	40	-
01	1	41	J
02	2	42	K
03	3	43	L
04	4	44	M
05	5	45	N
06	6	46	O
07	7	47	P
10	8	50	Q
11	9	51	R
12	0	52	None
13	=	53	$
14	"	54	*
15	'	55	γ
16	δ	56	None
17	α	57	d
20	blank	60	+
21	/	61	A
22	S	62	B
23	T	63	C
24	U	64	D
25	V	65	E
26	W	66	F
27	X	67	G
30	Y	70	H
31	Z	71	I
32	None	72	π
33	,	73	.
34	(	74	)
35	∫	75	β
36	Σ	76	±
37		77	?