CIRCLE OF LIGHT

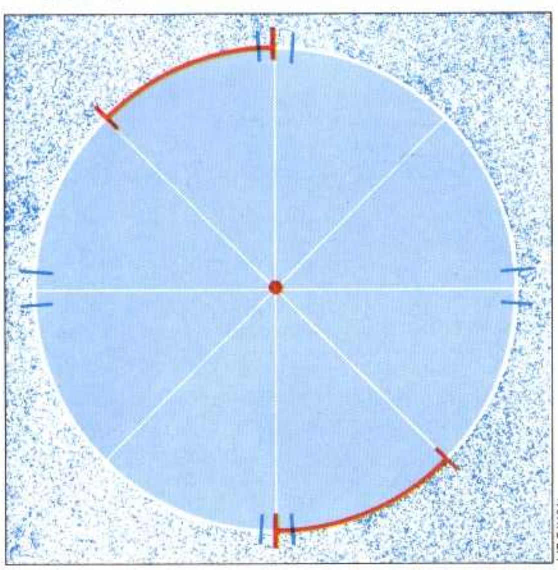
BBC BASIC is graced with a range of commands to help create impressive graphics. Yet it lacks one important facility — a command that enables you to draw circles. Here, we show you how to write a machine code routine to draw circles, which can be used from BASIC or incorporated as a routine in other programs.

Drawing a circle may seem like a simple task, but coming up with a mathematical equation to plot the points on a circle with speed is a fairly difficult exercise. The simplest way to draw a circle is by using the COS and SIN functions, as in this example:

```
DEF PROCCIRCLE (XORG,YORG,R)
  MOVE XORG+R, YORG
  FOR THETA =0 TO 2*PI STEP PI/32
    X=R*COS(THETA)
    Y=R*SIN(THETA)
    DRAW X+XORG,Y+YORG
  NEXT THETA
ENDPROC
```

Because this procedure uses the SIN and COS functions it takes some time to perform the calculations. The plotting of the circle is relatively slow. However, it can be speeded up considerably using the following algorithm, which can be derived from elementary geometry and differential calculus:

```
3 MODE1
5 PROCCIRCLE(500,600,200)
7 END
10 DEF PROCCIRCLE (XORG,YORG,R)
20 Y=R
30 FOR X=1 TO Y*.707
40 Y=Y-X/Y
50 PROCPOINTS(X,Y)
60 NEXT
70 ENDPROC
80 DEF PROCPOINTS (X,Y)
90 PLOT 69,XORG+X,YORG+Y
100 PLOT 69,XORG-X,YORG+Y
110 PLOT 69,XORG-X,YORG-Y
120 PLOT 69,XORG+X,YORG-Y
130 PLOT 69,XORG+Y,YORG+X
140 PLOT 69,XORG-Y,YORG+X
150 PLOT 69,XORG-Y,YORG-X
160 PLOT 69,XORG+Y,YORG-X
170 ENDPROC
```



LIZ DIXON

Mirror Images
The machine code program uses an equation to plot the circle that is specially chosen to be fast.. All the same it would be slow to use it to draw all the way around a circle, calculating each point. This is because the top half of a circle is a mirror image of the bottom half, so we only need calculate the equation for the top half and as we do this produce its mirror image below. This same principle can be extended because the left half of a circle is a mirror image of the right half.

In practice, the program only calculates one eighth of a circle. Each point on this eighth is copied seven times onto other parts of the circle to build up the whole circle

This routine draws the circle in eight sections at the same time, which helps make the plotting much quicker. This algorithm is also better than our initial routine because it doesn't require sine and cosine values to be calculated for each point. However, it does need to calculate a division for each point, which is slow to perform.

An alternative that doesn't need any complicated maths is:

```
10 MODE4
20 PNUM=69
30 PROCCIRCLE(500,600,200)
40 END
50 :
60 DEF PROCCIRCLE(X,Y,R)
70 VDU29,X;Y;:REM SET GRAPHICS ORIGIN
80 X=0:Y=R:D=3-2*R:REM VARIABLES
90 REPEAT
100 PROCCPLOT
110 IFD<0:D=D+4*X+6:ELSED=D+4*(X-Y)+10:Y
    =Y-4
120 X=X+4
130 UNTIL X>Y:ENDPROC
140 :
150 DEF PROCCPLOT
160 PLOT PNUM,X,Y
170 PLOT PNUM,Y,X
180 PLOT PNUM,Y,-X
190 PLOT PNUM,-X,Y
200 PLOT PNUM,-X,-Y
210 PLOT PNUM,-Y,-X
220 PLOT PNUM,-Y,X
230 PLOT PNUM,X,-Y
240 ENDPROC
```

This is known as Breshen's algorithm. It is much quicker because it only has to perform addition, subtraction, and multiplication by two or four (both of which can be implemented by bit shifting). This is the algorithm that we use in our machine code circle drawing program.