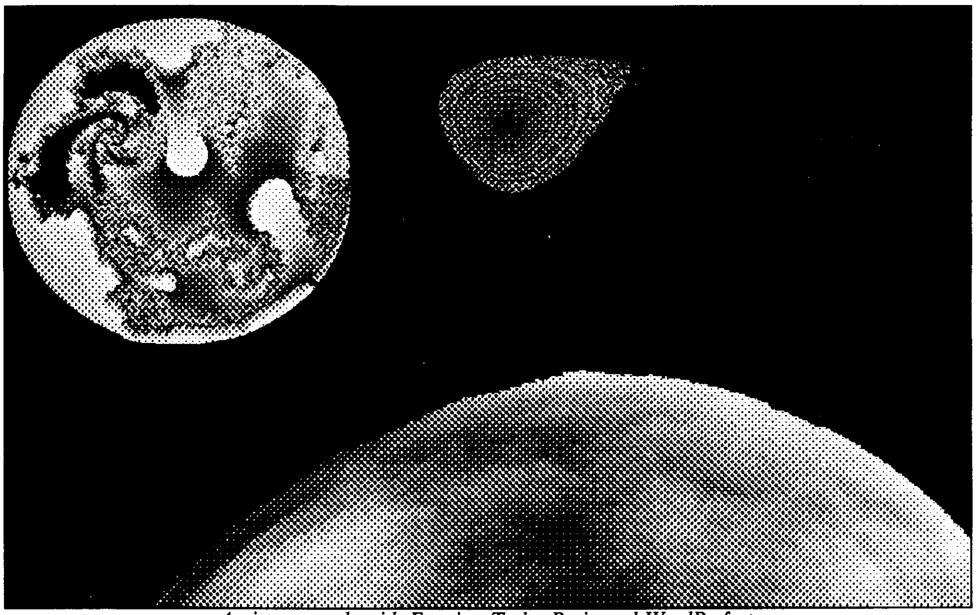
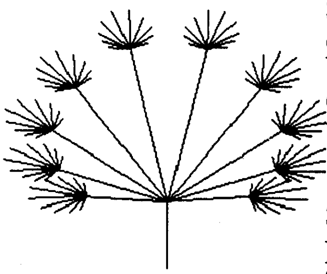
ISSN 0964 – 5640

FRACTAL REPORT 19



An image made with Fractint, Turbo Basic and WordPerfect.



Single copy rate £2. Subscription rates six issues: – £10 (UK only) £12 Europe £13 elsewhere. Cheques in British Pounds should be drawn on a UK bank and should be made payable to "Reeves Telecommunications Laboratories Ltd." Alternatively, dollar checks for $23 can be accepted if drawn on a U.S. bank and made payable to "J. de Rivaz".

Subscribers who are successful in getting one or more articles with programs published in a given series of six issues get the next volume of six issues free of subscription. All new subscriptions are backdated to the start of the current volume.

Fractal Report is published by Reeves Telecommunications Laboratories Ltd.,
West Towan House, Porthtowan, Truro, Cornwall TR4 8AX, United Kingdom.
Volume 4 no 19 First published February 1992. ISSN 0964 – 5640.

IMAGES FROM HALLEY'S METHOD John Topham

In Chapter 15 of the book 'Computers, Pattern, Chaos and Beauty', Dr Clifford Pickover mentions maps derived from Halley's iterative method for finding roots of a polynomial. The method starts with an initial single point similar to Newton's method. The function is as follows:

$$z_{k+1} = z_k - \frac{F(z_k)}{F'(z_k) - (F''(z_k)F(z_k)/\ 2F'(z_k))} \qquad (1)$$

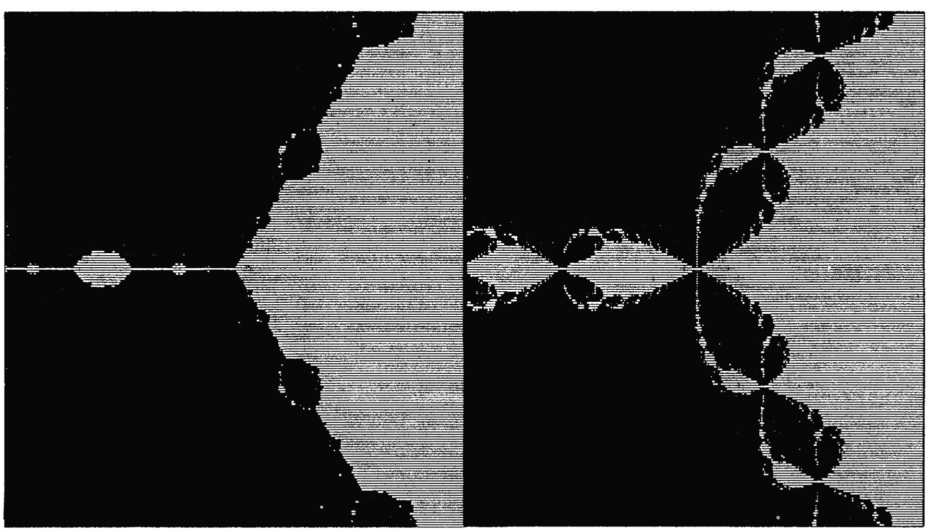
where z_k is the previous point, z_{k+1} is the next point, $F'(z_k)$ and $F''(z_k)$ are the first and second derivative of the polynomial $F(z_k)$ respectively. Dr Pickover applied it to the function: $F(z_k) = z_k((z_k)^6 - 1)$.

If the method was applied to polynomials with rational exponents, a whole variety of images can be derived. A third order polynonial with rational exponents is formulated as follows:

$$F(z) = z^{(m/n)} + c_1 z^{((m-n)/n)} + c_2 z^{((m-2n)/n)} + c_3 z^{((m-3n)/n)} \qquad (2)$$

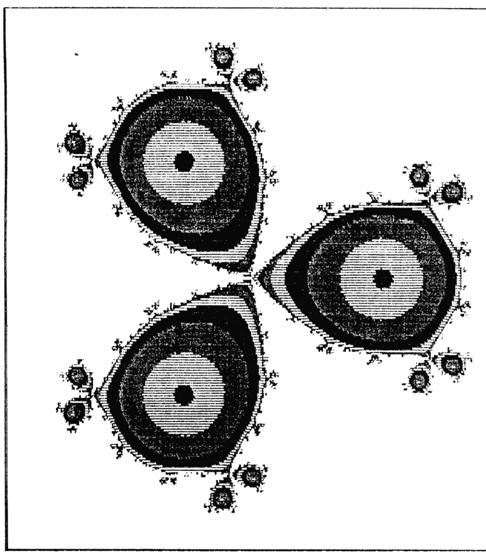
where c_1,c_2 and c_3 are constants which can be derived from roots, and m and n are numerator and denominator of power which 'z' is raised to.

With m=3, n=1, $c_1=c_2=0$ and $c_3=-1$ we get the simple function: $z = z^3 - 1$ (3). Applying this function to Halley's method the resulting map is shown below. When compared to the map of the same function derived from Newton's method, the first map below shows a simpler stucture. Those of you who are sharp eyed will have noticed that Halley's Map of $F(z) = z^3 - 1$ is the same as Newton's Map of $F(z) = z^2 - z^{-1}$. Why should this be so?



If m=0, n=1, $c_1=c_2=0$ and $c_3=-1$ we get the function: $z = 1 - z^{-3}$ (4). In Issue 13 of Fractal Report I showed that when this function was applied to Newton's method, it produced a radically different shape. With Halley's method both functions $F(z) = z^3 - 1$ and $F(z) = 1 - z^{-3}$ produce the same image. The same iterative formula appears for each case. However, when m=-1, n=1 we then get a Halley image akin to the map for function (4) with Newton's method. See the the first image below.

Why use a polynomial with rational exponents? Its seems that if a rational number is used as the exponential index, gradually varying images between integers can be drawn. Using the roots from the previous two images, the images below vary from a fractally sparse structure to one that begins to fill the plane with a rich complexity. By using rational numbers, intermediate phases can be drawn showing how the images develop.



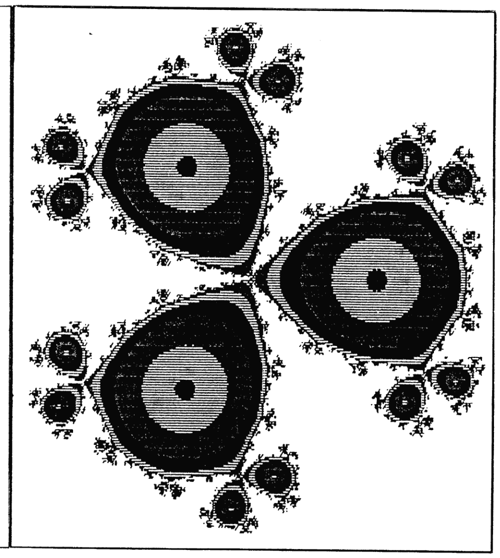
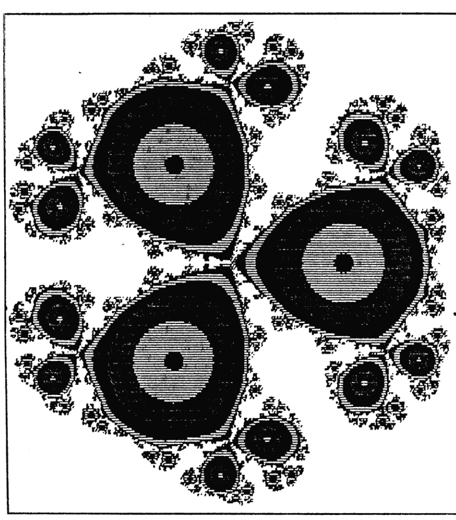


Figure 3 m=-1, n=1 **Figure 4 m=-11, n=16**



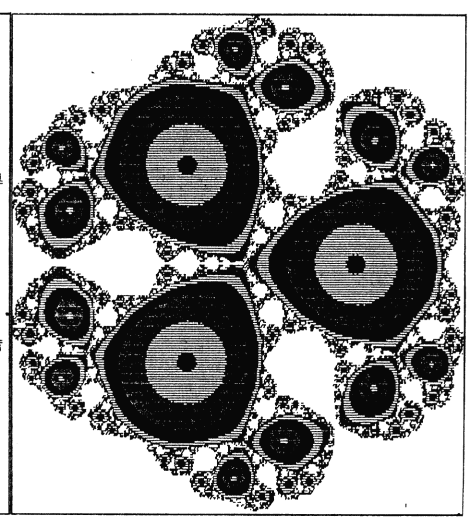


Figure 5 m=-21, n=32 **Figure 6 m=-81, n=128**

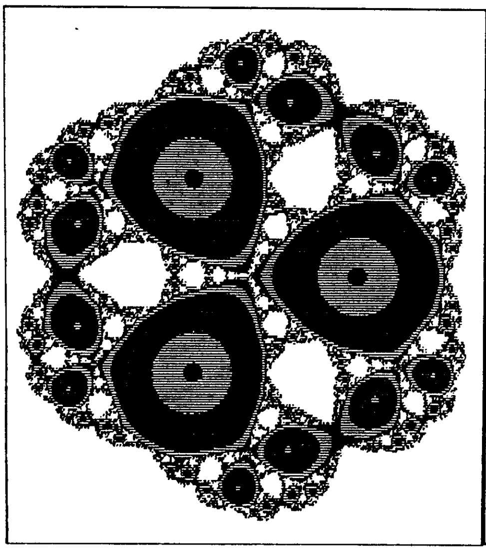


Figure 7   m=-5, n=8

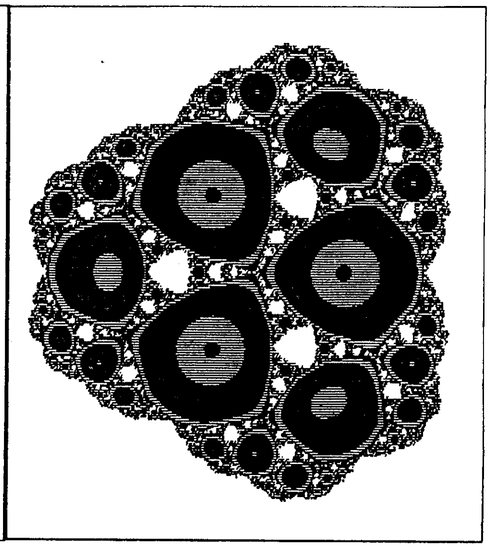


Figure 8   m=-1, n=2

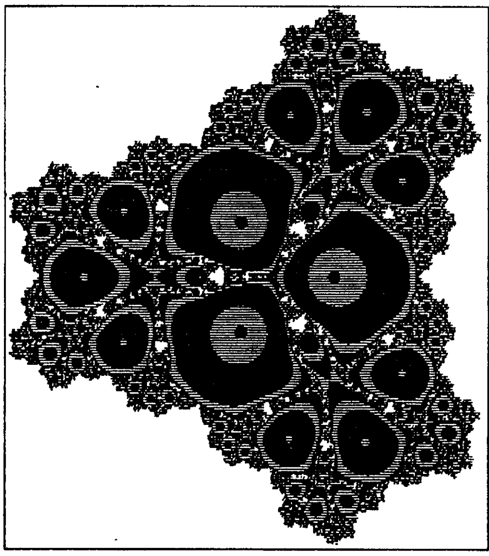


Figure 9   m=-1, n=4

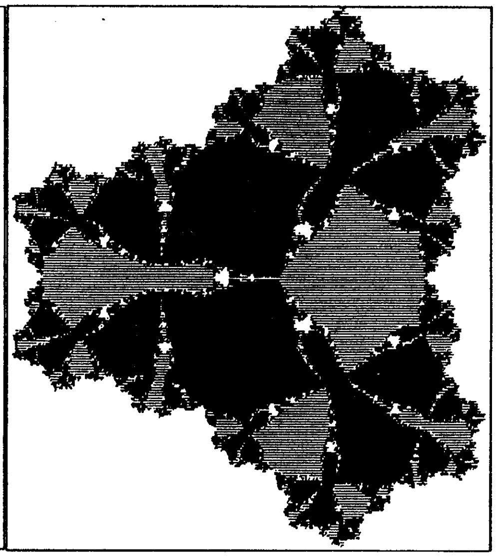


Figure 10   m=-1, n=4

Other images from Halley's method also hold a certain fascination. For example, the first image below shows an extrodinary amount of regularity when the basins of attraction are given only one colour. Two interlocking circles, each circle centred around a root can easily drawn with a compass and a ruler. In my experience, it is quite rare for fractals images of this type to have such a regularity . The image beside it is a Newton's method map for the same root set. It also shows a certain regularity buried in a lot of peripheral fractal

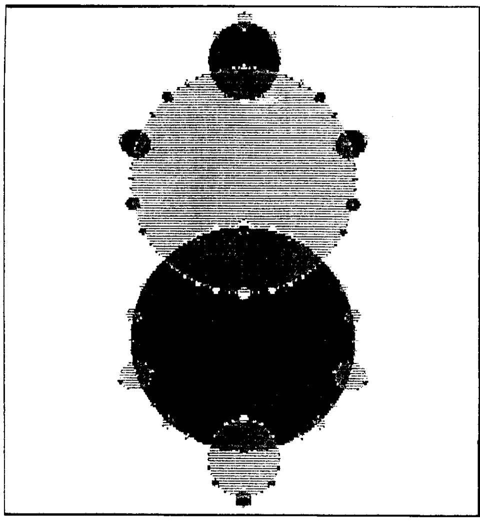


Figure 11 m=-1, n=1

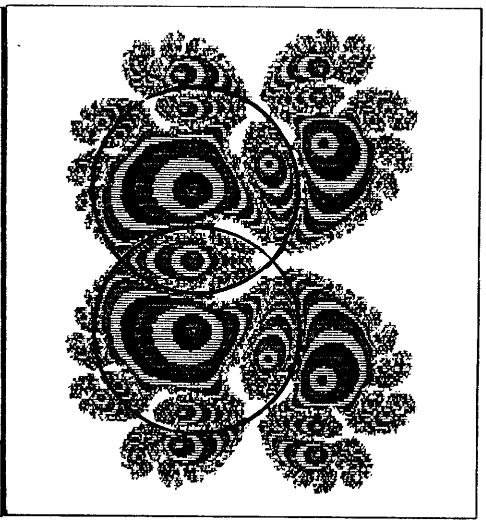


Figure 12 m=0, n=1

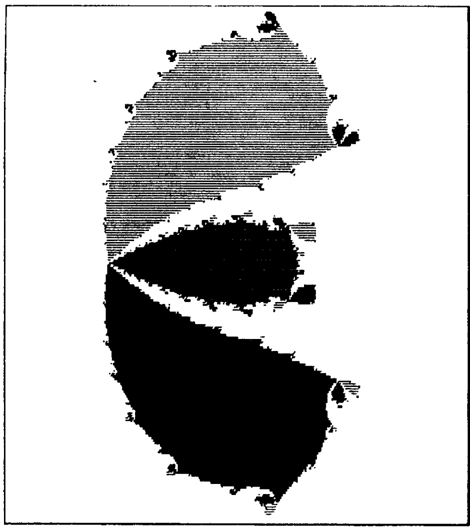


Figure 13 m= -1, n= 2

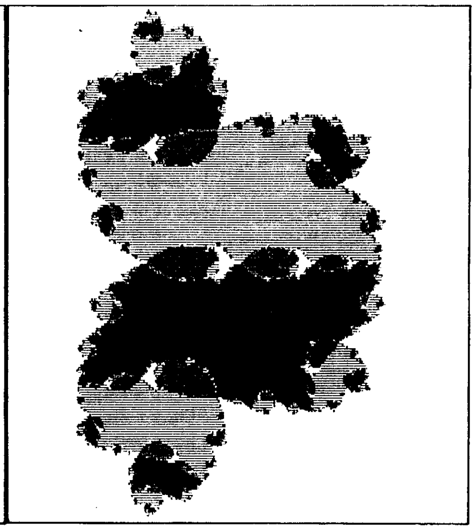


Figure 14 m=-3, n=2

structures. The heavy lines show again interlocking circles. In contrast, the next two pictures, which have the same root set as figure 14 and where in the first m=-1 and n=2 and in the second m=-3 and n=2, have lost all the regularity apart from their symmetry about the X-axis.

If anyone is interested in experimenting with other images I include a rather large QL 'Superbasic' program which draws all the Halley's method examples given above.

```
100 REMark  Halley's Method
110 REMark  General 3rd order
120 REMark  polynomial with
130 REMark  rational exponents
140 :
150 CLS
160 m=3:n=1
170 PP_CALC
180 xmin=-2:xmax=2
190 ymin=-2:ymax=2
200 across=200:down=200
210 REMark ROOTS
220 DIM rr(3),ri(3)
230 rr(1)=1:ri(1)=0
240 rr(2)=-.5:ri(2)=.866
250 rr(3)=-.5:ri(3)=-.866
260 COEFFS
270 dx=(xmax-xmin)/across
280 dy=(ymax-ymin)/down
290 :
300 FOR yp=0 TO down-1
310  FOR xp=0 TO across-1
320   xn=xmin+(dx*xp)
330   yn=ymin+(dy*yp)
340   FOR k=1 TO 40
350    IF ABS(xn-0)<1E-7
360    xn=0:END IF
370    IF ABS(yn-0)<1E-7
380    yn=0:END IF
390    EQU_I
400    IF c=0 AND d=0
410    EXIT k:END IF
420    xm=(a*c+b*d)/(c^2+d^2)
430    ym=(b*c-a*d)/(c^2+d^2)
440    IF ABS(xm)>1E6
450     IF ABS(ym)>1E6
460      EXIT k:END IF
470     END IF
480    END IF
490    IF ABS(xm-xn)<1E-3
500     IF ABS(ym-yn)<1E-3
510      FOR i=1 TO 3
520      IF ABS(xm-rr(i))<1E-3
530      IF ABS(ym-ri(i))<1E-3
540       col=i*2
550      u=xp+1:v=down-yp
560      BLOCK 1,1,u,v,col
570       EXIT k
580      END IF
590      END IF
600      END FOR i
610    END IF
620   END IF
630   xn=xm:yn=ym
640   END FOR k
650  END FOR xp
660 END FOR yp
670 STOP
680 :
690 DEFine PROCedure EQU_I
700 s=xn:t=yn
710 ss=s*s+t*t
720 IF ss=0:RETurn :END IF
730 :
740 ra1=s/ss
750 rb1=t/ss
760 ra2=  ra1*ra1-rb1*rb1
770 rb2=  2*ra1*rb1
780 ra3=  ra2*ra1-rb2*rb1
790 rb3=  ra2*rb1+rb2*ra1
800 ra4=  ra3*ra1-rb3*rb1
810 rb4=  ra3*rb1+rb3*ra1
820 pa2=s*s-t*t
830 pb2=2*s*t
840 pa3=pa2*s-pb2*t
850 pb3=pa2*t+pb2*s
860 :
870 a1=  pp1*pa3
880 a2=  pp2*(c1r*pa2-c1i*pb2)
890 a3a= pp3*(c2r*s-c2i*t)
900 a3b= pp4*(c11r*s-c11i*t)
910 a3=a3a+a3b
920 a4a= pp5*c3r
930 a4b= pp6*c12r
940 a4=a4a+a4b
950 a5a= pp7*(c13r*ra1+c13i*rb1)
960 a5b= pp8*(c22r*ra1+c22i*rb1)
970 a5=a5a+a5b
980 a6=  pp9*(c23r*ra2+c23i*rb2)
990 a7=  pp10*(c33r*ra3+c33i*rb3)
1000 a=a1+a2+a3+a4+a5+a6+a7
1010 :
1020 b1=  pp1*pb3
1030 b2=  pp2*(c1i*pa2+c1r*pb2)
1040 b3a= pp3*(c2r*t+c2i*s)
1050 b3b= pp4*(c11r*t+c11i*s)
1060 b3=b3a+b3b
1070 b4a= pp5*c3i
1080 b4b= pp6*c12i
1090 b4=b4a+b4b
1100 b5a= pp7*(c13i*ra1-c13r*rb1)
1110 b5b= pp8*(c22i*ra1-c22r*rb1)
1120 b5=b5a+b5b
1130 b6=  pp9*(c23i*ra2-c23r*rb2)
1140 b7=  pp10*(c33i*ra3-c33r*rb3)
1150 b=b1+b2+b3+b4+b5+b6+b7
1160 :
1170 c1=  pq1*pa2
1180 c2=  pq2*(c1r*s-c1i*t)
1190 c3a= pq3*c2r
1200 c3b= pq4*c11r
1210 c3=c3a+c3b
1220 c4a= pq5*(c3r*ra1+c3i*rb1)
1230 c4b= pq6*(c12r*ra1+c12i*rb1)
1240 c4=c4a+c4b
1250 c5a= pq7*(c13r*ra2+c13i*rb2)
1260 c5b= pq8*(c22r*ra2+c22i*rb2)
1270 c5=c5a+c5b
1280 c6=  pq9*(c23r*ra3+c23i*rb3)
1290 c7=  pq10*(c33r*ra4+c33i*rb4)
1300 c=c1+c2+c3+c4+c5+c6+c7
1310 :
1320 d1=  pq1*pb2
1330 d2=  pq2*(c1r*t+c1i*s)
1340 d3a= pq3*c2i
1350 d3b= pq4*c11i
1360 d3=d3a+d3b
1370 d4a= pq5*(c3i*ra1-c3r*rb1)
1380 d4b= pq6*(c12i*ra1-c12r*rb1)
1390 d4=d4a+d4b
1400 d5a= pq7*(c13i*ra2-c13r*rb2)
1410 d5b= pq8*(c22i*ra2-c22r*rb2)
1420 d5=d5a+d5b
1430 d6=  pq9*(c23i*ra3-c23r*rb3)
1440 d7=  pq10*(c33i*ra4-c33r*rb4)
1450 d=d1+d2+d3+d4+d5+d6+d7
1460 END DEFine
1470 :
1480 DEFine PROCedure COEFFS
1490 r1r=rr(1):r1i=ri(1)
1500 r2r=rr(2):r2i=ri(2)
1510 r3r=rr(3):r3i=ri(3)
1520 :
1530 c1r=-(r1r+r2r+r3r)
1540 c1i=-(r1i+r2i+r3i)
1550 c2ra=r1r*r2r-r1i*r2i
1560 c2rb=r1r*r3r-r1i*r3i
1570 c2rc=r2r*r3r-r2i*r3i
1580 c2r=c2ra+c2rb+c2rc
1590 c2ia=r1r*r2i+r1i*r2r
1600 c2ib=r1r*r3i+r1i*r3r
1610 c2ic=r2r*r3i+r2i*r3r
1620 c2i=c2ia+c2ib+c2ic
1630 c3ra=-r1r*r2r*r3r
1640 c3rb=r1i*r2i*r3r
1650 c3rc=r1i*r2r*r3i
1660 c3rd=r1r*r2i*r3i
1670 c3r=c3ra+c3rb+c3rc+c3rd
1680 c3ia=-r1r*r2r*r3i
1690 c3ib=r1i*r2i*r3i
1700 c3ic=-r1i*r2r*r3r
1710 c3id=-r1r*r2i*r3r
1720 c3i=c3ia+c3ib+c3ic+c3id
1730 :
1740 c11r=c1r*c1r-c1i*c1i
1750 c11i=2*c1r*c1i
1760 c22r=c2r*c2r-c2i*c2i
1770 c22i=2*c2r*c2i
1780 c33r=c3r*c3r-c3i*c3i
1790 c33i=2*c3r*c3i
1800 c12r=c1r*c2r-c1i*c2i
1810 c12i=c1r*c2i+c1i*c2r
1820 c13r=c1r*c3r-c1i*c3i
1830 c13i=c1r*c3i+c1i*c3r
1840 c23r=c2r*c3r-c2i*c3i
1850 c23i=c2r*c3i+c2i*c3r
1860 END DEFine
1870 :
1880 DEFine PROCedure PP_CALC
1890 pp1=m*(m-n)
1900 pp2=2*m*(m-2*n)
1910 pp3=2*(m*m-3*m*n-n*n)
1920 pp4=(m-n)*(m-2*n)
1930 pp5=2*(m*m-4*m*n-3*n*n)
1940 pp6=2*(m-n)*(m-3*n)
1950 pp7=2*(m*m-5*m*n+3*n*n)
1960 pp8=(m-2*n)*(m-3*n)
1970 pp9=2*(m-2*n)*(m-4*n)
1980 pp10=(m-3*n)*(m-4*n)
1990 pq1=m*(m+n)
2000 pq2=2*(m-n)*(m+n)
2010 pq3=2*(m*m-m*n-3*n*n)
2020 pq4=m*(m-n)
2030 pq5=2*(m*m-2*m*n-6*n*n)
2040 pq6=2*m*(m-2*n)
2050 pq7=2*(m*m-3*m*n-n*n)
2060 pq8=(m-2*n)*(m-n)
2070 pq9=(m-3*n)*(2*m-2*n)
2080 pq10=(m-2*n)*(m-3*n)
2090 END DEFine
```


References: 'Computers, Pattern, Chaos and Beauty'
Dr Clifford Pickover, Alan Sutton Publishing 1990
ISBN 0-86299-792-5

Rabbits and Foxes

by Dolores García García
Corredera de San Lorenzo, n.1, 2A
15706 Santiago
SPAIN

I have found one of the most curious nonlinear systems working with the predator-prey model of Volterra. These equations represent a population of two competing species, say rabbits and foxes:

Rnew=R+a*R-b*R*R-c*R*F
Fnew=f-d*F+e*R*F

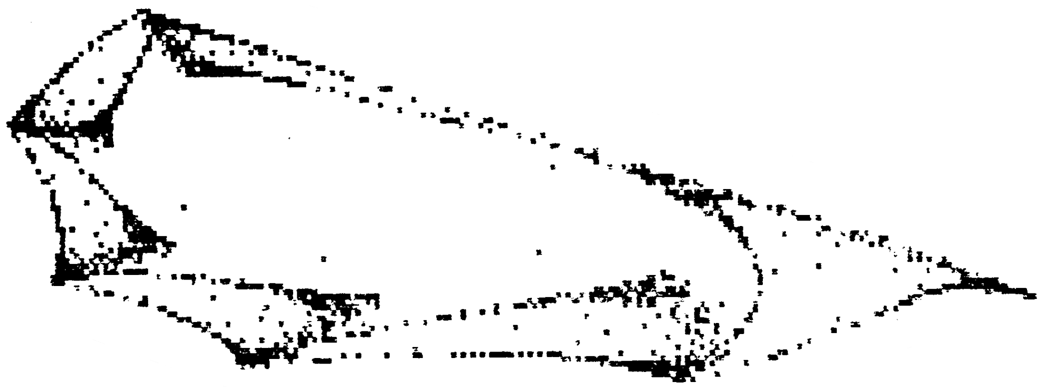


Figure 1 - Result after 2000 iterations of the Volterra equations, with a=3.1, b=0.01, c=0.03, d=0.5, e=0.0058

Rnew is the new rabbit population
R is the old rabbit population
a*R is the number of new-born rabbits, that depends on the number of existing rabbits
b*R*R is the number of rabbits that die from natural causes, that is, not eaten by foxes
c*R*F is the number of eaten rabbits, that depends not only on the number of existing rabbits ,but also on the number of foxes
Fnew is the new fox population
F is the old fox population
d*F is the number of foxes that die
e*R*F is the number of new-born foxes. The more rabbits are eaten, the more foxes are born.

I experimented with my computer with different values for a, b, c, d and e, and used graphics to see how the populations increased and decreased. I represented the rabbit population in the x axis and the fox population in the y axis. Usually you get a spiral, that means that the number of foxes and rabbits reaches an equilibrium where the two populations stay constant. But sometimes, I get a lot of small and big loops, and the numbers never repeat, for example with a=3.1, b=0.01, c=0.03, d=0.5 and e=0.0058. After many iterations, the points form strange patterns.

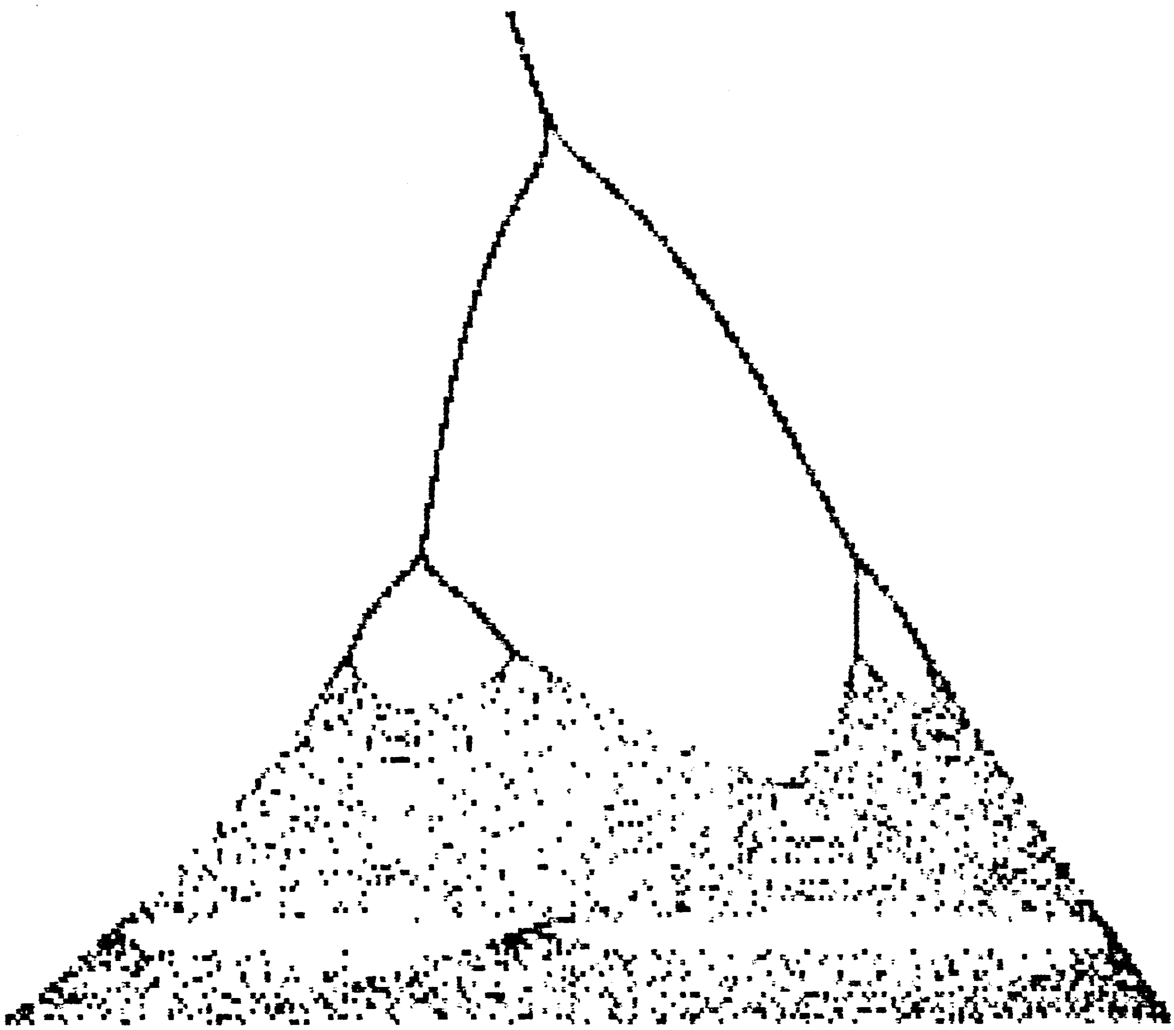


Figure 2 - Bifurcation diagram obtained with the modified equations, with a=3, b=0.015, c=0.03, d=0, e=0.001

But this isn't the only thing I discovered. A friend wondered if the rabbits-foxes equations were realistic enough. When rabbits decline, foxes should decline proportionally. So I changed the second equation to:

Fnew=F-d*F/R+e*F

When I put a pair of foxes in a group of 300 rabbits, and plotted the result, surprise! I saw the bifurcation diagram. After some time, I found out why does that happen. The rabbits equation is almost the equation of the bifurcation diagram. On each iteration, the foxes equation changes slightly the parameters of the rabbits equation, turning it into a less chaotic mode. In fact, in the equation for foxes, -d*F/R is always quite small and can be suppressed (setting d=0). What I have discovered is a new fast algorithm to plot the bifurcation diagram. The results are better with small values for e, for example e=0.001. (Figure 2)

Of course, this isn't a good model for the rabbits-foxes system, because the number of foxes is independent of the number of rabbits. So I kept the term -d*F/R and tried with different values for a, b, c, d and e, until I got something that seemed more realistic. For most values, the populations increase and decrease in a cycle, which is displayed as a closed loop in the screen. But some values are more interesting. When a=3.6, b=0.012, c=0.04, d=2.4 and e=0.15, half the loop is the same in all the cycles, but the other half is different each time. I understand that certain equations, like the Lorenz attractor, when iterated with slightly different starting values, can end up in very different points, but how is it possible that they can reach again and again the same values after the chaotic part of the loop?

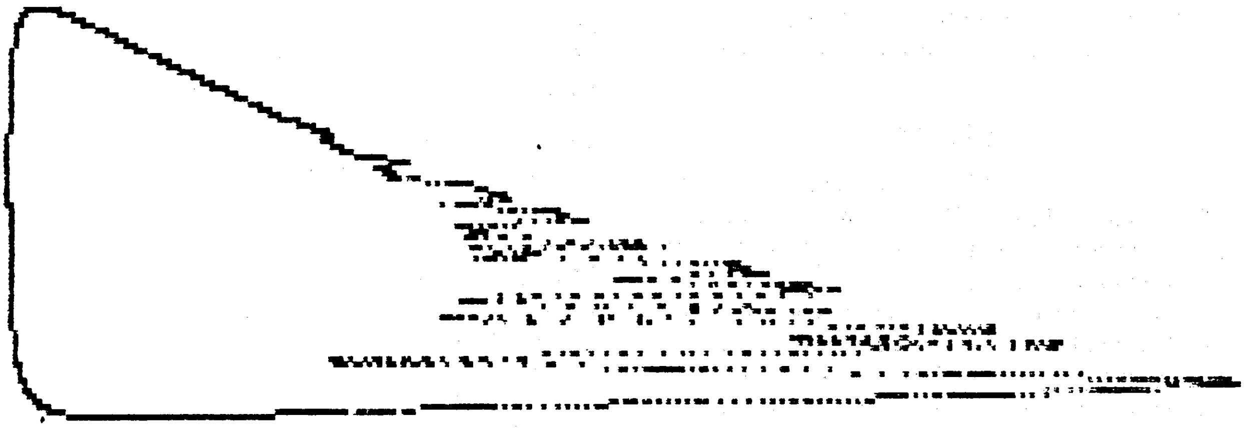


Figure 3 - Result after 2000 iterations of the modified equations, with a=3.6, b=0.012, c=0.04, d=2.4, e=0.15

LISTING FOR THE RABBITS-FOXES PROGRAM

```
10 REM INPUT CONSTANTS
20 INPUT "a ";A:INPUT "b ";B:INPUT "c ";C:INPUT "d ";D:INPUT "e ";E
30 INPUT "R (1st value)";R1:INPUT "F (1st value)";F1
40 INPUT "(1) Volterra equations  (2)Modified equations ";EQ
50 CLS:SCREEN 1:REM GRAPHICS SCREEN
60 R=R1:F=F1
70 FOR N=1 TO 2000
80 RNEW=R+A*R-B*R*R-C*R*F
90 IF EQ=1 THEN FNEW=F-D*F+E*R*F
100 IF EQ=2 THEN FNEW=F-D*F/R+E*F
110 R=RNEW:F=FNEW
120 PSET (R,199-F):REM PLOT PIXEL
130 NEXT N
```

L-systems

by John Sharp

In the last article in Fractal Report [6], I showed some examples of fractals which were generated using L-systems. This article describes basic L-systems, together with a simple set of programs for creating them. Fractint (from version 15) also includes them and allows you definitions in a file. I have used a set of small programs rather than one "do-it-all" program so that I can study evolution and variations. Since L-systems are string rewriting systems, basically carrying out substitutions (a sophisticated search and replace) the successive levels can also be generated with a word processor or editor. This leads to more flexibility and yet more variations.

The L in L-systems

The basic principles of L-systems go back to the turn of the century when von Koch, Peano, Hilbert and Sierpinski were generating new types of curves. (For details of these curves, see [5].) They upturned the conventions of the time since they produced curves which had infinite length, finite area and had no tangent at any point (von Koch's snowflake) or contained every point of a square yet had an area of only half of the square (Sierpinski curve).

From the 1960s fractals were arising from a number of directions. Vector fractals such as we are describing here were a natural output from Turtle graphics which was an outcome of work by Seymour Papert on the LOGO language at MIT summarised in [1]. In 1968 A. Lindenmayer introduced a system for describing the growth of plants. His work has been much expanded since and the methods he pioneered have become known as L-systems after him. There is some detail in [2], but the book "The Algorithmic Beauty of plants" [3] is the bible on their history and use, with some superb examples in two and three dimensions. Such systems are the basis for computer graphic generation of plant forms just as other fractals are used to generate mountains for example see [4].

Shareware/PD for L-Systems

On the PC, I know of two programs for studying L-systems. "Fractint" includes formulae files to define and plot L-system fractals and output them as HPGL (Hewlett Packard Graphics Language) files. "Fractal paint" is a Windows program which allows you to produce displays using many L-systems as well as other similar fractals.

Why then did I take this route. Well, firstly when I started this project, these facilities were not available. Secondly, the output from Fractal Paint was not precise enough for me to manipulate the values it gave in its HPGL files. Also, neither gave me the flexibility to experiment as much as I would like.

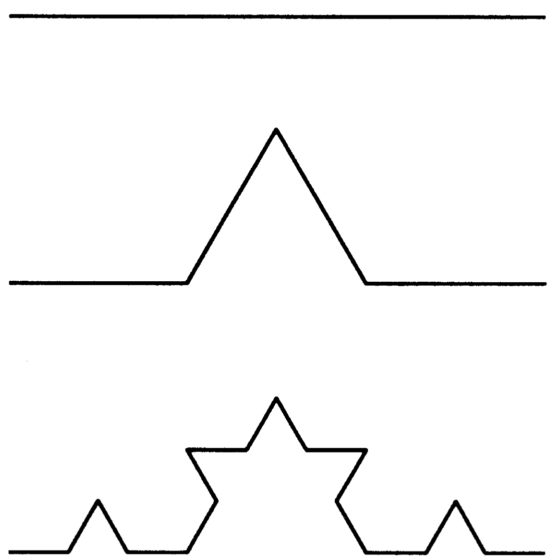
String rewriting

The essence of an L-system is a set of strings (that is character strings) which are used to build up the object defined by the string. For example `F` means move forward, `+` means turn left a defined number of degrees `-` turn right and so on. To get the next level of the object, the string is rewritten according to a set of rules. The new string is then used to plot the object in the turtle geometry fashion of LOGO.

As a simple example, consider the following:
Syntax: `F` move forward a fixed length
`+` turn left 60 degrees
`-` turn right 60 degrees
Starting string (the axiom) is: `F`
Production rules are:
rewrite `F` to `F-F++F-F`
rewrite `+` to `+`
rewrite `-` to `-`
The initial and two subsequent levels are shown in fig 1, and the strings at each level are:

```
Level 1   F
Level 2   F+F--F+F
Level 3   F+F--F+F+F+F--F+F
          --F+F--F+F+F+F--F+F
```

Fig 1



There are many more complicated sets of syntax and there are more sophisticated substitution rules. I am only going to use very basic ones. Look in the Fractint documentation, or in refs [2] and [3] for more ideas. Some books reverse the direction of turning for + and -, so you may get different results by using the same string.

Other common syntax characters are:
G move forward without drawing a line
[save the current drawing position and continue
] restore the last stored drawing position
| turn 180 degrees.

In drawing the object, the size needs to be reduced for the movement as you go up each level, otherwise it becomes too big to see. Also the position of the centre is unpredictable. This requires some calculation. These calculations explain why Fractint asks you to wait and then draws the complete object.

Using a wordprocessor/editor

Since the production rules are a set of search and replace sequences, you could use a wordprocessor (preferably with a macro facility to avoid errors and save the rules) to build up subsequent levels. There are a number of pitfalls, but there are a number of benefits which give more control and which allow some interesting variations which would be tedious to program. The method of continuous substitution and saving the file at each stage which I have used in the program is compatible with this generation method. In both cases it avoids having to carry out recursive programming which can be confusing, and which is also not possible with some versions of the BASIC language.

Possible pitfalls I have come across and their solution are as follows:

1) You must remember to save successive levels if you want to plot them; this also helps you remember which level you have reached.

2) When you save, make sure you save in ASCII format (DOS, text, non-document mode or whatever your wordprocessor/editor calls it), not as the wordprocessor's formatted file.

3) Using successive searches on the whole file means that there is dependence on ones previously replaced. If you are subsituting with more than one rule, for example both F and + get replaced, and you search and replace in that order, you need to ensure that only the + in the original string (not the ones you have substituted) get changed. You can overcome this by converting the original + to something else first and then search for that character instead of the +. On the other hand this opens up possibilities for evolving fractals.

String rewriting using a program

The following program (LSYSGEN.BAS) was written in QuickBasic. It takes a file rule (with extension .RUL) of the following format:

angle in degrees
axiom
translate_from1 translate_to1
translate_from2 translate_to2
translate_from3 translate_to3
translate_from4 translate_to4

so a typical file is:

```
22.5
F
F -F++F-
- -
+ +
G G
```

the program is:

```
' program LSYSGEN for generating
' L system files by John Sharp

pi = 4 * ATN(1)

File$ = "FRACT1"
level = 10    : ' generate up to level

OPEN File$ + ".RUL" FOR INPUT AS #1
' get angle and axiom
INPUT #1, angle: angle = pi * angle / 180
INPUT #1, AXIOM$

FOR k = 1 TO 4
INPUT #1, rule$
rulein$(k) = LEFT$(rule$, 1)
ruleout$(k) = RIGHT$(rule$, LEN(rule$) - 2)
NEXT k

' get characters from axiom and output level 2
n = 2: n$ = STR$(n)
ext$ = RIGHT$(n$, LEN(n$) - 1)
OPEN File$ + ext$ FOR OUTPUT AS #n
FOR j = 1 TO LEN(AXIOM$)
c$ = MID$(AXIOM$, j, 1)
GOSUB rule:
NEXT j
CLOSE #n

' now create other levels
FOR j = 3 TO level
    n = j: n$ = STR$(n)
    ext$ = RIGHT$(n$, LEN(n$) - 1)
    m = j - 1: m$ = STR$(m)
    prevext$ = RIGHT$(m$, LEN(m$) - 1)
    OPEN File$ + prevext$ FOR INPUT AS #m
    OPEN File$ + ext$ FOR OUTPUT AS #n
        WHILE NOT EOF(m)
                c$ = INPUT$(1, #m)
                GOSUB rule:
        WEND
CLOSE #n
NEXT j
```

```
CLOSE

END
rule:
SELECT CASE c$
        CASE rulein$(1)
                PRINT #n, ruleout$(1)
        CASE rulein$(2)
                PRINT #n, ruleout$(2)
        CASE rulein$(3)
                PRINT #n, ruleout$(3)
        CASE rulein$(4)
                PRINT #n, ruleout$(4)

END SELECT
RETURN
```

The levels are written to files because BASIC cannot handle strings longer than 255 characters and the number of characters can be quite long after even a few translations.

The fractal can either be displayed on the screen or output as a vector graphics file for printing. Desktop publishing or wordprocessors can produce the best quality on your printer without your having to write special drivers or just do a screen dump.

Plotting the fractal

The main part of the program for plotting is reading the L-system file and turning it into a set of coordinates. These coordinates are saved to another file for a number of reasons. Firstly so that the fractal can be centred on the screen. Secondly, so that the coordinates can be handled in a number of ways. In the examples shown here they are either displayed on the screen or plotted to a file for printing in this article. In the next article, they are transformed to yield the new type of fractal revealed in the last issue of Fractal Report. The program plots from a subroutine, so I will describe different methods of plotting once I have the coordinates in a file. This program could be simplified, but I have tried to make it as easy to understand as possible; writing around some of the potholes presented by QuickBasic has also made it longer.

```
' L system plotting

angF = (4 * ATN(1)) / 180 'degs to radians
file$ = "C": level = 8
n = level: n$ = STR$(n)
ext$ = RIGHT$(n$, LEN(n$) - 1)
OPEN file$ + ext$ FOR INPUT AS #n

' also get angle from rule file
OPEN file$ + ".RUL" FOR INPUT AS #1
INPUT #1, angle

'reduce length as level gets larger
dist = 200 / n

OPEN "temp.pts" FOR OUTPUT AS #20

heading = 0: x = 0: y = 0: rang = 0
      PRINT #20, USING " #####.##"; x;
      PRINT #20, USING " #####.##"; y

' decode and write coords to temp file
WHILE NOT EOF(n)
      c$ = INPUT$(1, #n)
      GOSUB tempry:
   WEND

CLOSE

' read in temp file and write again with
' header of centre of gravity and max/min

OPEN file$ + ext$ + ".pts" FOR OUTPUT AS #21
OPEN "temp.pts" FOR INPUT AS #1
PRINT #21, USING " #####.## "; xcount / count
PRINT #21, USING " #####.## "; ycount / count
PRINT #21, USING " #####.## "; Xmax
PRINT #21, USING " #####.## "; Ymax
PRINT #21, USING " #####.## "; Xmin
PRINT #21, USING " #####.## "; Ymin
WHILE NOT EOF(1)
INPUT #1, A$
PRINT #21, A$
WEND
CLOSE

GOSUB plotxxx:
END

tempry:

SELECT CASE c$

  CASE "F"
     x = x + dist * COS(rang)
     y = y + dist * SIN(rang)
       PRINT #20, USING " #####.##"; x;
       PRINT #20, USING " #####.##"; y
           count = count + 1
           xcount = x + xcount
           ycount = y + ycount
           IF Xmax  x THEN Xmax = x
           IF Xmin  x THEN Xmin = x
           IF Ymax  y THEN Ymax = y
           IF Ymin  y THEN Ymin = y

    CASE "+"
      heading = (heading + angle) MOD 360
      rang = heading * angF:  'radians

    CASE "-"
      heading = (heading - angle) MOD 360
      rang = heading * angF

END SELECT
RETURN
```

Reasons for programming in this way, include the following. Most forms of Basic put spaces either side of any printed number. This means adding a number to a filename has to have them removed. Printing to a file the second time removes the leading spaces at the beginning of a line. This allows a space to be used as a delimiter for the x and y coordinates and searched for when plotting or reading in the file. Print using is required to avoid problems with small (or large) numbers). If numbers get too large for the number of hashes, you get a % sign in front of the number. Reduce the constant in the variable `dist`.

Plotting to the screen

Plotting to the screen is relatively straightforward. You have the maximum and minimum coordinates and the centre at the top of the coordinate file, so you can position and the scale the fractal to fit on the screen as you want it. The coordinates for positioning are single and the ones for the points are pairs separated by a comma. To decode them, you may find the HPGL program below of help.

Plotting the fractal to an HPGL file

Apart from enabling plotting to a Hewlett Packard plotter, HPGL files can be imported into a number of graphics packages and desktop publishing programs. Some laser printers and ink-jets can also plot from HPGL files. You could also plot to a PostScript file, but then you are more limited in the output available to you. These methods allow you to get much better quality graphics than screen dumps.

The HPGL language has many options, but you only need to know a few of them as follows and very little syntax. The plotter must be initialised, and you must select a pen. For black printing any pen will do. Plotting is then achieved by moving the pen; `PU` raises the pen and the following coordinates are the position to move to. To draw, simply put the pen down with `PD`. Commands are separated with a semi-colon and coordinates with a comma. So a simple file to plot two lines, lift the pen and plot another one is as follows (spaces and carriage returns are ignored:

```
IN;SP1; PU 100, 100; PD 150, 200; PD 300,
300; PU 120, 180; PD 10, 10;
```

Because a graphics/DTP/wordprocessor program handles centring and scaling, the program is less complicated than plotting on the screen.

```
plotHPGL:

OPEN file$ + ext$ + ".pts" FOR INPUT AS #21
OPEN file$ + ext$ + ".HPG" FOR OUTPUT AS #19

' first six lines are centre, max and min
' not needed for this file
FOR j = 1 TO 6: INPUT #21, dummy$: NEXT

' initialise plotter
        PRINT #19, "IN;SP1;"

WHILE NOT EOF(21)
   INPUT #21, A$
        p = INSTR(A$, " ")
        q = LEN(A$) - p
        x = VAL(LEFT$(A$, p - 1))
        y = VAL(RIGHT$(A$, q))
           IF flag = 0 THEN
                 plot$ = "PU"
                 flag = 1
              ELSE plot$ = "PD"
           END IF
        PRINT #19, plot$;
        PRINT #19, USING "#####.##"; x;
        PRINT #19, ",";
        PRINT #19, USING "#####.##"; y;
        PRINT #19, ";"
   WEND
CLOSE
RETURN
```

Results

The main purpose of the programs here was to generate coordinate files for the fractals so that they can be converted to Sharp fractals [6] in the next article in the series. Hoever, some interesting points arise at this stage. Figs 2 and 3 show a fractal generated to level 4 using the rule file:

```
angle
F
F -F++F-
+ +
- -
G G
```

With *angle* as 45 degrees in fig 2 and 60 degrees in fig 3. The same conditions apply for figs 4 and 5, except that a "genetic mutation" was added to the string at level 3 to change one `F` to an `FF`.

Simply changing the angle for plotting can give more dramatic changes in the resulting fractal. The following rule file:

```
angle
F
F F-F++F-F
+ +
- -
G G
```

has been plotted with *angle* as 60 degrees in fig 6, 72 degrees in fig 7 and 90 degrees in fig 8.

Figs 9, 10 and 11 show the effect of using a search and replace in WordPerfect, all to level 2. The rule file is:

```
angle
F
F F++F
+ +FF+
- -
G G
```

In figs 9 and 10, the rewriting is done normally, with *angle* is 45 degrees and 60 degrees respectively. Fig. 11 is the file generated in WordPerfect with first the `F` replace and then the `+` replace. Going to level 4 generates a very large file. Although for 60 degrees the fractal varies as the level increases, this is not the case for 45 degrees. Fig 11 shows the result for all levels from 2 on.

Finally, figures 12 and 13 show the results of having first one and then two successive rules. The rule file is :

60

```
F
F F++FF-
+ +
- -
G G
```

alone for figure 12, but alternating with the following rule :

```
60
F
F F--FF+
+ +
- -
G G
```

for the even rewriting in fig 13. (Fig 14 is the same as 13 but with the angle changed to 90 degrees.)

Like most fractals, life is too short to explore all the possibilities. I have just given a few ideas at this stage. In the next article I will open up yet more new Sharp fractals.

References

[1] Harold Abelson and Andrea diSessa, "Turtle Geometry", MIT 1981

[2] H-O Peitgen et al, "The Science of Fractal Images", Springer-Verlag 1988

[3] P.Prusinkiewicz, A. Lindenmayer "The Algorithmic Beauty of plants", Springer 1990.

[4] Chris Sangwin, "The Blancmange", Fractal Report 18, Dec 1991

[5] David Wells, illustrated by John Sharp, "The Penguin Dictionary of Curious and Interesting Geometry", Penguin 1991

[6] John Sharp, "Chaos in new clothes and some new fractals", Fractal Report 18

Fig 2

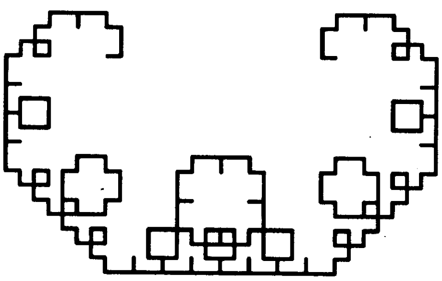


Fig 3

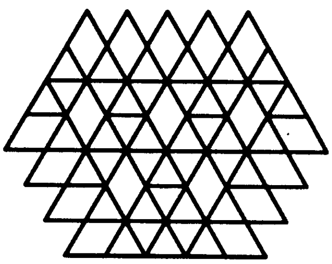


Fig 4

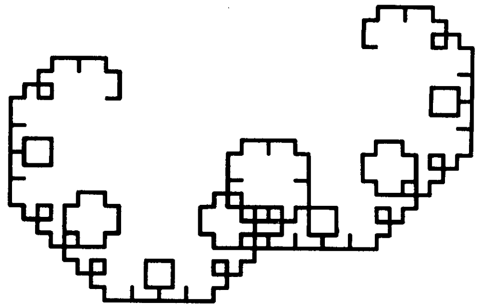


Fig 5

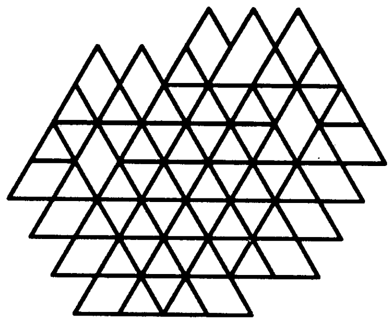


Fig 6

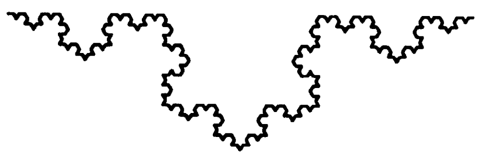


Fig 7

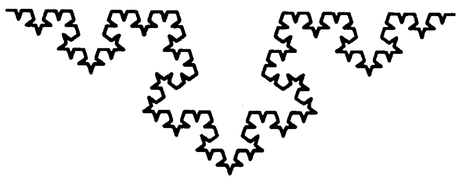


Fig 8

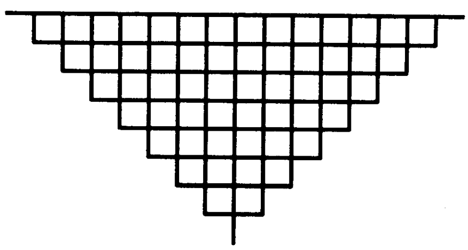


Fig 9

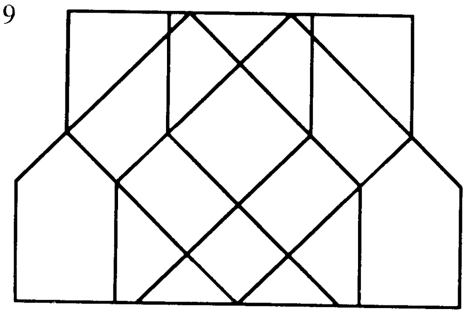


Fig 10

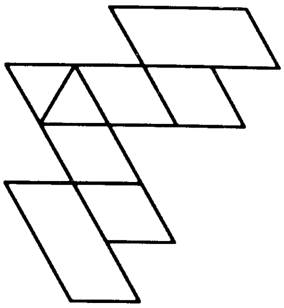


Fig 11

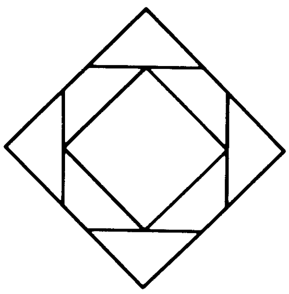


Fig 12

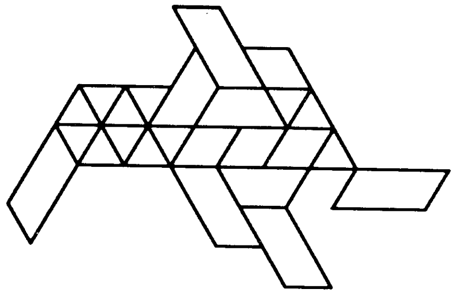


Fig 13

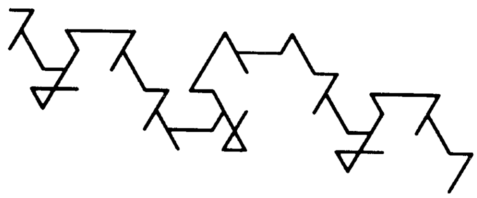
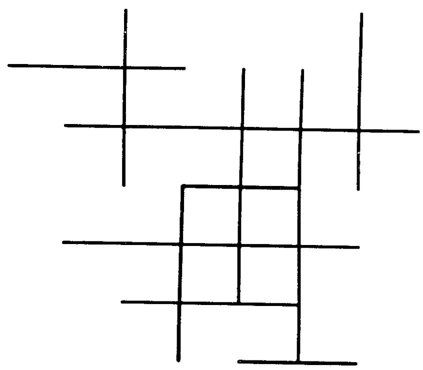


Fig 14



Editorial

The article pile recovered a bit, but again is now fairly low, and I would be grateful for some contributions for the next issue, preferably before the end of February. The articles were a bit longer this time, but I hope that you enjoy them and the graphic images. I have regretfully had to hold over a few articles this time, so if one was yours, don't despair – it may appear next time!

Because of the popularity of *Fractint*, many readers would benefit from a regular column of articles on using *Fractint,* especially the Formula function. I look forward to receiving suitable material for publication in future issues. In the meantime I would highly recommend *Fractal Creations,* promoted with issue 17. We still have a few leaflets left if you have lost yours.

I would also be interested in material concerning the use of the Roland LAPC – 1 PC sound card to generate fractal music, preferably using a compiled BASIC.

The results of our £300 advertisement were very poor from a financial point of view, with only a few subscriptions resulting. However there were over 50 replies, so one can't blame *New Scientist*. It appeared on 10 October.

Despite the poor timing of its launch, we have received a number of enquiries for paid space in our free – to – readers sheet *Fractal Shopper* and it is hoped that it will appear twice a year. It is hoped that the revenue earned by this will support *Fractal Report* and also enable us to remind people who have dropped out or failed to subscribe of *Fractal Report*'s continued existence.

I hope that Dr Michael Ecker of *REC* won't mind me repeating a bit of his editorial in its latest issue, but I strongly agree with his sentiment. At the same time it will show what good value *Fractal Report* offers its subscribers: *Besides, I think the world has gone mad with the proliferation of those overpriced newsletters. Do any of you get the same "offers" that I get? One I got recently was from the Seybold group, with just one of its monthlies for a "mere" $295 per year, $100 off the usual $395. Another company offers its monthly newsletter of business communication for "only $432"; or Software Direct Marketing Newsletter at "just $495".*

To which I would add that I have noted how many of those prices end in "95". Speaking of direct software marketing, how many of you get those tempting offers from Borland International, with hundreds of pounds knocked off their software prices. But they still seem to see a need to hide the true cost by quoting the price as so much plus a huge packing bill plus VAT. I am pretty sure that if you can't see round that marketing policy you'd never stand a chance of getting to grips with products such as their *C+ +*.

Announcements

Reader's Hall of Fame

No entry again for magazine articles, I am afraid. I wish someone could get an article in *New Scientist* mentioning *Fractal Report*! But we are going to get another mention from Dr Pickover in a new book title unknown to appear in August. I have been sent a preview chapter concerning π and DNA. It would appear that this book will be as good a read as *Computers and the Imagination.*

More from REC

REC, the recreational and educational computing newsletter produced a double issue in October. Featured on the front cover was a Halley map configured to produce a little "man" in its centre with arms, legs, ears, eyes, nose and two other round bits in a humorous position! This heralded a long article inside on the fundamentals of BLOAD/BSAVE and plotting images with GWBASIC. Maybe some think that in the days of *Fractint* all this is unnecessary, but those interested in education will find such articles essential reading. And this also applies to those who like to have the fullest possible control of their computers! A fractal Valentine Card program was also included as the usual diet of mathematical puzzles and similar material. Still no mention of the PI theorem though, the one that lets you calculate the digits from the nth digit onwards. (909, Violet Terrace • Clark's Summit • PA18411 • USA)

And Another Issue of Amygdala

Amygdala 25 starts with a reprint of its piece of Math Fiction *The Amygdalan Sects* – slightly reminiscent of the strange doom – SF novels of J G Ballard. Of particular interest to *Fractal Report* readers is its main article, which details various methods of getting images from the interior of the Mandelbrot set, although many of these have also been covered in our pages. The other article is a description of The Secant Method, a mathematical process similar to Newton's Method, and which also generates a range of fractal images. These are presented in this issue as black and white prints and also in the colour slides which come with the newsletter. Their $30/rectangle advertisement page had nothing that I recognised as being new, and their 1991 fractal calendar is still for sale at $4.95 plus postage. It is well worth it for the colour images, some of which are by Ian Entwistle, and these can always be cut out and framed. I should have thought that they cost far more than $5 *each* to print. (Box 219 • San Christobal • NM 87564 • USA)

As of 18 July 1991, their circulation had fallen to 518 subscribers. The circulation of *Fractal Report* at the end of this volume was 25% down on that at the end of the last volume, and maybe this heralds a decline in interest in fractals amongst computer enthusiasts.

Romanian Computer Magazine

From that issue of *Amygdala*, some news of a computer magazine starting in Romania. The organisers request help in the form of articles (or money!): Mr Marius F. Danca • ProInformatica • PO Box 524 • 3400 Cluj – 9 • Romania.

A Rival to Fractint?

A UK produced public domain PC program has appeared that claims speed improvements over *Fractint* in certain configurations. *CAL* fits on a 360K disk and will run in 250k of free memory. At present only the Mandelbrot Set and Henon Attractor are supported, but the program has been written so that other types will be introduced in subsequent editions (and possibly by the time this report appears). Also improvements scheduled will include 30,000 decimal places arithmetic for extreme zooms, data compression of saved images, more accurate image timing, and improved colour palette design. Copies are available for a blank disk and SAE, or £2.20 to include disc and postage. Overseas readers are asked to send a blank disk plus IRCs. Mr Timothy Harris • 5, Burnham Park Road • Peverell • Plymouth • Devon • PL3 5QB.

More on FracGen 2

The Amiga program *FracGen 2* now has different marketing arrangements since the last issue went to press. It is not public domain itself, but is sold by public domain libraries for a fixed price of £3.50. (IE although it is priced the same as the copying fee on some PD or shareware, you are not authorised to copy it and give it away to your friends.) *Fractal Report* readers may buy copies by post from Deja-Vu Software, 25, Park Road, Wigan WN6 7AA, ask for *FracGen 2,* disk no LPD36 and send £3.50 plus your name and address. I am told that the disk also contains a *Fractal Report* advertisement!

Chaos Theory Bulletin Board

Mr Joe Pritchard is setting up a bulletin board dedicated to Chaos Theory. He will send an information sheet on protocols for an SAE or contact via Compuserve. 27, Walkley Crescent Road • Walkley • Sheffield S6 5BA: Compuserve 100010,2243.

Shareware by Telephone

Mr B.R. Holgate sent us details of the Cyberspace Gateway, a bulletin board system that claims the biggest shareware library in Europe, totalling 6 Giga Bytes. There are also teleconferencing systems, 200 sub-bulletin boards, shopping, classified ads, E mail, MPGs, MUGs, and virus checking. The system is available 24 hrs/day. The annual charge is a pound less than £100, or consumers can pay monthly @ £10 (annual total £120) (plus telephone costs to BT or Mercury, of course.) The modem setting is 300, 1200 or 2400 baud and Cyberspace will supply 2400 baud modems for 5p under £50. Tel 071-580-6433 (modem) or 071-323-1552 (voice), address 1, Malcom Drive • Surbiton • Surrey KT6 6QS.

Physics Software Includes Chaos Titles

Professor John S. Risley is the editor of *Physics Academic Software* and he has sent details of his range of products that include three titles that offer demonstrations of chaos theory for college lecturers and students. Two of them are collections of "interactive colourful demonstrations illustrating chaos in physics and biological systems". The third is described as a "utility that allows the advanced college physics student or researcher to perform interactive numerical experiments on nonlinear systems modeled by ordinary differential equations." Each title costs a shade under $70 and is for the PC. Educational users are warned that if more than one copy of the program is to be used by a class a site license must be bought. For ten users this costs about three times the cost of one program. Some of the other titles are being issued for the Macintosh and therefore it is reasonable to speculate whether this will happen to the chaos programs as well. For further details please write to him at the Dept of Physics • North Carolina State University • Raleigh • NC 27695-8202 • U. S. A.

Fractals in the Pop Music World

Mr Robert Wahlstedt sent a letter with his renewal joining in the discussion about fractal artwork on albums and CDs. He said that *The Soup Dragons* doesn't use fractal generated music, but they have used fractals in their videos. He said that many groups are using fractal art, including Mike Oldfield. There is a genre of music called "Techno" in Europe which uses a lot of fractal art, and fractal verbal imagery is used by a Swedish group *The Butterfly Effect.* Their titles include *Deep Julia Dreaming* and *Seahorse Valley.* But Mr Wahlstedt does not know of any group specifically using fractal methods to generate music. He suggests *Fractal Report* readers generating fractal music send in examples that we can put together to form a *Fractal Report* music cassette.

Shareware Shake-Up

PC-Star has issued a circular stating that it has further reduced prices to £1.50 per disk for one to one copying at 360K. If you chose 1.2M disks, they cost £2.50, but can contain the contents of up to three 360K disks. However there is a £1 surcharge per orders not paid by postal orders or cash. This is due to bank charges for inpayment of cheques, and furthermore non-members also have to pay an additional £1 per order. But if you do pay by cheque, make it out to "Paul Perera" not PC Star, as the company account will be closed. (Charges are higher for company accounts. – This goes to show bankers and other professionals that money grubbing loses custom in the long run.)

User supplied disks will be filled at a copying charge of £1 per disk.

Mr Perera believes that his operation offers the lowest prices ever for Shareware, but he is obviously having problems maintaining his business as he plans to review whether to continue at the end of 1992. So if you support low prices and business efficiency, then please support his company with your shareware needs this year.

He has an enormous range of shareware, including games, clip art, windows applications, programming languages and even a special "hot sheet" for "adults", available on special request.

Mr Perera also has some second hand books and PC software for sale at low prices. Contact him at PC Star, PO Box 164, Cardiff CF5 4SF, tel 0222 593476.

DOUBLE POSITION FOR FUNCTIONS OF A COMPLEX VARIABLE

Paul Gailiunas

In *Fractal Report 17* I suggested the method of double position as a source of computer images. Since I had arrived at this idea after thinking about John Topham's article in the previous issue it is not too surprising that he should come up with the same idea at the same time. Extending the method to complex numbers is obvious enough in principle (although in practice there are a few small problems) and leads to interesting fractal images.

The algorithm produces an improved estimate, x_3, of a root of an equation, $f(x)=0$, given two estimates, x_1 and x_2, by applying the formula:

$$x_3 = \frac{x_1 f(x_2) - x_2 f(x_1)}{f(x_2) - f(x_1)}$$

This becomes very messy if complex numbers are used because the quotient must be converted to a real number by multiplying by its complex conjugate. Using $\mathcal{R}$ and $\mathcal{I}$ to indicate real and imaginary parts respectively the formula becomes:

$$\mathcal{R}x_3 = \frac{\mathcal{R}x_1 |f(x_2)|^2 + \mathcal{R}x_2 |f(x_1)|^2 - \text{cross}_1 (\mathcal{R}x_1 + \mathcal{R}x_2) + \text{cross}_2 (\mathcal{I}x_1 - \mathcal{I}x_2)}{|f(x_2) - f(x_1)|^2}$$

$$\mathcal{I}x_3 = \frac{\mathcal{I}x_1 |f(x_2)|^2 + \mathcal{I}x_2 |f(x_1)|^2 - \text{cross}_1 (\mathcal{I}x_1 + \mathcal{I}x_2) - \text{cross}_2 (\mathcal{R}x_1 - \mathcal{R}x_2)}{|f(x_2) - f(x_1)|^2}$$

where

$$\text{cross}_1 = \mathcal{R}f(x_1)\mathcal{R}f(x_2) + \mathcal{I}f(x_1)\mathcal{I}f(x_2)$$

$$\text{cross}_2 = \mathcal{R}f(x_1)\mathcal{R}f(x_2) - If(x_1)\mathcal{I}f(x_2)$$

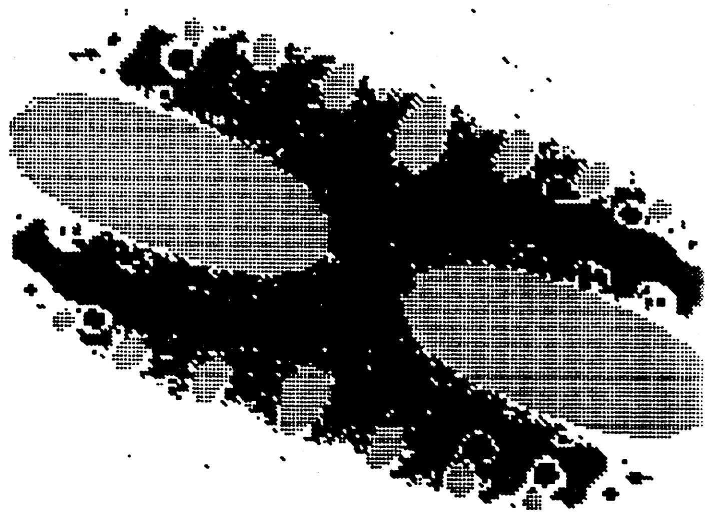
$$|z|^2 = (\mathcal{R}z)^2 + (\mathcal{I}z)^2$$

It would take four dimensions to display fully the way this algorithm behaves, and there are obviously many ways of taking a 2-D slice through this space to get a simple image. It might also be possible to take 3-D sections and to display aspects of them, given the computing power. However I am satisfied if I can get a reasonable image in as straightforward a way as possible. The simplest function which is likely to produce anything interesting is the cubic, $f(x)=x^3-1$ (i.e. $\mathcal{R}f(x)=(\mathcal{R}x)^3-3\mathcal{R}x(\mathcal{I}x)^2-1$, $\mathcal{I}f(x)=3(\mathcal{R}x)^2\mathcal{I}x-(\mathcal{I}x)^3$), and the most obvious strategy is to fix one of the starting values, allow the other to vary, and colour the corresponding point in the complex plane according to which (if any) of the cube roots of unity is produced. Slightly different images occur depending on whether the fixed starting point is x_1 or x_2. In either case a value of zero produces a symmetric, six-pointed star-like pattern (what if $f(x)=x^4-1$?), and it is immediately obvious that the algorithm fails to converge for most of the complex plane (unlike Newton's method). If the fixed starting point is not zero, then a distorted version of the star shape is produced. I have not investigated these distortions, but animation techniques might be useful.

Images from Newton's method applied to the one parameter cubic, $f(x) = x^3+(c-1)x-c = (x-1)(x^2+x+c)$, are well known. There is a problem, however, in using the above strategy with it: where should the fixed starting point be to get an undistorted image? Interesting behaviour occurs near turning points of the function when double position is used with real variables, and it turns out that these are the values to use. Turning points occur when $3x^2+c-1=0$.

Once again the algebra gets messy, because of the need to find the square roots of complex quantities. It is easy if the complex number is in the form $r(\cos\theta+i\sin\theta)$, but unfortunately this is inconvenient to use in the iteration formula. If c = a+ib then the roots of the equation are 1 and -½±(SQR((SQR((4a-1)²+16b²)+1-4a)/8) + iSQR((SQR((4a-1)²+16b²)-1+4a)/8)), and some care is needed over the signs of the square roots! The turning points are at x = ±(SQR((SQR((a-1)²+b²)-a+1)/6) + iSQR((SQR((a-1)²+b²)+a-1)/6)). I have given a listing for Amstrad PCW with Streamlined BASIC extension. It should be easy to modify for different starting points or different functions.

The general form of the image does not seem to depend on the value of c (except when c is close to 1 and a more or less star shape results), nor on which of the two turning points is used, nor on whether the fixed starting point is used for x_1 or x_2, although the details change. Considering the effect of c suggests the Mandelbrot type of plot recalled by John Topham in *Fractal Report 16* (vary c instead of the starting point). Again there is a problem over which starting points to use, and again an answer is to use the turning points (both of them this time). This means that the starting points depend on c, unlike the image from Newton's method where the iteration always starts from 0. Nevertheless what is produced looks remarkably similar to it, although it differs in detail. In particular I can find no copies of the Mandelbrot set.



```
10 PRINT"Parameter?"
20 INPUT"real";a
30 INPUT"imaginary";b
40 r1=1;i1=0
50 disc=SQR((4*a-1)*(4*a-1)+16*b*b)
60 u=SQR((disc+1-4*a)/8);v=SQR((disc-1+4*a)/8)
70 IF b<>0 THEN v=-v*SGN(b)
80 r2=-0,5+u;i2=v
90 r3=-0,5-u;i3=-v
100 PRINT"Coordinates of centre ?"
110 INPUT"real";rx0;INPUT"imaginary";ix0
120 INPUT"Range from centre";range
130 LDW d;LDW n;LDW c
135 rdisc=SQR((a-1)*(a-1)+b*b)
136 rstart=SQR((rdisc-a+1)/6);istart=SQR((rdisc+a-1)/6)
137 IF b<>0 THEN istart=-istart*SGN(b)
140 FOR x=0 TO 358 STEP 2
150 FOR y=0 TO 254 STEP 2
160 rx1=rstart;ix1=istart
170 rx2=rx0+(x-179)*range/179;ix2=ix0+(127-y)*range/159
180 rfx1=rx1*(rx1*rx1-3*ix1*ix1)+(a-1)*rx1-b*ix1-a;ifx1=ix1*(3*rx1*rx1-ix1*ix1)+b*rx1+(a-1)*ix1-b
190 FOR j%=1 TO 20
200 rfx2=rx2*(rx2*rx2-3*ix2*ix2)+(a-1)*rx2-b*ix2-a;ifx2=ix2*(3*rx2*rx2-ix2*ix2)+b*rx2+(a-1)*ix2-b
210 IF ABS(rfx2)>1E+10 OR ABS(ifx2)>1E+10 THEN GOTO 360
220 denom=(rfx2-rfx1)*(rfx2-rfx1)+(ifx2-ifx1)*(ifx2-ifx1)
225 IF ABS(denom)<1E-10 THEN GOTO 360
230 f12=rfx1*rfx1+ifx1*ifx1;f22=rfx2*rfx2+ifx2*ifx2
240 cross1=rfx1*rfx2+ifx1*ifx2
250 cross2=rfx1*ifx2-ifx1*rfx2
260 rz=rx1*f22+rx2*f12-cross1*(rx2+rx1)+cross2*(ix1-ix2);rz=rz/denom
270 iz=ix1*f22+ix2*f12-cross1*(ix2+ix1)-cross2*(rx1-rx2);iz=iz/denom
280 rx1=rx2;ix1=ix2;rx2=rz;ix2=iz
290 rfx1=rfx2;ifx1=ifx2
300 IF ABS(rx1-rx2)<0,00001 AND ABS(ix1-ix2)<0,00001 THEN GOTO 320
310 NEXT j%
320 REM
330 IF ABS(rx2-r1)<0,001 AND ABS(ix2-i1)<0,001 THEN GOSUB 390
340 IF ABS(rx2-r2)<0,001 AND ABS(ix2-i2)<0,001 THEN GOSUB 400
350 IF ABS(rx2-r3)<0,001 AND ABS(ix2-i3)<0,001 THEN GOSUB 410
360 NEXT y
370 NEXT x
380 END
390 LDW g,x,y;LDW g,x+1,y;LDW g,x,y+1;LDW g,x+1,y+1;RETURN
400 LDW g,x,y;LDW g,x+1,y+1;RETURN
410 LDW g,x,y;RETURN
420 LDW g,x,y;LDW g,x+1,y;LDW g,x,y+1;RETURN
```

TAKING PHOTOGRAPHS OF COMPUTER SCREEN IMAGES USING FOCAL PLANE SHUTTER CAMERAS

By Roger Castle-Smith FIEE C(Eng)

I refer to the excellent article written by Mr Howard Jones in FRACTAL REPORT 17. There is just one piece of information which I would like to add as a result of my own experiences.

To start with I was getting light or dark bands across my pictures. The penny soon dropped in that parts of the film were being exposed more than others. If the speed setting of a focal plane shutter is not precisely divisible by the period of one complete CRT scan then clearly the scanning spot will traverse the film (n) times in some areas and (n+1) times in the remainder. If (n+1)/n is large, say 2, then the exposure difference between two areas will be quite noticeable. After some trial and error I found that an exposure difference of 5:4 could just be tolerated without exposure difference bands being objectionable.

The position and widths of the bands will depend on the camera speed setting, the timing of the shutter starting to open relative to the CRT spot position at that instant and the characteristics of the individual focal plane shutter. The complexity of the situation is compounded by the fact that up to a given speed setting, typically 1/60th second, one shutter blind opens fully and this is then followed by a pause before the trailing blind starts to close. Above this speed the trailing blind starts to close before the leading blind has finished moving; thus a slit traverses the film to give ever increasing speeds as it gets narrower. I wrote a short programme to demonstrate the effect. It would be a waste of space to give a full print out but the following low resolution copy demonstrates the idea.

```
2222222222222222
2222222222222222
2222222222222223
2222222222222333
2222222222233333
2222222223333332
2222222333333222
2222233333322222
2223333332222222
2333333222222222
```

For a particular speed setting and a random shutter opening time the CRT spot scanned the film twice in the areas shown as "2" and thrice in the remainder. Re-runs using a random number generator to determine the shutter opening instant in relation to the start of the CRT scan moved the band and with certain timings it split into two as might be expected; diagonally across the top left hand and bottom right hand corners. In practice the bands may be slightly curved due to the shutter acceleration characteristics. So a golden rule is to select a speed setting which is not less than about 5 times as long as the CRT scan period. If the former is then not en exact multiple of the latter exposure differences over the film area will not be noticeable. A half second or longer usually meets this requirement. Incidently Hewlett Packard recommend bracketing the exposure around 1 second at f5.6 with ASA64 film; this advice accords almost exactly with Mr Jones' experience.

The contents of this article are perhaps obvious. But if, in conjunction with Mr Jones' article, it saves other people wasted film and a few rude mutterings when film is received back from processing then it will have served its purpose !