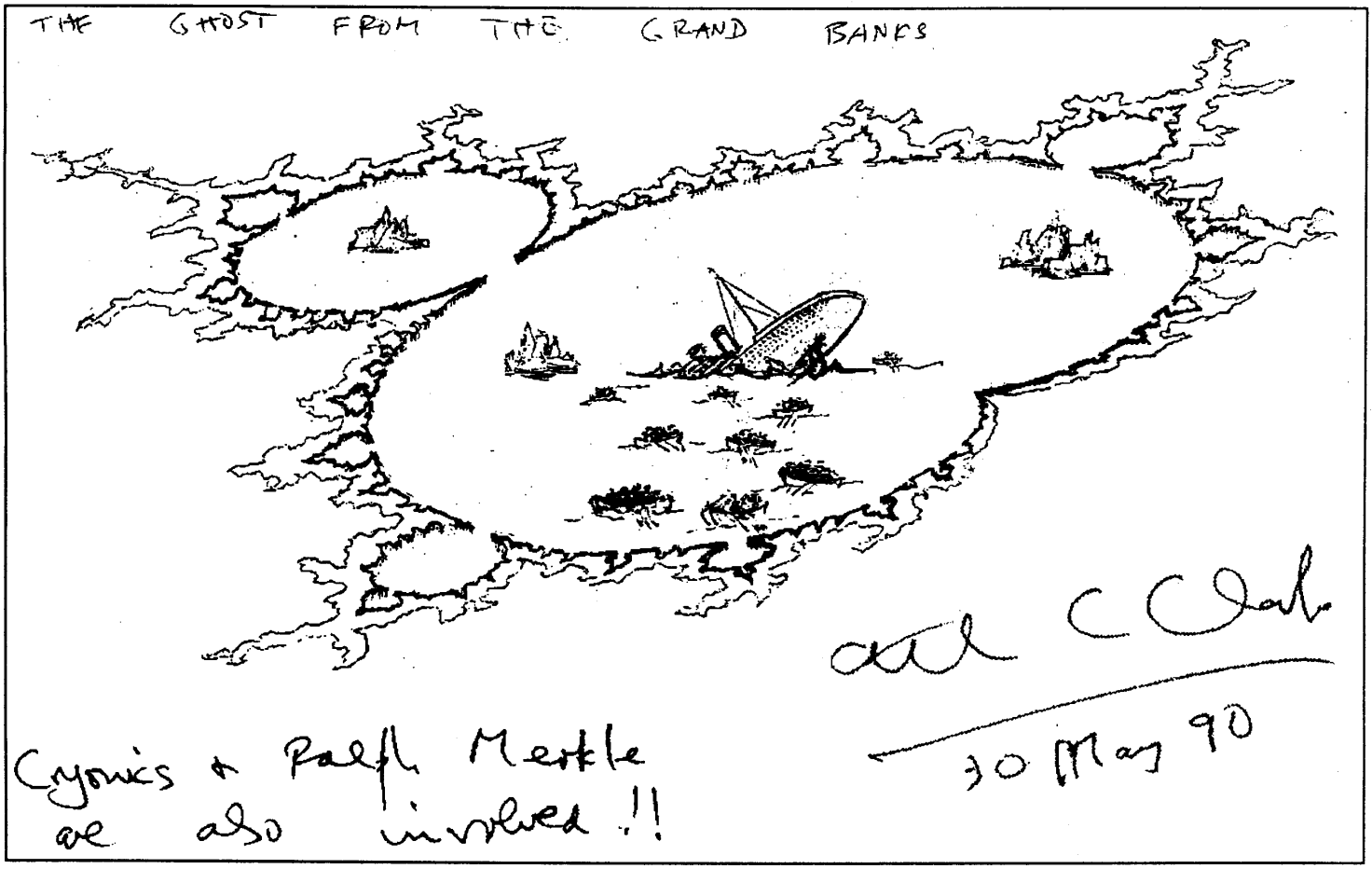
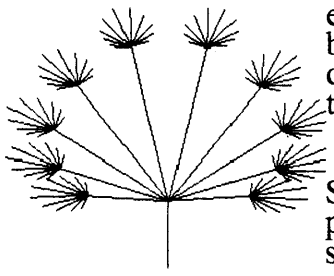
FRACTAL REPORT 10



Arthur C. Clarke's novel *The Ghost from the Grand Banks,* featuring the Mandelbrot set, cryonics, and the *Titanic* will be published by Gollancz in October, and Bantam in December, 1990

Wallpaper	Peter Moon	2
Julia Sets – Part II	Dr Ian Entwistle	4
Fractal File Formats	Larry Cobb	7
More Martin's Mappings	Dr Hugh N. Daglish	8
Bifurcated Sounds	Leon Heller	13
Beyond Mandelbrot	Tom Marlow	14
Ginger Bread Men	Tom Marlow	15
Feigenbaum Diagram in C for the PC	Leon Heller	16
Editorial	John de Rivaz	17
Announcements	John de Rivaz	18



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".

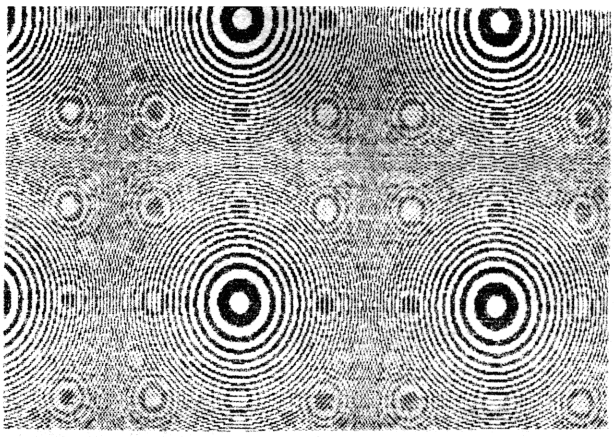
Subscribers who are successful in getting one or more articles with programs published in a given series of six issues get the next volume of six issues free of subscription. All new subscriptions are backdated to the start of the current volume.

Fractal Report is published by Reeves Telecommunications Laboratories Ltd., West Towan House, Porthtowan, Truro, Cornwall TR4 8AX, United Kingdom.
Volume 2 no 10 First published August 1990. ISSN applied for.

WALLPAPER

There is an extremely simple program due to John E. Connett which is capable of producing surprisingly complex patterns resembling wallpaper, persian carpets and tesselated pavements. You can find the details in "Scientific American" for July 1986 but essentially, all the program does is to take take the x and y co-ordinates of all the points in a gridded square, calculate x^2 + y^2 for each point, truncate the result to an integer, and plot the point if the result is even, blank if odd. So that you can rush to your keyboard and make some wallpaper instantly, here is the simplest version of the program (CIRCLE1), without any frills or refinements. Compared with the creeping progress of the Mandelbrot, this one runs like a train.

```
10 REM CIRCLE1 P.W.H. Moon 1990.  Compaq BASIC.
20 INPUT "Side ";SIDE: REM Try 20 for a start and
30 REM increase in steps of 50%, eg 20,30,45 ...3000.
40 REM Avoid multiples of 200.
50 KEY OFF: REM Gets rid of key list on screen line 25.
60 SCREEN 9: CLS: REM For the highest resolution.
70 FOR I=0 TO 200
80   FOR J=0 TO 200: REM Use whatever number of pixels
90     REM your display provides.  200 x 200 gives me
100    REM a small plot for quick evaluation.
110    X=I*SIDE/200: Y=J*SIDE/200
120    C=INT(X*X+Y*Y): D=C/2
130    IF D-INT(D)>.1 GOTO 180
140    REM Lines 120-130 avoid the use of C MOD 2 to
150    REM test for even numbers, which would result
160    REM in overflow for values of C > 32,767.
170    PSET(I+50,J+50),7: REM Plotting instruction.
180  NEXT J
190 NEXT I
```



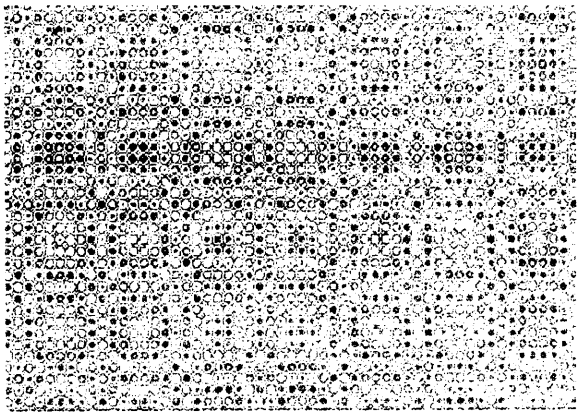
Line 170 plots the point, using colour 7 (white), but you can have fun with colours if you calculate a variable called KOLOR depending on I and/or J and put KOLOR instead of the constant 7 in the plot instruction. The smaller the value of SIDE the larger the magnification of the pattern. As you increase SIDE you are in effect standing further and further away from the pattern. All sorts of elements of larger scale patterns, superimposed on the basic one, appear in shadowy outline as you back away. The patterns are not true fractals, and there is a limit to which you can magnify before detail disappears, but as you increase SIDE, larger patterns of similar form become visible.

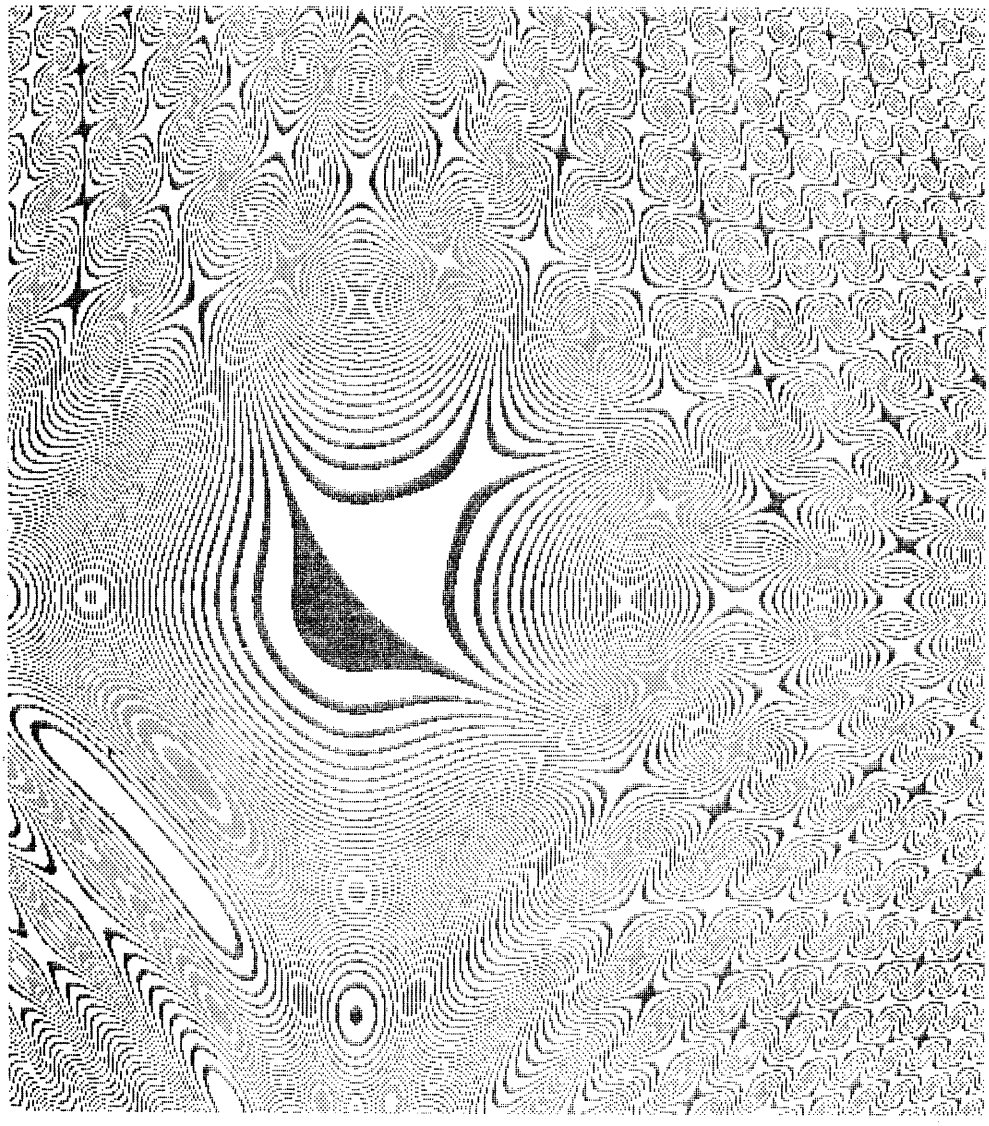
Connet's program used x^2 + y^2, but lateral thinkers will explore x^2 - y^2, cubics, and trigonometrical functions. All seem determined to produce some regular pattern. The one which looks like a spacecraft doing a re-entry manouevre was based on a complicated function involving squares and cubes. If there is too much "ink" in the design, plot only points corresponding to D divisible by 3 or 4. Those without a screen dumping facility can get superbly detailed print-outs by programming bit image plotting on a dot matrix printer. My programs for doing this are too long to print here but if anyone wants listings I'll provide two in Compaq BASIC suitable for an Epson FX or compatible printer (SAE + 20p for the photocopier please).

Finally a slightly more elaborate program (CIRCLE2) which allows you to displace the starting corner to a position away from the origin, and prints the details of the run - handy if you do a lot of screen dumps and can't remember what numbers you used to get them.

Peter Moon, 2 Greymouth Close, Stockton-on-Tees, Cleveland TS18 5LF.

```
10 REM CIRCLE2 P.W.H. Moon 1990.
20 INPUT "SIDE, X CORNER, Y CORNER ";SIDE, CX,CY
30 TSTART=60*(VAL(MID$(TIME$,4,2)))+VAL(RIGHT$(TIME$,2))
40 KEY OFF: SCREEN 9: CLS
50 FOR I=0 TO 200
60   FOR J=0 TO 200
70     X=I*SIDE/200+CX: Y=J*SIDE/200+CY
80     D=INT(X*X+Y*Y)/2
90     IF D-INT(D)>.1 GOTO 110
100    PSET(I+150,J),7
110  NEXT J
120 NEXT I
130 PRINT "SIDE ";SIDE: PRINT "XCORNER ";CX
140 PRINT "Y CORNER ";CY
150 TFIN=60*(VAL(MID$(TIME$,4,2)))+VAL(RIGHT$(TIME$,2))
160 PRINT "TIME  ";TFIN-TSTART: PRINT "  SECONDS"
```





Julia sets

Part II: Mapping of the function Z^2+U in the 1/Z and 1/U planes

By Ian D. Entwistle

Readers of Fractal Report who have studied the text and illustrations of Peitgen's book "The Beauty of Fractals" will be familiar with the inverse Mandelbrot set(see page 75).Fewer readers will be familiar with illustrations of the mapping of the function Z^2+U Julia sets in the inverse plane.For those who are not familiar with inverse mappings a brief explanation of the idea of inversion geometry is needed in order to provide the background information for the partial Basic Listings described to produce the Figs(4 to 8).

In Fig(1) the point P is the inverse of the point Q with respect to the centre of the circle O.Conversely Q is the inverse of P.Clearly every point has its own unique inverse with respect to any given circle and the inverse of the point O will be at infinity.

To perform an inversion with respect to a given circle all the points in the plane are moved to the locations of their inverses.So for the chess-board in Fig(2) inversion produces the Fig(2a) from which it is noted that the petal shaped centre is the whole of the plane outside the sixty four squares ,while the four inner squares of the chess-board now stretch to infinity and are only partially illustrated in Fig(2).From the foregoing it should now be clear as to how the Fig(3),the inverse "M" set relates to the shape of the familiar Mandelbrot set.Note that inversion preserves angles and so all the familiar curves of the "M"set are retained.Magnified parts of the "M"set are not easily distinguished from the normal equivalents.The basic rules are that circles invert to circles except those which pass throuh the centre of inversion.Lines invert to circles unless they pass through the inversion centre.

The utility of this geometry for computing the mapping data lies in the observation (based on '0' level geometry) that the line OQ in Fig(1) is the reciprocal of OP for a circle of unit radius.So for construction of complex function maps Z becomes 1/Z.The real and imaginary parts of Zo need to be derived and then inverted as they are inputted into the iteration loop of the Listing(I).Using conventional complex algebra the equation (1) gives the required values of Zo when Z=a+ib.

1/Z=1/(a+ib)=(a-ib)/(a^2+b^2)=a/(a^2+b^2)-ib/(a^2+b^2)-----Equation(1)

In the Listing(1) a and b are therefore replaced by a/(a^2+b^2) and b/(a^2+b^2) for the starting values of Zo.It should be noticed that 1/(a^2+b^2) need only be calculated once.Running any Julia set listing with the corresponding values of a and b modified will give inverse maps.Inverse mappings derived from points inside the "M"set and those Julia sets that are "dusts" tend to look rather disappointing.This is caused by the large areas out to infinity(or the edge of the map) being unpatterned since they are part of the Julia set.An alternative strategy was therefore adopted for Figs(5-8) in order to achieve pattern out to the map edge.Where a full range of colours is available the effect is quite unusual and appealing.

The minimum value of |Z|^2(CI) is trapped in line 660 of Listing(II) and its value used to control the "colour" of the points in the inverse plane which do not diverge(see references for other examples of this strategy).The values of CI are reals which range over tens of powers of ten so some adjustments are required to scale them to integer values usable to obtain ranges suitable for controlling the colours or printer black/white.Experiments will demonstrate the most interesting values of CI.Those between 50 and 300 are quite acceptable when using double precision arithmetic.

Figs(3-8)were obtained using the data in the Table.In Fig(4) the low iteration values were not printed in order to achieve the propellor shape.Investigation of the window size produces a few surprises.

Fig(3) was obtained by inputting inverse values of real and imaginary parts of U into the standard listing for generating the "M "set.

Table

Fig.	Point in M	Window size		Iterations	Integer
	real imag.	x-min/max	y-min/max		Factor CC%
3		6.1,4.6	3.6,4.1	100	-------
4	0.11031,-0.071	-5.3,5.3	-4.6,4.6	40	100
5	0.15652,-1.03225	-8.0,8.0	-6.0,6.0	40	100
6	0.233 ,0.5378	-8.0,8.0	-6.0,6.0	40	100
7	0.27324,0.00742	-8.0,8.0	-6.0,6.0	40	100
8	-0.481762,-0.531657	-8.0,8.0	-6.0,6.0	40	100

Appendix

For the Listings NI%= y axis of output,MI%= x axis of output,TI%=Iteration value(max IT%),CI=minimum value of |Z|^2,CC%=Integer factor,a=real value of Zo,b=imaginary value of Zo,RC=real value of u,IC=imaginary value U.Figs(4-8) were run at 420*756 pixels.Fig(3) at 920*420 pixels.At A3 size some of these prints look magnificent even in black and white.

References

Comput.& Graphics 13(3),389-392,1989,Julia Set Art and Fractals in the Complex Plane,I.D.Entwistle.

Comput.& Graphics 13(4),549-551,1989,Methods of displaying the Behavior of the Mapping Z--Z^2+U,I.D.Entwistle.

Algorithm 1.1,9-12,1989,Inside the Mandelbrot Set,C.A.Pickover.

Listing 1

```
480 FOR NI%=ZT% TO Z1%
481 BB1=b^2:
490 FOR MI%=0 TO XA%
491 BB2=a^2
500 BB=BB1+BB2
501 Y=b/BB
510 X=a/BB
520 FOR TI%=1 TO IT%
531 X3=X*X:Y3=Y*Y:XY=X*Y
540  X=X3-Y3+RC
550  Y=XY+XY+IC
580CS=X3+Y3
600 IF CS>10 THEN GOTO 680:REM GOTO PRINTER OR SCREENOUTPUT STATEMENT
640  NEXT
```

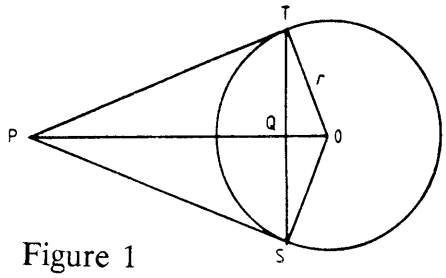


Figure 1

Listing 2

```
530 FOR NI%=ZT% TO Z1%
540 BB1=b^2:
550 FOR MI%=0 TO XA%
560 BB2=a^2
570 BB=BB1+BB2
580 Y=a/BB
590 X=b/BB
600 CI=25
610 FOR TI%=1 TO IT%
611
620 X3=X*X:Y3=Y*Y:XY=X*Y
630  X=X3-Y3+RC
640  Y=XY+XY+IC
650CS=X3+Y3
660  IF CS<CI THEN CI=CS
670 IF CS>10 THEN GOTO 720
680  NEXT
690CI=INT(CI*CC%)
700    REM:PUT OUTPUT TO PRINTER HERE THEN GOTO 730
710
720 REM: PUT OUTPUT TO PRINTER HERE FOR DIVERGENT POINTS
730 NEXT MI%
```

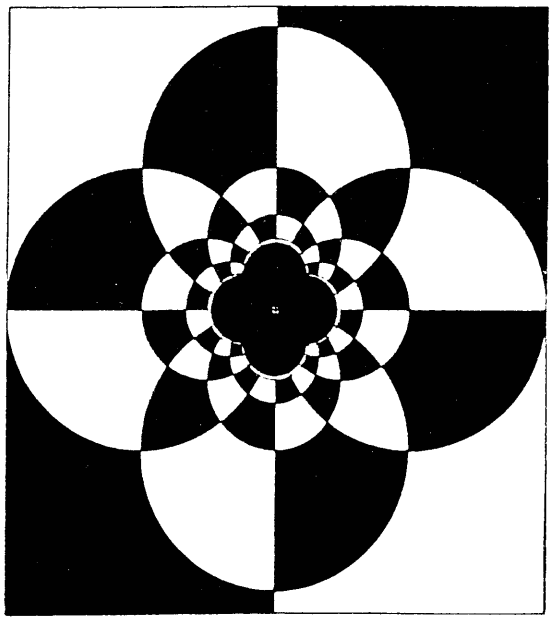


Figure 2a

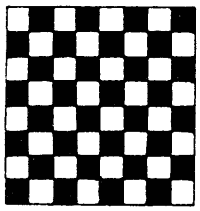


Figure 2

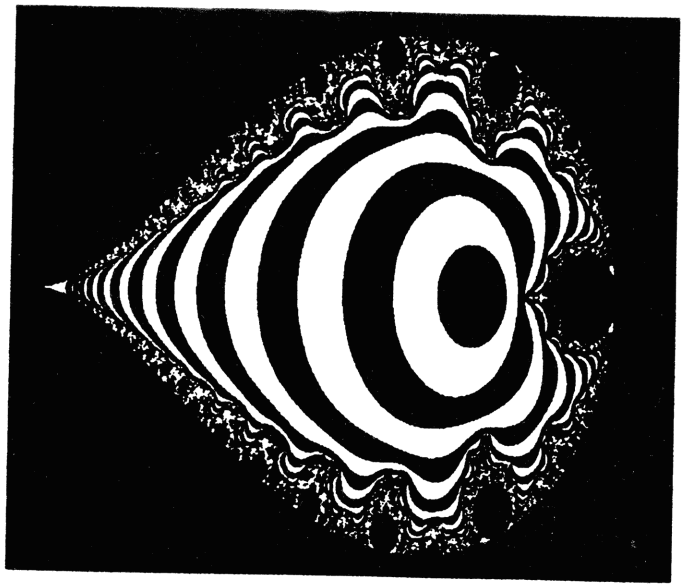


Figure 3

Figure 4

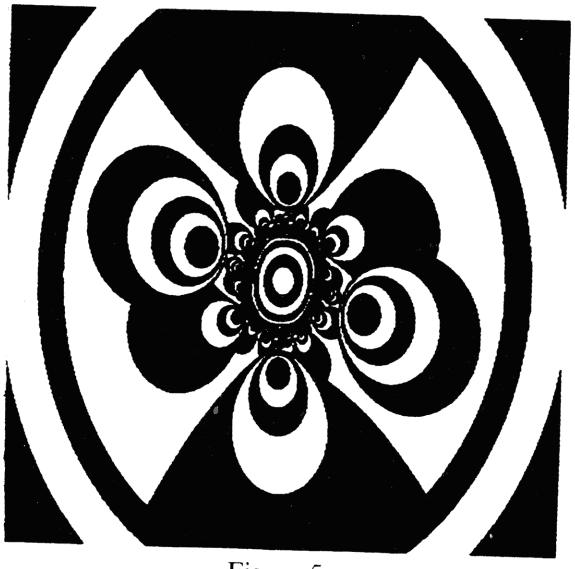


Figure 5

Figure 6

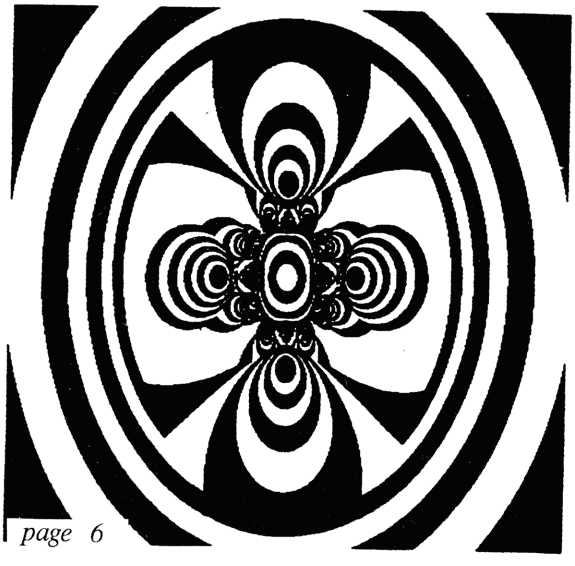


Figure 7

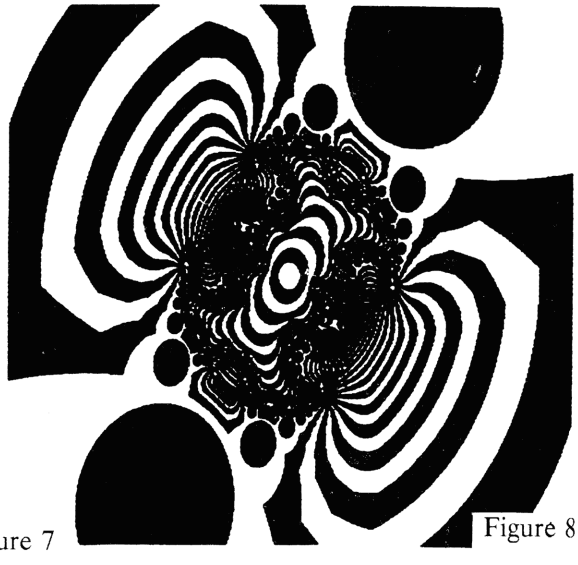


Figure 8

Fractal File Formats

by Larry Cobb

All graphics files tend to have one thing in common – they take up a lot of disk space, and fractal images are no exception! If a VGA image were represented by one of a possible 256 colours for each pixel on a VGA screen, then the file size would be 480 x 640 bytes, requiring a minimum 307,200 bytes of disk space. (In practice there would be extra bytes needed to describe the graphics parameters.)

The current version of my Fractal Investigation Program, DRAGONS 2, uses a relatively simple method of reducing the file size called Run Length Coding (RLC). Raster scanned images often have areas of constant colour and so, instead of describing the screen point by point, RLC defines the colour and then says how many adjacent pixels have that same colour. In this way a run of 256 pixels with the same colour can be compressed from 256 bytes to 2 bytes. Of course this is an extreme example but file compressions of 5 are common with fractal images. The weakness of this method is that if the colour changes every pixel, RLC actually uses more space that the uncoded data – two bytes per pixel instead of one.

There are more powerful compression techniques that construct a look-up table of the patterns that are present in the data and assign a code to each. When the pattern recurs in the data, it is only necessary to look up the appropriate code in the table and send this. One system using this technique is called the Graphics Interchange Format (GIF) which was designed by CompuServe. They run a large bulletin board or forum service and so have a particular interest in reducing file sizes and hence storage requirements. Also, when you are paying telephone and connection charges, with data rates around 1200 baud, smaller files can save a lot of money!

The GIF standard has many attractive features, including:

- It is one of the most efficient compression techniques available, producing files that are typically half the size of an RLC file.
- It is a hardware independent specification and decoders are available for most popular computers.
- It is freely available to all programmers to use without licence or royalty payments.
- It is widely recognised and more and more programs use it as their file specification, (eg Autodesk's Animator).

Even if it is not directly supported by a graphics program, there are many Shareware converters available. For instance, I have used VPIC to convert GIF to Dr Halo CUT format so that I can add text to a fractal image. Also, there are many Shareware programs that can print and produce slide shows using GIF files. One of the best I know for the IBM PC is called CSHOW.

And of course it is the best way to send fractal files all around the world on the CompuServe network to compare results with other enthusiasts.

Perhaps its main drawback is its complexity – it is not easy to write coders and decoders for the GIF standard. But if you want more information to allow you to write your own coder and decoder, I can supply a copy of the GIF 87a specification on floppy disk or you can get it from CompuServe directly. Also available are the Shareware programs for IBM compatibles mentioned above, namely VPIC and CSHOW. Please remember that Shareware is not free. You can try out these programs for a short time but, if you continue to use them, you must send a registration fee to the author. They are very reasonably priced though, and supporting good Shareware authors encourages them to write more programs.

As I mentioned before, DRAGONS 2 uses RLC files but also on the GIF disk set is a DRAGONS to GIF converter. Watch out for a new release of DRAGONS in the Autumn that will support both standards and have other enhancements – DRAGONS 3 is coming soon!

Finally, I would recommend that anyone who has a modem should try the CompuServe network. There is a whole forum devoted to computer art (GO COMART), a section of which is devoted to fractals. The Fractint program is widely used here and the latest version (13?) can be downloaded, as can any of the graphics files. If you get bored with fractals, there are lots of other forums to explore including computer hardware and games. In fact there seems to be a forum on just about every topic but the bias is towards the American user at the moment (weather, travel etc). It becomes so addictive searching through the forums that you have to keep a careful eye on the connection time to avoid the charges piling up.

In future I intend to upload the fractal competition prize winner to the Computer Art Forum. I think our images are every bit as good as the ones I've seen on CompuServe. Not only national but international fame awaits the winner, so keep those entries coming!

If you are tempted to try CompuServe, you can contact them (free) on 0800 289 378. An introductory kit costs £29.95 but comes with $25 of credit. And if you want to send me a message on the network, my user id is 100016,421.

I would be particularly interested to hear how users of computers other than the IBM PC compatibles get on with GIF files, either by electronic mail or the more conventional post. GIF should be a way to break some of the hardware barriers but I have only tried it on IBM Pcs.

The special GIF software package for IBM PC compatibles contains:

ITSAGIF.EXE	DRAGONS to GIF converter for DRAGONS 2 users
GIFSTD.TXT	A description of the GIF 87a file standard
VPIC.EXE	A Shareware GIF display and format conversion program for IBM PCs
CSHOW.EXE	A Shareware GIF display, printing and slideshow program for IBM Pcs

together with the documentation files and as many fractal GIF files as I can fit onto the disks.

Send a cheque or PO for £5 to cover costs to:
Larry Cobb, Bay House, Dean Down Drove, Littleton, Winchester, Hants, SO22 6PP and don't forget to say whether you want 5¼ or 3½ inch disks.

Send details of your Mandelbrot or Julia set fractals for the competition entries to the same address, please. Despite my pleas, there have been no Julia Set entries for the competition.

Julia sets are slightly more difficult to find, so here are some tips to help those who would like to try. The trick is to use the Mandelbrot set as a map to where the Julia dragons are hiding. Find the co-ordinates by moving a cursor around the edge of the Mandelbrot set, in and out of the infinite sprouts or caves. This will give you the complex c value that is needed to calculate the Julia set at this point. If you move deeper into the set (the black inner area) then the Julia set becomes fatter. These are the best ones for "tiling" – a method where the inside of the set is coloured depending on the attractor or root value. If you cross the boundary into the area of chaos, the dragon begins to break up in "Fatou" dust. This too gives an attractive result.

The number of filaments which lead from the Mandelbrot sprout, determines the number of heads the dragon has, while the swirls of the filaments are transferred to its writhing body. So, with a little experiment with the Mandelbrot map, some fascinating mythological creatures can be found. Julia sets also have the advantage of being quicker to calculate, giving you more time to get some spectacular results.

The winner will receive a high resolution, colour print of their entry and international fame!

More Martin's Mappings

Hugh N Daglish

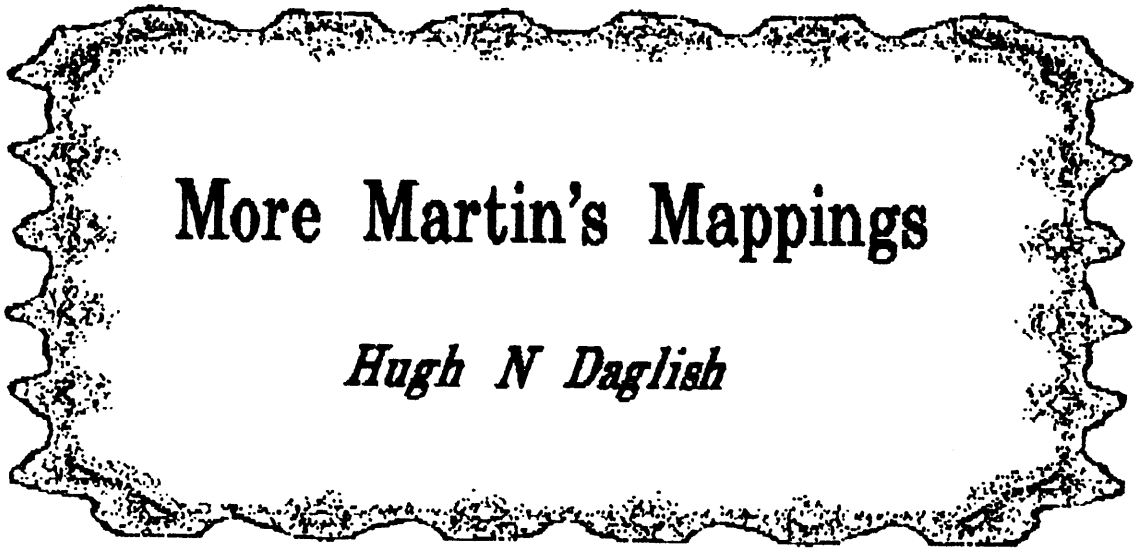


Figure 1

The range of fascinating and decorative patterns which can be produced by the iterative technique named after Dr Barry Martin seems endless. Quite a few such patterns have already appeared in *Fractal Report*. Some further patterns are illustrated here: perhaps they might inspire further exploration!

The illustrations were produced using a fairly standard CBASIC program, which consists of three parts. The first part sets up the graphic procedures for my particular machine, and reads in the starting parameters. The third part provides a safe way of stopping the program to enable the screen to be copied to the printer. The central loop in the program actually draws the picture, and takes the following form.

```
X1  =  Function [ X, Y, PEE, QUE]
Y  = ACON  -  X
X  =  X1
U  =  T * X  +  T * Y
V  = -T * X  +  T * Y
XP%  =  INT%(XOFFSET + U * XSCALE)
YP%  =  INT%(YOFFSET + V * YSCALE)
PSET (XP%,YP%)
IF INKEY$ <> "" THEN GOSUB 8000 'Subroutine to stop display'
LOOP
```

PEE, QUE and CON are the main parameters controlling the pattern. It is also necessary to input initial values for X and Y. The offset and scaling factors are fixed in the program, having been selected to suit my screen and printer. T is always 1/SQR(2).

The notes below should enable other readers to recreate these pictures, although it is important to realize that when one of these programs is transferred to another language or to another machine, the detail of the picture is likely to be affected. Arithmetic precision, rounding conventions and the algorithms used for particular functions can significantly alter the details of the patterns, and also the number of iterations needed. There is plenty of scope for experimentation!

FIGURE 1 (The frame surrounding the title)

```
Function:   X1 = Y - SGN(X)*SIN(X)*SIN(X)
Parameters: PEE and QUE not used.
            CON = PI            XINIT = 20  YINIT = 0
Iterations: 13000
```

FIGURE 2

```
Function:   X1 = Y - SGN(X) * QUE / (1.0 - ABS(PEE * X))
Parameters: PEE = 0.5  QUE = 0.5
            CON = 5.0  XINIT = 0  YINIT = 0
Iterations: 36000
```

FIGURE 3

```
Function:   X1 = Y - SGN(X) * QUE/ (0.000001 + ABS(PEE * X))
Parameters: PEE = 1  QUE = 1
            CON = 1  XINIT = 25  YINIT = 0
Iterations: 22000
```

FIGURE 4

```
Function:   X1 = Y - QUE * EXP(-ABS(PEE * X))
Parameters: PEE = 1  QUE = 1
            CON = 3  XINIT = 2.871  YINIT = 1
Iterations:  22000 (Curve heads for infinity)
```

FIGURE 5

```
Function:   X1 = Y - QUE * LOG(ABS(PEE * X))
Parameters: PEE = 0.5  QUE = 0.5
            CON = 0.5  XINIT = 1  YINIT = 1
Iterations: 7280 (Curve heads for infinity)
```

FIGURE 6 (This is one of a family of similar patterns where the function includes SIN(X) raised to an integer power. Where the power is 1, the function is that published as AXHEAD.)

```
Function:   X1 = Y - (SIN(X))^N     when N is odd; or
           X1 = Y - SGN(SIN(X)) * (SIN(X))^N when N is even.
Parameters: N = 2
            PEE and QUE not used
            CON = PI + 0.7        XINIT = 1  YINIT = 1
Iterations: 20000
```

FIGURE 7 (Another of the same series)

```
Parameters: N = 5
            PEE and QUE not used
            CON = PI + 0.5        XINIT = -5  YINIT = 0
Iterations: 13000
```

NOTES

Although Figure 7 was plotted using the CBASIC program already described, W E Thomson and I have plotted exactly the same function on three different machines, using four different dialects of Basic and one of Pascal. In each case, the overall pattern was the same, but the fine detail different. In some versions, the central four elements appear emphasized, with outer patterns quite feint, whereas in the others, the central elements appear only as ghost outlines.

In most of these examples, the number of iterations is large, but this is necessary if the full pattern is to develop, because the programs often reach apparent stability, followed by further periods of rapid change. For example, in Figure 6, the four central elements build up rapidly, and then little appears to happen other than a thickening of the lines, until at about 7200 iterations, the pattern begins to move outwards to colonize the whole screen.

As a further experiment, this program was translated to QuickBasic, compiled and rerun on a PC, using the same parameters. With this version, the four initial elements grew to eight after 3400 iterations. These eight then remained with little change, until after 11400 iterations, expeditions rapidly set forth towards the four corners, and disappeared from the screen. **FIGURE 8** shows the result after 20000 iterations.

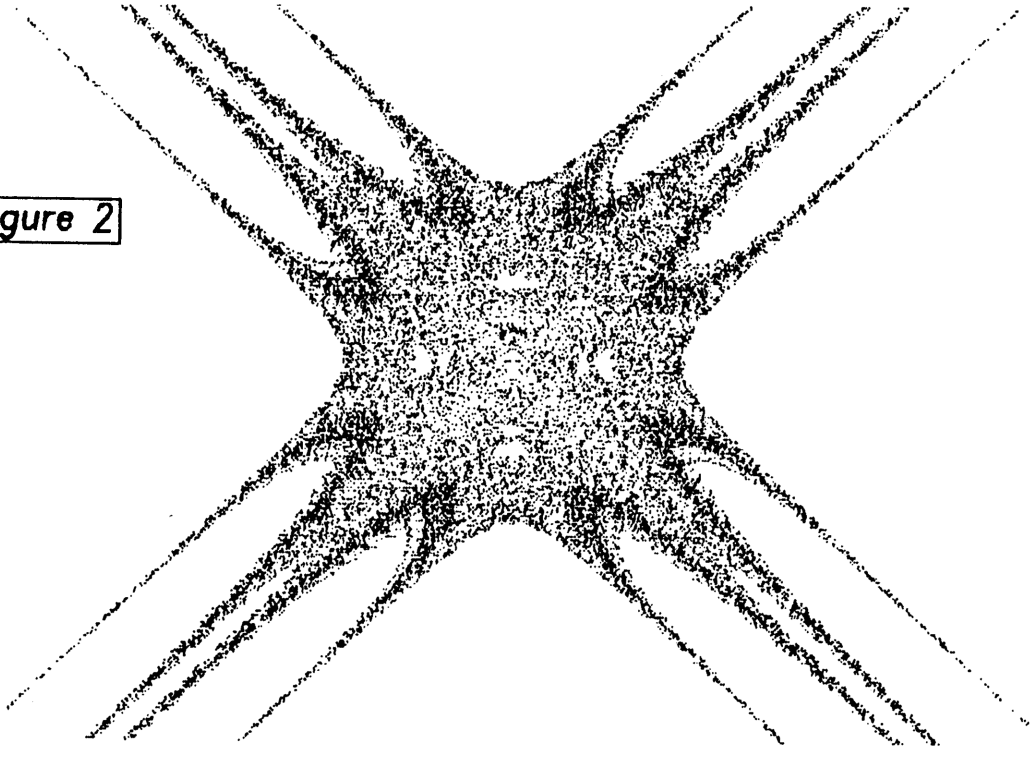


Figure 2

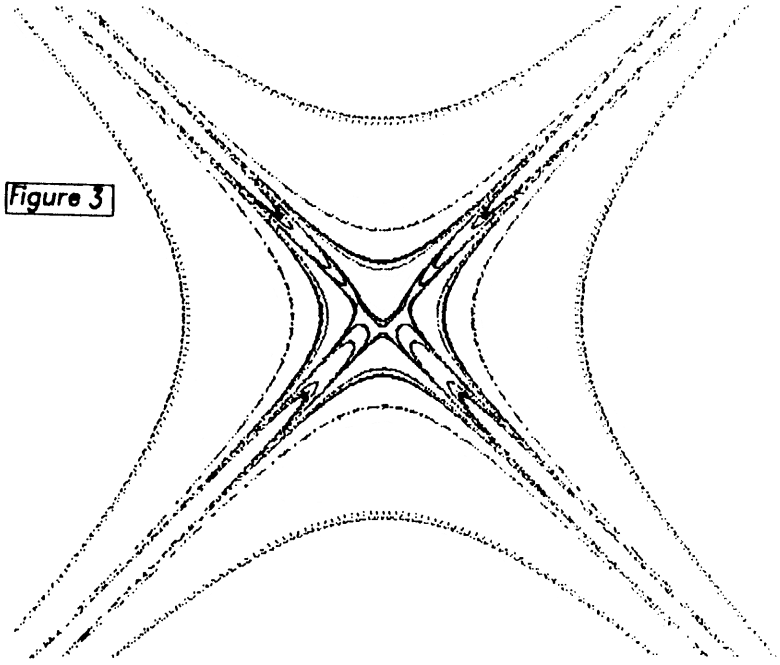


Figure 3

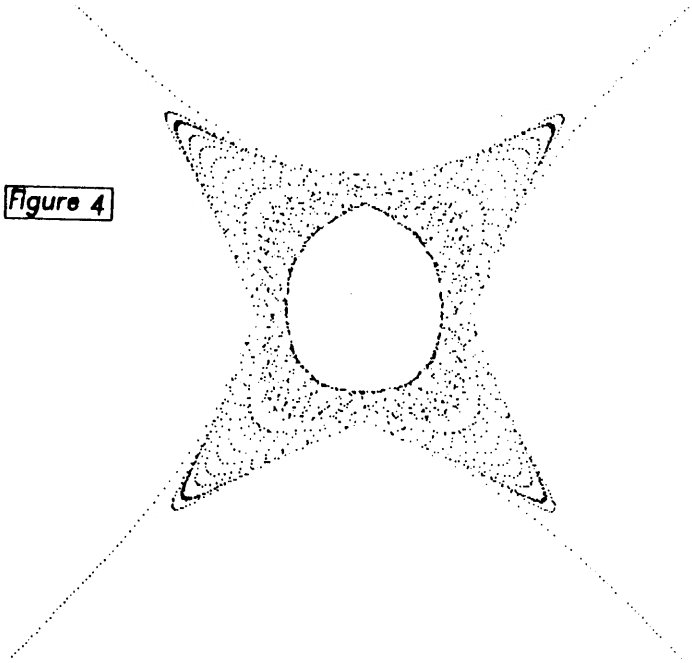


Figure 4

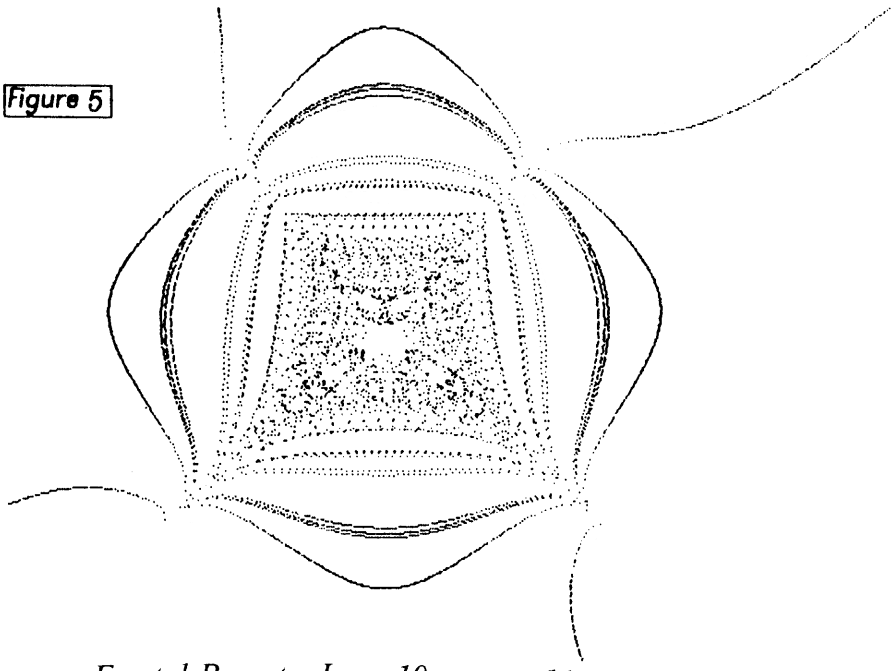


Figure 5

Figure 6

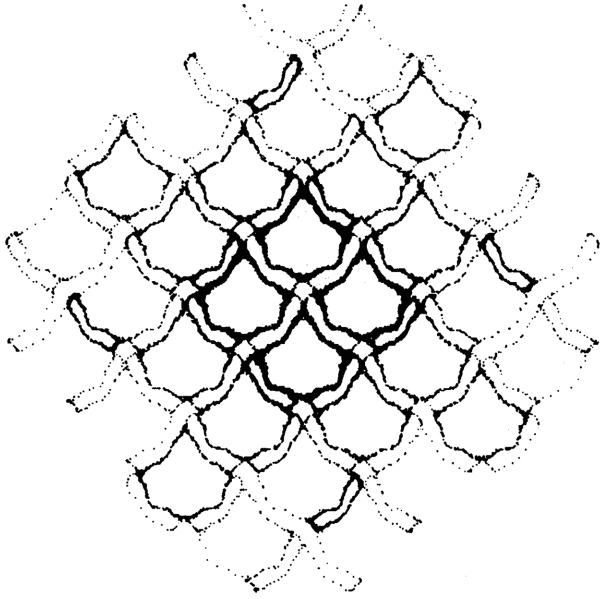


Figure 7

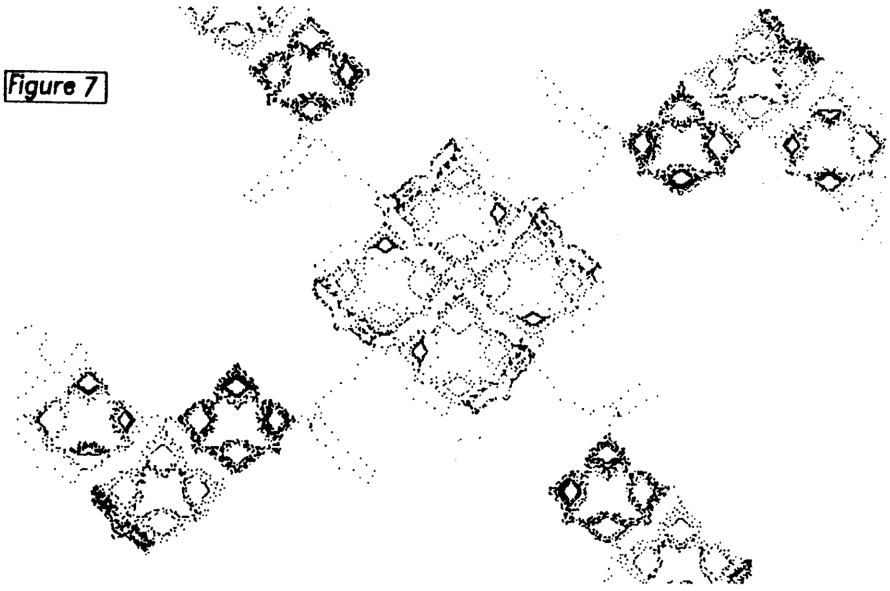
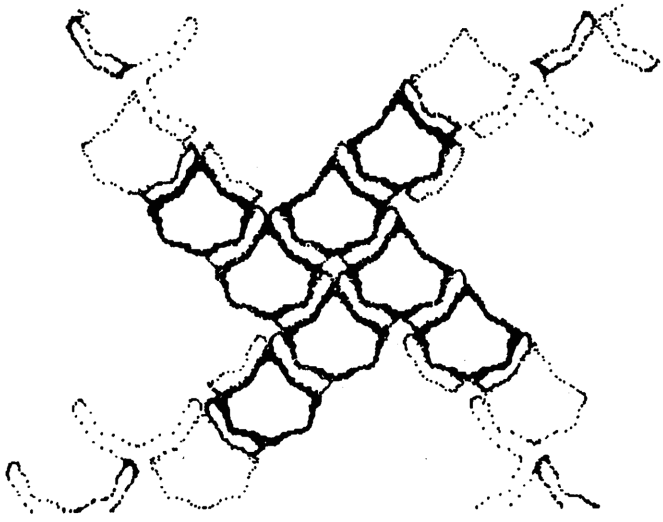


Figure 8



BIFURCATED SOUNDS

by Leon Heller

Following a suggestion in Becker and Dörfler's book, "Dynamical Systems and Fractals", that the reader presents the values generated by a Feigenbaum – type system acoustically, as a series of tones, I wrote the following little program, using the Zortech C compiler, for my Opus PC clone.

```
£include <stdio.h>
main()
{
    float x, r;
    int n;
    printf("Initial value of x: ");
    scanf("%f", &x);
    printf("\nInput growth factor: ");
    scanf("%f", &r);
    printf("x = %f r = %f\n", x, r);
    for (; ;)
    {
        x = r * x * (1 - x);
        n = (int) 100 * x;
        printf("%d\n", n);
        sound_tone(1000, n, n);
    }
}
```

The program in C

Try an initial value of 0.5. Growth factors of more than 3.57 result in chaotic output.

The results don't sound particularly musical (Babbage, my Burmese cat, started howling!), and the recurrent clicks resulting from the PC internal interrupt are a bit annoying, but listening to chaos makes a change!

The first sound tone function parameter is the number of cycles, and the second and third parameters are the output up time and down time, respectively. Making them the same as the number generated by the recurrence formula ensures a 50% duty cycle.

With a MIDI interface and a synthesiser, something more interesting could no doubt be generated. Perhaps someone with some decent musical hardware could follow this up.

Incidentally, the aforementioned book has some material relevant to John Topham's article on bifurcation diagrams, in the region where the behaviour becomes chaotic.

```
start:
if rlast=0 then
  xlast=.5:  rlast=3.8
  fblast=100:ftlast=5000
  dlast  =1  :eflast=1000
end if

screen 2
cls
Print "(enter 999 to quit)"
Print "x must be less than 1"
Print "Initial value of x if not ";xlast; :input x
if x=999 then end
if x>=1 then x=.9
print "Growth factor must be less than 4."
print "Growth  factor  if  not ";rlast; :input r
if r>4 then r=3.9
print "Top  frequency  if  not ";ftlast;:input ft
print "Bottom  frequency  if not ";fblast;:input fb
print "Expansion  factor  if not ";eflast;:input ef
print "Duration   if      not ";dlast; :input d
if d=999 then end

if  d=0 then  d=dlast
if ft=0 then ft=ftlast
if fb=0 then fb=fblast
if  x=0 then  x=xlast
if  r=0 then  r=rlast

if ef=0    then ef=eflast
if fb<37   then fb=37
if ft>32767 then ft=32767

dlast=d:ftlast=ft:fblast=fb:xlast=x:rlast=r:eflast=ef

cls
print "x=";x;" r=";r:Print:print "Invalid Notes: (if any)"
while not instat
  x=r*x*(1-x)
  n=int(ef*x)
  if n>fb and n<ft then
    sound n,d
  else
    locate 4,1:print n ;"  "
  end if
  if getkey$=chr$(27) then end
  if instat then goto start
wend
```

An extended editorial version of the program in Turbobasic

BEYOND MANDELBROT

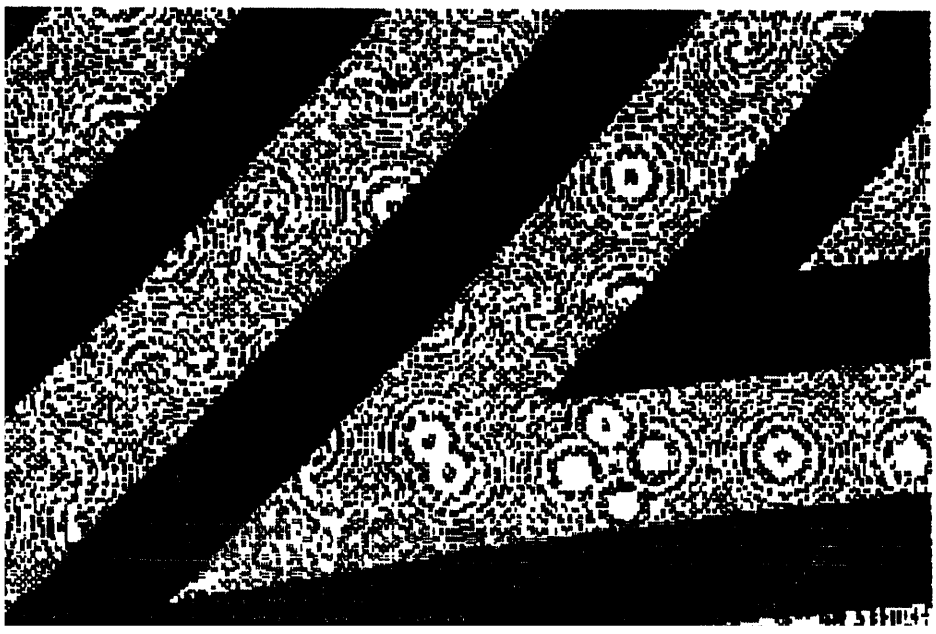
by Tom Marlow

I would like to discuss a Mandelbrot phenomenon which I have not seen described elsewhere.

As we move outwards from the border of the set the 'contour lines' form approximate circles. Large areas between are identified as outside the set with the same (low) number of iterations of the standard equation. However I have seen irregularities here when a programme is run to show some of this outer area. At the scale of a window 3 units wide there are a few tiny blips on the contour lines, each covering only a few pixels and hardly noticeable. When enlarged they produce patterns which are reminiscent of finger prints or of abstract art They seem to be fractal in nature as every magnification produces more detail which echoes that seen previously but which is unlike the fringe of the Mandelbrot set. The figure below is an example and comes from a window .000178074321 wide and with the bottom left corner set at (-1.87617849,.25480849).

It is possible that all this arises from a programme error or from errors due to rounding when numbers are made to fit the computer registers; but in either case it would seem suprising that fractals result.

Have these patterns been reported before and are there any suggestions on why they arise?



GINGER BREAD MEN

by Tom Marlow

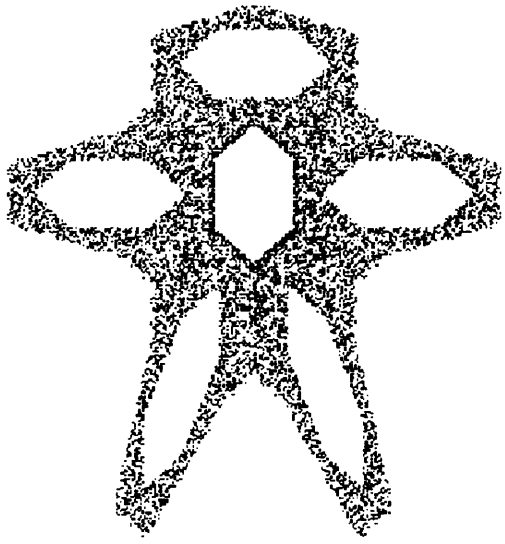
In The Science of Fractal Images, Devaney quotes a chaotic set which he aptly describes as a gingerbread man, due to the shape that it covers. It is generated by the equations:-

$$NewX = 1-Y+Abs(X)$$
$$NewY = X$$

The programme below will plot out this set. The programme is written for BBC Basic but little change is likely to be needed for any other system. The variables H and V in line 20 hold the number of horizontal and vertical pixels of the screen and are used in lines 80 and 90 to set the man in the centre of the screen. The calculations on X and Y in the same lines are designed to rotate the set so that the man is standing upright on his legs. The initial values for X and Y given in line 30 refer to a point within the set and any other such point would serve equally well. Note that all this assumes that the origin of the screen co-ordinates is at the lower left. If it is at the upper left then in lines 80 and 90 change every + to - and every - to +. Also change V*.725 to V*.275 in line 90

An interesting feature is that there is another distinct chaotic set surrounding the ginger bread man. This set can be displayed by substituting the initial values of (8.56,3.76) for X and Y in line 20, when a larger, hollow man appears. Further, there is a succession of shells beyond this second set. The next two can be seen if (6.1,6) and (8.3,8.02) are successively used for X and Y and apparently the series goes on indefinitely. A striking effect is produced if several sets are generated on the same screen each in a different colour.

```
 10  REM Ginger Bread Men
 20  H=1280:V=1024
 30  X=-.1:Y=0
 40  FOR L=1 to 10000
 50  NewX=1-Y+ABS(X)
 60  NewY=X
 70  X=NewX:Y=NewY
 80  A=H/2-X*18+Y*18
 90  B=V*.725-X*18-Y*18
100  PLOT 69,A,B
110  NEXT
```



Editorially produced using Turbobasic, and Rotated for effect!

FEIGENBAUM DIAGRAM IN C FOR THE PC

The following program is a translation into Zortech C of the Pascal program on page 38 of "Dynamical Systems and Fractals" by Becker and Dörfler. It is written for a PC with EGA (640 X 350 resolution). A monochrome, white on black, image is produced.

Some of the images produced differ from those reproduced in the book, presumably because Zortech C, like most C compilers, uses double precision 64 bit floating point arithmetic routines.

When I've worked out how to access 8086 interrupts from the INMOS transputer development system, I'll be able to run this type of program (translated into Occam) on an array of transputers, without the need for an expensive transputer – based graphics system, although I hope to have such a system before long.

If anyone tries this program, and doesn't have Becker and Dörfler's book, the following values will get you started:

Left = 2.5, Right = 2.8, Top = 1.4, Bottom = 0.9, Invisible = 50, Visible = 50.

With these values, I was intrigued to see a ghostly parabola in the chaotic region on the right of the image. Of course, the generating function is parabolic, but I don't see why the "ghost" appears. Has anyone got an explanation for this?

```
£include <stdio.h>
£include <math.h>

struct {
    int ax;
    int bx;
    int cx;
    int dx;
    int si;
    int di;
    int cflag;
    } regs;

double modf();

£define XSCREEN 640
£define YSCREEN 350

/* globals */
float left, right, bottom, top;
int visible, invisible;

main()
{
    /* get input values from user */
    printf("Left: "); scanf("%f", &left);
    printf("Right: "); scanf("%f", &right);
    printf("Bottom: "); scanf("%f", &bottom);
    printf("Top: "); scanf("%f", &top);
    printf("Invisible: "); scanf("%d", &invisible);
    printf("Visible: "); scanf("%d", &visible);

    /* set video mode to EGA graphics (640X350 pixels) */
    regs.ax = 0x0010;
    int86(0x10, &regs, &regs);

    /* plot Feigenbaum diagram */
    feigenbaum_iteration();

    /* wait for keypress */
    getch();

    /* reset video mode to mono alphanumeric */
    regs.ax = 0x0002;
    int86(0x10, &regs, &regs);
}

/* plot point at co-ordinates (x, y) */

plot(x, y)
int x, y;
{
    regs.ax = 0x0C0F;
    regs.bx = 0x0000;
    regs.cx = x;
    regs.dx = y;
    int86(0x10, &regs, &regs);
}

/* plot Feigenbaum diagram */
feigenbaum_iteration()
{
    int range, i;
    float p, k, deltax_per_pixel;
    deltax_per_pixel = (right - left) / XSCREEN;
    for (range = 0; range != XSCREEN; range++)
    {
        k = left + range * deltax_per_pixel;
        p = 0.3;
        for (i = 0; i != invisible; i++)
            p = p + k * p * (1 - p);
        for (i = 0; i != visible; i++)
        {
            set_universal_point(k, p);
            p = p + k * p * (1 - p);
        }
    }
}

/* machine-independent function for setting a pixel with real co-ords */

set_universal_point(xu, yu)
float xu, yu;
{
    float xs, ys;

    xs = (xu - left) * XSCREEN / (right - left);
    ys = (yu - bottom) * YSCREEN / (top - bottom);
    plot(round(xs), round(ys));
}

/* round a real value */
/* modf extracts integral and fractional parts of a real number */

round(x)
float x;
{
    double d1, d2;

    d1 = modf(x, &d2);
    if (d1 >= 0.5)
        return ((int) d2 + 1);
    else
        return ((int) d2);
}
```

Leon Heller,
30 Baldslow Road,
Hastings,
East Sussex TN34 2EY.
Tel: (0424) 714790

Editorial Note: Although we don't normally print C programs, as this article is clearly aimed at putting a well known algorithm into C I hope that we won't offend readers who don't use the language. As it was supplied on disk, we were able to fit it in without overuse of paper.

Editorial

At the end of volume 1 we had over 400 subscribers, and that the total number of subscribers for that volume must now be well over 400 as we continue to sell it. The fact that we have only 320 or so subscribers to the current volume may suggest that interest in fractals cannot be maintained over a long period. The initial take up on enquiries for volume 1 was of the order of 40%, which is very high for an operation of this type. But if the replenishment rate is less than the new intake each year, then the lifetime of the newsletter will definitely be finite! Subscribers are reassured though, that when and if the time comes the newsletter will be wound up properly, with a "last volume," and not just suddenly stopped leaving unfulfilled subscriptions, as do many small DTP computer journals.

I anticipate *Fractal Report* will run to several volumes at least, and I would like to keep the early ones in print. The possibility exists of offering early ones in microfilm, if microfilm readers can be obtained and offered to subscribers at a reasonable price. Any help on this idea would be appreciated.

Under *Announcements* we cover further *Fractint* news, and the current version, 13.0, which although massive in size and number of features, is beginning to show a sort of sameness that may be dulling people's initial enthusiasm.

Another avenue for exploration may be to graft a program like Microsoft Flight Simulator onto fractal landscapes or 3D sets and mappings. Flight Simulator has a facility for scenery disks to enable flight over different countries, so anyone who knows how these are made up may be able to give us a clue as to how to input fractal images instead.

We still await an article explaining clearly how to make fractal music. Many readers have asked for this.

A regular publishing schedule for *Fractal Report* is now anticipated, with issues appearing at the end of the following months:

March, May, July, September, November, January.

The cover date will be the following month, so for example the issue that appears at the end of July will be cover-dated August. Renewal notices will go out with the January issue. Issues will go to press about five weeks before despatch. Time sensitive copy should never be left until the last minute though, as each issue is gradually filled during the month. If there are problems that disrupt this schedule, more frequent publication will be used until it is re-established.

There's Money in Chaos

I may have mentioned this before, but I feel that it is worth mentioning again. There are many computer magazines (over fifty at a quick count) and computer user groups, all of which require articles. *Fractal Report* contains programs and algorithms that can be adapted to most makes of computer. Once you have got such a program up and running on your machine, why not try submitting it as an article to a computer magazine? (Preferably with due acknowledgement to *Fractal Report* and the original author, mentioning the address for sample copy.) Most news stand magazines pay quite highly for articles now, and the payment could certainly cover your *Fractal Report* subscription and leave you some change afterwards.

I would also like to start a regular feature *Readers' Published Work* listing any articles that have appeared in news stand magazines mentioning *Fractal Report*. Therefore any reader who gets such an article printed is asked to send (preferably) a photocopy of the published article and/or the details of the magazine it appeared in.

This is another deal where everyone benefits. The reader/author gets money, *Fractal Report* gets extra publicity, and the target magazine gets an interesting article with program AND a mention in *Fractal Report*.

Fractal Prose

At the outset of this project I suggested the possibility of Fractal Prose and Fractal Poetry. I have an idea for the former which may be relevant. One would need a long list of common first names and last names, sufficiently long so that the probability of two coming together more than once is slight. Then a basis for a novel is devised on the main problem facing those proposing cryonics - if you can

get Mr A to agree, he may not want to go if he cannot afford to take Mrs A, and she may not opt for the process if she can't take her mother, who won't do it unless she can take her spouse and other children and so on. A list of situations with suitable verbs and nouns etc would be needed, and the "story" could proceed for ever.

Postal Inflation

I have heard rumours that the Post Office plan the most savage increases in postal prices since the introduction of the modern postal service by Rowland Hill. The *increase* in first class letters will be more than what it cost to post a first class letter only a few years ago! I am hopeful that I can maintain my prices of £10 (UK) for newsletters in the face of this increase by way of further economies and improved circulation. If the content of *Fractal Report* can be maintained over the next issues, then we should be able to retain and expand our readership. Rather than put cassette prices up, we will probably start mailing them second class or parcel post.

For what it's worth, my view of the Post Office is that they should look to economies internally instead of increasing prices so dramatically. Do people really need two deliveries per day? How about a 3rd class rate for bulk advertising mail, to be delivered during slack periods only? As there is no delivery on Sunday (and we don't need one) do we need collections on Saturdays? I hope that many *Fractal Report* readers will write to their MPs making constructive suggestions about how to keep postage costs down. For widely spread minority groups like fractal enthusiasts, these continuing increases are a serious threat.

If everyone could act as vigorously against rising costs as they do against poor wages, then we would have a far healthier economy.

Announcements

CAD Newsletter

Jo Gedrych has a nice little newsletter for those who use Sun Workstations and high specification PC systems. Entitled "*top*ics" it describes various products for the professional CAD user. Consisting of 4 A4 sides in full colour, it comes out quarterly and is free, but is obviously intended for business users rather than hobbyists. (Intelligent Computer Solutions Ltd., London House, 26 – 40, Kensington High Street, London W8 4PF.)

Video Services

Mr Nigel Woodhead, of Scarab Software, 38, Midship Point, West Ferry Road, London E14 8SW, has expressed an interest in helping with the *Fractal Report* video, and says that he can provide the following services to anyone interested, by negotiation, but extremely competitively.

- Custom Video Services. Transfer from any format PC image program to PAL VHS videotape, either single frame or real time, plus access to further editing and production services at a professional London video company.

- Animation, graphics or special effects and titling – available on disk, videotape, using supplied or created artwork, including scanning from video frames or hard copy.

Fractint Corner

We must all offer a vote of thanks to Robin Harvey who sent me *Fractint* 12.0, and Cade Roux who sent *Fractint* 13.0. Robin Harvey and a number of others pointed out that *Fractint* is public domain rather than shareware. I mailed copies to Jo Gedrych and Adam Case, whom I hope passed them on to any enquirers.

Those people who had already obtained version 9.1 from Jo and Adam are asked to send in another blank disk and return postage if they want the latest version. Due to the volume of mail, they may not be able to let everyone know.

This program also contains a formula input system. You merely type in a formula, like:

```
Mandelbrot(XAXIS) = {
z = Pixel:  z = sqr(z) + pixel, |z| <= 4
}
```

and your fractal can be produced with the full set of features of *Fractint*! For full details see the *Fractint* documentation.

User's formulae may be added to the next release of the program, with attribution in the .frm file. Let's see plenty from *Fractal Report* readers. Like *Longevity Report* readers, Woody Allen prefers to live on in his apartment, but for those of you who are happy to live on in your works, then this is your chance of immortality!

Fractint 13.0 also has a facility whereby one can create a "virtual screen" of up to 2048x2048x256

and print to a HP laserjet compatible printer. The resulting images equal or exceed those produced as black and white prints in glossy fractal books. If Larry Cobb can get his colour printer to print these images in colour, then surely this will add a new lease of life to his competition. I tried a Julia set, and it took six hours to produce and one and a half hours to print! (And that was with a reasonably fast PC – a standard PC would take several days, and Digital Precision's QL PC Emulator several weeks!)

Fractint is written in C and assembler. The source code for both is available, and therefore it should be possible to get versions going for other machines, such as the Amiga, ST and Archimedes. But this task should only be considered by people with a lot of experience and/or time.

Fractint is regarded by its originators as "Stone Soup Software" and here is part of the manual that explains the concept:

Stone soup with pixels: the authors

Once upon a time, somewhere in Eastern Europe, there was a great famine. People jealously hoarded whatever food they could find, hiding it even from their friends and neighbours. One day a peddler drove his wagon into a village, sold a few of his wares, and began asking questions as if he planned to stay for the night.

[No! No! It was three Russian Soldiers! – Lee Crocker] [Wait! I heard it was a Wandering Confessor! – Doug Quinn]* [Well *my* kids have a book that uses Russian Soldiers! – Bert] [Look, who's writing this documentation, anyway? – Monte] [Ah, but who gets it *last* and gets to upload it? – Bert]

"There's not a bite to eat in the whole province," he was told. "Better keep moving on."

"Oh, I have everything I need," he said. "In fact, I was thinking of making some stone soup to share with all of you." He pulled an iron cauldron from his wagon, filled it with water, and built a fire under it. Then, with great ceremony, he drew an ordinary-looking stone from a velvet bag and dropped it into the water.

By now, hearing the rumour of food, most of the villagers had come to the square or watched from their windows. As the peddler sniffed the "broth" and licked his lips in anticipation, hunger began to overcome their scepticism.

"Ah," the peddler said to himself rather loudly, "I do like a tasty stone soup. Of course, stone soup with CABBAGE – – that's hard to beat."

Soon a villager approached hesitantly, holding a cabbage he'd retrieved from its hiding place, and added it to the pot. "Capital!" cried the peddler. "You know, I once had stone soup with cabbage and a bit of salt beef as well, and it was fit for a king."

The village butcher managed to find some salt beef...and so it went, through potatoes, onions, carrots, mushrooms, and so on, until there was indeed a delicious meal for all. The villagers offered the peddler a great deal of money for the magic stone, but he refused to sell and travelled on the next day. And from that time on, long after the famine had ended, they reminisced about the finest soup they'd ever had.

***

That's the way FRACTINT has grown, with quite a bit of magic, although without the element of deception. (You don't have to deceive programmers to make them think that hours of painstaking, often frustrating work is fun... they do it to themselves.)

It wouldn't have happened, of course, without Benoit Mandelbrot and the explosion of interest in fractal graphics that has grown from his work at IBM. Or without the example of other Mandelplotters for the PC. Or without those wizards who first realized you could perform Mandelbrot calculations using integer math (it wasn't us – we just recognize good algorithms when we steal – – uh – – see them). Or those graphics experts who hang around the Compuserve PICS forum and keep adding video modes to the program. Or...

* Does this refer to the Doug Quinn who is the founder of the Cryonics Society of Canada, does anyone know?

Amiga News

Cade Roux, addresses in last issue, advises that some of his prices for Amiga products have been adjusted downwards, so readers are asked to request his latest list before ordering. Some new items have also been added. The first edition of his Amiga disk *Ami-FX, Amiga Fractal Exchange* is being put together now, and includes animated sequences of a 3-D Sierpinski tetrahedron rotating and bouncing, a full colour Julia walk, and an animation based on changing the boundary value of the algorithm used to generate Mandelbrot Set images.

Fractals in the News

Mr Anthony Bayliss enquired on 25 May re *Fractal Report*, his interest in fractals and chaos being inspired by a television programme called *The Late Show*. Has anyone got a VHS video of this I could borrow, please? Apparently Dr Benoit Mandelbrot was interviewed on the programme.

Mike Zehse of Peckham has a favourite pastime of scouring London's dustbins for thrown out newspapers, and one of his finds was a report on fractals in The Sunday Times of 29 April.

Rebecca Fowler described a new fractal culture. This includes T-shirts with $x^2 + c = x$ emblazoned on their arms. (I wonder what happened to the imaginary component!) These are marketed by a company called K-OS, who also plan *Tripatrom* a fractal computer game with no end, no losing and no winning. Founder James Ashbey is quoted as stating "With fractals, a pattern is always just at its most complex as a new order is reached. Just as we are getting to the end of this century we are reaching the maximum complexity of chaos and turbulence." (Why? The numbering of the years is purely arbitrary. The turn of the century is only meaningful if people choose to make it so. Come to think of it, people are motivated by the publishing media, so I suppose they will make it so! If the results from the space telescope make it possible to give an accurate age to the universe, then only this could give a sensible datum to a numbering of years, but even then, one could argue that the orbital elements of the Earth are still arbitrary in this context. And of course it is only half the age of the universe as a whole. {?I think!} If the Quantum Theory gives a smallest possible moment of time {I think 10^{-24} of a second has been suggested} then measuring the age of the universe in these units, with a suitable multiplier prefix to make the numbers manageable, may be the best way if you are looking into any deep meaning in numbers. There again, if whole numbers have significance, why choose the decimal system. Perhaps the significant whole numbers are those when they are expressed in a system to base 131, for example!)

Also mentioned is *Chaos* comic, and an expression "I'm off on a fractal" is reported to be in common (if hardly accurate) usage to mean the speaker is going to let his thoughts follow one another in a random fashion.

Ms Fowler appears to have discussed all this interest with Dr Mandelbrot who is directly quoted as saying "I view myself to be a scholar removed from these movements."

Another video

New reader maths teacher Costel Harnasz tells us that he has seen a supplement to *Scientific American* called *Scientific European.* (April). In it there was an advertisement for a video entitled *Fractals: an Animated Discussion,* However he sent no further details. He also sent in a colour photo of his Casio 7000G graphic scientific calculator showing the Mandelbrot set. He said it took two hours to plot it, and 24 hours to plot map 44 in *The Beauty of Fractals.*

Amygdala News

I received issue 20 on 19 June, and it's eight pages contained two mathematical articles: *Computing Sections of the Cubic Connectedness Locus* and *Tutorial on Complex Analytic Geometry.* So this is where those who require this material should go. The address is Box 219, San Christobal, New Mexico 87464, U. S. A. Their circulation is up to 953 subscribers, which perhaps explains why they charge $30 for announcements, and didn't mention *Fractal Report* despite our coverage of their magazine. A current subscription to *Amygdala*, covering issues 19-28, costs $45 to UK readers, and *Amygdala* will accept Visa and MasterCard (Access etc). Credit cards usually give a fairer rate of exchange than banks.

Slide sets are $32 extra. Do not underate these slides. If you could get or make an automatic back-projector (and have somewhere suitable to put it) they would make a great decoration! They are slides of views from the Mandelbrot Set and other fractals, and do not necessarily relate to material in the accompanying newsletter, although there are usually short text descriptions of what is on each slide.

Amygdala will also supply the Zooms cassette, but the price to UK readers is $35.20 including insured airmail on top of which, if you are unlucky, you will have to pay a mixture of UK taxes, charges and duties. The advantages of getting it from *Amygdala* it is quicker than via *Fractal Report.* We keep the price down by leaving it until we have a waiting list before ordering so we don't have to hold stock, and by ordering via surface mail and at a special bulk rate so the duties and taxes are worked out on a much smaller base figure. The price people pay for saving a lot financially by getting it from us is a long wait!

Music News

Mr Kobus Nieuwmeijer performed live on Saturday 16 June at London's Earth Signals - *The Chaos Happening Amphalos Gallery.* They used a Mac for real time fractal music (with help from Mr A. McLean) and an Atari with midi (Array) stored in a seq. He says it went down well. We still have a few of his *Fractal Fissures* cassettes left at £4. (post included.) - First come, first served.