FRACTAL REPORT 11

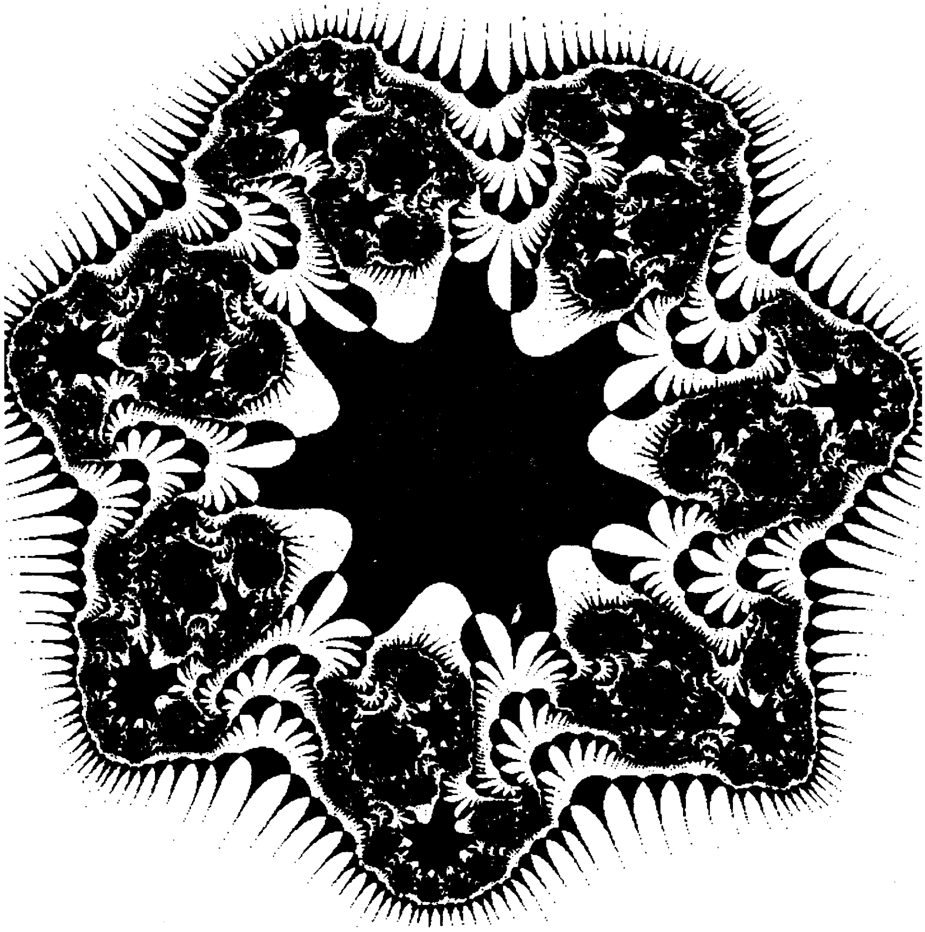
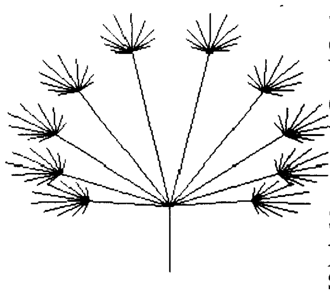


Image by Dr I.D. Entwistle

Hyperbolic Patterns	Dr Uwe Quasthoff	2
Stereoscopic Mandelbrot	Ed Hersom	5
Variations on the Mandelbrot Formula	Kate M. Crennell	8
Book Reviews	Dr Ian Entwistle	11
Shareware and Public Domain Programs for the PC	Jake Davies	12
Announcements	John de Rivaz	14
Three View Lorenz	José E. Murciano	16
Fractal Convolutions	Terry Spitz	17
A Mandelbrot Monster	Paul Gailiunas	18
The Fifth Larry T. Cobb Prize Awards	Larry T. Cobb	20



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 11 First published October 1990. ISSN applied for.

Hyperbolic Patterns

By Uwe Quasthoff, Leipzig

1. Introduction

When looking on M. C. Escher's famous pictures you probably noticed that in some cases there is a connection with fractals. For instance, have a look at Circle Limit I-IV. In this article we want to make the computer draw similar tilings using a chessboard pattern instead of Escher's beautiful shapes.

From the mathematical point of view these pictures are symmetric with respect to some reflections. There are both reflections with respect to straight lines and reflections with respect to circles. Our figures can be obtained using these reflections in an iterated function system with condensation.

2. Reflections

The symmetry properties of a reflection with respect to a straight line g are well known: If a point X is given and X' is its image under the reflection with respect to g, then X and X' lie on different sides of g (or both at g), have the same distance from g, and the line XX' is perpendicular to g.

Next we recall the reflection with respect to a circle of radius r and centre M. If the point X is given, X' lies on the ray through X starting in M and the distances satisfy

$$MX \quad MX' = r^2 .$$

Hence, points on the reflecting circle are again fixed under reflection and the line XX' is again perpendicular to the circle.

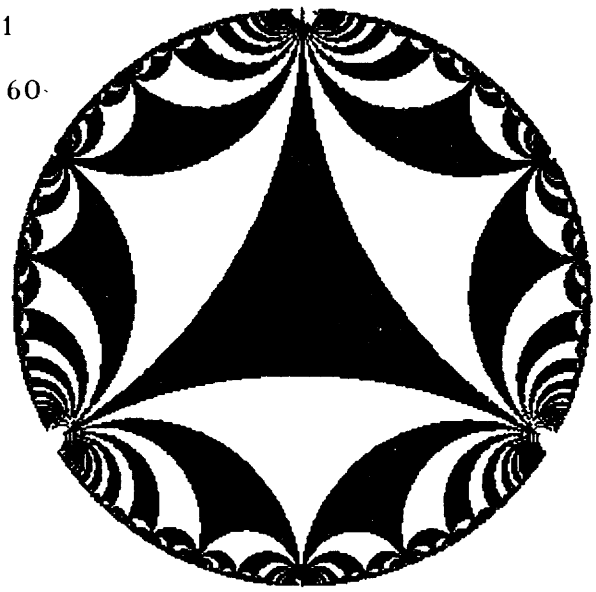
In this article we always start with three reflections. We will apply these reflections in an arbitrary order and maybe several times to a given starting point. So we get an orbit of this point which will be drawn. Of cause, this orbit will depend on the particular reflection choosen at a certain time. But always it will approach a certain limit set which in our cases is a circle.

To get a more structured picture we repeat this procedure with different starting points. There is a so-called fundamental region such that the orbits of all points of this fundamental region fill the whole limit circle. To get a chessboard pattern we start with all points in the fundamental region but draw only every second point of an orbit.

3. Example

We consider three reflections with respect to just-touching circles of equal radii. Lines 40,50 calculate the centers of these circles. Line 60 chooses an starting point (x,y) and in lines 70-100 is checked whether it is contained in the fundamental region. In lines 150 to 230 an orbit of length 10 is calculated, every second point is drawn.

```
10 DIM xx(3),yy(3)
20 DATA 3,3.75,.433
30 READ num,scale,r
40 FOR i=0 TO num-1
50 xx(i)=COS(i*2*PI/num)/2:yy(i)=SIN(i*2*PI/num)/2:NEXT i
60 x=RND-0.5:y=RND-0.5
70 flg=0
80 FOR i=0 TO num-1
90 IF (x-xx(i))^2+(y-yy(i))^2«r^2 THEN flg=1
100 NEXT i
110 IF flg=1 OR x^2+y^2»0.25^2 THEN GOTO 60
120 xb=100*scale*x+100:yb=100*scale*y+100
130 DRAW xb,yb
140 odd=1
150 FOR i=1 TO 10
160 odd=1-odd
170 j=INT(3*RND)
180 rr=r*r/((x-xx(j))^2+(y-yy(j))^2)
190 x=(x-xx(j))*rr+xx(j)
200 y=(y-yy(j))*rr+yy(j)
210 xb=100*scale*x+100:yb=100*scale*y+100
220 IF odd THEN DRAW xb,yb
230 NEXT i
240 GOTO 60
```



The result is given in fig. 1

4. A better algorithm

The above program needs a long time to fill the black areas. The following tree parsing algorithm gives much better results.

For all points of the fundamental region do the following:

STEP 1. [Initialize]. Set even–flag=TRUE and put the point of the fundamental region and even–flag on the stack.

STEP 2. [End?]. If the stack is empty, the processing of the considered starting point is finished.

STEP 3. [Next]. Remove a point and it's even–flag from the stack. If even–flg=TRUE then draw the point. Compute all of its images under the allowed reflections.

STEP 4. [Test]. For every image point look whether it is already drawn. If not, put in and NOT(even–flag) on the stack.

STEP 5. [Loop]. Goto STEP 2.

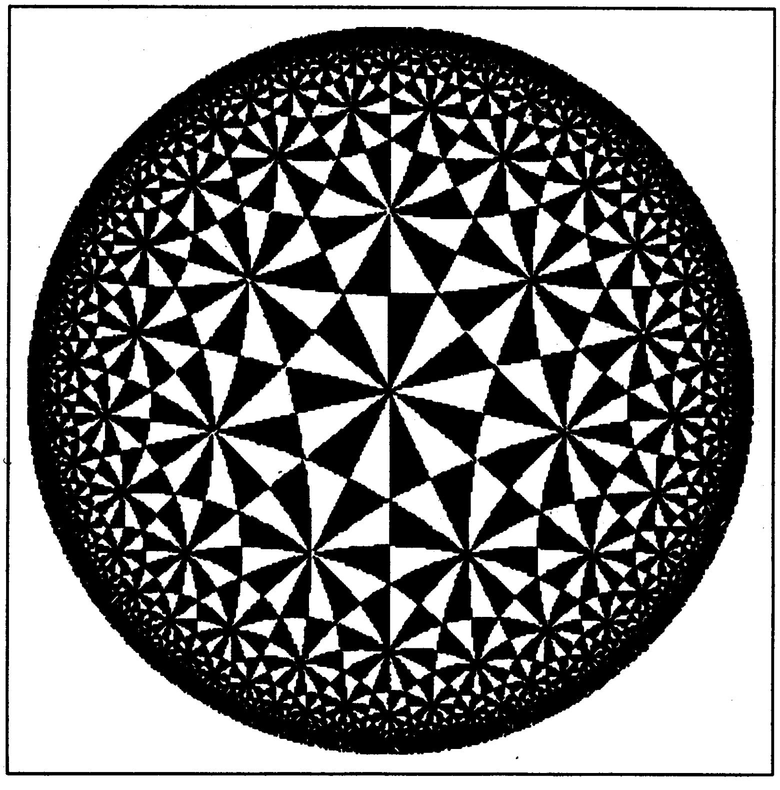
5. More data

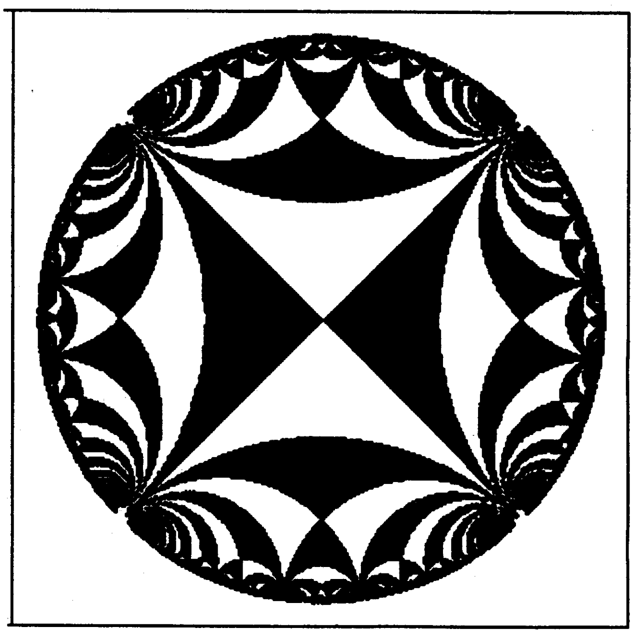
We give the transformations needed to generate some more patterns.

Fig. 1 (right):
Reflection wrt. the straight line through (0,0) and (1,1);
Reflection wrt. the straight line through (0,0) and (1,-1);
Reflection wrt. the circle of radius 1/SQR(2) around (1,0).
As fundamental region you can take a large black triangle near the center of the picture. It has the angles (π/2,0,0).

Fig. 2:
Reflection wrt. the straight line through (0,0) and (1,0);
Reflection wrt. the straight line through (0,0) and (cos(π/7), sin(π/7));
Reflection wrt. the circle of radius 2*sin(π/7) around (1,0).
As fundamental region you can take a large black triangle near the center of the picture. It has the angles (π/2,π/3, π/7).

For your first experiment take the following:
Reflection wrt. the straight line through (0,0) and (cos(π/3,sin(π/3);
Reflection wrt. the straight line through (0,0) and (cos(π/3,-sin(π/3);
Reflection wrt. the circle of radius 10 around (10.668041,0).





STEREOSCOPIC MANDELBROT

Ed Hersom

Inspired by earlier contributions on displaying MANDELBROT in 3D (Ettrick Thomson and Lewis Siegel in Fractal Reports no. 8 and 5 respectively), I have attempted to obtain an even more realistic 3D effect using the old principle of stereoscopic viewing. There are several other ways of getting a 3D effect, such as creating a picture with strong perspective or one which is made to rotate so that foreground and background become obvious. The first is a non-starter for Mandelbrot since, almost by definition, a Fractal cannot have perspective! Making a dynamic model is possible, but it requires a powerful processor to create multiple images fast enough to be satisfactory, so I settled for a method which requires only two images to be made; one for the left eye and one for the right.

Siegel and Thomson and many others create their images by starting with the position of the pixel they are about to display and then calculating the x-y co-ordinates of the Mandelbrot function. If the method is repeated for a neighbouring image, i.e. one for the other eye or for the next one in a dynamic sequence, the same pixel on the screen corresponds to a slightly different point for the Mandelbrot function. This would be fine if the function were continuous, but with Mandelbrot many points would show abrupt changes and the display would be a mess. Instead, I create one model from the Mandelbrot and then rotate or view this model from different directions. Numerically this means taking a grid of a suitable size, laying it over a chosen area of the Mandelbrot plane and then calculating and storing the 'dwell' times for each grid point. From then on, I forget about Mandelbrot and work only with these grid values.

The BASIC program shows how I achieve this, but it does not show the original calculation of the grid values nor how they are read back into Grid% as this is too system dependent and, in any case, it should be straight-forward. Because the plotting routine is required twice, it is made a subroutine (210 - 350). It is first called from 205 which sets the origin over to the left and with Delta set to zero. The second call, from 206, moves the origin 270 pixels to the right and with Delta = 0.2. Delta is used to change the view-point (see later). The significance of 270 is that the two views, when printed on a dot-matrix printer of 120 dots/inch, are 2 1/4" apart, which is approximately my eye separation.

The plotting is similar to that given by Ettrick Thomson, but with some notable differences. Lines 210 and 230 show that I do a blanket coverage of the screen. This avoids the calculation of the diagonal limits of my grid which, because of the small rotation required for the right-hand picture, could have annoying rounding problems. Also, in place of calculating the x-y co-ordinates for the Mandelbrot function, I determine the grid indices, I%, J%. Since my grid has a maximum of 160 for J% and this is going to be displayed on the screen with Yy up to 100, J% is obtained from Yy by a simple scaling (240). Line 250, with Delta = 0, is just Sx% = I% + Yy% (the scale of I% is the same as Sx%), but for the right picture, Delta is non-zero, and so the foreground is shifted to the left relative to the background to give a view appropriate for the right eye.

At line 255 the program detects when the calculation goes above the grid in the top-left corner. If so, the loop is terminated after which grey is plotted, (subroutine 400 - 470). At line 256, if we are in the bottom right corner, however, I regard that area as "ground" rather than "sky", and so it is plotted in white rather than grey. Finally, I have modified the calculation of z so that it varies from 0 to 1, (260). This avoids the rather uninteresting step from zero to the minimum of z. It requires the minimum of the dwell values, but this is readily available since all the points have been pre-calculated.

In the BASIC program which follows, % indicates an integer, which takes half the storage space of a real and so is important in the case of Grid%. POINTS X,Y is the high resolution version of plotting a ".".

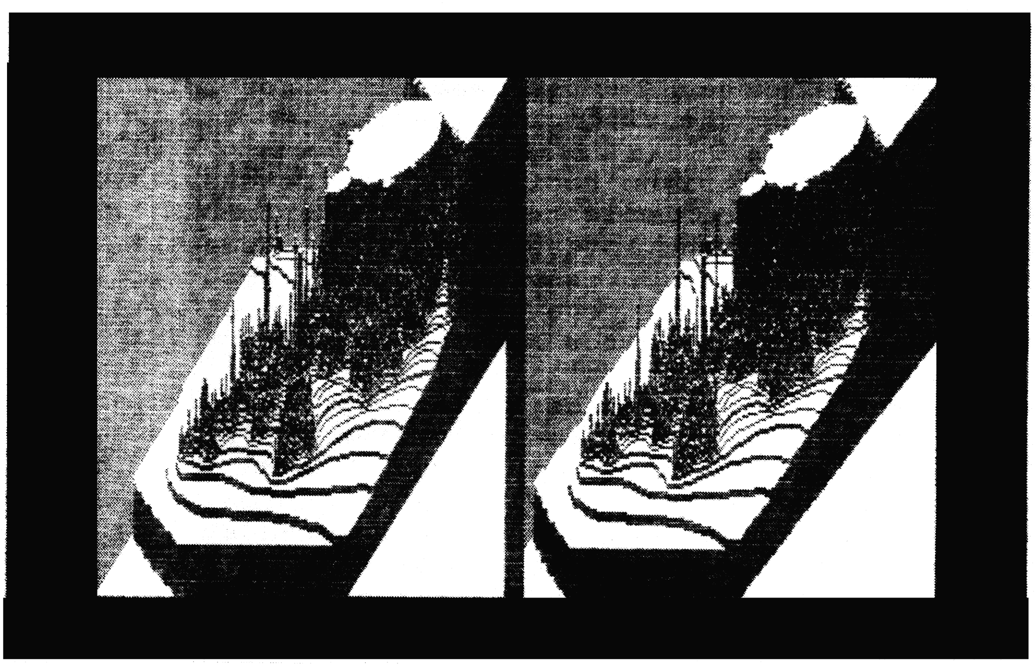
```
 15 Imax%=160 : Jmax%=120
 20 DIM  Grid%(Imax%,Jmax%)
   ( Here is the program to read into Grid% the values
     previously calculated.  In my case the maximum dwell
     time, L%, is read from the file, but the minimum value,
     Minl%, is entered in line 200 below)
150 CLS
200 Minl% := 4 : Px% := 260
201 Yymax% := 100 : Scaley :=  1.2 : Py% := 200
205 SET ORIGIN 30, 20 : Delta := 0 : GOSUB 210
206 SET ORIGIN 300, 20 : Delta := 0.2 : GOSUB 210
207 STOP
210 FOR Sx% := 1 TO Px%
220   Symax% := - 1
230   FOR Yy% := 1 TO Yymax%
240     J% := Yy% * Scaley + 0.5 (+.5 and truncation = rounding)
250     I% := Sx% - Yy% - Delta * (Jmax% - J%) +  0.5
255     IF (I% < 1) THEN Yy% := Yymax% : GOTO 300
256     IF (I% > Imax%) THEN POINTS Sx%, Yy% : GOTO 300
260     Z := (Grid%(I%, J%) - Minl%) / (L% - Minl%)
270     Sy% := Yy% + 100 * SQRT(SQRT(Z)) +  0.5
280     IF Symax% >= Sy% GOTO 300
290     POINTS Sx%, Sy% : Symax% := Sy%
300   NEXT Yy%
310   GOSUB 400
320 NEXT Sx%
350 RETURN
400 REM Plot grey above the figure
410 FOR K% := Symax% + 1 TO Py%
420   IF Sx% + K% AND 1 THEN 450
430   POINTS Sx%, K%
450 NEXT K%
470 RETURN
```

We now come to the tricky bit; how do we view the dual pictures? The essential point is that each eye is to see its corresponding picture and for the brain to think they are looking at the same object. For some people, there is no problem; they can simply hold up the paper before them, their eyes click into position and the image leaps out of the paper!

If you are not so fortunate, you might acquire a pair of (extra) spectacles so that you can focus the pictures with your eyes about 7" from the paper and place a piece of dark card between your nose and the join of the two pictures. Alter-natively, a thin strip of paper or Sellotape can be stuck on your lenses alongside the nose and parallel to it. In fact, use any way of preventing one eye from seeing the other eye's picture. Some winking and blinking might help! Other optical devices have been available and may be still be around. In the 1930's there were viewers, rather like the cheapest slide-viewers currently available. They were simply frames with two lenses and clips to hold a pair of photographs about 4" or 5" from the eyes. I have a pair of plastic "lorgnettes" in which each lens is a combination of a positive lens of 10" focal length and a small prism to make the light beams diverge. The images of a point at infinity are spaced 3 3/4" apart compared with the lens spacing of only 2 3/8". This nicely compensates for the natural tendency for the two eyes to converge to the centre when looking close-to.

Finally, you can have different values for the spacing of the pictures, i.e. something different from my 2 1/4". It is easy to increase this distance, but you will have to reduce the size of the pictures, if you decrease it too much, to avoid overlap.

The pictures are from the Mandelbrot plane of $-0.6 <= x <= -0.4$, $-0.625 <= y <= -0.5$ and the maximum dwell was set to 500. The top right-hand corner is actually part of the main cardioid of the Mandelbrot function.



02 Jul 1990 Thoughts on Variations on the Mandelbrot Formula

Kate M. Crennell

SOME USEFUL BOOKS AND JOURNALS

In the last Fractal Report you asked (page 20) for a general program to draw an image of an object illuminated from a particular viewpoint, and viewed from another with hidden line removal. Such programs are described in Ian Angell and Brian Jones' book "Advanced Graphics with the BBC model B microcomputer" ISBN 0 333 35052 published by Macmillan. Complete program listings are given for those who want to type in the programs. When I bought the book about 5 years ago the programs were also available from the publisher on cassette.

Philip Hickin's last sentence on Page 16 says he would like to have the time to transform his program on to the surface of a sphere. A good explanation of how to do this type of tranformation is given in "The Art of Microcomputer Graphics for the BBCmicro/electron" by McGregor and Watt published by Addison Wesley ISBN 0 201 14435 2 This book has many useful program listings. Chapter 7 describes recursion, fractals and natural patterns. Chapters 8 and 9 discuss 3D manipulations including the mapping of 2D patterns on to sheres and cylinders with hidden line removal. Parts of the book and some of the programs were published in 'Acorn User' about 5 years ago. There is an almost identical volume with the same programs but with listings for the IBM PC.

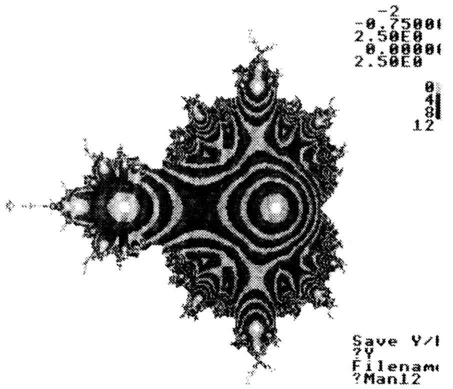
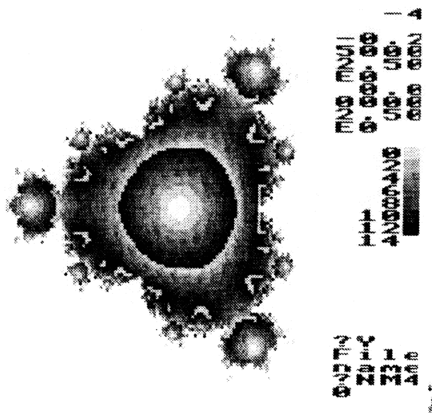
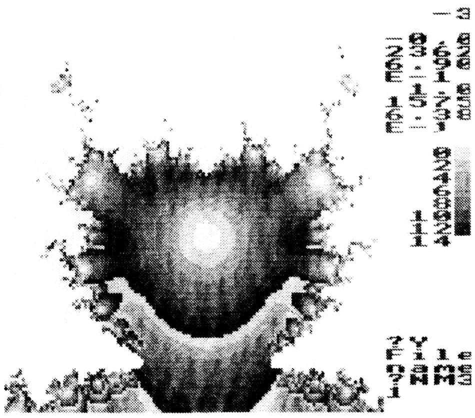
BBC micro owners may be aware of Beebug, the user group which has published several articles on fractals, a recent issue, Vol 9 no 1 May 1990, has a fractal pattern generator which makes all kinds of trees. In Vol 6 no 8 Jan/Feb 1988 they published a program of mine, Hopalong, which draws the Martin's Mappings refered to by Paul Gailumas in Fractal report 9 page 2. In Vol 5 no 1 they published "Mandelbrot Graphics" with a program to plot both the usual law with powers of 2, and extra cubic and quartic ones refereed to by R.J.F.Stewart in Fractal report 6. Andy Lunness article (Issue 8 Page 16) shows a general formula for calculating these higher power Mandelbrot like Sets. This may save the programmer some thought, but for small powers (say less than 6 or so) it takes very much longer because it involves an arctangent, a power (equivalent to a log and an exponential), a sine and a cosine in each iteration. I have tried this using the 4th power as an example, when it takes between 2 and a half and three times as long as the explicit algebra.

A more general journal 'Algorithm' edited by A.K.Dewdney, tries to explain how to program algorithms in simple terms so that personal computer owners can have the fun of programming interesting plots themselves. They do not give complete program listings for any particular computer, so it should interest any fractal enthusiast, because it has several articles by Cliffor Pickover who has written for Fractal report.

Following his suggestions I have made a simple program to plot both the usual Mandelbrot set with coloured bands outside and black inside the set and a variation with black outside and coloured bands inside. Figure 1 shows the interior of the set given by using the normal square law. These plots are made by keeping the smallest value reached at each point inside the set, and using it to decide how to colour that point. The only difficulty is how to map small values on to the 8 possible colours of the BBC micro. In the program the calculations are all done at line 210, with small procedures Man2, Man3, Man4 for the usual plots of the exterior of the Mandelbrot Set, and Man2i, Man3i, Man4i for the plots of the interior. The enclosed plots show a few of the thousands of possible patterns. The concentric stripes surround mathematical features of the set called 'zeros'.

```
10REM"Mandelbrot BASIC plots 26 Jun 90
20REM"Program subject to copyright
30REM"K.M.Crennell   using BBC BASIC for a Model B or Master
40ON ERROR GOTO 320
50MODE7:PRINTTAB(4,10)"MANDELZOOM"''"for square, cubic and quartic
  laws"'" outside or inside the set.":PRINT''''''"Press SPACEBAR
  to start":IFGET
60@%=8:MODE7
70INPUT"Type one of "'"  2 for square law"'"  3 for cubic"'"  4 for
  quartic"'" -2 for square interior"'" -3 cubic interior"'" -4  quartic
  interior",L%
72PRINT"Default Sizes "'" Law 2 X,Y -0.75,0 width 2.5"'
  " Law 3 X,Y  0,0 width 3"'" Law 4 X,Y -0.25,0 width 2.5"
80INPUT''"ENTER centre X"XC:INPUT"ENTER centre Y"YC:
  INPUT"Enter width ",W:INPUT"Enter height ",H:Xmin=XC-W/2:Ymin=YC-H/2
90INPUT" Are these values OK Y/N",A$:IFA$<>"Y"GOTO70
100Dx=W/1024:Dy=H/1024:P%=14:B%=2:Eps=0.002:Limit=(2^(1/(ABS(L%)-1)))^2:
  m2=0.13:Norm=B%*P%/(m2-Eps):REM P% is max no of iterations; B% is no
  of colour repeats, Eps=small value, Limit=max value inside set,
  m2=limit of variation in small
110MODE2:VDU28,16,31,19,0:@%=8:
  PRINT'"Xc",XC;'"YC",YC;'"WIDE",W;'"HITE",H;'" LAW",L%:@%=4
120VDU28,16,31,19,30:REM text window
130Stp%=8:Min%=4:Max%=1020:REM step size and screen limits
140IF W>H Nx%=1+(Max%-Min%)DIVStp%:Ny%=Nx%*H/W ELSE
  Ny%=1+(Max%-Min%)DIVStp%:Nx%=Ny%*W/H
150Dx=W/(Nx%*Stp%):Dy=Dx:Nx%=Nx%*Stp%-Stp%DIV2:Ny%=Ny%*Stp%-Stp%DIV2
160FOR Y%=Min%TONy%STEPStp%:PRINTY%DIVStp%+1;:REM print row number
170Y=Y%*Dy+Ymin
180FOR X%=Min%TONx%STEPStp%
190X=X%*Dx+Xmin
200N%=0:U=X:V=Y
210IF L%= 2 PROCMan2  ELSE IF L%= 3 PROCMan3  ELSE IF L%=4 PROCMan4 ELSE
  IF L%=-2 PROCMan2i ELSE IF L%=-3 PROCMan3i ELSE PROCMan4i
220GCOL0,1+(N%DIV2)
230IFN%<P%PLOT69,X%,Y%-2:PLOT69,X%,Y%+2
240NEXT,
250VDU7,28,16,31,19,26:CLS:REM Beep and reset text window
260INPUT"Save Y/N",F$:IF F$="Y"INPUT"Filename";,F$:GOTO270 ELSE GOTO 290
270PRINT"          "
280OSCLI"SAVE "+F$+" FFFF3000 +5000"
290INPUT"MORE Y/N",F$:IF F$="Y"GOTO60
300END
310:
320IF ERR=17GCOL0,7:GOTO 250
330MODE7:PRINT" ERROR ";ERR," AT LINE ";ERL:REPORT
340END
350:
360DEFPROCMan2:REM calculate power law 2
370REPEATA=U*U:B=V*V:IFA+B>Limit UNTILTRUE:ENDPROC
380V=Y+2*U*V:U=X+A-B
390N%=N%+1:UNTIL N%>P%
400ENDPROC
410:
420DEFPROCMan3:REM calculate power law 3
440REPEATA=U*U:B=V*V:IF A+B>Limit UNTILTRUE:ENDPROC
```

```
450V=Y+V*(3*A-B):U=X+U*(A-3*B)
460N%=N%+1:UNTIL N%>P%
470ENDPROC
480:
490DEFPROCMan4:REM calculate power law 4
500REPEATA=U*U:B=V*V:IF A+B>Limit UNTILTRUE:ENDPROC
510V=Y+4*U*V*(A-B):U=X+A*A+B*B-6*A*B
520N%=N%+1:UNTIL N%>P%
530ENDPROC
540:
550DEFPROCMan2i:REM interior of set law 2
560Small=4
570REPEAT:A=U*U:B=V*V:C=A+B:IFC<Small Small=C
580IFC>Limit N%=P% :UNTILTRUE:ENDPROC
590V=Y+2*U*V:U=X+A-B
600N%=N%+1:UNTIL N%=P%
610IFSmall<Eps N%=P% ELSE IF Small>m2 N%=P%-1 ELSE
   N%=(Small-Eps)*Norm MOD P%
620ENDPROC
630:
640DEFPROCMan3i:REM interior of set law 3
650Small=100.
660REPEAT:A=U*U:B=V*V:C=A+B:IF C<Small Small=C
670IF C>2 N%=P%:UNTILTRUE:ENDPROC
680V=Y+V*(3*A-B):U=X+U*(A-3*B)
690N%=N%+1:UNTIL N%=P%
700s=Small-Eps:IF s<0 N%=P% ELSE N%=(s*Norm) MOD P%
710ENDPROC
720:
730DEFPROCMan4i:REM interior of set law 4
740Small=100.
750REPEAT:A=U*U:B=V*V:C=A+B:IF C<Small Small=C
760IF C>2 N%=P%:UNTILTRUE:ENDPROC
770V=Y+4*U*V*(A-B):U=X+A*A+B*B-6*A*B
780N%=N%+1:UNTILN%=P%
790s=Small-Eps:IF s<0 N%=P% ELSE N%=(s*Norm) MOD P%
800ENDPROC
```

Fractal Geometry –
Mathematical Foundations and Applications
by Kenneth Falconer,
Wiley, Chichester, New York, 1990, hbk, 310 pp, £19.95.

A review by Ian D. Entwistle

The contents page lists eighteen chapters. Their headings range from Hausdorff measure and dimension, Dynamical systems, Projection of fractals, Dynamical systems, Iteration of complex functions Julia sets to Random fractals and Multifractal measures. Most topics are treated by a section on background followed by theorems or propositions. Many examples follow. Each chapter is ended with a list of exercises. Although the consequences of many of the proofs may be of interest to the general fractal enthusiast this is a text book for mathematicians or mathematics students. Many of the diagrams which include Koch curves, coastlines, Julia sets and various attractors will be familiar to most casual readers. The accompanying maths text will not be so easily followed. The substantial reference list alphabetical by author is up to date. This book will convince most mathematicians unfamiliar with fractal geometry that fractals are not just the pretty pictures being generated by computers to please the media and pattern generators. They are seen to be complex and fascinating mathematics. This book should be all a mathematician needs for conviction. In the generally more readable introductory chapter the author adopts a more rigid view about fractal definition than has developed in other recent texts. His detailed mathematical treatise therefore carefully avoids the wider application of fractals declared by B. Mandelbrot. For the *Fractal Report* readers who are fractal generators the book cover illustration will be of interest. A nice coloured rendering of the Julia set for $z^2 + 0.233 + 0.5378i$ certainly attracts the eye. **This book is not recommended unless you are a trained mathematician.**

Computers, Pattern, Chaos and Beauty: Graphics from an Unseen World by
Clifford A. Pickover, St. Martins Press, New York.

A review by Ian D. Entwistle.

A considerable number of books about fractals have been published. The earliest brought the existence of fractals and some of their more aesthetic properties into attention. As more scientists recognised the utility of the measurement of fractal dimension for classifying complex shapes texts appeared on the wider use of the geometry and dynamics of fractal behaviour. More recently books describing computer programs for computation of the more appealing fractal patterns have appeared. A few more advanced texts on the pure mathematics of fractals have also been published. *Fractal Report* subscribers would mostly have appreciated the high quality fractal pictures and the practical help to generate their own versions in these books. They probably didn't find much in the way of readable explanation which would help them be more creative in developing new fractals. Dr Pickover's book goes some way towards filling this gap. Additionally it puts fractal pattern generation into a new perspective. The use of computation to generate patterns, observing their beauty by computer graphics and studying their relevance to art designs and natural shapes is given a grand airing in this book. The text is a comprehensive compilation of the author's publications in a wide field of computer graphical studies. As a researcher whose interest is in showing how computer generated data can be used to compose imaginative, realistic and useful new images he has few peers. Dr.Pickover's work has been published in a wide range of journals. The book's content reflects his interest in addressing a wider audience. Some parts of the descriptive text requires a recognition of mathematical processes and some mathematical understanding is needed to write listings to generate some of the graphics described, e.g. Halley maps. There are no mathematical proofs to struggle with! The contents pages are quite detailed and the seventeen chapters are each prefaced with a full page graphic most of which are computer generated. These are the author's own originals and for many readers they will be the main visual impact of the book. There are hundreds more illustrations, some 29 in colour, all of which are high resolution. Most enthusiastic readers will wish to try reproducing some of them. Direct descriptions of how to do this are not always given. The origins of most are however discussed and with the help of the numerous algorithms/pseudo codes listed throughout the chapters the persistent reader will be able to generate a number of them. Some tips on the contents of the algorithms would have been some help in this. Throughout the book the author writes to encourage lateral thinking readers to generate their own work.

Topics of particular interest to fractal enthusiasts are not located in a single section but are found and discussed as they fit into the main themes which are reflections of the book's title. If you wish to read everything related to Julia sets then you need to read through a number of chapters such as *Genesis Equations* (or Biological Feedback forms), *More Beauty from Complex Variables* (Julia set of cosh(z) is described in detail) and *Mathematical Chaos*. A few Mandelbrot set illustrations are to be found but the author uses them only to demonstrate alternative methodology. In the chapter Synthesising Nature Julia sets of an unfamiliar design are discussed. The author's wider interest in generating data for meaningful graphics does occupy a substantial part of the book. In these chapters such as *Image Processing, Tessellation Automata, Number Theory and Fourier Transforms* most readers will find some fascinating images. Items such as the Shroud of Turin, snowflakes and genes are also covered. The author frequently cites reference works and has compiled a very comprehensive section of them. An additional reading list and an impressively long list of the author's own publications (>100 refs.) from which this book has been compiled is included. Some help in coping with the large number of scientific disciplines encountered is covered by a detailed glossary of terms. Amidst all the exciting graphics there are numerous quotations serious and humorous and curious artistic illustrations designed to put the clinical aspects of computing into a more human environment. The author has a great enthusiasm for striking images whatever their origin. One of the best quotes by Rudolf Carnap "It is indeed a surprising and fortunate fact that nature can be expressed by relatively low order mathematical functions" is apt also for the computer enthusiast attempting to generate exciting patterns. If you are not already familiar with the author's widely published work then this book should afford you inspiration and challenges and should stimulate the real computer graphics enthusiasts to experiment more widely. For fractal enthusiasts it should widen your scope since many of the interesting graphics require a lot less CPU time than the Mandelbrot set. There are however no instant recipes for generation of fractals. The book needs to be read and understood if you wish to convert many of the listed codes into working computer listings. Those who produce distributed software could perhaps take up some of Dr. Pickover's ideas. At the cover price of $29.95 the book is really good value (over 390 pages and all those images!). The early orders direct to St. Martins Press paid only $23.95 post paid. A real steal! Even at the imported UK price of £25 it has to be a good buy even for the pictures! [From Alan Sutton Publishing, Phoenix Mill, Far Thrupp, Stroud, Glos GL5 2BU – ed]

Shareware and Public Domain Programs for the PC.

Jake Davies

Most of my fractal explorations are based on my own code, but I find other authors programs beneficial and educational, either in the documentation, or in the source code sometimes supplied with registration. Other peoples programs also throws light on the varied methods of representing the data, differing colouring and contouring strategies etc. Sometimes the disks are worth getting simply for the sample files and co-ordinate data included.

Among my collection of programs are the following selection of Shareware and PD software. The quality varies, but registering your software usual brings benefits. For each program I give the system requirements, video adapters needed or supported, whether the program makes use of a maths co-processor, amount of documentation supplied on disk, the benefits and price of registration, and strengths of the program.

Cell Systems (100k)

Requirements: 128K
Video: CGA
Co-processor: No
Documentation: 6k
Registration: $15 for latest version and complete documentation.

Cell Systems is a menu driven cellular automata generator. It comes with 62 sample files which can be edited, or you can start from scratch. Its quite fast, and well behaved, and its easy to save your own designs. The program is the fastest automata generator I know of.

Cellular Automata (900k)

Requirements:
Video: CGA
Co-Processor: NO
Documentation: 36k
Registration: None required, Public Domain

Cellular Automata is a whole suite of programs (25) to generate one dimensional cellular automata. C source code for all the programs is included. There is a mass of background material, and the programs go quite heavily into the mathematical background. Generating images though is fairly straightforward.

Mandelbrot Super (340k)

Requirements: 256K
Video: CGA
Co-processor: No
Documentation: 4K
Registration: $40. Gives you a version supporting 8087 co-processor, and C source code.

Mandelbrot Super comes with lots of sample files, but is fairly simple, offering windowing and colour changing. Fairly fast.

Mandelex (90k)

Requirements: BASICA or equivalent
Video: CGA
Co-processor: No
Documentation: 10k
Registration: $20 for latest version and printed documentation

Mandelex uses an unusual "contour following" algorithm that is much faster than the normal "brute force" method. It plots iteratively in stages of increasing resolution, allowing you to see an approximation of the plot very quickly. Needs Basic to run, but means the source is easily alterable. I have started to convert to compiled basic, but no success yet.

Superman. (250K)

Requirements: 256k
Video: CGA
Co-processor: NO
Documentation: 21K
Registration: ?$. Gets you the latest version, plus math co-processor support, and full Pascal source code.

Superman is an early program, but works well, within the limitations of CGA. Saves datasets, that can then be recoloured using different banding strategies. Colour cycle animation is quite extensively catered for.

Jset & Mset (120K)

Requirements: 256k
Video: Hercules, CGA, EGA & extended EGA
Co-processor: Yes
Documentation: 18K
Registration: $35. This gets you the latest version that allows you to save and reload screens, and to customise the palettes.

Generates both the Mandelbrot and Julia Set. A variety of palettes are available, but the program doesn't allow you to save the screens.

Zoomlens (70k)

Requirements: 64K
Video: CGA
Co-processor: No
Documentation: 19k
Registration: $20 gets a far more advanced program, with Pascal source code.

This is the most unusual Mandelbrot Set generator I have yet seen, and seems ideally suited to children and beginners. Unfortunately it only runs in CGA. The whole set is stored in compressed form to several levels of magnification, so upon zooming the screen is instantly redrawn. When you go deeper than what is already calculated the program calculates and then adds to the original file. It is therefore very easy to set up sequences of animation, zooming and panning in real-time. When calculating it plots every 10 pixels in each axis, then every fifth, and so on thereby enabling you to see an approximation of the plot very quickly.

Mandel (310K)

Requirements:
Video: EGA or VGA
Co-processor: No
Documentation: 16K
Registration: $20 for latest version, co-processor support, manual, and pascal source.

Mandel is a nice menu driven system. Values are stored, and the program gives a lot of control over the colouring. Files are compressed and not too large. A nice feature is the ability to generate a graph of the distribution of dwell values, and their associated colours, making it easy to see where to change the colours.

Mandelbrot Magic (360K)

Requirements: 256K.
Video: CGA, EGA, VGA.
Co-processor: YES
Documentation: 98K
Registration: $15 Registration brings the latest version and a superb program BACKMAGIC, (see below)

This is a real gem of a program for exploring Mandelbrot and Julia sets, and I consider it the best of the bunch. It is the most versatile program I have come across (equal to Larry Cobb's Dragons program, although each has quite different strengths,) This program offers the most versatility in mapping and presenting images I have seen. The program creates a dataset of the mapping, not just a screen map, so consequently any file can be called to the screen in an enormous combination of palettes, colour modes and region definitions. Once on screen, any aspect of its appearance can be instantly changed, and all views can be animated. Because the program generates a dataset, the files tend to get quite large, the deeper you go, EGA screens typically being 100K+ and VGA screens 150K+. The second most useful facility is the ability to just use a small part of the screen to generate a "thumbnail". The whole program is menu driven and very user-friendly, the documentation is clear and informative (38 pages). Highly Recommended.

Backmagic is a program you get with registering your copy of *Mandelbrot Magic*. It is an "Add-in", and you need *Mandelbrot Magic* to run it. *Backmagic* lets your machine get on with the calculations in the background, while you get on with more mundane things with your computer, (like earning a living).
This is easily worth $15, to free your machine.

Just in

VGAdemo (280k)

Requirements: 256k
Video: VGA
Co-processor: No
Documentation: None
Registration: None

This disk is an animated slide show of VGA Mandelbrot images. The animation is by colour-cycling, and in 256 colours.

Fractint (640k) *Version 13*

Requirements:
Video: Hercules, CGA, EGA, VGA, extended VGA, Targa, in fact just about any graphics adapter.
Co-processor: Yes
Documentation: 125k
Registration: None required, this is Public Domain

This must be THE ultimate fractal generator, it generates over FIFTY different types of fractal. If you are a C programmer then this is a gold mine, as there is the complete C source code with lots of help. Just about everything is customisable. You can create disk files up to 2000 by 2000 resolution for printing out on a Laserjet. If you have a VGA then this program really lets you go to town on the application of Colour. It would take a whole issue of Fractal report to list everything *Fractint* does.
Interested in 3D Fractals ?.... maps fractals as landscapes or onto spheres... Barnsley IFS's in two and Three dimensions? This program really flies along as well, using several speed up routines (all of which are fully explained). This program makes me wish I could write C.

Fractkal (150k)

Requirements: 192k
Video: EGA
Co-processor: Yes
Documentation: 24k
Registration: $15 gets the latest version with lots of enhancements, $35 get the same plus full Turbo Pascal source code.

Fractkal is an unusual program. It uses affine transformations to generate random kaleidoscopic displays, with a high degree of user control and interaction.

Recursive Realm (500k)

Requirements: 256k
Video: EGA/VGA
Co-processor: YES
Documentation: 39K
Registration: $20 gets you the latest version

Recursive Realm is a nice integrated menu driven package. As well as the classic Mandelbrot and Julia Sets, it also generates Julia sets for two other functions, Newton's method for nine functions, and four types of phases of magnetism maps. Well worth it for these extra functions. files can be saved in mid-calculation, and finished later. File sizes are 130k for EGA and 170k for VGA. Quite a lot of control over colouring and banding strategies.

Mndlzoom (140k)

Requirements: Dos 2.1 or higher
Video: EGA
Co-processor: NEEDED
Documentation: 9K
Registration: $15 get the latest version

This is the only program I haven't tried out, as it needs a maths co-processor, and I don't yet have one. It is a fairly recent program (1989) and seems straight forward enough. The Fortran and assembly code is included, to make it easy to translate to other languages.

Some of the above mentioned programs are available from the larger shareware outlets, although others I recently bought back from the states, so may not be easily available over here. If you cannot find these programs from your usual sources then they are available from myself at the address below. I will fit as many programs as possible onto a disk, so for instance 6 programs may fit onto 1 3.5"disk

PRICES: 360k (5 1/4 ")....£5 720K (3 1/2")....£6

For those who wish to save time in generating early zooms, or who want pre-generated screens/Data to start from, I can supply disks (360k or 720k) filled with screens for all of the above programs. £1-50 (£2-50 for 3.5) each disk.

Jake Davies
Higher Trengove
Constantine
Falmouth
Cornwall TR11 5QR

Announcements

Mr Leon Heller kindly sent us the following notes:

Transputer I/F for Archimedes

Andy Lunness is working on an interface for his Archimedes which will enable him to connect one or more of my transputer modules to it.

DIY Parallel Processing

I mentioned Simon Langridge's parallel Z80 system for Mandelbrot set computation some time ago. I gave Simon some spare 68008s I had, and he has got four of them running in parallel. Mandelbrot set calculations are performed at about the same speed as a 68000/68881 combination, but his system is a lot cheaper, especially as the 68008s were free! I'll see if I can get him to write it up for you.

New Motorola Number Cruncher

Motorola recently announced the DSP96002 96-bit General Purpose IEEE Floating-Point Dual Port Processor. This is probably the fastest number cruncher around at the moment. According to the data booklet I got from Motorola, it will execute up to 165 MOPS and 50 MFLOPS, with a 33.3 Mhz clock. Calling it a 96-bit device is a bit naughty, as each of the parallel execution units (the Data ALU, the Address Generation Unit, and the Program Control Unit) are only 32-bit sub-systems.

The 50 MFLOPS refers to single-precision IEEE arithmetic. Future devices will perform double-precision calculations at this rate, or even faster!

The dual-port feature means that arbitrarily long pipelines of 96002s can be constructed, with no external logic. Also, interfacing the chip to other processors is made very easy, especially processors with multiplexed data/address busses, like the transputer.

The device has 1024 words of on-chip data RAM and 1024 words of on chip program RAM, and programs can be loaded, executed and debugged via a high speed clocked serial port. Fractal calculations could therefore be performed with virtually no external circuitry, with a simple three wire interface to a host processor. I haven't been able to find how much the device costs, but I doubt if you will get much change out of £1000.

Editorial Comments

When the transputer interface for the Archimedes is completed, it will be interesting to see the resulting speed up. As far as I can recall the combined cost of the transputer module and software to run is it not trivial!

On the subject of cost, the Motorola number cruncher will surely come down in cost if one waits a few years. The bulk of the costs are in design and tooling, and once the chip has been superseded by something better, Motorola have little to lose and something to gain by selling it at a lower price.

DRAGONS 3 Release

By the time you read this DRAGONS 3 should have completed its Beta testing and the documentation update will be done (I hope!). As well as all the well known features, like cursor searches for Julia sets and automatic co-ordinate transforms, DRAGONS 3 will have:

Full support for GIF fractal files (both 87a and 89a). These are half the size of the previous DRAGONS files, and provide a standard interface to other programs and computers.

Now included is the Ikenaga fractal and its Julia sets – best described as a "smudged and crushed" Mandelbrot. The regular order of the Mandelbrot set is lost and a much more abstract image results.

The Cubic fractal and its Julia sets are now supported, bringing the total to 10 forms. The Cubic is a more orderly fractal with threefold symmetry.

Also including many minor improvements, DRAGONS 3 is a bargain at £16 (or £10 for registered users of DRAGONS 2). Send a cheque, or a SAE for more details, to:

Larry Cobb,
Bay House, Dean Down Drove,
Littleton, Hants, SO22 6PP

And don't forget to keep the competition entries coming too!

More from Mr McLaren

Mr Jon McLaren sent us another letter at the end of July. The correct page reference for the fractal he wants to generate, and offered £25 for the code to do it, is on page 144 of the November 1987 *Scientific American*. An image in similar vein appears as plate 35 in *Science of Fractal Images,* and it is discussed on page 212.

The function is $P(a,b) = z^3 - 3a^2z + b$, where z,a,b are all complex.

He also mentions that many of the ideas in articles in *Fractal Report* can be found in other literature in public libraries. I am sure that our authors don't claim originality in most cases. What they do offer is explanation of the fractals in terms that enable home computer users to actually enter the programs into their machines and experiment with them. Of course we always hope that we will find originality, but obviously we won't get five or six original fractals ideas every issue.

Fractint Corner

Dr Ian Entwistle suggested that *Fractint* would be better if one could "control the X,Y, pixelation from iteration and so view small resolution plots directly before committing all the CPU time." He still felt that many commercial fractal offerings were much better than *Fractint*, some as much as ten times better. Personally, I would like to see a file handler added, similar to the one in WordPerfect, so that you get a list of suitable files and can select one by moving a cursor when recovering a file or doing a 3d projection.

Mr Anthony Giles sent a combined letter to both *Fractal Report* and *Longevity Report*. With regards to the former he said he has access to a Pixar Image Computer and may be able to make some fractals videos from it when he has discovered how to get PAL video out of it. Apparently there seems to be a facility for this. It runs at 80 MIPS. Also he is to work on an MSc project in a year's time that may involve fractal analysis of X-rays, which may produce a few spin-off articles for *Fractal Report*. This will hopefully help to satisfy those readers who ask for a practical purpose to all these pretty pictures!

Archimedes Notes

Another note from Dr Ian Entwistle mentions some software he has got that converts Archimedes sprites into .GIF and .TIF MSDOS files. He also has other file conversion software that will convert almost anything but is at the time of writing having difficulty in using them. No doubt by the time this appears all problems will have been solved. The main difficulty seems to be in getting his Archimedes to read Amiga and other disk formats. Archimedes owners are invited to contact John Kortink, Middlehuistr 17, 7422 El Haaksbergen, The Netherlands for further information on this file conversion program.

Apple Program

Dr Entwistle sent us a photocopy of an advertisement for an Apple educational program that displays a number of fractal images. Further information from EduTech, 1927, Culver Road, Rochester, NY 14609, USA, or $75 gets you two disks, a manual and a workbook, entitled *Chaos Plus*.

Fractal Conferences

Fractal entrepreneur Mr Andy Lunness (69, Bronté Avenue, Bury, Lancs, BL9 9RN) is now thinking of holding a fractal symposium. He suggests that a fairly central location for the venue would be in South Wales, and expects the cost to be £50 – £60 per delegate for a weekend running from Friday evening to Sunday lunchtime. The cost will include meals but not travel. On offer will be video shows, slide shows and a variety of other demonstrations together with discussion groups and talks. It is intended that slides, videos and software will be on sale together with books and magazines.

As readers of *Longevity Report* know, one of my eccentricities is to abhor travel and crowds, so I will not be attending this. However both newsletters are intended to reflect the interests of the readership generally, and so I give coverage of such events. (As a matter of interest immortalists hold more conferences and have a higher turnout of members than most other special interest groups.) I will keep you all informed as to progress with the project. Anyone wishing to register an interest and/or offers of help is invited to write to Mr Lunness direct.

Mr Lunness also informs us that he has received a flood of enquiries for his newsletter *Chaos and Complex Cartography* each time it is mentioned in *Fractal Report*.

Another item for readers wanting a trip, is a one day colloquium at the IEE's Savoy Place, London, on Monday, 3 December. The colloquium is on the production of fractals on cheap computers for image processing. Further details from Professor R.J. Clarke, Dept of Electrical and Electronic Engineering, Heriot – Watt University, 31 – 35, Grassmarket, Edinburgh, EH1 2HT. (He obviously doesn't think locally!)

Mac Music

Mr Adam McLean, of *The Hermetic Journal,* PO Box 375, Headington, Oxford OX3 8PW, has a Pattern Generator and Music Generator program suite available for the Apple Macintosh for £35 or $60. The Pattern Generator appears similar to Fractint, producing a wide range of different fractal patterns. However it can write these fractals to a file which can be used by the Music Generator to produce music through the Mac's MIDI port. Notes are sent on channels 1 to 4. The player chooses with a mouse a region of the fractal displayed on the screen, and the program then generates the music. The notes can be played in the full cycle of modes Major, Minor, Lydian, Wholetone etc., and the music can be transposed to various keys and a number of rhythmic patterns can be chosen from a menu. A demonstration version is available with a number of features cut out, and it times out after a certain number of sessions.

Kobus Nieuwmeijer has tried the program and says it is very versatile and easy to use.

More on Videos

Dr Ian Entwistle sent in some more information on the *Scientific European* fractal video. It costs a whopping DM99 from *Scientific European,* Mönchhofstrasse 15, D – 6900 Heidelberg, West Germany. He says you should send no money, but you will be invoiced. Authors of animated video fractals sequences discuss fractals. The history and prospects of fractals are discussed, and the sequences illustrate important concepts in chaos, self – similarity and music composed according to fractal principles.

The new Art Matrix 2 hour video is available from them for $50 plus postage. According to Dr Entwistle reports indicate that it has to be seen to be believed. Art Matrix accept credit cards, which give a fair rate of exchange, and their address is: PO Box 880, Ithaca, New York 14851 – 0880, USA. UK and European readers should ask for the PAL version.

Peitgen and Saupe in Scientific American

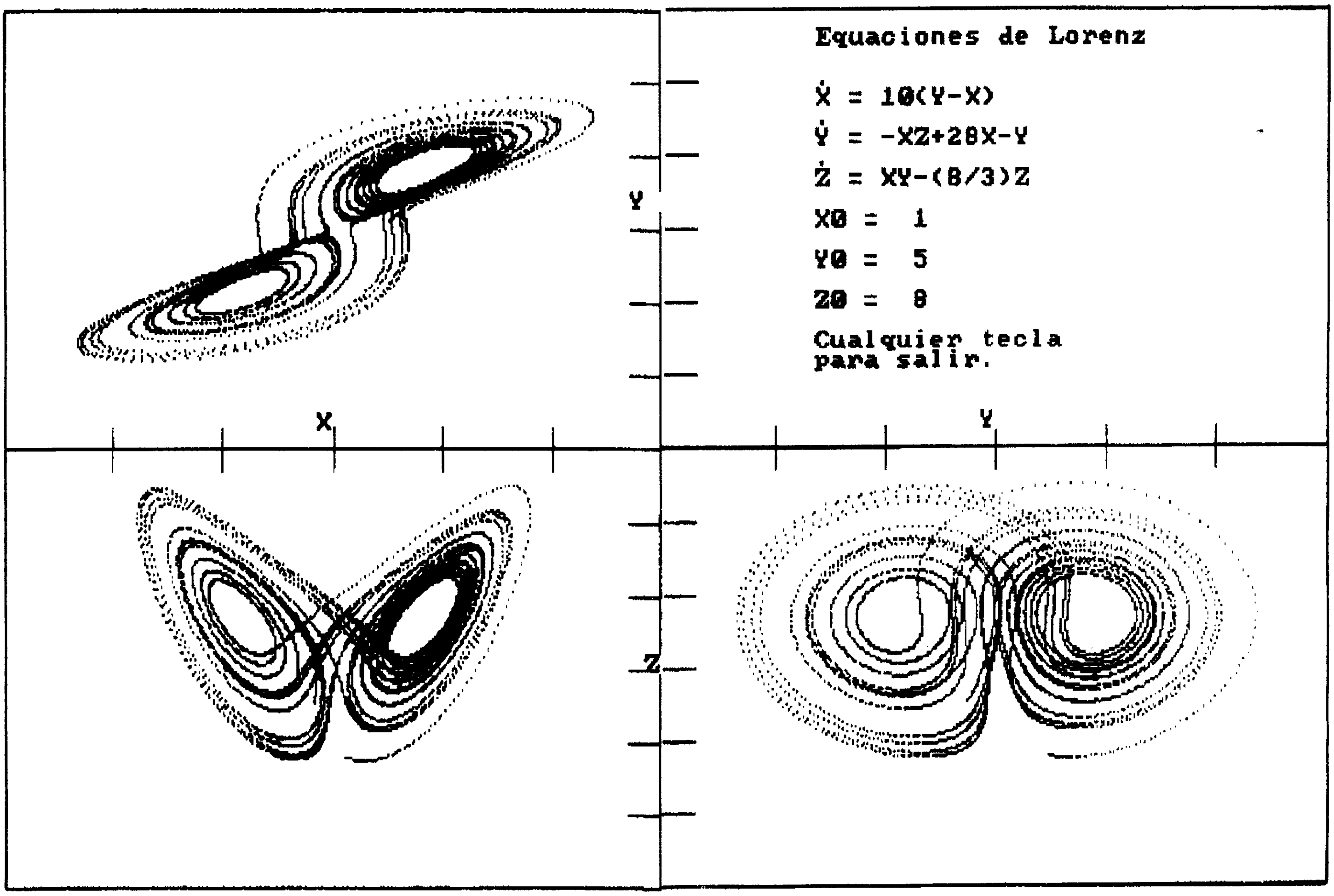
Also from Dr Entwistle: The August 1990 *Scientific American* contains an article by Peitgen and Saupe *The Language of Fractals.* It covers most areas, especially IFS, mountains and image coding. There are lots of Julia Sets and colour pictures.

Mandelbrot Set "Noise"

The "problem" mentioned by Mr Tom Marlow resulted in various items of correspondence. Dr Entwistle has used this to good advantage in the past. Some years ago he sent Rollo Silver a number of colour versions. They were part of his "Persian Carpet Set". The effect results from lack of precision and therefore misplacement of z. It is one of the pitfalls when trying to make integer algorithms work.

Three View Lorenz by José Enrique Murciano

```
Programa:     Lorenzo1.BAS
Función:      Desarrolla las tres ecuaciones de
              Lorenz al mismo tiempo.
Compilador:   QBasic y con muy ligeras
              modificaciones en en Turbo Basic
              de Borland.
Autor:        José Enrique Murciano & Dan
              Lufkin Colleagues
REM $DYNAMIC
DIM Pathx(5000)
REM $DYNAMIC
DIM Pathy(5000)
REM $DYNAMIC
DIM Pathz(5000)

CLS
SCREEN 9
WIDTH 80, 43
VIEW
WINDOW (-60, -60)-(60, 60)

LINE (-60, -60)-(60, 60), 2, B
LINE (0, -60)-(0, 60), 2
LINE (-60, 0)-(60, 0), 2

FOR I = -60 TO 60 STEP 10
  LINE (I, -3)-STEP(0, 6), 2
NEXT I

FOR J = -60 TO 60 STEP 10
  LINE (-3, J)-STEP(6, 0), 2
NEXT J
COLOR 14, 1

LOCATE 2, 50: PRINT "Equaciones de Lorenz"
LOCATE 4, 50: PRINT "."
LOCATE 5, 50: PRINT "X = 10(Y-X)"
LOCATE 6, 50: PRINT "."
LOCATE 7, 50: PRINT "Y = -XZ+28X-Y"
LOCATE 8, 50: PRINT "."
LOCATE 9, 50: PRINT "Z = XY-(8/3)Z"
SIG = 10
R = 28
B = 8 / 3
DELT = .004
T = 0
P = 0
Init:
  RANDOMIZE TIMER
  X = INT((26) * RND)
  Y = INT((26) * RND)
  Z = INT((26) * RND)
LOCATE 11, 50: PRINT "X0 = "; X
LOCATE 13, 50: PRINT "Y0 = "; Y
LOCATE 15, 50: PRINT "Z0 = "; Z
LOCATE 17, 50: PRINT "Cualquier tecla"
LOCATE 18, 50: PRINT "para salir."
LOCATE 10, 39: PRINT "Y "
LOCATE 21, 20: PRINT "X"
'LOCATE 23, 20: PRINT "0"
LOCATE 21, 60: PRINT "Y"
'LOCATE 23, 60: PRINT "0"
LOCATE 33, 40: PRINT "Z"
DO
  GOSUB STEPUM
  C = (T MOD 6) + 9
  PSET (Y - 30, X + 30), C
  PSET (X - 30, Z - 50), C
  PSET (Y + 30, Z - 50), C
     Pathx(P) = X
     Pathy(P) = Y
     Pathz(P) = Z
     OLDX = Pathx(P + 1)
     OLDY = Pathy(P + 1)
     OLDZ = Pathz(P + 1)
  PSET (OLDY - 30, OLDX + 30), 0
  PSET (OLDX - 30, OLDZ - 50), 0
  PSET (OLDY + 30, OLDZ - 50), 0
  IF P > 4998 THEN P = 0
  P = P + 1
  T = T + DELT
LOOP WHILE INKEY$ = ""
Finish:
  WIDTH 80, 25
  CLS
  LOCATE 12, 35: PRINT CHR$(173); "Hasta
luego!"
  FOR N = 1 TO 5000
  NEXT N
  END

STEPUM:
  X1 = SIG * (Y - X)
  Y1 = -(X * Z) + R * X - Y
  Z1 = X * Y - B * Z
  X = X + X1 * DELT
  Y = Y + Y1 * DELT
  Z = Z + Z1 * DELT
RETURN
END
```



Fractal Convolutions

by Terry Spitz – St Edmund Hall, Oxford.

I am writing following your plea for articles, with a number of small points which may interest *Fractal Report* readers. Starting with strictly the least relevant, my attention was drawn to mention of a new comic to be entitled *The Mandelbrot Set* by Alan Moore (of *Watchmen* fame) and Bill Sienkiewicz. Further investigations reveal it has been released under the actual name *Big Numbers* though still sporting an M – Set on the front, and the first of twelve promises excellent and original illustration supported by fractally convoluted plot(!) Details from *Mad Love* (Publishing) Ltd. PO Box 61, Northampton.

Second, a quick mention of the lectures conducted at a number of universities by Professor Benoit B. Mandelbrot himself on the subject of fractals. Though hindered by a thick accent, he covered basic ideas of fractals and their relations to nature and included many brilliant slides. My un – initiated friends liked the pretty pictures!

After reading the last few *Fractal Reports* I suddenly noticed the uncorrected proof of Dr Clifford Pickover's *Computers, Pattern, Chaos, and Beauty* which has been sitting on my shelf for a few months. Unfortunately my prepublication copy only has the pictures included separately at the end with big gaps in the text into which to fit them, which makes it patience – trying to browse, but when time permits I'll put some work into trying it out.

A tip for Archie owners, especially those like myself suffering in monochrome. Forget colour for adding that extra dimension to your investigations, try the real thing: time! Using the Archie's amazing memory powers you can load large hi – res mode 0 screensaved sequences as sprites and simply and speedily plot them to your screen. Not for those of you at home without the proper equipment!

Finally a BASIC landscape zoom almost like those ones you see on the telly. Archie – exclusive bits are in square brackets and alternatives have been included where necessary. It runs using a midpoint displacement method quoted in Peitgen and Saupe's *The Science of Fractal Images* (a real must which includes all those 3D rendering methods you can't do without). The fractals line is then zoomed in to twice magnification followed by a small(!) delay as the more detail is calculated. It unfortunately can't manage islands since it is essentially just 1D brown noise, but the resolution and interestingly the fractals dimension can be optimised.

```
10REM >LAND
[20MODE0]:REM graphics screen mode
[30ON ERROR MODE0:REPORT:PRINT ERL:END]
40M=7:MAX=2^M      : REM resolution
45H=.6             : REM fractal dimension 0< H <1
50DIM A(MAX),TRANS(MAX)
60S=TRUE:L=1:LEV=0:I=3/8*MAX    : REM TRUE is -1
70

80REM************ INTERPOLATE
90REPEAT
100   LL=MAX/2^L
110   FOR X%=LL TO MAX-LL STEP LL*2
120     A(X%)=(A(X%-LL)+A(X%+LL))/2 + FNG/2^(L*H)*SQR(1-2^(2*H-2))*1000
130   NEXT
140   L+=1
150UNTIL L>M
160

170L=1:REM******* DISPLAY AND ZOOM
180REPEAT
190   II=I*(1-1/L)
200   AA=A(I) * (L-1)
[210   *FX19 :REM VDU sync]
220   CLS
230   POINT I*1280/MAX,1023 :REM plot point (quite important)
240   X=0 : MOVE 0,500+ (A(II + X/L) - AA) *SQR(L)
250   FOR X=0 TO MAX
260     DRAW X*1280/MAX,500+ (A(II + X/L) - AA) *SQR(L) :REM line to point
270   NEXT
280   FILL I*1280/MAX,0 : REM floodfill, if you can't, don't
[290   S=NOT S:IF S THEN PROCSCR1 ELSE PROCSCR2: REM switch screens]
300   L+=.2
310UNTIL L>2
320II=I*(1-1/L)
330

340REM************ RECALC DATA
[350A()-=A(I):I+=(RND(MAX)-MAX/2)/20]
REM not for those of you at sans Archie:
    try FOR X=0 TO MAX:A(X)=A(X)-A(I):NEXT
    and I=I+(...    instead
[360TRANS()=A() ] :REM sorry! try FOR X=0 TO MAX:TRANS(X)=A(X):NEXT
370FOR X=0 TO MAX STEP 2^(L-1) : A(X)=TRANS(II+X/L)*SQR(2) : NEXT
380LEV+=1 :L=M : REM LEV=LEV+1, ok.
390GOTO 90
400END
410

420DEF FNG:SM=0:FOR P=1 TO 10:SM+=RND(1):NEXT:=(SM-5)/5:REM gaussian RND
   REM if this wont do then just replace the function call FNG earlier with
      a random number in the range 0 to 1.

[430DEFPROCSCR1:*FX 112,1: REM switch screens
440 *FX 113,2
450 ENDPROC
460DEFPROCSCR2:*FX 112,2
470 *FX 113,1
480 ENDPROC ]
```

A MANDELBROT MONSTER

Paul Gailiunas

Chaos is concerned with those situations where a very small difference to begin with makes a big difference later, so maybe it is not surprising that small changes in the way that pictures of the Mandelbrot set are calculated can completely alter their final appearance. The image is not accurate so that technically the program which produces it has a bug, and the picture represents some kind of monstrous deformation of the mathematically precise version. Pickover has written about how a programming error produced interesting forms (*Computers, Pattern, Chaos and Beauty*, Alan Sutton, 1990, p. 102) and he suggests a variety of convergence tests for Julia sets which alter the appearance of the final image (op. cit. pp.123-127), and Mandelbrot has published many pictures which were produced by programs with bugs (in *The Fractal Geometry of Nature*).

Images of the Mandelbrot set are notorious for the time they take to generate and I have tried several ideas to speed things up. Usually this involves programming at a low level because the most significant gains will probably come from speeding up the innnermost loop of the program, the part that actually calculates z = z↑2 + c. (It is obviously not possible to write a routine in BASIC which will multiply faster than the existing one which is in machine code.)

I have tried using FORTH to produce faster programs, with some success, but what is more interesting is the effect of truncation errors in the algorithm on the final picture. FORTH will only support integer arithmetic, either on single precision numbers (16 bit = 2 byte = 1 word) or on double precision numbers (32 bit = 4 byte = 2 words), and although it can multiply two single precision numbers to give a double precision number (mixed precision multiplication), it is not able to multiply two double precision numbers.

My idea was to use double precision numbers, equal to actual numbers multiplied by a large scale factor and truncated to an integer, and to write a program to square these numbers (rather than multiplying them). If a double precision number, x, is written as a high word, hi(x), plus a low word, lo(x), then multiplying two numbers involves:

$$x*y = (hi(x) + lo(x))*(hi(y) + lo(y))$$
$$= hi(x)*hi(y) + hi(x)*lo(y) + lo(x)*hi(y) + lo(x)*lo(y)$$

This produces a 64 bit number, but only the most significant 32 bits are needed. This means that the lo(x)*lo(y) part can be ignored, and three mixed precision multiplications are needed. Squaring a number only requires two mixed precision multiplications:

$$x*x = (hi(x) + lo(x))*(hi(x) + lo(x))$$
$$= hi(x)*hi(x) + 2*(hi(x)*lo(x)) + lo(x)*lo(x)$$
$$\simeq hi(x)*hi(x) + 2*(hi(x)*lo(x))$$

(There are quick ways of doing the doubling - see † below).

All that is needed is to square x, y and (x+y) and the usual escape time algorithm can be easily implemented:

if $x^2 + y^2$ > four then leave
  $x = x^2 - y^2 + a$
  $y = (x+y)^2 - x^2 - y^2 + b$

It is not quite so simple because these numbers are actual numbers multiplied by a large scale factor, and this needs to be taken into account (four in the above algorithm is 4 * scale factor). The first thing to do is to work out a suitable factor. It is fairly easy to work out that the biggest number that can ever occur is less than 8 (if x = y and $x^2 + y^2 = 4$, then x = sqrt(2), $(x+y)^2 = 8$), so that

$$2^{32} = 8 * \text{scale and scale} = 2^{29} = 536870912$$

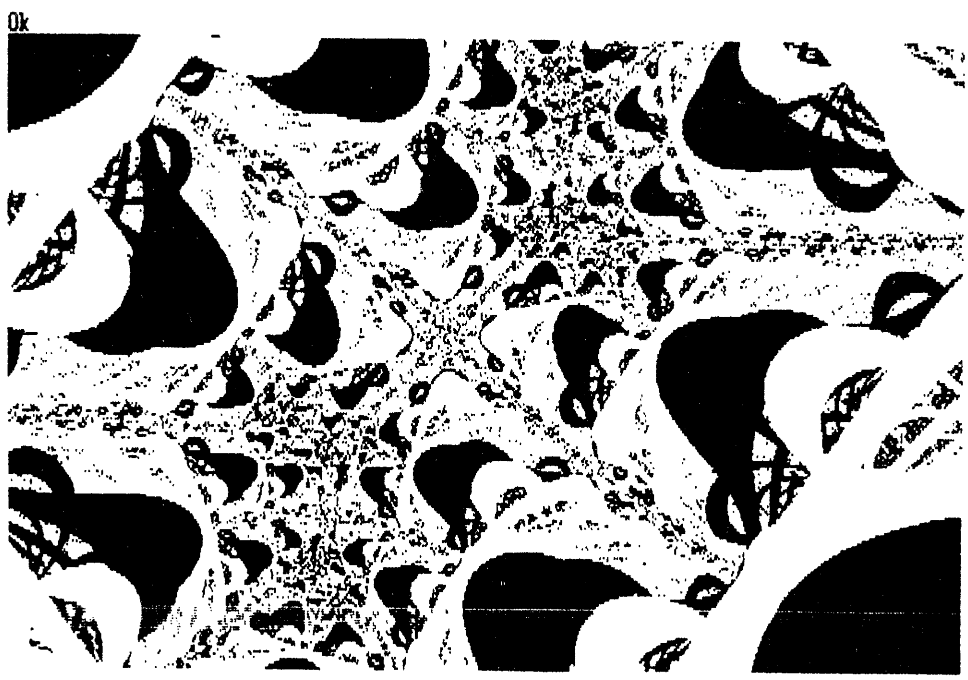
Now think about what happens to the number 1. It becomes scaled to 536870912, which in binary is 00010000 00000000 00000000 00000000. When this is squared it becomes a 64 bit (8 byte) number, but the least significant 4 bytes are ignored, so it becomes 00000010 00000000 00000000 00000000. Of course it should be the scaled version of 1 again. In other words everything needs to move three places to the left. It is actually easier to do this in assembler, but it just amounts to multiplying by 8. The complete algorithm looks like this:

◦ take the absolute value of the number
(The sign is irrelevant, and will complicate things.)
◦ calculate hi(x)*hi(x)
◦ shift the 4 byte number 3 places left
(or multiply by 8 = add to itself 3 times)
◦ calculate hi(x)*lo(x)
(The 4 most significant bytes are actually 2^{16} times too big compared with the 4 bytes of hi(x)*hi(x), since the low part of the number is 16 places to the right of the high part. So this number needs to be shifted 16 places to the right, 3 places to the left as before, and doubled. † This amounts to shifting 12 places to the right, or dividing by 4096.)
◦ Add the two results.

This produces reasonably accurate results. When used in the escape time algorithm the number of iterations is quite near to the value produced by a more precise method.

I use an Amstrad PCW with a monochrome monitor and the colouring algorithm I use is:
input an integer (colourmask)
form the logical AND of colourmask and escape time
plot a point if this is 0.
This means that an error of just 1 in the value of the escape time can make the difference between black or white on the printout. The example I have given is in a region which Larry T Cobb mentioned in *Fractal Report* issue 8 : side = 3.3e-05, centre = (-1.941245, 0.001852). This translates into scaled numbers : step = 25, bottom left corner = (-1042206832, 987789). I used a colourmask of 3, ie a point is plotted only if the escape time is divisible by 4. It is dominated by structures which are entirely absent from an accurate version of this picture.

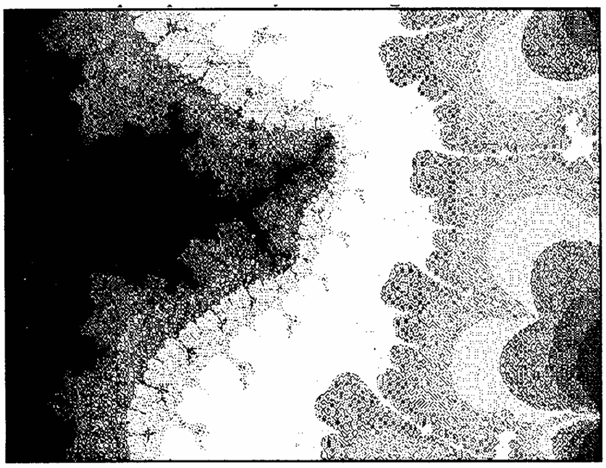
I have tried translating this program into BASIC but it is very slow, mainly because BASIC is not designed for this kind of low level bit-manipulation, and I don't think it's worth bothering with, but I will be happy to provide a stand-alone version of the program for PCW produced using HiSoft FORTH if people send me a CF2 format 3" disc. It will only run if copies of GSX.SYS and DDSCREEN.PRL from the Amstrad system disc are present along with a text file called ASSIGN.SYS which consists of the line: 01 @:DDSCREEN. Alternatively I can provide a listing of the FORTH definitions, although some modification will probably be needed for other machines, since there is some assembler, as well as GSX graphics.



The Fifth "LARRY T COBB PRIZE" Awards

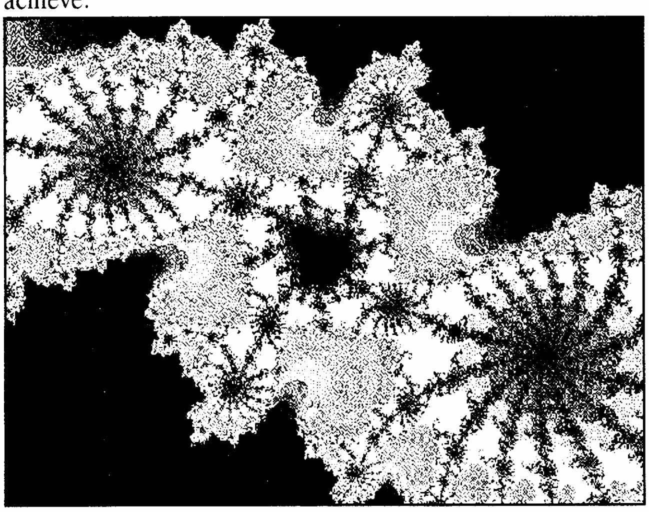
The competition hasn't been run for some time because the number of entries was too low. But now the scorching summer seems to be at an end and the rains have returned, readers are finding more time to search for fractals.

Marco Verhoeven, from the Netherlands, sent in several entries which are represented by Mandel40 and Mandel41. I spent some time trying to reproduce the attractive effects in his printouts and I hope my examples do them justice, but going from CGA to VGA often radically alters the image. Perhaps readers with a CGA display would find these examples particularly interesting.



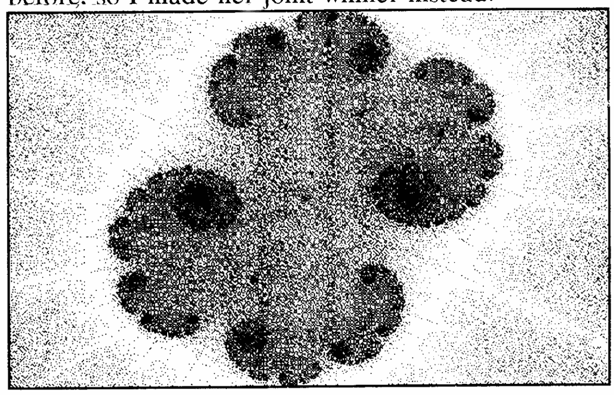
Mandel41: VGA graphics driver with 16 colours
Mandelbrot routine using Log colour mapping
offset = 0, start = 50, stop = 200, Z value = (0,0)
side = 6.6e − 6, centre = (− 1.7444456, − 0.022026364)
and, not illustrated,
Mandel40: Extended VGA graphics, 256 colours
Mandelbrot routine using Alternate colour mapping
offset = 170, start = 0, stop = 250, Z value = (0,0)
side = 0.0002966, centre = (− 1.7444508, − 0.022017214)

Sue Ilsley also got some interesting results from a low resolution system, as you can see from Mandel42. Sue was using an Amstrad CPC and I am often surprised at the results that the dedicated (and patient?) enthusiast can achieve.



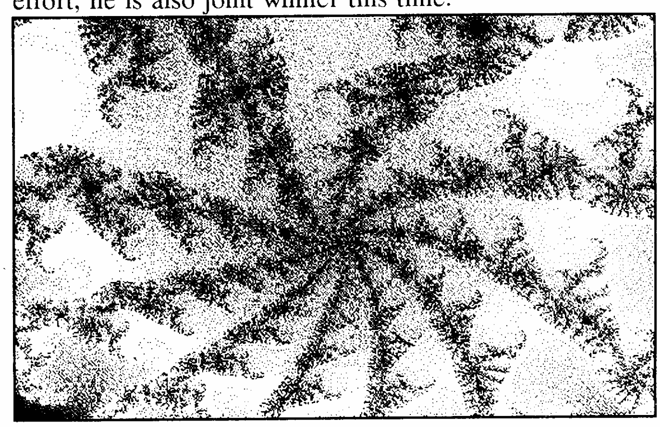
Mandel42: VGA graphics driver with 16 colours
Mandelbrot routine using Log colour mapping
offset = 0, start = 40, stop = 200, Z value = (0,0)
side = 0.002 , centre = (− 0.742575,0.208924)

Kate Crennell is becoming a regular in this competition and I am always pleased to see the excellent standard of her entries. She says that the image in Mandel45 is called "Charybdis Egg", after the legendary Greek whirlpool. (Perhaps someone will find the Scylla in the Julia set?) I avoided making her the winner because she has won before, so I made her joint winner instead!



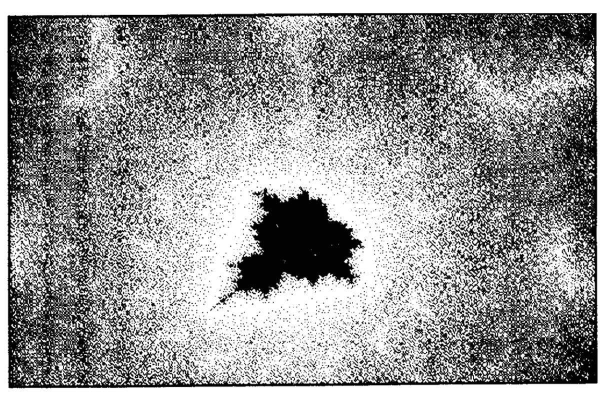
Mandel45: Extended VGA graphics, 256 colours
Mandelbrot routine using Log colour mapping
offset = 0, start = 55, stop = 250, Z value = (0,0)
side = 1.4e − 07,
centre = (− 1.94048108403, 1.61952215048e − 05)

The entry from Michael Collins illustrates the immense impact that the palette can have on the image. Michael sent a colour print of Mandel49 using the default palette, calling it "Peacock Eyes". When I generated it using DRAGONS spectrum palette, I got a lovely red tinged fern! Try it with both and see which you prefer. For this effort, he is also joint winner this time.



Mandel49: Extended VGA graphics, 256 colours
Mandelbrot routine using Log colour mapping
offset = 0, start = 20, stop = 255, Z value = (0,0)
side = 0.0048 , centre = (0.3349158,0.046965473)

The last competitor, Cade Roux, has taken palette modifications to the extreme. On his Amiga, he imported the fractal Mandel50 into a paint program, called DeluxPaint III, and manipulated the palette colours with it. Then he used its print option to produce a beautifully subtle and effective print on a Rainbow NX − 1000, (a US version of the Star LC10).



Mandel50: Extended VGA graphics, 256 colours
Mandelbrot routine using Log colour mapping
offset = 0, start = 200, stop = 1000, Z value = (0,0)
side = 3.5e − 09, centre = (− 0.1434794512, − 1.018965407)