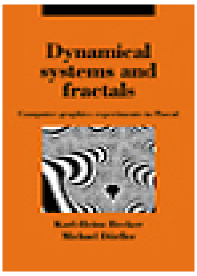


Dynamical Systems and Fractals

Computer Graphics Experiments with Pascal

Karl-Heinz Becker, Michael Dörfler, Translated by I. Stewart

Book DOI: http://dx.doi.org/10.1017/CBO9780511663031

Online ISBN: 9780511663031

Hardback ISBN: 9780521360258

Paperback ISBN: 9780521369107

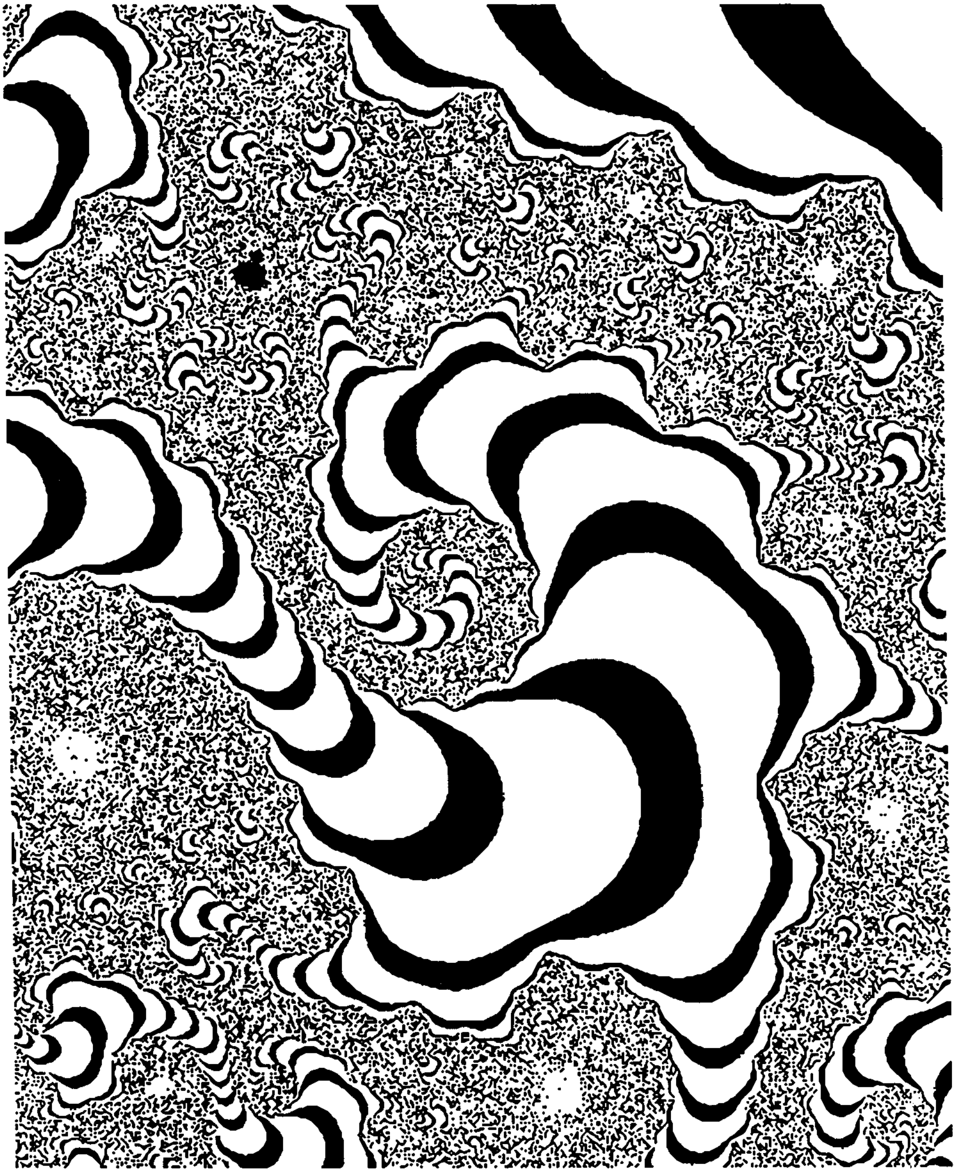
Chapter

6 - Encounter with the Gingerbread Man pp. 127-178

Chapter DOI: http://dx.doi.org/10.1017/CBO9780511663031.008

Cambridge University Press

6 Encounter with the Gingerbread Man



Pictures like grains of sand on the seashore... the graphics which we can generate by the methods of the previous chapter are as different or as similar as sand. Each complex number produces another picture, sometimes fundamentally different from the others, sometimes differing only in details. Despite the self-similarity (or maybe because of it?) new surprises appear upon magnification.

6.1 A Superstar with Frills

From all this variety of apparent forms we will pursue only one property. It concerns the question: is the picture connected or does it split apart?[1] But instead of investigating the connectivity with particular magnifications in unpredictable places, we wihall use a trick. Perhaps something has already occurred to you in connection with the experiments in Chapter 5?

As an example, consider the two complex numbers

$$c_1 = 0.745\,405\,4 + i*0.113\,006\,3$$

and

$$c_2 = 0.745\,428 + i*0.113\,009.$$

We have already looked at the corresponding pictures in Chapter 5 (Figures 5.2-9 to 5.2-16), and discovered that the Julia set corresponding to c_2 is connected. In contrast c_1 produces pictures in which the basin of the finite attractor splits into arbitrarily many pieces. It suffices to show that the figure is not connected together at one place, but falls into two pieces there. The more general conclusion follows by self-similarity.

Consider Figures 6.1-1 and 6.1-2, corresponding to the above c-values. They show at about 40-fold magnification the region in the neighbourhood of the origin of the complex plane, which lies symmetrically placed in the middle of the entire figure.

The question whether the set is connected can now be answered quite easily. If you look in Figure 6.1-1 in the middle of the striped basin, it is clear that no connection can exist between the lower left and the upper right portions of the figure. That is a fundamentally different observation from Figure 6.1-2. There the stripes in the basin of the attractor at infinity do not approach near enough to each other.

In the middle there appears, between them, a relatively large region from the other attractor. We have already encountered something similar in Figures 5.2-14 and 5.2-16. You can show for yourself that such circle-like forms appear at very many places in this Julia set. But this region round the origin of the complex plane is fairly large. Thus we can pursue the investigation without any pictures at all, reducing the question to a single point, namely the origin. In the sequence for c_1 (Figure 6.1-1) we have drawn the required boundary after about 160 iterations, and thereby made sure that the origin belongs to the basin of the attractor ∞. But even if the computer takes a week, we cannot determine the boundary for c_2 (Figure 6.1-1).

[1] We call a set *connected* if there is a path within it, by means of which we can reach any point without leaving the set. If this is not the case, the set must divide into two distinct parts.

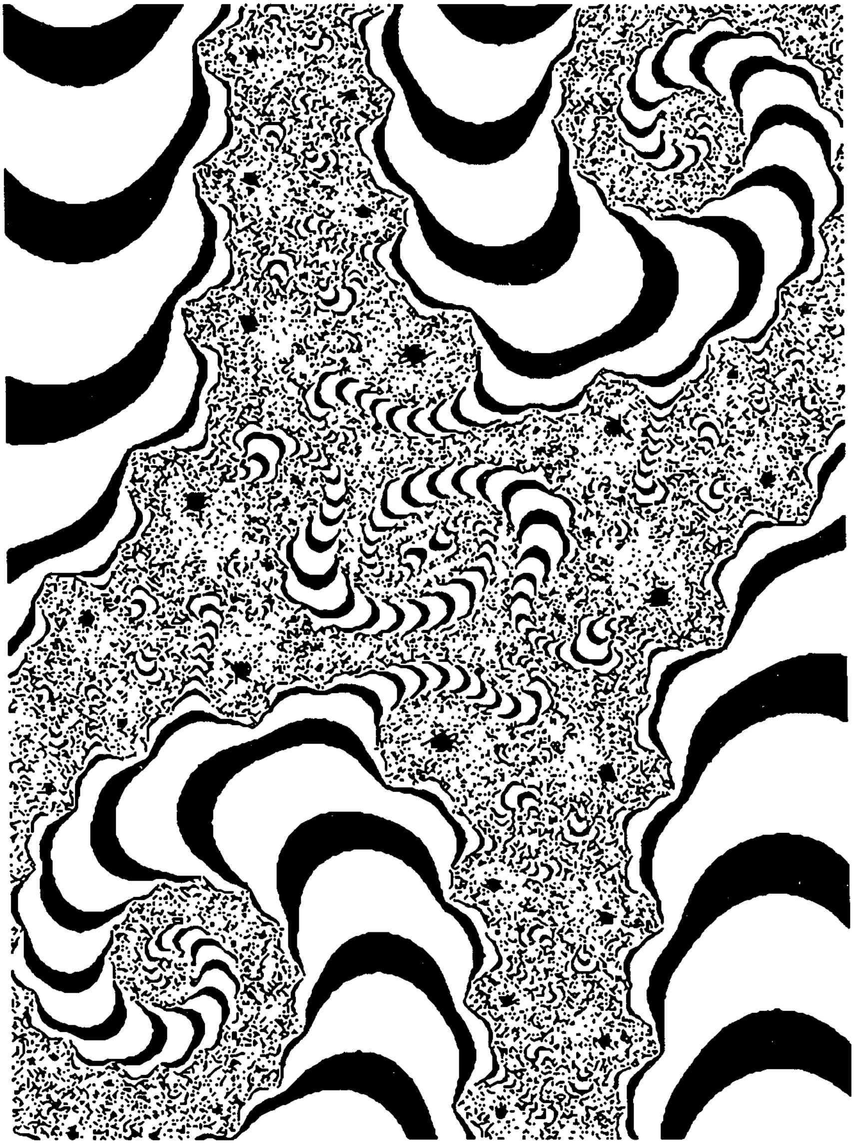


Figure 6.1-1 Julia set for c_1, section near the origin.

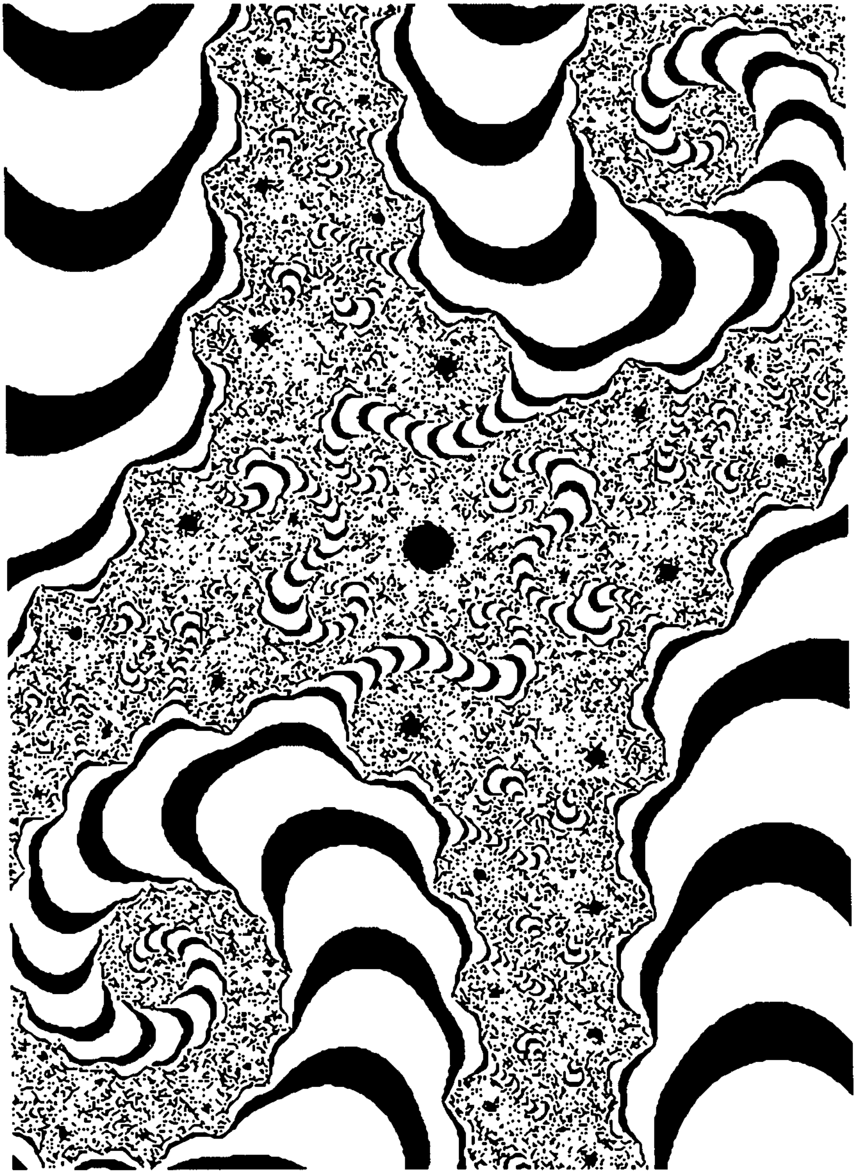


Figure 6.1–2 Julia set for c_2, section near the origin.

In other words, the Julia set for a particular number c in the iteration sequence

$$z_{n+1} = z_n^2 - c$$

is connected, provided the sequence starting from

$$z_0 = 0$$

does not diverge.

We shall not pursue the mathematics of this relationship any further. All iteration sequences depend only on c, since $z_0 = 0$ is predetermined. If you insert this value into the iteration sequence, you obtain in turn:

$$z_0 = 0,$$
$$z_1 = -c,$$
$$z_2 = c^2 - c,$$
$$z_3 = c^4 - 2c^3 + c^2 - c,$$
$$z_4 = c^8 - 4c^7 - 2c^6 - 6c^5 + 5c^4 - 2c^3 + c^2 - c,$$

etc.

Whether this sequence diverges depends upon whether the positive and negative summands are of similar sizes, or not. Then, when the modulus of z exceeds a certain bound, squaring produces such a large increase that subtracting c no longer leads to small numbers. In the succeeding sequence the z-values grow without limit.

Quadratic Iteration / c real

1	2	3	4	5	6	7	8	9	10
c →	0	1,00	-1,00	-0,50	-0,25	1,50	2,00	2,10	1,99
n ↓	Z n	Z n	Z n	Z n	Z n	Z n	Z n	Z n	Z n
0	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1	0,00	-1,00	1,00	0,50	0,25	-1,50	-2,00	-2,10	-1,99
2	0,00	0,00	2,00	0,75	0,31	0,75	2,00	2,31	1,97
3	0,00	-1,00	5,00	1,06	0,35	-0,94	2,00	3,24	1,89
4	0,00	0,00	26,00	1,63	0,37	-0,62	2,00	8,37	1,59
5	0,00	-1,00	677,00	3,15	0,39	-1,11	2,00	68,00	0,53
6	0,00	0,00	*****	10,44	0,40	-0,26	2,00	*****	-1,71
7	0,00	-1,00	*****	109,57	0,41	-1,43	2,00	*****	0,94
8	0,00	0,00	*****	*****	0,42	0,55	2,00	*****	-1,11
9	0,00	-1,00	*****	*****	0,42	-1,19	2,00	*****	-0,75
10	0,00	0,00	*****	*****	0,43	-0,08	2,00	*****	-1,43
45	0,00	-1,00	#NUM!	#NUM!	0,48	-1,45	2,00	#NUM!	-1,98
46	0,00	0,00	#NUM!	#NUM!	0,48	0,60	2,00	#NUM!	1,92
47	0,00	-1,00	#NUM!	#NUM!	0,48	-1,13	2,00	#NUM!	1,68
48	0,00	0,00	#NUM!	#NUM!	0,48	-0,21	2,00	#NUM!	0,84
49	0,00	-1,00	#NUM!	#NUM!	0,48	-1,45	2,00	#NUM!	-1,28
50	0,00	0,00	#NUM!	#NUM!	0,48	0,62	2,00	#NUM!	-0,35

Table 6.1-1 Iteration sequence $z_{n+1} = z_n^2 - c$ for purely real c-values.

Nobody can carry out this kind of calculation with complex numbers in his head. For a few simple cases we will work out the boundary of the non-divergent region, before we try to obtain an overview of all possible c-values at once.

To begin with we will use purely real numbers c with no imaginary part. At each iteration step the current value is squared and c is subtracted. Thus the number only stays small if the square is not much larger than c. Furthermore, we will carry out the computation in a spreadsheet program.[2] In Table 6.1-1, we see the development for different real values of c.

In column 1 we find the variable n for $n = 0$ to 10. In order to get a glimpse of the later development, the values $n = 45$ to $n = 50$ are shown below. Next come the z_n-values, computed using the value of c listed in the first row.

Here is a brief commentary on the individual cases.

- Column 2, $c = 0$: z_n remains zero, no divergence.
- Column 3, $c = 1$: switching between $z = 0$ and $z = -1$, again no divergence.
- Column 4, $c = -1$: already after 5 steps the answers cannot be represented in the space available (#####) and after 15 steps they become larger than the biggest number with which EXCEL can work (#NUM!), namely 10^{200}. Certainly we have divergence here!

  Now we must investigate the boundary between $c = -1$ and $c = 0$.
- $c = 0.5$ (column 5): surely not small enough, divergence.
- $c = -0.25$ (column 6): the first new case in which the c-values do not grow beyond reasonable bounds.
- The remaining case (columns 6-8) show that the upper boundary lies near $c = 2.0$.

Conclusion: the iteration sequence diverges if $c < -0.25$ or $c > 2.0$. In between we find (as in the Feigenbaum scenario in Chapter 2) simple convergence or periodic points, hence finite limiting values.

We have here carried out the iteration in great detail, in order to

- give you a feel for the influence of the c-values,
- show you how effectively a spreadsheet works,
- prepare you for the next step in the direction of complex numbers.

The investigation for purely imaginary parameters is not so easily carried out. Even when we square, we create from an imaginary number a negative real one, and upon subtracting we obtain a complex number! In Tables 6.1-2 and 6.1-3 we always list the real and imaginary parts of c and z next to each other. Otherwise both are constructed like Table 6.1-1.

---

[2] We use EXCEL, German version. As a result the decimal numbers have commas in place of the decimal point.

Quadratic Iteration / c imaginary

1	2	3	4	5	6	7	8	9
c →	0,00	0,50	0,00	-0,50	0,00	1,00	0,00	1,10
n ↓	z-real	z-imag	z-real	z-imag	z-real	z-imag	z-real	z-imag
0	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1	0,00	-0,50	0,00	0,50	0,00	-1,00	0,00	-1,10
2	-0,25	-0,50	-0,25	0,50	-1,00	-1,00	-1,21	-1,10
3	-0,19	-0,25	-0,19	0,25	0,00	1,00	0,25	1,56
4	-0,03	-0,41	-0,03	0,41	-1,00	-1,00	-2,38	-0,31
5	-0,16	-0,48	-0,16	0,48	0,00	1,00	5,55	0,35
6	-0,20	-0,34	-0,20	0,34	-1,00	-1,00	30,66	2,83
7	-0,08	-0,36	-0,08	0,36	0,00	1,00	931,80	172,70
8	-0,13	-0,44	-0,13	0,44	-1,00	-1,00	#####	#####
9	-0,18	-0,39	-0,18	0,39	0,00	1,00	#####	#####
10	-0,12	-0,36	-0,12	0,36	-1,00	-1,00	#####	#####
45	-0,14	-0,39	-0,14	0,39	0,00	1,00	#NUM!	#NUM!
46	-0,14	-0,39	-0,14	0,39	-1,00	-1,00	#NUM!	#NUM!
47	-0,14	-0,39	-0,14	0,39	0,00	1,00	#NUM!	#NUM!
48	-0,14	-0,39	-0,14	0,39	-1,00	-1,00	#NUM!	#NUM!
49	-0,14	-0,39	-0,14	0,39	0,00	1,00	#NUM!	#NUM!
50	-0,14	-0,39	-0,14	0,39	-1,00	-1,00	#NUM!	#NUM!

Table 6.1-2 Iteration sequence for purely imaginary c-values.

- Columns 2 and 3 (c = 0.5i) show that a non-divergent sequence occurs.
- In columns 4 and 5 (c = -0.5i) we obtain the same numbers, except that the sign of the imaginary part is reversed.
- What we observe by comparing columns 2 and 3 with 4 and 5 can be expressed in a general rule: two conjugate complex[3] numbers c generate two sequences z which are always complex conjugates.
- In column 6 and 7 we see that c = 1.0i provides an upper limit... .
- ... which is clear by comparing with columns 8 and 9 (c = 1.1i).

To see that the behaviour on the imaginary axis is not as simple as on the real axis, take a look at

- Columns 10 and 11: for c = 0.9i the sequence diverges!

It is a total surprise that a small real part, for example

- c = 0.06105 + 0.9i (columns 12 and 13),

again leads to an orderly (finite) sequence (at least within the first 50 iterations).

The last two examples, columns 14 and 15 (c = 0.5+0.6i) and 16 and 17 (c = -0.3+0.5i) should convince you that non-divergent sequences can be found when c is well removed from the axes. Indeed the number -0.3+0.5i lies further to the left

[3]If you have difficulty with this concept, re-read the fundamental ideas of calculation with complex numbers in Chapter 4.

Quadratic Iteration / c imaginary

1	10	11	12	13	14	15	16	17
c →	0,00	0,90	,06105	0,90	0,50	0,60	-0,30	0,50
n ↓	z-real	z-imag	z-real	z-imag	z-real	z-imag	z-real	z-imag
0	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1	0,00	-0,90	-0,06	-0,90	-0,50	-0,60	0,30	-0,50
2	-0,81	-0,90	-0,87	-0,79	-0,61	0,00	0,14	-0,80
3	-0,15	0,56	0,07	0,47	-0,13	-0,60	-0,32	-0,72
4	-0,29	-1,07	-0,28	-0,84	-0,84	-0,45	-0,12	-0,04
5	-1,07	-0,28	-0,68	-0,43	0,01	0,15	0,31	-0,49
6	1,06	-0,30	0,22	-0,31	-0,52	-0,60	0,16	-0,81
7	1,03	-1,52	-0,11	-1,03	-0,58	0,02	-0,33	-0,75
8	-1,27	-4,03	-1,12	-0,68	-0,16	-0,63	-0,16	-0,01
9	-14,65	9,35	0,73	0,62	-0,87	-0,40	0,33	-0,50
10	127,30	*****	0,08	0,00	0,10	0,09	0,16	-0,82
45	#NUM!	#NUM!	0,25	0,36	0,03	0,15	0,32	-0,50
46	#NUM!	#NUM!	-0,13	-0,72	-0,52	-0,59	0,15	-0,82
47	#NUM!	#NUM!	-0,56	-0,72	-0,58	0,02	-0,35	-0,74
48	#NUM!	#NUM!	-0,26	-0,09	-0,17	-0,62	-0,13	0,02
49	#NUM!	#NUM!	0,00	-0,85	-0,85	-0,40	0,32	-0,50
50	#NUM!	#NUM!	-0,79	-0,90	0,07	0,08	0,15	-0,82

Table 6.1-3 Iteration sequence for purely imaginary and complex c-values.

than the boundary at $c = -0.25$, which have have discovered on the real axis.

We certainly will not obtain a complete overview using the tables - the problem is too complicated. We must resort to other means, and use graphical representation.

Every possible Julia set is characterised by a complex number. If we make the c-plane the basis of our drawing, each point in it corresponds to a Julia set. Since the point can be coloured either black or white, we can encode information there. As already stated, this is the information on the connectivty of the Julia set. If it is connected, a point is drawn at the corresponding screen position. If the appropriate Julia set is not connected, the screen remains blank at that point. In the following pictures, all points of the complex c-plane are drawn, for which that value of c belongs to the basin of the finite attractor. A program searches the plane point by point. In contrast to the Julia sets of the previous section, the initial value is fixed at $z_0 = 0$, while the complex number c changes.

The corresponding program is developed from Program Fragment 5.2-2. To start with, we have changed the name of the central functional procedure. It must throughout be called from the procedure `Mapping`. The above parameter will be interpreted as `Creal` and `Cimaginary`. We define `x` and `y` as new local variables, intialised to the value 0.

Program Fagment 6.1-1.

```
FUNCTION MandelbrotComputeAndTest (Creal, Cimaginary :
               real) : boolean;
  VAR
    iterationNo : integer;
    x, y, xSq, ySq, distanceSq : real;
    finished: boolean;
  PROCEDURE  StartVariableInitialisation;
  BEGIN
    finished := false;
    IterationNo := 0;
    x := 0.0; y := 0.0;
    xSq := sqr(x);  ySq := sqr(y);
    distanceSq := xSq + ySq;
  END; (* StartVariableInitialisation *)
  PROCEDURE compute;
  BEGIN
    IterationNo := IterationNo + 1;
    y := x*y;
    y := y+y-Cimaginary;
    x := xSq - ySq -Creal;
    xSq := sqr(x); ySq := sqr(y);
    distanceSq := xSq + ySq;
  END; (* compute *)

  PROCEDURE test;
  BEGIN
    finished := (distanceSq > 100.0);
  END; (* test *)

  PROCEDURE distinguish;
  BEGIN (* Does the point belong to the Mandelbrot set? *)
    MandelbrotComputeAndTest : =
       (IterationNo = MaximalIteration);
  END;  (* distinguish *)

 BEGIN (* MandelbrotComputeAndTest *)
   StartVariableInitialisation;
   REPEAT
     compute;
     test;
   UNTIL (IterationNo = MaximalIteration) OR finished;
```

```
      distinguish;
    END;   (* MandelbrotComputeAndTest *)
```

The figure that this method draws in the complex plane is known as the *Mandelbrot set*. To be precise, points belong to it when, after arbitrarily many iterations, only finite z-values are produced. Since we do not have arbitrarily much time, we can only employ a finite number of repetitions. We begin quite cautiously with 4 steps.

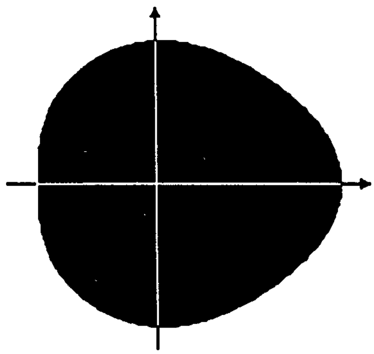


Figure 6.1-3 Mandelbrot set (4 repetitions).

As we see from the coordinate axes that are included, the resulting shape surrounds the origin asymmetrically. In the calculation in Table 6.1-1 we have already noticed that the basin stretches further in the positive direction. After two further iterations the egg-shaped basin begins to reveal its first contours.

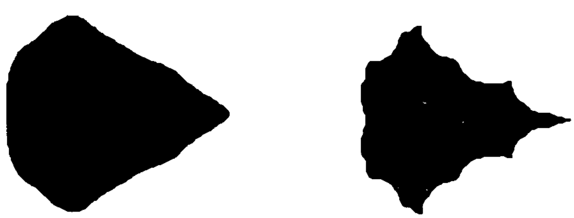
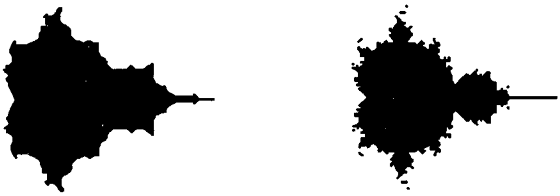


Figure 6.1-4 and 5 Mandelbrot set (6, respectively 8 repetitions).

It gradually emerges that the edge of the figure is not everywhere convex, but that in some places it undergoes constrictions.

The mirror symmetry about the real axis is obvious. It means that a point above the real axis has the same convergence behaviour as the corresponding point below. The complex numbers corresponding to these points are complex conjugates. In Table 6.1-2 we have already seen that such complex numbers have similar behaviour. They produce conjugate complex sequences.



Figures 6.1-6 and 7 Mandelbrot set (10 - 20 repetitions).

In Figure 6.1.6 some unusual points can be identified. The right-hand outermost point corresponds to $c = 2$. There the figure has already practically reduced to its final form: it exists only as a line along the real axis. However, magnified sections soon show that it possesses a complicated structure there.

The left upper branch of the same figure lies on the imaginary axis, where c has the value $c = i$. If we move from there directly downwards, we leave the basin of attraction, as we already know from Table 6.1-3 (columns 10 and 11).

Even if the drawing at first sight appears to show the opposite, the figure is connected. On the relatively large grid that determines the screen, we do not always encounter the extremely thin lines of which the figure is composed at many places. We already know this effect from the pictures of Julia sets.

All pictures can be combined and lead to 'contour lines'. In the next picture we not only draw the sets. In addition we draw the points, for which we can determine, after 4, 7, 10, 13, or 16 iterations, that they do not belong to it.

At this point we should describe the form of the Mandelbrot set. But, you may well ask, are there really similes for this uncommonly popular figure, which have not already been stated elsewhere?

In many people's opinion, it is reminiscent of a tortoise viewed from above, Randow (1986). Others see in it a remarkable cactus plant with buds, Clausberg (1986). For some mathematicians it is no more than a filled-in hypocycloid, to which a series of circles have been added, on which sit further circles, Durandi (1987). They are surely all correct, as also is the Bremen instructor who was reminded of a 'fat bureaucrat' (after rotation through 90°). Personal attitudes and interpretations certainly play a role when describing such an unusual picture. Two perceptions are involved, in our opinion. First, a familiarity with the forms that resemble natural shapes, in particular those seen upon

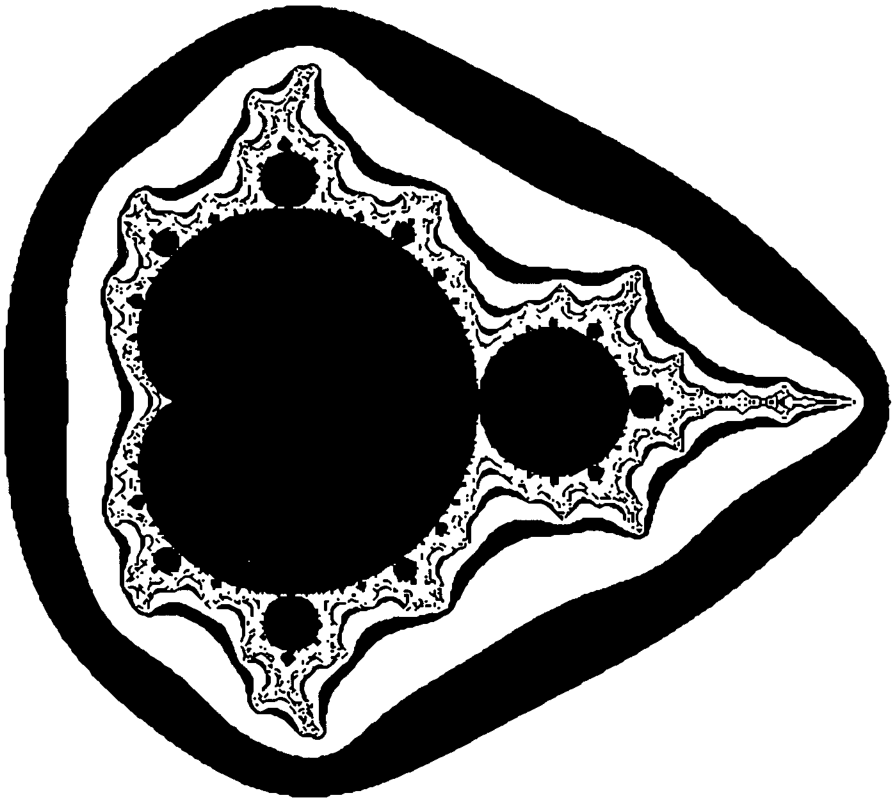


Figure 6.1-8 Mandelbrot set (100 repetitions, 'contour lines' up to 16).

magnification. On the other hand, the strangeness of this type of repetition is very different from that of, e.g. a leaf and a tree. The structure on different scales, whereby more features of the shape emerge, contradicts our normal vision. It makes us rethink our perceptions and eventually reach a new understanding.

In name-giving, and the fascination of form (and colour), we will follow the example of the Research Group at the University of Bremen, when in 1983 they were first able to produce the Mandelbrot set in their graphics laboratory. The name 'Gingerbread Man'[4] arose spontaneously, and we find it so appropriate that we will also use it.

The figure itself is not much older than the name. In the Spring of 1980 Benoit Mandelbrot first caught a glimpse of this graphic on a computer, hence its more formal

[4]*Translator's note*: 'Apfelmännchen' in the German, literally 'Little Apple Man'. The 'translation' in the text, traditional among some sections of the English-speaking fractal community, captures the style of the German term. It is also a near-pun on 'Mandelbrot', which translates as 'almond bread'.

name.[5]

Never before has a product of esoteric mathematical research become a household word in such a short time, making numerous appearances on notice-boards, taking up vast amounts of leisure time,[6] and so rapidly becoming a 'superstar'.

In order to avoid misunderstandings in the subsequent description, let us define our terms carefully.[7]

- The complete figure, as in Figure 6.1-9, we will call the *Mandelbrot set* or *Gingerbread Man*. It is the basin of the 'finite attractor'.
- The approximations to the figure, as shown in Figure 6.1-8, are the *contour lines* or *equipotentialsurfaces*. They belong to the basin of the attractor ∞.
- In the Mandelbrot set we distinguish the *main body* and the *buds*, or baby Gingerbread Men.
- At some distance from the clearly connected central region we find *higher-order Mandelbrot sets* or *satellites*, connected to the main body by *filaments*. The only filament that can be clearly seen lies along the positive real axis. But there are also such connections to the apparently isolated points above and below the figure.
- We speak of *magnification* when not the whole figure, but only a section of it, is drawn. In a program we achieve this by selecting values for the variables `Left`, `Right`, `Bottom`, `Top`.

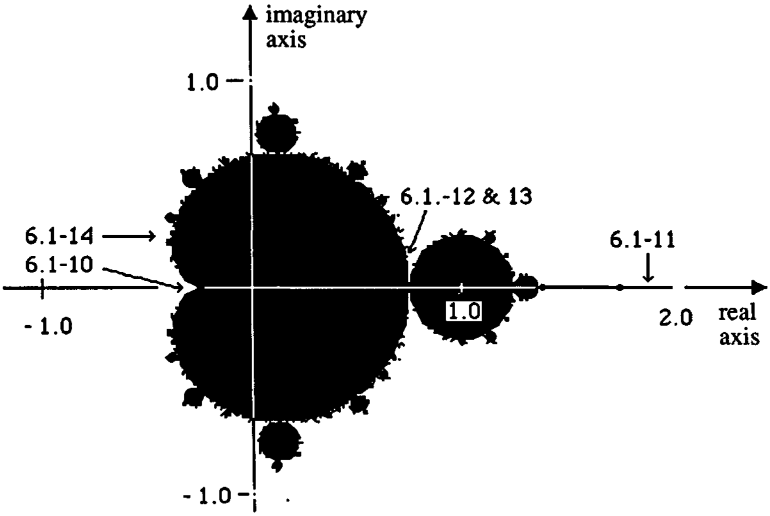


Figure 6.1-9 Mandelbrot set (60 repetitions).

[5]Compare the description in Peitgen and Richter (1986).

[6]If we can so describe the occupation with home and personal computers.

[7]They go back in particular to Peitgen and Richter (1986).

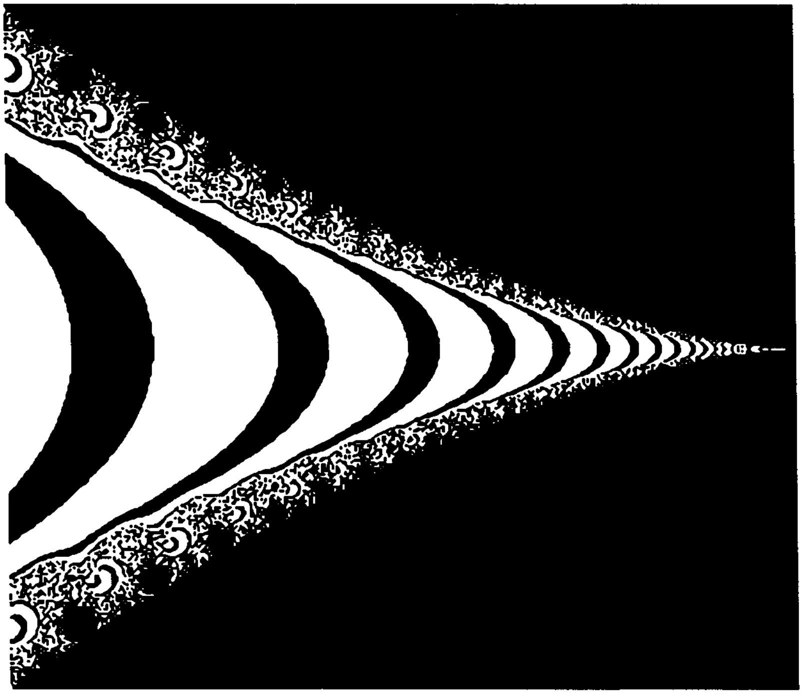


Figure 6.1-10 Mandelbrot set (section left of the origin).

The pictures that will now be drawn are all formed by sections of the original Gingerbread Man at different places and with different magnifications. In Figure 6.1-9, the regions concerned are marked by arrows.

This pictures shows a 'skewered Gingerbread Man' on the real axis. In order to obtain the contours more sharply, it is necessary to raise the iteration number. Here it is 100.

The magnification relative to the original is about 270-fold. Compare the central figure with the corresponding one in Figures 6.1-3 to 6.1-7.

Figures 6.1-12 and 6.1-13 show a section of the figure which is imprinted with spirals and various other forms. The massive black regions on the left side are offshoots of the main body. If the resolution were better, you would be able to identify them as baby Gingerbread Men.

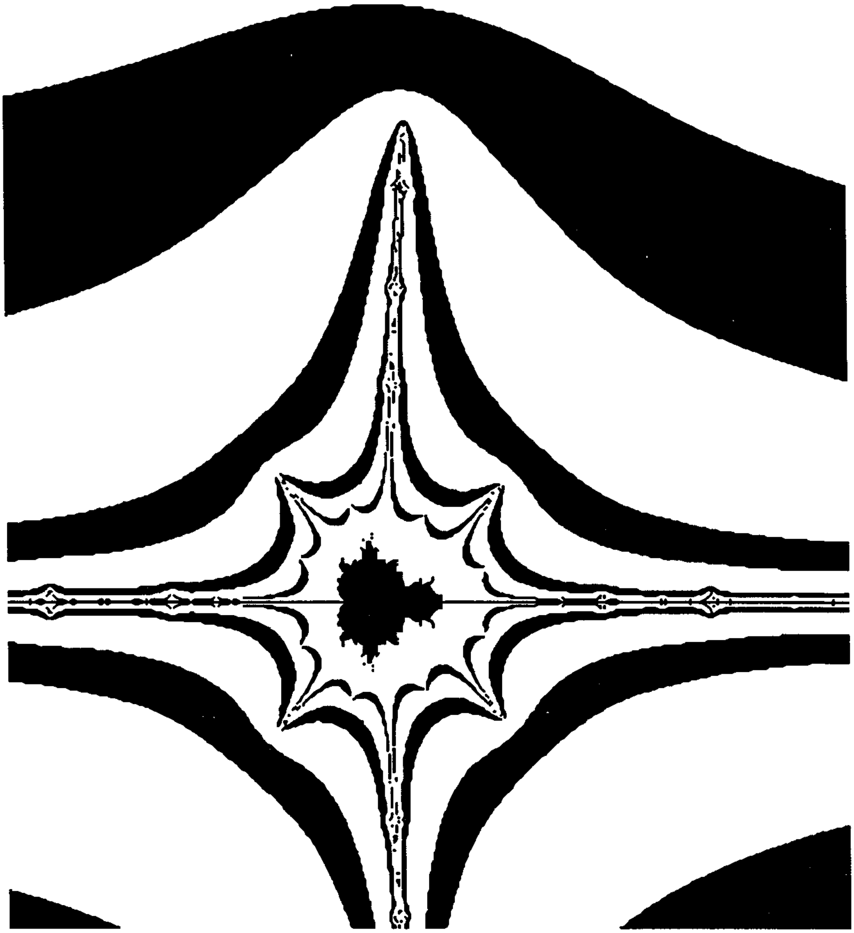


Figure 6.1-11 A Mandelbrot set of the second order.

The final picture, Figure 6.1-14, has been turned 90°. In the original position the tiny Mandelbrot set has an orientation almost opposite to that of the original. It is attached to a branch with very strongly negative values for c_{real}. The magnification is about 500-fold.

The self-similarity observed here is also known in natural examples. Take a look at a parsley plant. For many 'iterations' you can see that two branches separate from each main stem. In contrast to this, the self-similarity of mathematical fractals has no limit. In a lecture, Professor Mandelbrot briefly showed a picture with a section of the Gingerbread Man magnified $6*10^{23}$ times (known to chemists as Avogadro's Number), which quite clearly still exhibited the standard shape.

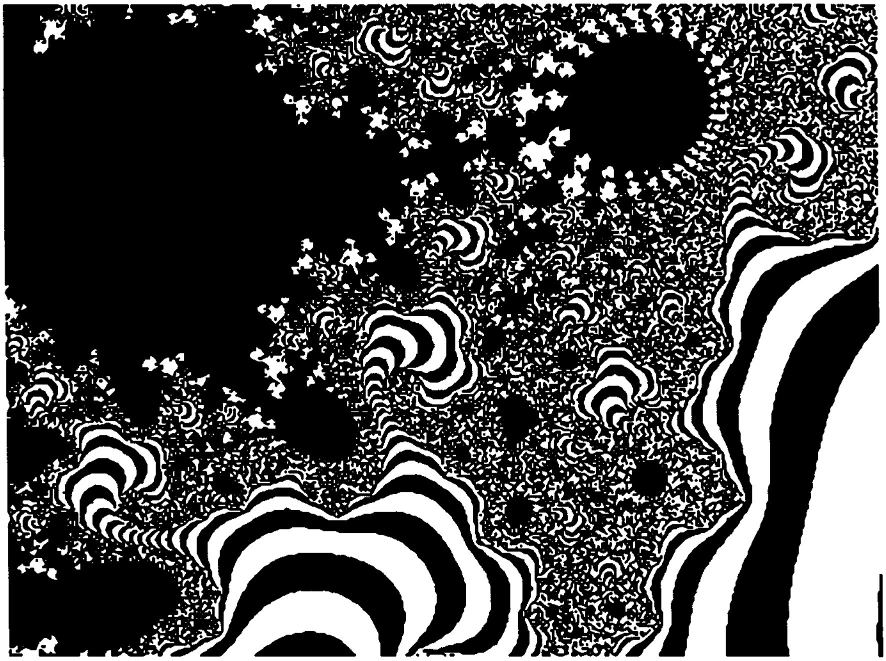


Figure 6.1-12 A section betwen the main body and a bud.

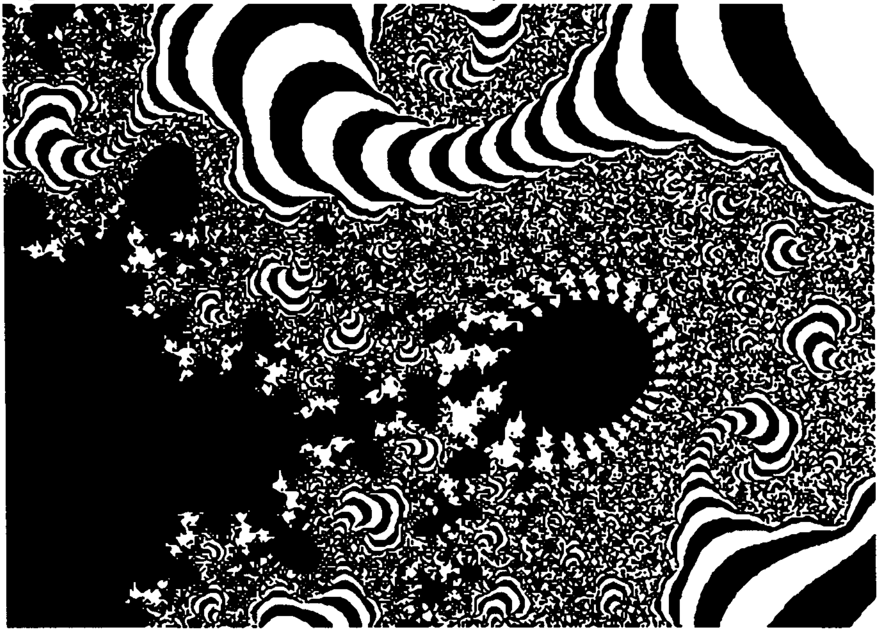


Figure 6.1-13 A section directly below Figure 6.1-12.

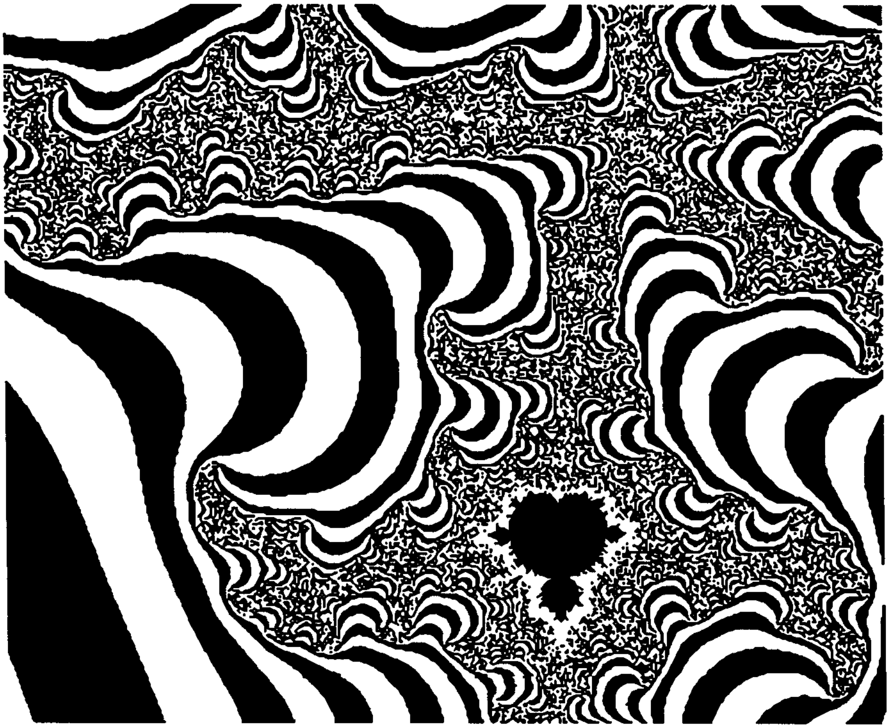


Figure 6.1-14 A satellite fairly far left.

Computer Graphics Experiments and Exercises for §6.1

Exercise 6.1-1

Using a spreadsheet calculator or a Pascal program, construct an instrument to represent iterative sequences in tabular form.

Verify the results of Tables 6.1-1 to 6.1-3.

Find out what changes if instead of the formula

$$z_{n+1} = z_n^2 - c$$

you use

$$z_{n+1} = z_n^2 + c.$$

Exercise 6.1-2

To make the processes on the real axis, and their periodicity, more obvious, draw a Feigenbaum diagram for this case. Draw the parameter c_{real} ($-0.25 \le c_{real} \le 2.0$) horizontally, and the z-value for 50 to 100 iterations vertically.

Exercise 6.1-3

Implement a Pascal program to draw the Mandelbrot set (Gingerbread Man).

Choose the limits for the region of the complex plane under investigation roughly as follows:

`Left` ≤ -1.0, `Right` ≥ 2.5 (`Left` $\leq c_{\text{real}} \leq$ `Right`)

`Bottom` ≤ -1.5, `Top` ≥ 1.5 (`Bottom` $\leq c_{\text{imaginary}} \leq$ `Top`).

For each screen pixel in this region start with the value

$$z_0 = x_0 + i * y_0 = 0.$$

Draw all points, for which after 20 iterations the value of

$$|f(z)|^2 = x^2 + y^2$$

does not exceed the bound 100.

It should take your computer about an hour to do this.

Exercise 6.1-4

When you have plenty of time (e.g. overnight), repeat the last exercise with 50, 100, or 200 steps. The more iteration steps you choose, the more accurate the contours of the figure will be.

Draw 'contour lines' too, to make the approximations to the Mandelbrot set more accurate.

Exercise 6.1-5

Investigate sections of the Mandelbrot set, for which you choose the values for `Left`, `Right`, `Bottom`, `Top` yourself.

Take care that the real section (`Right - Left`) and the imaginary section (`Top - Bottom`) always stay in the same proportion as the horizontal (`Xscreen`) and the vertical (`Yscreen`) dimensions of your screen or graphics window. Otherwise the pictures will appear distorted, and this strangeness can detract from their pleasing appearance. If you do not have a 1:1 mapping of the screen onto the printer output, you must also take care of this. For large magnifications you must also increase the iteration number.

In the pictures in this section you can already see that it is the boundary of the Mandelbrot set that is graphically the most interesting.

Worthwhile objects of investigation are:

- The filament along the real axis, e.g. the neighbourhood of the point $c = 1.75$,
- Tiny 'buds' that sprout from other larger ones,
- The regions 'around' the buds,
- The valleys between the buds,
- The apparently isolated satellites, which appear some distance from the main body, and the filaments that lead to them.

By now you will probably have found your own favourite regions within the Mandelbrot set. The boundary is, just like any fractal, infinitely long, so that anyone can explore new territory and generate pictures absolutely unseen before.

Exercise 6.1-6

The assumption that all iterations begin with

$$z_0 = x_0 + i * y_0 = 0$$

is based on a technical requirement. For our graphical experiments it is not a necessity. Therefore, try constructing sequences beginning with a number other than zero, with $x_0 \neq 0$ and/or $y_0 \neq 0$.

Exercise 6.1-7

The iteration sequence studied in this section is of course not the only one possible. Perhaps you would like to reconstruct the historical situation in the Spring of 1980, which B. B. Mandelbrot described in his article. Then you should try using the sequence

$$z_{n+1} = c * (1+z_n^2)^2 / (z_n^2 * (z_n^2 - 1))$$

or

$$z_{n+1} = c * z_n * (1 - z_n).$$

The boundaries and the remaining parameters you must find out for yourself by experimenting. Enjoy yourself: it will be worth it!

6.2 Tomogram of the Gingerbread Man

Julia sets and Mandelbrot sets, such a variety of forms, such fantastic patterns! And all that happens if you just iterate a simple nonlinear equation with a complex parameter.

If we resolve the possible parameters into their components, we find that we can influence the iteration mathematically in four places. The first two values available to us are the real and the imaginary component of the initial value z_0; the other two are the components of c.

Because these four quantities can be varied independently of one another, we obtain a fourfold range of new computational foundations and thus a fourfold system of new pictures, when we combine the quantities in different ways. The true structure of the attractor of the iteration formula

$$z_{n+1} = z_n^2 - c$$

is four-dimensional!

Most people experience severe difficulties thinking about three-dimensional situations. Everything that transcends the two dimensions of a sheet of paper discloses itself only with great difficulty, and only then when one has much experience with the subject under investigation. There is no human experience of four independent directions, and four mutually orthogonal coordinate axes cannot be represented artistically or technically. If people wish, despite this, to get a glimpse of higher-dimensional secrets, there is only one possibility: to make models that reduce the number of dimensions. Every architect or draughtsman reduces the number of dimensions of his real objects from three to two, and can thus put them on paper. Each photo, each picture

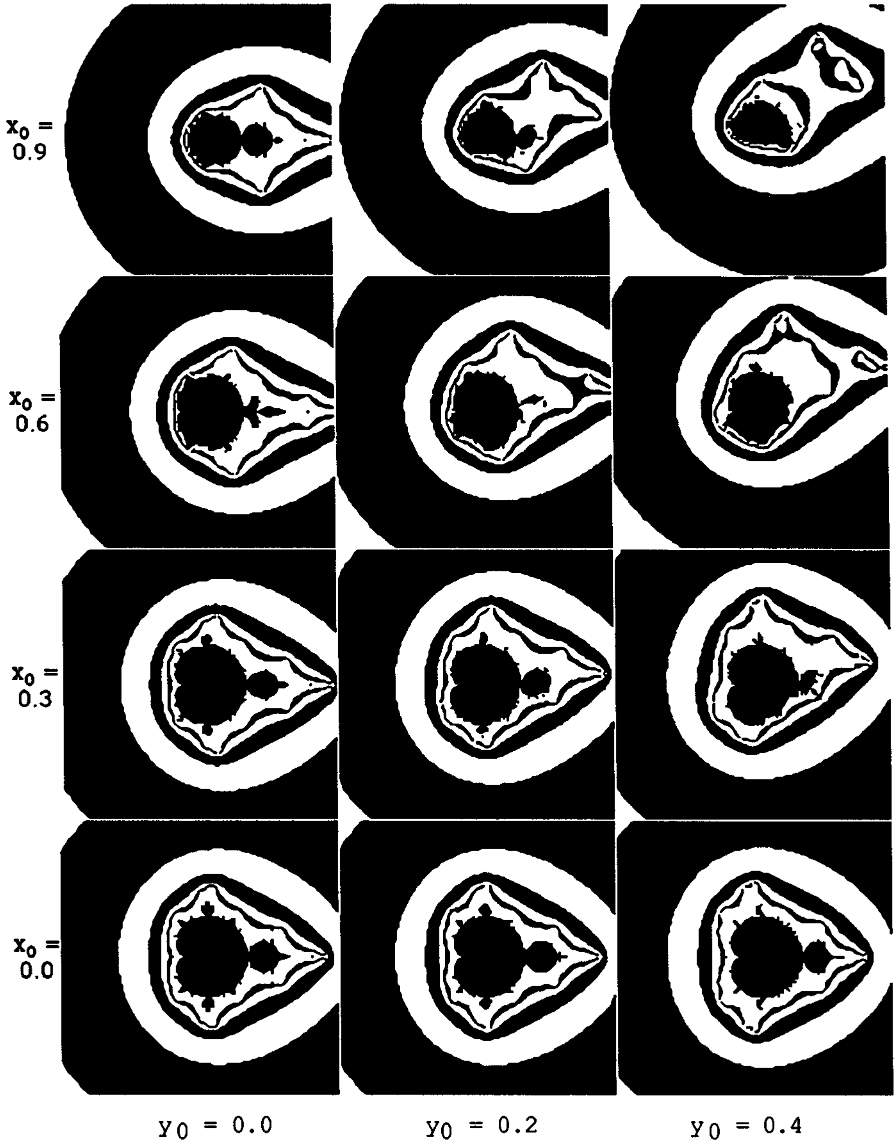


Figure 6.2-1 Quasi-Mandelbrot sets for different initial values.

basically does the same.

A particularly pretty example appears on the cover of Douglas R. Hofstadter's book *Gödel, Escher, Bach.* A shape, presumably cut from wood, has a shadow which from

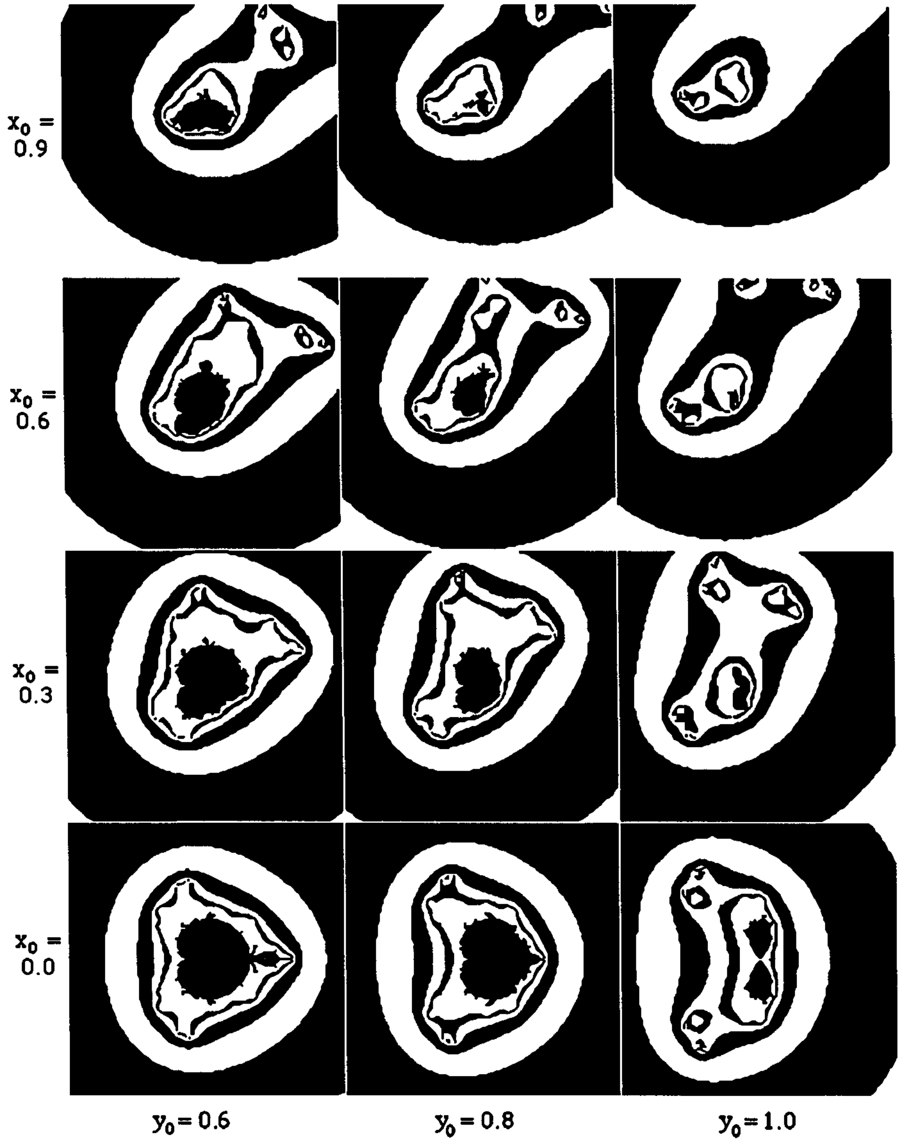


Figure 6.2-2 Quasi-Mandelbrot sets for different initial values.

different directions looks like the letters G, E, or B. Thus we see that such a reduction of dimension can colloquially be described as a *silhouette*, a *section*, or a *tomogram*. Architects and mathematicians alike build upon a three-dimensional coordinate system given by the three mutually perpendicular axes length, breadth, and height. In its simplest form we can create a section by fixing the numerical value in one coordinate direction – for example, in the plan of a house, looking at the height of the first storey. In this case the space that is graphically represented runs parallel to the two remaining axes. Complicated sections run obliquely to the axes.

The pictures on the previous two pages survey the form of the basin of attraction, when the iteration begins with the value shown round the edges. That is,

$$z_0 = x_0 + iy_0$$

is drawn in the middle of each of the 24 frames, together with the contours for the values 3, 5, and 7. In the frames, just as in the Mandelbrot set itself, c_{real} runs horizontally and $c_{imaginary}$ vertically. In the frame at lower left we see the standard Mandelbrot set.

In our previous computer graphics experiments we have always kept two variables fixed, so that of the four dimensions only two remain. This lets us draw the results on the screen without difficulty.

In §5.2 we fixed

$$c = c_{real} + i * c_{imaginary}$$

for every picture. The two components of

$$z_0 = x_0 + i * y_0$$

could then be changed, and provided the basis for drawing Julia sets.

For the Gingerbread man in §6.1 we did exactly the opposite, a mathematically 'perpendicular' choice. There

$$z_0 = x_0 + i * y_0$$

remained fixed, while

$$c = c_{real} + i * c_{imaginary}$$

formed the basis of the computation and the drawing.

Building on the four independent quantities x_0, y_0, c_{real}, $c_{imaginary}$ we will systematically investigate which different methods can be used to represent graphically the basin of the finite attractor.

	x_0	y_0	c_{real}	$c_{imaginary}$
x_0	X	1	2	3
y_0	1	X	4	5
c_{real}	2	4	X	6
$c_{imaginary}$	3	5	6	X

Table 6.2-1 Possibilities for representation.

In Table 6.2-1 we show the $4 \times 4 = 16$ possibilities, which can be expressed using two of these four parameters. The two quantities on the upper edge and down the side are kept constant in the graphics. The remaining two are still variable, and form the basis of the drawing. Because of the 'four-dimensional existence' of the basin of attraction, we select sections that are parallel to the four axes.

The four possibilities along the diagonal of Table 6.2-1 do not give a sensible graphical representation, because the same quantity occurs on the two axes. Corresponding to each case above the diagonal there is one below, in which the axes are interchanged. For the basic investigation that we carry out here, this makes no difference. Thus there remain 6 distinct types, with which to draw the basin of the iteration sequence.

By case 1 we refer to that in which x_0 and y_0 are kept fixed. Then c_{real} and $c_{imaginary}$ are the two coordinates in the plane that underlies each drawing. A special case of this, which we call case 1a, is when $x_0 = y_0 = 0$, and this leads to pictures of the Mandelbrot set or Gingerbread Man. Case 1b, for which $x_0 \neq 0$ and/or $y_0 \neq 0$, has already been worked out in Exercise 6.1-6. A general survey of the forms of the basins may be found in Figs. 6.2-1 and 6.2-2. The intial values x_0 and y_0 are there chosen from the range 0 to 1. Lacking any better name we call them *quasi-Mandelbrot sets.*

We recommend you to find out what happens when one or both components of the initial value are negative. Do you succeed in confirming our previous perception that the pictures of the basins becomes smaller and more disconnected, the further we go away from the starting value

$$z_0 = x_0 + i * y_0 = 0?$$

And that we obtain symmetric pictures only when one of the two components x_0 or y_0 has the value zero?

Another case, already used in §5.2 as a method for constructing Julia sets, is number 6 in the Table.

If you have already wondered why the Julia sets and the Gingerbread Man have so little in common, perhaps a small mathematical hint will help. In a three-dimensional space there are three possible ways in which two planes can be related. They are either equal, or parallel, or they cut in a line. In four-dimensional space there is an additional possibility: they 'cut' each other in a point. For Julia sets this is the origin. For the Gingerbread Man this is the parameter c, which is different for each Julia set.

The two real quantities x_0 and c_{real} are kept fixed throughout. The basis for the drawing is then the two imaginary parts y_0 and $c_{imaginary}$. The pictures are symmetric about the origin, which is in the middle of each frame.

The central basin is rather small near $c_{real} = 0.8$ Recall that at $c = 0.75$ we find the first constriction in the Gingerbread Man, where its dimension along the imaginary axis goes to zero. And this axis is involved in the drawing.

The central basin is fairly shapeless for small values of x_0 and $c_{imaginary}$. Perhaps this is just a matter of the depth of iteration, which for all the pictures here is given by `maximalIteration` = 100.

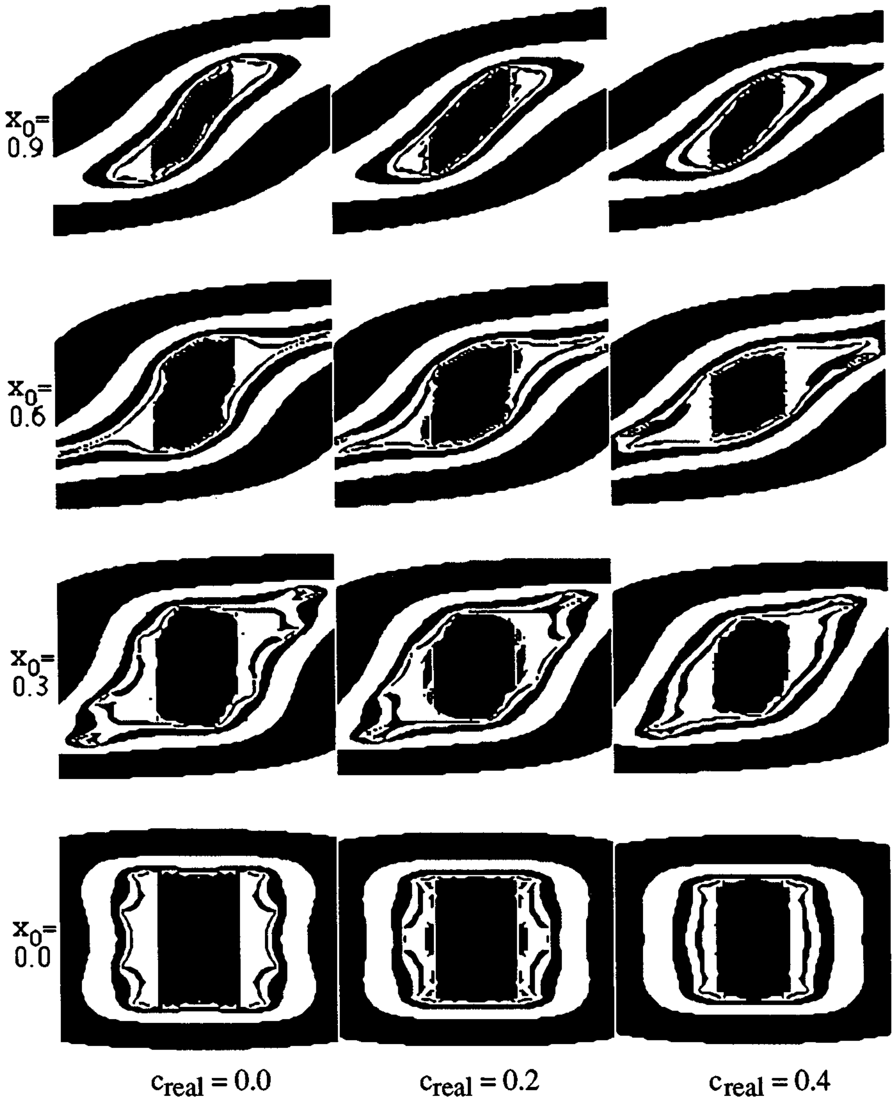


Figure 6.2-3 Diagram for case 2.

Interesting forms first arise for values such that the sets are rapidly disappearing. There it also looks as though the basin is no longer connected.

The pictures on the surrounding pages are symmetric about the origin. They all lie in the x_0-$c_{imaginary}$ plane, with $c_{imaginary}$ being drawn vertically. We also find this type of symmetry in Julia sets.

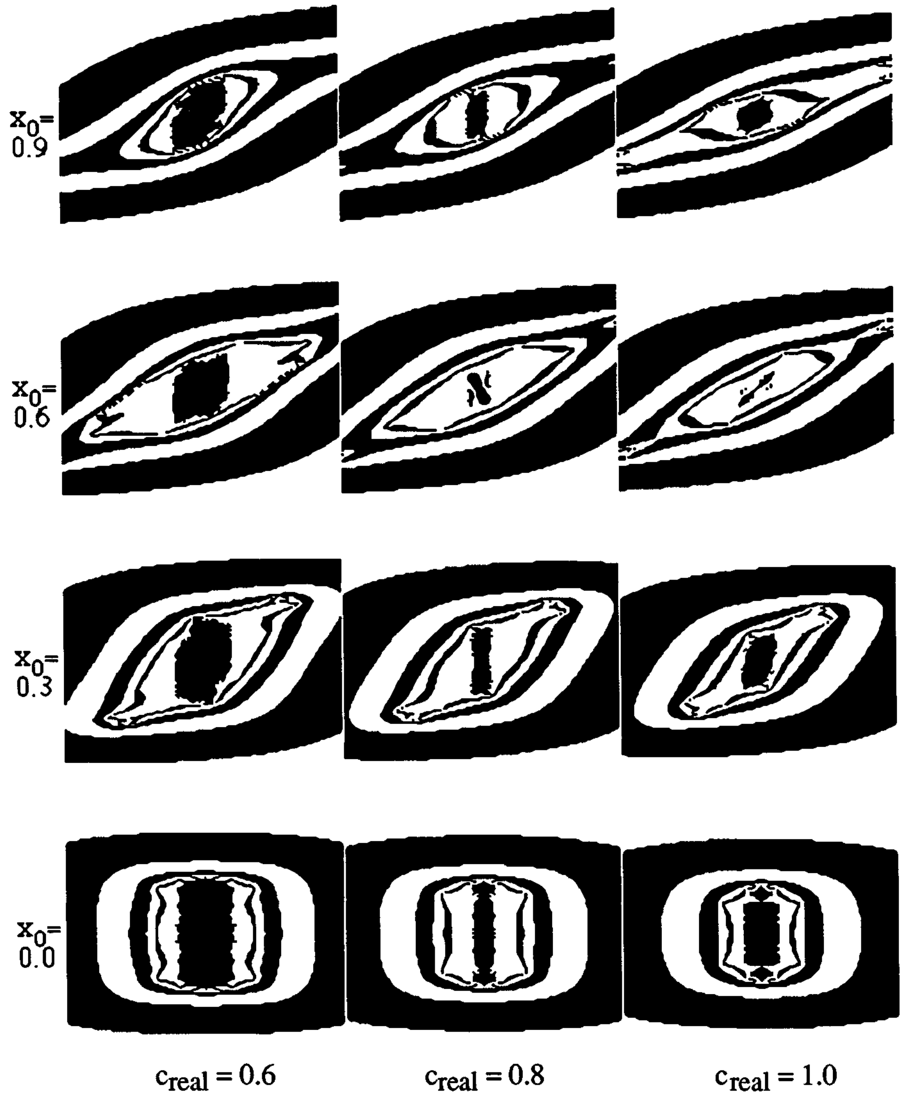


Figure 6.2-4 Diagram for case 2.

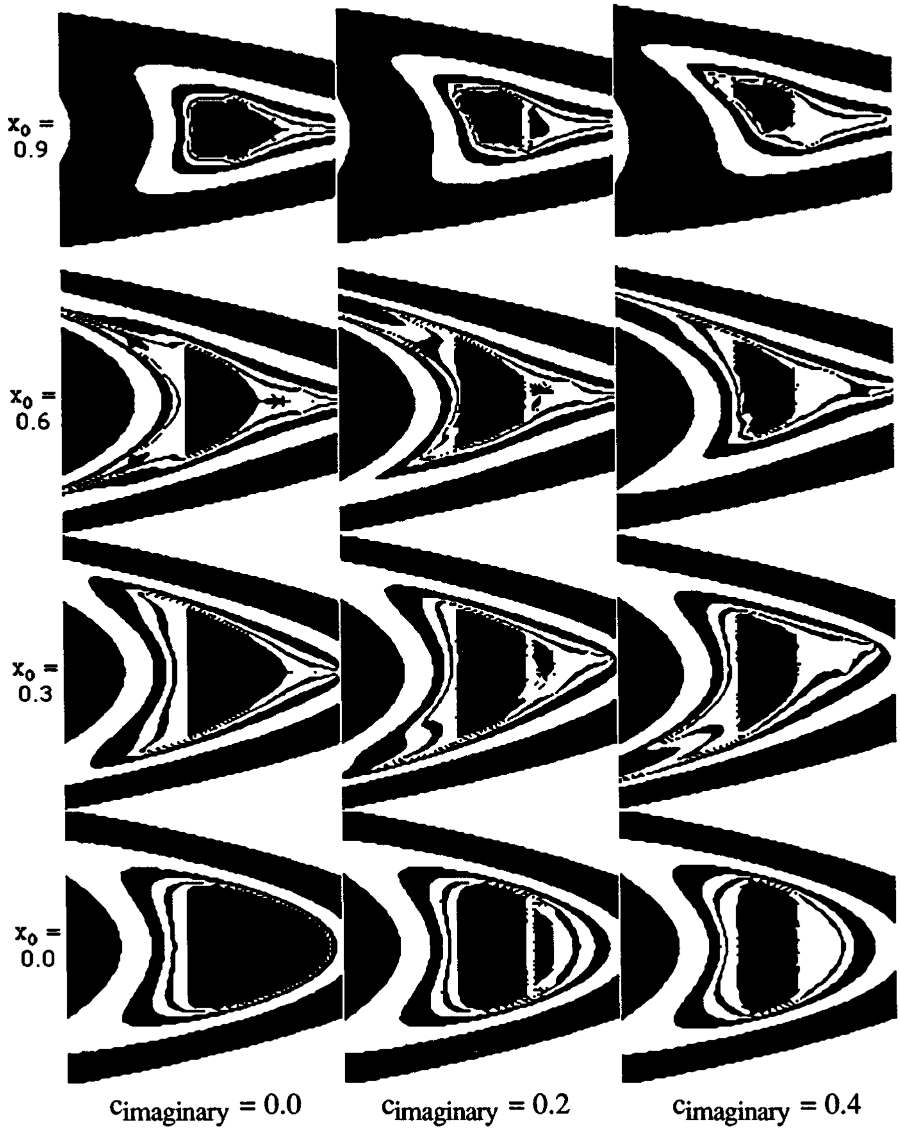


Figure 6.2-5 Diagram for case 3.

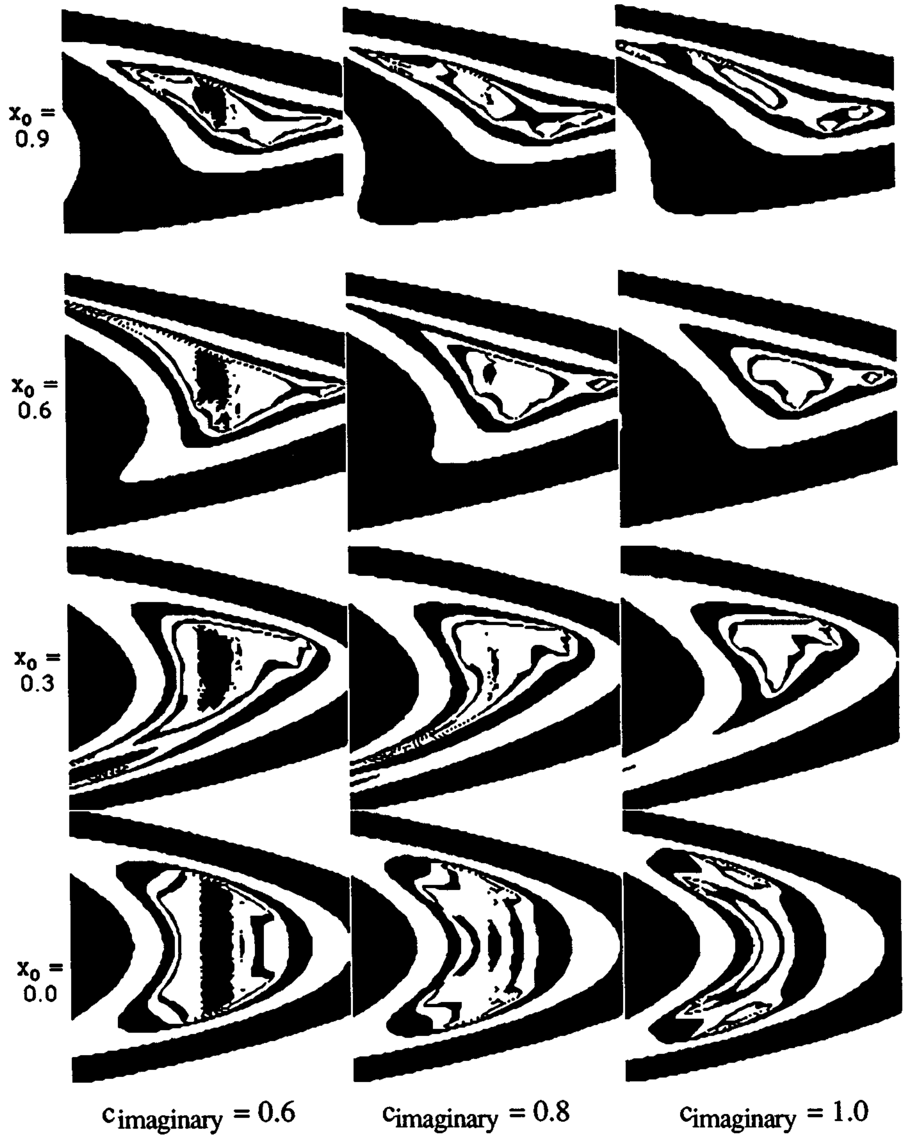


Figure 6.2-6 Diagram for case 3.

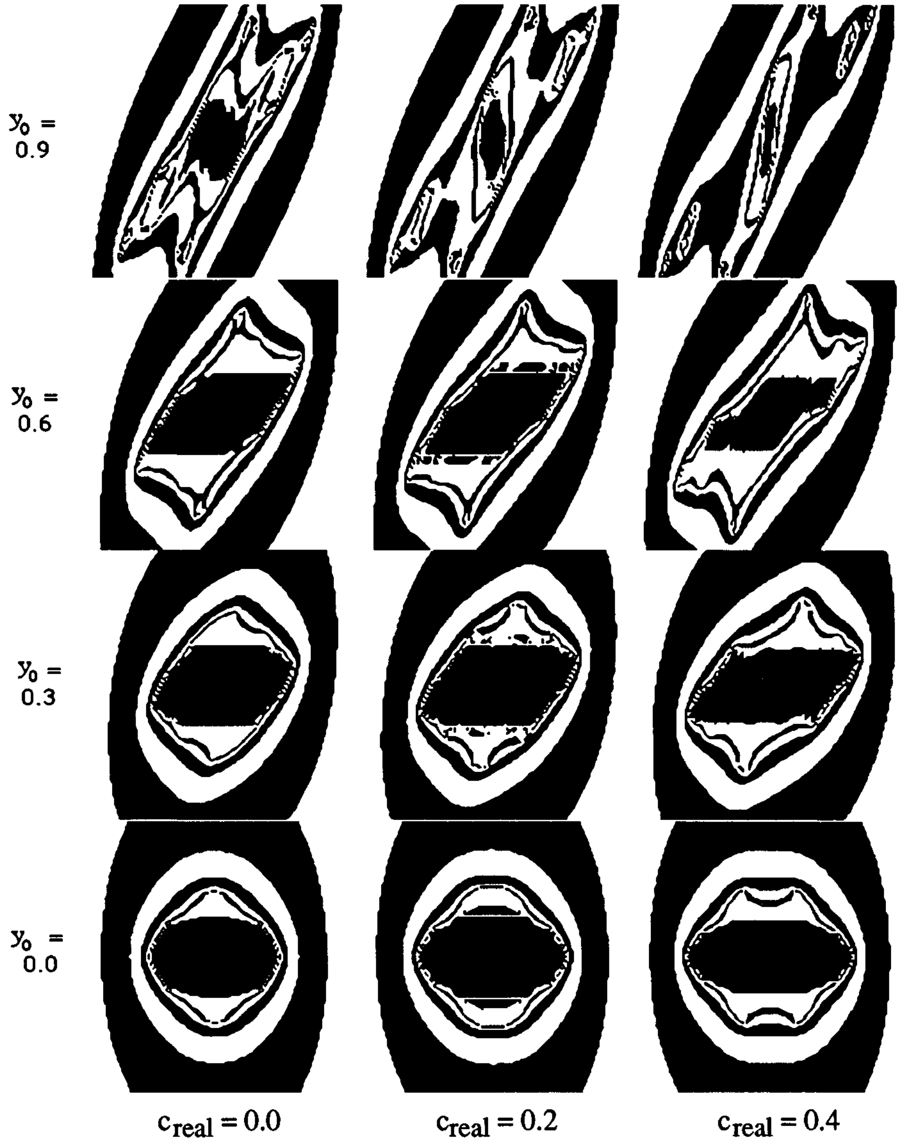


Figure 6.2-7 Diagram for case 4.

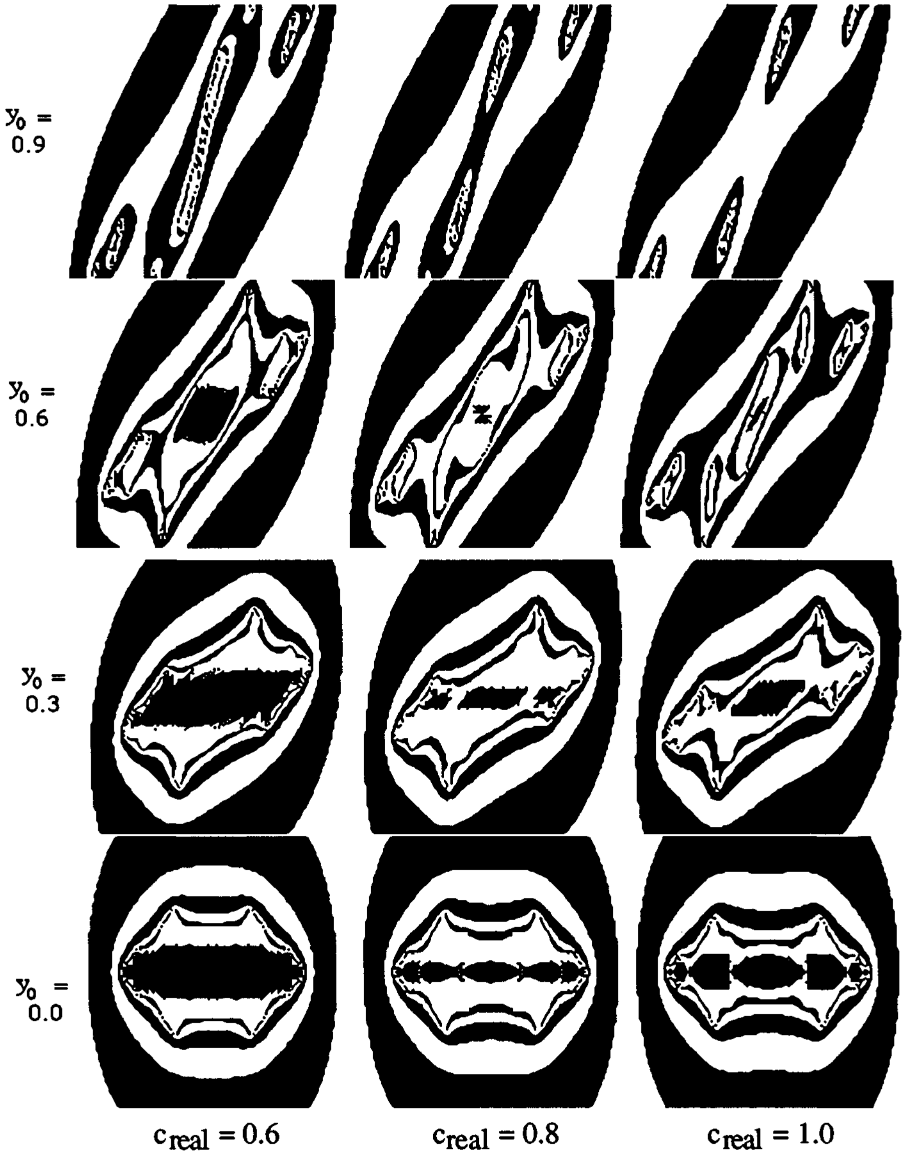


Figure 6.2-8 Diagram for case 4.

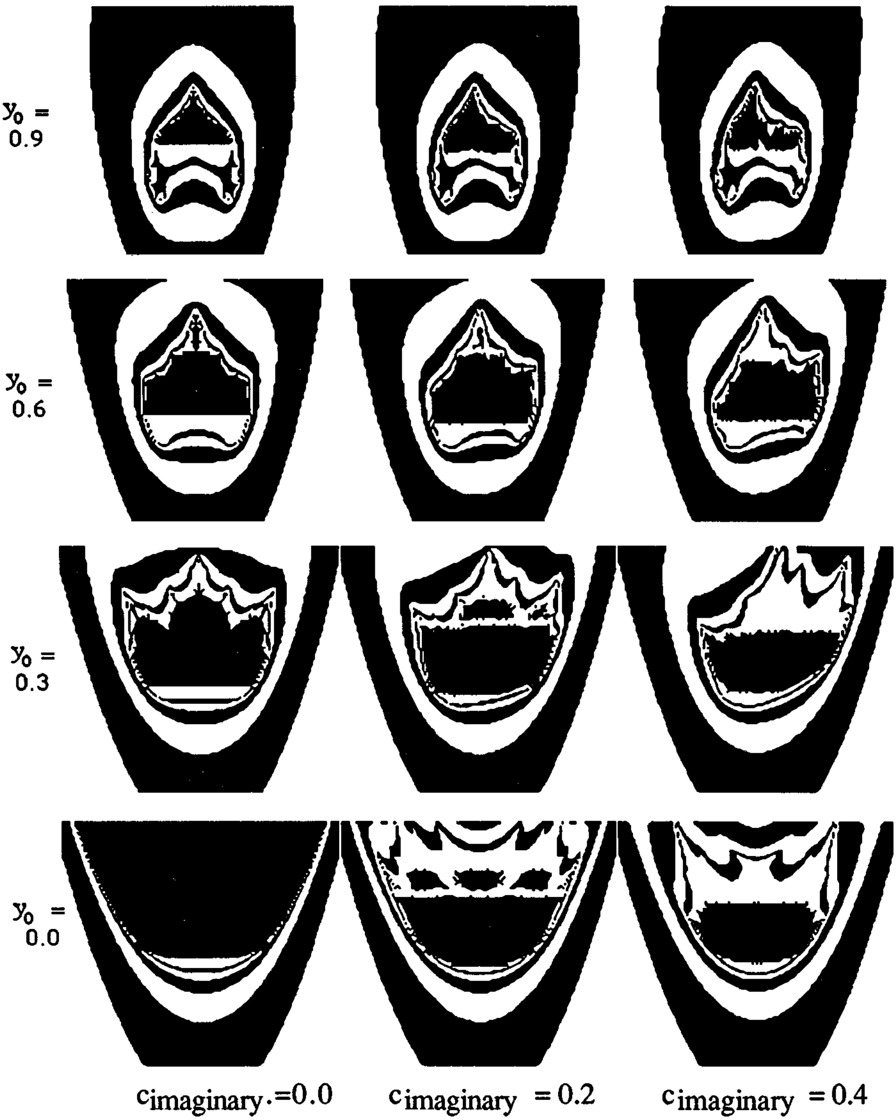


Figure 6.2–9 Diagram for case 5.

Figure 6.2-10 Diagram for case 5.

It is clear that the central basin is no longer connected. If you produce magnifications of the details, you must take into account the extreme irregularity (see Exercise 6.2-1). If we keep the two imaginary quantities y_0 and $c_{imaginary}$ constant, and in each picture change x_0 and c_{real}, we get the most remarkable and wildest shapes.

Provided $c_{\text{imaginary}} > 0$, the sets break up into many individual regions. In Fig. 6.2-11 you see a section of the boundary of such a set, drawn as in case 5. Its individual pieces are no longer joined up to each other.

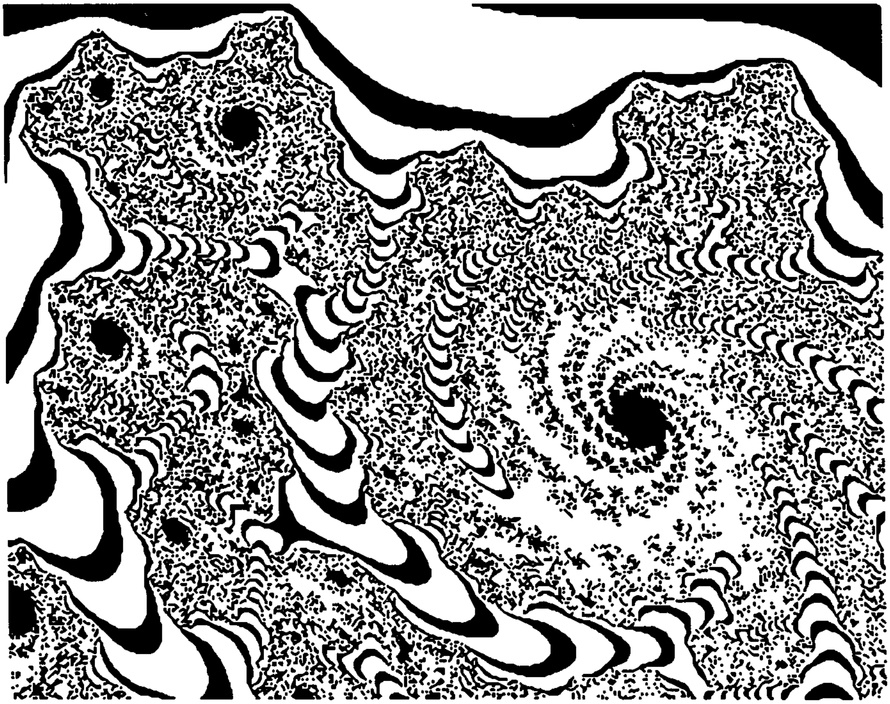


Figure 6.2-11 Detail with sevenfold spiral.

Computer Graphics Experiments and Exercises for §6.2

Exercise 6.2-1

To formulate specific exercises at this stage, you should avoid undervaluing your taste for adventure and fantasy. Simply cast your eyes a little further afield when you look at the previous pages. Negative and large parameters are wide open. Explore sections of the pictures shown. In some cases you must then employ different scales on the two axes, or else the pictures will be distorted.

We show you an attractive example above in Fig. 6.2-11. The two values $y_0 = 0.1$ and $c_{\text{imaginary}} = 0.4$ are fixed. From `Left` to `Right` the real starting values changes in the range $0.62 \le x_0 \le 0.64$. From `Bottom` to `Top` c_{real} varies: $0.74 \le c_{\text{real}} \le 0.8$. In the diagonal direction the original square figure is stretched by a factor of 3.

Exercise 6.2-2

Experiment with 'slanting sections'. From left to right change both c_{real} and $c_{\text{imaginary}}$ using a formula of the type

$$c_{\text{real}} = 0.5*c_{\text{imaginary}}$$

or similar. Or nonlinear expressions. From bottom to top x_0, say, varies, while y_0 stays fixed.

6.3 Fig-tree and Gingerbread Man

The definitive investigations that we have carried out for the Gingerbread Man in §6.1 have reminded us that in different regions of the complex plane it is possible to find different periodicities in the number sequences. It is therefore not so easy to distinguish between periodic behaviour and chaos - for instance, when the period is large. The highest number in respect of which we will consider periodicity is therefore 128. For purely computational procedures a whole series of problems arise. First, we must wait several hundred iterations before the computation 'settles down'. By this we mean it reaches a stage at which order and chaos can be distinguished. The next difficulty is that of comparison. The internal computer code for representing real numbers in Pascal is not completely unequivocal,[8] so that equality can only be tested using a trick. Thus we can investigate only whether the numbers differ by less than an assigned bound (e.g. 10^{-6}). In Pascal we can formulate a functional procedure like that in Program Fragment 6.3-1:

Program Fragment 6.3-1

```
FUNCTION equal (no1, no2: real) : boolean;
BEGIN
   equal := (ABS(no1-no2)) < 1.0E-6);
END;
```

For complex numbers, we must naturally check that both the real and the imaginary parts are equal:

Program Fragment 6.3-2

```
FUNCTION equal (z1Real, z1Imag, z2Real, z2Imag : real) :
            boolean;
BEGIN
   equal := (ABS(z1Real-z2Real)+ABS(z1Imag-z2Imag)
            < 1.0E-6;
END;
```

[8]The same sort of thing happens with the decimal numbers 0.1 and 0.0999..., which are equal, but written differently.

For example, if we have established that the 73rd number in a sequence is equal to the 97th, then we can conclude that a period of 24 then exists. This condition can be used to colour the uniformly black region inside the Gingerbread man.

We offer this as an example, and collect the results together:

- The constrictions in the Mandelbrot set (or, more poetically, the necks of the Gingerbread Man) divide regions of different periodicity from each other.
- The main body has period 1, that is, each number sequence $z_{n+1} = z_n^2 - c$ beginning with $z_0 = 0$, for c chosen within this region, tends towards a fixed complex limit. If c is purely real, so is the limit.
- The first circular region, which adjoins it to the right, leads to sequences of period 2.
- The further 'buds' along the real axis exhibit the periods 4, 8, 16, ...
- We find period 3 in the two next largest adjoining buds near the imaginary axis, near $c = 1.75$.
- For each further natural number we can find closed regions of the Mandelbrot set in which that periodicity holds.
- Regions that adjoin each other differ in periodicity by an integer factor. Adjacent to a region with periodicity 3 we find regions with the periods 6, 9, 12, 15, etc. The factor 2 holds for the largest bud, the factor 3 for the next largest, and so on.
- At the limits of the above regions the convergence of the number sequence becomes very poor, so that we need far more than 100 iterations to decide the question of periodicity.

We have already had a lot to do with this condition in Chapter 2, in the study of the Feigenbaum phenomenon. First, as a bridge-building exercise, we show that the new graphics display a similar state of affairs. Thus even the formula that underlies the Mandelbrot set can be drawn in the form of a Feigenbaum diagram.

The parameter to be varied here is, in the first instance, c_{real}. To begin with, the imaginary part $c_{\text{imaginary}}$ will be held at a constant value of 0. The program that we used in §6.1 to draw the Mandelbrot set changes a little.

To make as few alterations to the program as possible, we relinquish the use of the global variables `Visible` and `Invisible`. Their role will be played by `boundary` and `maximalIteration`, respectively.

Program Fragment 6.3-3

```
PROCEDURE Mapping;
  VAR
    Xrange : integer;
    deltaXPerPixel : real;
    dummy : boolean;
```

```
FUNCTION  ComputeAndTest (Creal, Cimaginary : real) :
                boolean;
  VAR
    IterationNo : integer;
    x, y, xSq, ySq, distanceSq : real;
    finished : boolean;

  PROCEDURE StartVariableInitialisation;
  BEGIN
    x := 0.0; y := 0.0;
    finished := false;
    iterationNo := 0;
    xSq := sqr(x); ySq := sqr(y);
    distanceSq := xSq + ySq;
  END; (* StartVariableInitialisation *)

  PROCEDURE ComputeAndDraw;
  BEGIN
    IterationNo := IterationNo + 1;
    y := x*y; y := y+y-Cimaginary;
    x := xSq - ySq - Creal;
    xSq := sqr(x); ySq := sqr(y);
    distanceSq := xSq + ySq;
    IF (IterationNo > Bound) THEN
               SetUniversalPoint (Creal,x);
  END; (* ComputeAndDraw *)

  PROCEDURE test;
  BEGIN
    finished := (distanceSq > 100.0);
  END; (* test *)

BEGIN (* ComputeAndTest *)
  StartVariableInitialisation;
  REPEAT
    computeAndDraw; test;
  UNTIL (IterationNo = MaximalIteration) OR finished;
END (* ComputeAndTest *)

BEGIN
  deltaxPerPixel := (Right - Left)/Xscreen;
```

```
      x := Left;
      FOR xRange := 0 TO Xscreen DO
      BEGIN
         dummy := ComputeAndTest (x, 0.0);
         x := x + deltaxPerPixel;
      END;
   END; (* Mapping *)
```

As you see, we now need a loop which increments the running variable `xRange`. As a result, variables for the other loops become superfluous. A new introduction is the variable `dummy`. By including it we can call the functional procedure `ComputeAndTest` as before. The drawing should be carried out during the iteration, and would be called from `Mapping` and built into `ComputeAndTest`.

As global parameters we use

```
Left := -0.25; Right := 2.0; Bottom := -1.5; Top := 1.5;
MaximalIteration := 300; Bound := 200;
```

and away we go!

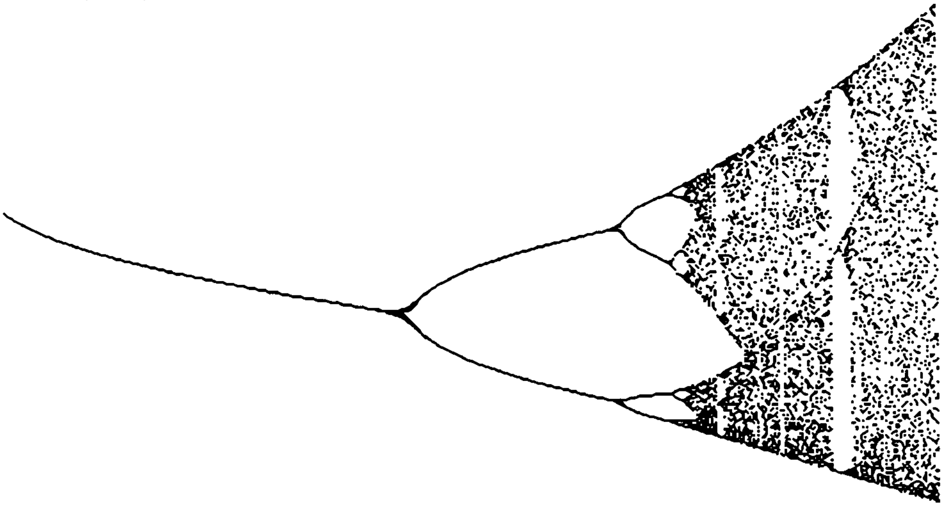


Figure 6.3-1 Feigenbaum diagram from the Mandelbrot set.

The result in Figure 6.3-1 appears very familiar, when we recall Chapter 2. It shows that the Feigenbaum scenario with bifurcations, chaos, and periodic windows is present in all respects along the real axis of the Mandelbrot set.

The next figure, 6.3-2, illustrates this by drawing the two diagrams one above the other: the periods 1, 2, 4, 8, ... etc. can be identified in the (halved) Mandelbrot set just as well as in the Feigenbaum diagram. And the 'satellite' Gingerbread Man corresponds in the Feigenbaum diagram to a periodic window with period 3! The diagram is only defined where it lies within the basin of the finite attractor, $-0.25 \leq c_{\mathrm{real}} \leq 2.0$. For other values of c_{real} all sequences tend to $-\infty$.

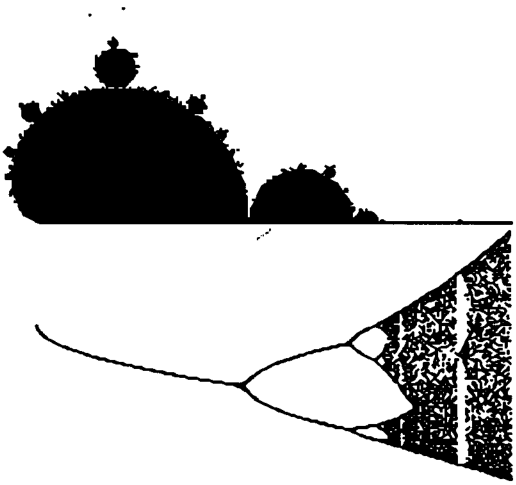


Figure 6.3-2 Direct comparison: Gingerbread Man and Fig-tree.

Now the real axis is a simple path, but not the only one along which a parameter can be changed. Another interesting path is shown in Figure 6.3-3.

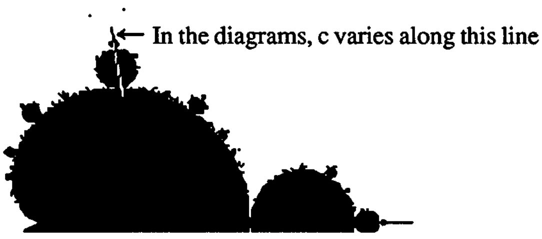


Figure 6.3-3 A parameter path in the Mandelbrot set.

This path is interesting because it leads directly from a region of period 1 to a bud of period 3, then on to 6 and 12.

This straight line is specified by two points, for example those at which the buds touch. Using a program to investigate the Mandelbrot set we have discovered the coordinates of the points P_1 (0.1255, 0.6503) between the main body and the first bud, and P_2 (0.1098, 0.882) where the next bud touches. The equation of the line can then be specified in two-point form. Using the parameter xRange, which runs from 0 to 400, we can travel along the desired section.

Mapping then takes the following form:

Program Fragment 6.3-4

```
(Working part of the procedure Mapping)
BEGIN (* Mapping *)
  FOR xRange := 0 TO xScreen DO
    dummy := ComputeAndTest
      (0.1288 - xRange*6.767E-5,          {Creal}
      0.6 + xRange*1.0E-3);               {Cimaginary}
  END ; (* Mapping *)
```

Now we must clarify what should actually be drawn, since ultimately both the real and imaginary parts of z are to be studied. We try both in turn. To draw the real part x, the appropriate part of ComputeAndTest runs like this:

Program Fragment 6.3-5 (Drawing commands in ComputeAndTest)

```
IF (IterationNo > Bound) THEN
    SetUniversalPoint (Cimaginary, x);
```

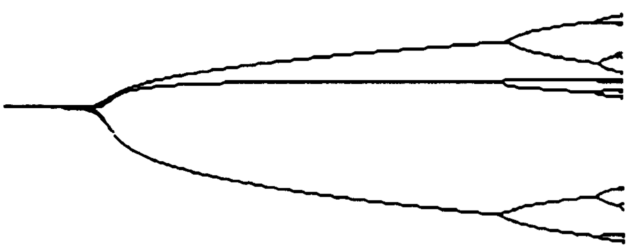


Figure 6.3-4 Quasi-Feigenbaum diagram, real part.

And as a matter of fact we see in this picture an example of 'trifurcation', when the period changes from 1 to 3. It looks rather as if first two paths separate, and then a further one branches off these. But that is just an artefact of our viewpoint.

In the next figure we look at the imaginary part y. For Figure 6.3-5 the relevant commands run like this:

Program Fragment 6.3-6 (Drawing commands in ComputeAndTest)

```
IF (IterationNo > Bound) THEN
    SetUniversalPoint (Cimaginary, y);
```

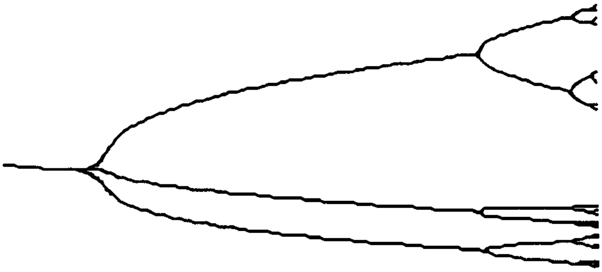


Figure 6.3-5 Quasi-Feigenbaum diagram, imaginary part.

We obtain a fairly complete picture using a pseudo-three-dimensional representation, as follows:

Program Fragment 6.3-7 (Drawing commands in `ComputeAndTest`)

```
IF (IterationNo > Bound) THEN
    SetUniversalPoint (Cimaginary - 0.5*x, y+0.866*x);
```

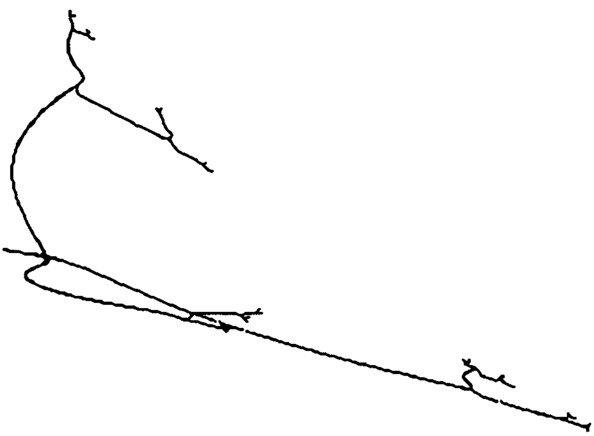


Figure 6.3-6 Pseudo-three-dimensional representation of trifurcation (oblique view from the front).

Here we get a complete view of the Quasi-Feigenbaum diagram. Two of the three main branches are shortened by perspective. The 24 tiny twigs appear at greater heights to come nearer the observer, whereas the first few points of the figure (period 1) are towards the rear. We can clearly see that the threefold branching occurs at a single point.

Computer Graphics Experiments and Exercises for §6.3

Exercise 6.3-1

Using your Gingerbread Man program, or your own variation on it, find other boundary points at which buds touch. We have already looked at the thinnest constriction and done additional computation at its midpoint.

Exercise 6.3-2

The method for obtaining the parameters for the path can be described here in general terms.[9]

We have found the two points P_1 (x_1,y_1) and P_2 (x_2,y_2). Thus, a 400-pixel line between them can be seen schematically as follows:

```
         P1                                  P2
--------+-----------------------------------+----------
0       50                                 350      400
```

Let the variable t run along this line (in the program t becomes `xRange`). Then for the two components we have

$$c_{real} = x_1 - (x_2-x_1)/6 + t*(x_2-x_1)/300,$$
$$c_{imaginary} = y_1 - (y_2-y_1)/6 + t*(y_2-y_1)/300.$$

Change your program to this general form, and investigate other interesting paths.

You might consider starting from the period 3 region and finding a further sequence leading into the region of period 9.

Or perhaps you can find a 'quintufurcation' - a fivefold branch-point?

Exercise 6.3-3

Naturally it is also possible to follow other lines than straight ones, which can be represented parametrically. Or those that can be obtained by joining straight segments. Take care not to leave the Gingerbread Man, however. If you do, the iteration will cease, and nothing will be drawn.

Exercise 6.3-4

We have obtained the pseudo-3D effect using the following trick.

In principle we have drawn the $c_{imaginary}$ diagram. But we have added to or subtracted from each of the two components a multiple of the x-value. The numbers 0.5 and 0.866 come from the values of the sine and cosine of 30°.

Experiment using other multiples.

[9] In program Fragment 6.3-4 we have proceeded somewhat differently.

6.4 Metamorphoses

With the Gingerbread Man and Julia sets constructed from the quadratic feedback mapping

$$z_{n+1} = z_n^2 - c$$

we have reached some definite conclusions. For investigations that go beyond these we can only provide a few hints. The questions that arise are so numerous that even for us they are for the most part unsolved and open. Treat them as problems or as further exercises.

First of all, we should point out that the quadratic mapping is not the only possible form for feedback. To be sure, it is the simplest that leads to 'nontrivial' results. In this book we have tried to avoid equations of higher degree, for which the computing time increases steeply, and we have only done so when it illustrates worthwhile principles.

In Peitgen and Richter (1986) p. 106, the authors describe the investigation of rational (that is, fractional) functions, which occur in physical models of magnetism. As above, there is a complex variable z which is iterated, and a constant c which in general is complex. The appropriate equations are:

$$z_{n+1} = \left(\frac{z_n^2+c-1}{2z_n+c-2}\right)^2 \qquad \text{Model 1,}$$

$$z_{n+1} = \left(\frac{z_n^3+3(c-1)z_n+(c-1)(c-2)}{3z_n^2+3(c-2)z_n+c^2-3c+3}\right)^2 \qquad \text{Model 2.}$$

Again there are two methods of graphical representation. We draw in either the z-plane or the c-plane, so that in the first case we choose a fixed c-value, and in the second case we begin with $z_0 = 0$. You should decide for yourself the c-values, which can also be real, the boundaries of the drawings, and the type of colouring used. Of course there is nothing wrong in experimenting with other equations or modifications.

We encounter rational functions if we pursue an idea from Chapter 4. There we applied Newton's method to solve a simple equation of the third degree. It can be shown (see Curry, Garnett and Sullivan, 1983) that we can investigate similar cubic equations with the formula

$$f(z) = z^3+(c-1)*z - c.$$

Here c is a complex number.

We begin the calculations once more with

$$z_0 = 0,$$

insert different c-values, and apply Newton's method:

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)} = z_n - \frac{z_n^3+(c-1)z_n+c}{3z_n^2+c-1}.$$

We find three different types of behaviour for this equation, depending on c. In many cases, in particular if we calculate with c-values of large modulus, the sequence converges to the real solution of the equation, namely

$$z = 1.$$

In other cases the sequence converges to another root. In a few cases the Newton method breaks down completely. Then we get cyclic sequences, that is, after a number of steps the values repeat:

$$z_{n+h} = z_n,$$

where h is the length of the cycle.

We can draw all of the points in the complex c-plane corresponding to the first case, using a Pascal program.

The calculation requires so many individual steps that it can no longer, as in the previous examples, be programmed 'at a walking pace'. For that reason we provide here a small procedure for calculating with complex numbers. The complex numbers are throughout represented as two 'real' numbers.

The components deal with addition, subtraction, multiplication, division, squaring, and powers. All procedures are constructed in the same way. They have one or two complex input variables (`in1r` stands for 'input 1 real part' etc.) and one output variable as VAR parameter.

In respect of division and powers we must take care of awkward cases. We do not, for example, consider it sensible to stop the program if we inadvertently divide by the number zero. Thus we have defined a result for this value too.

In your program these procedures can be defined globally, or locally within `ComputeAndTest`.

Program Fragment 6.4-1

```
PROCEDURE compAdd (in1r, in1i, in2r, in2i: real; VAR outr,
                  outi: real);
BEGIN
  outr := in1r + in2r;
  outi := in1i + in2i;
END;  (* compAdd *)

PROCEDURE compSub (in1r, in1i, in2r, in2i: real; VAR outr,
                  outi: real);
BEGIN
  outr := in1r - in2r;
  outi := in1i - in2i;
END;  (* compSub *)

PROCEDURE compMul (in1r, in1i, in2r, in2i: real; VAR outr,
```

```
                  outi: real);
BEGIN
   outr := in1r * in2r - in1i * in2i;
   outi := in1r * in2i + in1i * in2r;
END;  (* compMul *)

PROCEDURE compDiv (in1r, in1i, in2r, in2i: real; VAR outr,
                  outi: real);
   VAR numr, numi, den: real;
BEGIN
   compMul (in1r, in1i, in2r, -in2i, numr, numi);
   den := in2r * in2r + in2i * in2i;
   IF den := 0.0 THEN
      BEGIN
         outr := 0.0; outi := 0.0;  (* emergency solution *)
      END
   ELSE
      BEGIN
         outr := numr/den;
         outi := numi/den;
      END;
END;  (* compDiv *)

PROCEDURE compSq (in1r, in1i : real; VAR outr, outi: real);
BEGIN
   outr := in1r * in1r - in1i * in1i;
   outi := in1r * in1i * 2.0;
END;  (* compSq *)

PROCEDURE compPow (in1r, in1i, power: real; VAR outr, outi:
                  real);
   CONST
      halfpi := 1.570796327;
   VAR
      alpha, r : real;
BEGIN
   r := sqrt (in1r*in1r + in1i * in1i);
   IF  r > 0.0 then r := exp (power * ln(r));
   IF ABS(in1r) < 1.0E-9 THEN
      BEGIN
         IF in1i > 0.0 THEN alpha := halfpi;
```

```
                              ELSE alpha := halfpi + Pi;
        END ELSE BEGIN
          IF inlr > 0.0 THEN alpha := arctan (inli/inlr)
                           ELSE alpha := arctan (inli/inlr) + Pi;
        END;
      IF alpha < 0.0 THEN alpha := alpha + 2.0*Pi;
      alpha := alpha * power;
      outr := r * cos(alpha);
      outi := r * sin(alpha);
    END;  (* compPow *)
```

Having equipped ourselves with this utility we can now carry out an investigation of the complex plane. Replace the functional procedure `MandelbrotComputeAndTest` in your Gingerbread Man program by one based upon the above procedures. But do not be surprised if the computing time becomes a bit longer.

Program Fragment 6.4-2 (Curry-Garnett-Sullivan Method)

```
    FUNCTION ComputeAndTest (Creal, Cimaginary : real) :
                          boolean;
      VAR
        IterationNo : integer;
        x, y, distanceSq, intr, inti, denr, deni : real;
        (* new variables to store the denominator *)
        (* and intermediate results *)
        finished : boolean;

      PROCEDURE StartVariableInitialisation;
      BEGIN
        finished := false;
        IterationNo := 0;
        x := 0.0;
        y := 0.0;
      END; (* StartVariableInitialisation *)

      PROCEDURE compute;
      BEGIN
        IterationNo := IterationNo + 1;
        compSq (x, y, intr, inti);
        compAdd (3.0*intr, 3.0*inti, Creal-1.0, Cimaginary,
                          denr, deni);
        compAdd (intr, inti, cReal -1.0, Cimaginary, intr,
                          inti);
```

```
      compMul (intr, inti, x, y, intr, inti);
      compSub (intr, inti, Creal, Cimaginary, intr, inti);
      compDiv (intr, inti, denr, deni, intr, inti);
      compSub (x, y, intr, inti, x, y);
      distanceSq := (x-1.0) * (x-1.0) + y * y;
   END;  (* compute *)

   PROCEDURE test;
   BEGIN
      finished := (distanceSq < 1.0E-3);
   END (* FurtherTest *)

   PROCEDURE distinguish;
   BEGIN (* does the point belong to the set? *)
      ComputeAndTest := iterationNo < maximalIteration;
   END;  (* distinguish *)

BEGIN (* ComputeAndTest *)
   StartVariableInitialisation;
   REPEAT
      compute;
      test;
```

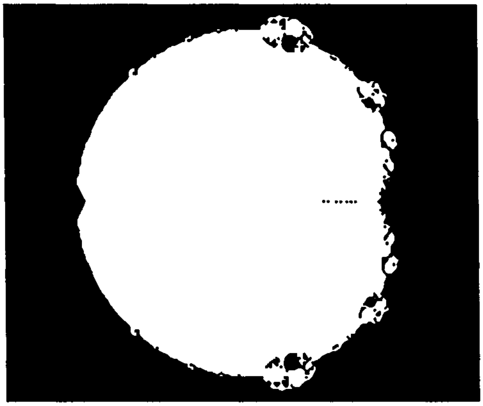


Figure 6.4-1 Basin of the attractor $z = 1$.

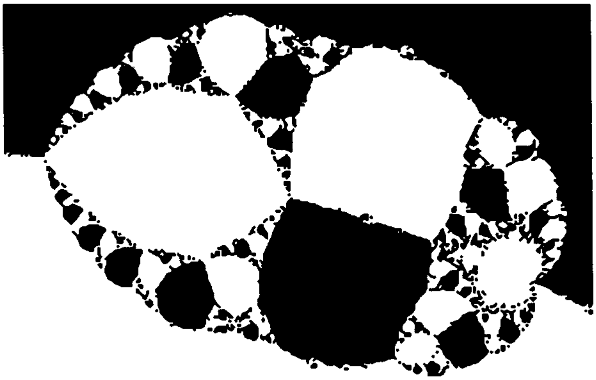


Figure 6.4-2 Section from Figure 6.4-1 (with a surprise!).

```
    UNTIL (IterationNo = MaximalIteration) OR finished;
    distinguish;
  END;  (* ComputeAndTest *)
```

As you see, the computing expense for each step has grown considerably. As a result you should not choose too large a number of iterations.

Figure 6.4-1 has a very clear structure and displays some interesting regions which are worth magnifying. For example, you should investigate the area around

$c = 1$,

$c = 0$,

$c = -2$.

The elliptical shape in the neighbourhood of

$c = 1.75i$

is enlarged in Figure 6.4-2.

- Black areas correspond to regions in which z_n converges to $z = 1$.
- Most white regions mean that case 2 holds there. The sequence converges, but not to $z = 1$.
- A white region at the right-hand end of the figure is an exception. This is a region where the sequences become cyclic.

Check this for c-values near

$c = 0.31 + i*1.64$.

You can already discern the result in Figure 6.4-2. In fact what you get is a close variant of the Gingerbread Man from §6.1.

This resemblance to the Gingerbread Man is of course no accident. What we named the 'finite attractor' at the start of Chapter 5 is mostly a cyclic attractor, as in this computation.

If the change to third powers already leads to such surprising results, what will we find for fourth or even higher powers? To get at least a few hints, we generalise the simple iteration equation

$$z_{n+1} = z_n^2 - c.$$

Instead of the second power we use the pth:

$$z_{n+1} = z_n^p - c.$$

We carry out the power computation using the procedure `compPow` (Program Fragment 6.4-1), which involves the global variable p. The changes to the Gingerbread man program are limited to the procedure `Compute`.

Program Fragment 6.4-3

```
PROCEDURE compute;
  VAR
    tempr, tempi: real;
BEGIN
  IterationNo := IterationNo + 1;
  compPow (x, y, p, tempr, tempi);
  x : = tempr - Creal;
  y := tempi - Cimaginary;
  xSq := sqr(x);
  ySq := sqr(y);
  distanceSq := xSq + ySq;
END;  (* compute *)
```

A brief assessment of the results shows that for the power $p = 1$ the finite attractor is confined to the origin. Every other c-value moves further away, that is, to the attractor ∞.

For very high values of p we can guess that the modulus of c plays hardly any role compared with the high value of z^p, so that the basin of the finite attractor is fairly close to the unit circle. Inside this boundary, the numbers always get smaller, and hence remain finite. Outside it they grow beyond any bound.

In the next few pages we will attempt to give you an overview of the possible forms of the basins of the finite attractor. For non-integer values of p we observe breaks in the pictures of the contour lines, which are a consequence of the way complex powers behave.

Because the calculation of powers involves complicated functions such as exponentials and logarithms, these computations take a very long time.

In each frame we see the central basin of attraction, together with contour lines for 3, 5, and 7 iterations. The attractor for $p = 1.0$, virtually a point, at first extends relatively diffusely, but by $p = 1.6$ acquires a shape which by $p = 2.0$ becomes the familiar Gingerbread Man. Between $p = 2.0$ and 3.0 it gains a further protuberance, so

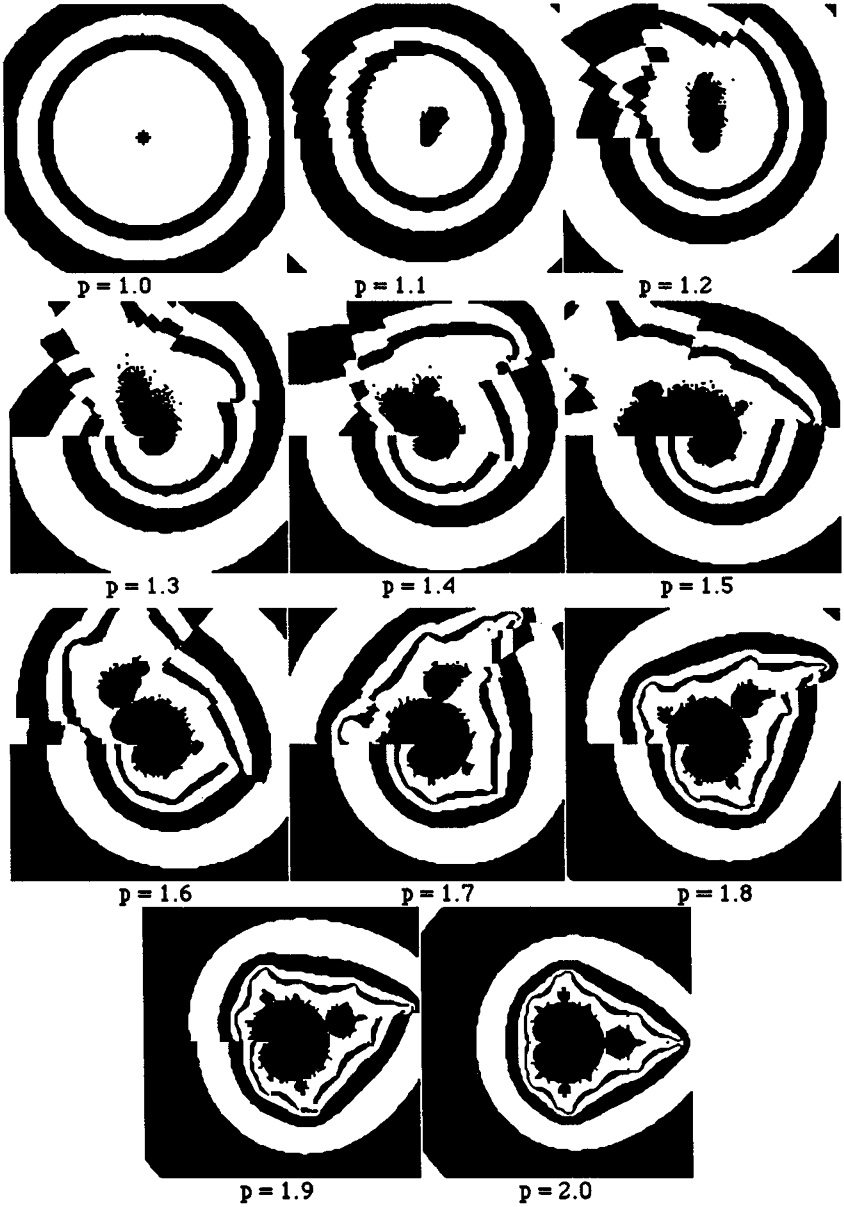


Figure 6.4–3 Generalised Mandelbrot set for powers from 1 to 2.

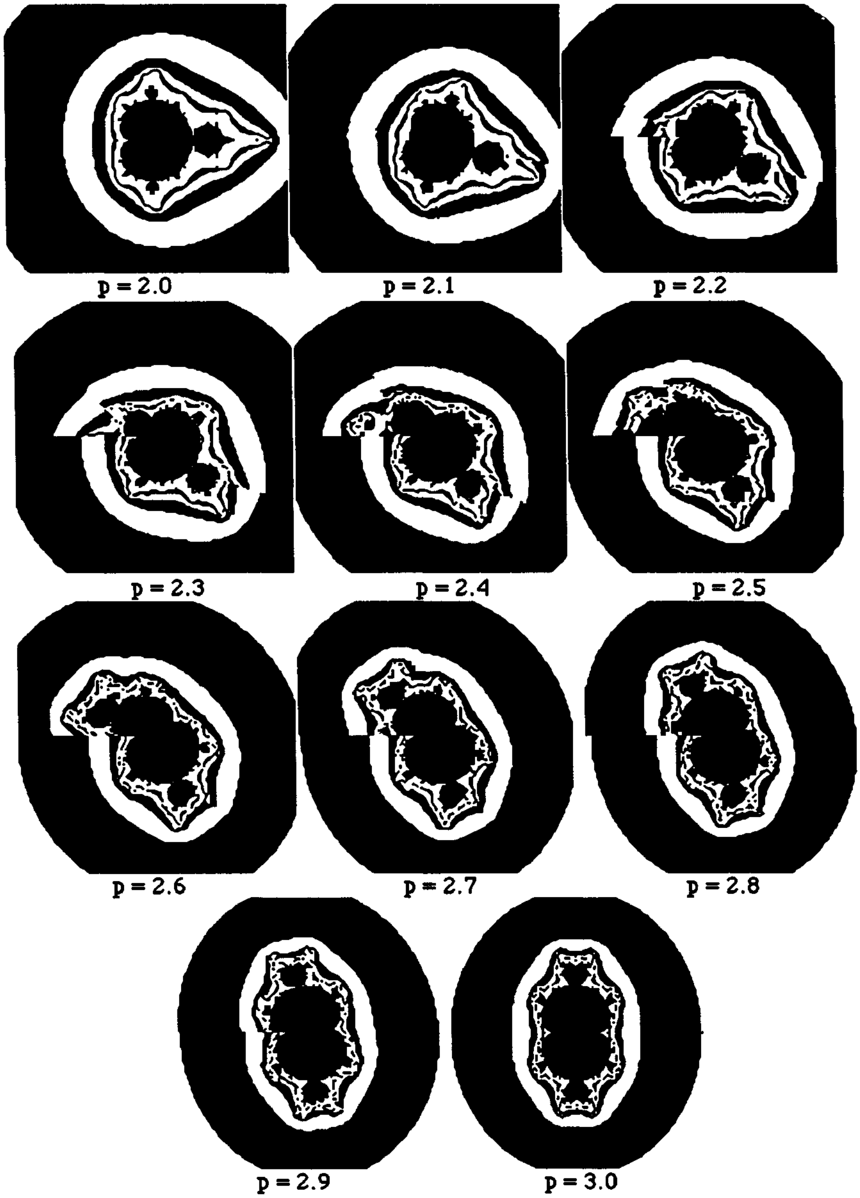


Figure 6.4–4 Generalised Mandelbrot set for powers from 2 to 3.

that we eventually obtain a very symmetric picture. The origin of the complex plane is in the middle of each frame.

The powers increase further, and at each whole number p a further bud is added to the basin. As already indicated, the figure grows smaller and smaller, and concentrates around the unit circle. We leave the investigation of other powers as an exercise.

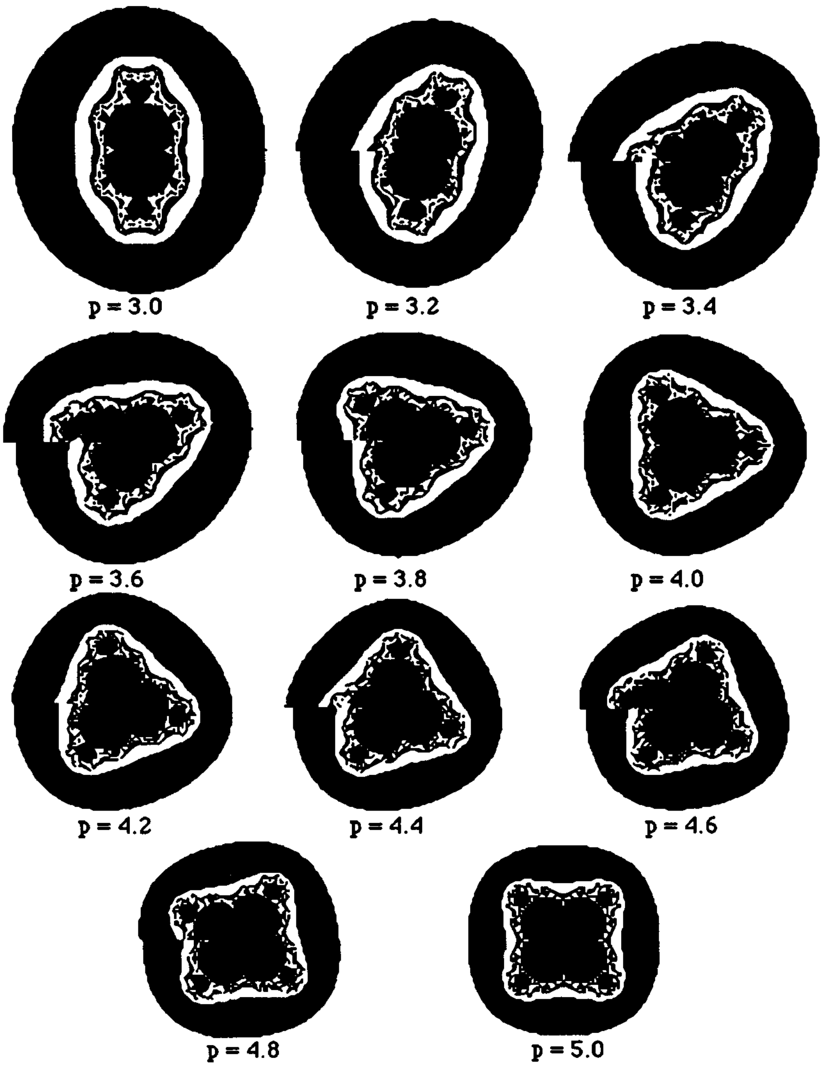


Figure 6.4–5 Generalised Mandelbrot set for powers from 3 to 5.

Computer Graphics and Exercises for §6.4

Exercise 6.4-1

Write a program to compute and draw the basin according to Curry, Garnett, and Sullivan. Either use the given procedures for complex operations, or try to formulate the algorithm one step at a time. This is not particularly easy, but has an advantage in computational speed.

Investigate the regions recommended in connection with Figure 6.4-2. An example for the region at $c = 1$ is shown in Figure 6.4-6.

Does the picture remind you of anything else you have seen in this book?

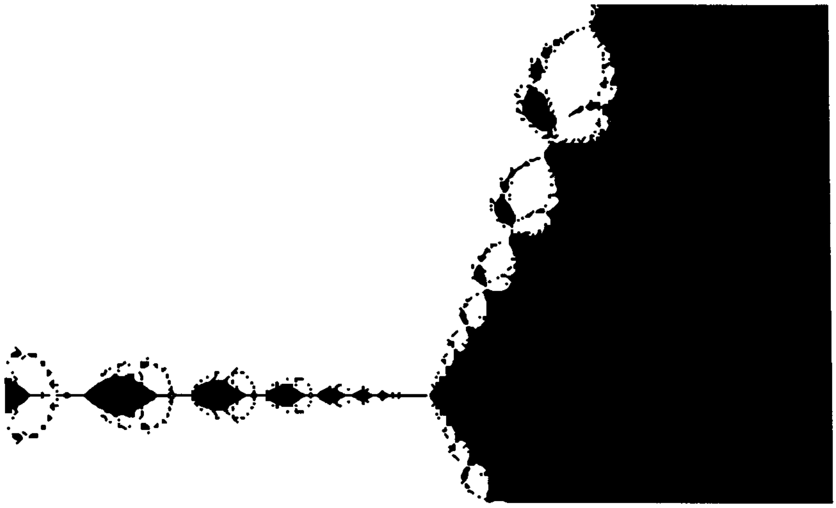


Figure 6.4-6 Section from Figure 6.4-1 near $c = 1$.

Exercise 6.4-2

Modify the program so that in the interesting region of Figure 6.4-2 a distinction is drawn between convergent and cyclic behaviour of the number sequences. Draw the figure corresponding to the Gingerbread Man. It corresponds to the numbers c for which Newton's method does not lead to a solution, but ends in a cyclic sequence.

Compare the resulting figure with the original Mandelbrot set. Spot the difference!

Exercise 6.4-3

Investigate further the elliptical regions on the edge of the 'white set' in Figure 6.4-1. Compare these. What happens to the Gingerbread Man?

Exercise 6.4-4

Develop a program to illustrate the iteration formula

$$z_{n+1} = z_n^p - c$$

graphically. Use it to investigate the symmetries of the basins for $p = 6$, $p = 7$, etc.

Try to formulate the results as a general rule.

Exercise 6.4-5

Naturally it is also possible to draw Julia sets for the iteration equation

$$z_{n+1} = z_n^p - c.$$

To get connected basins, you should probably choose a parameter c belonging to the inner region of the sets found in Exercise 6.4-4 or shown in Figures 6.4-3 and 6.4-5.

The changes to the program are fairly modest. Concentrate on the differences between the Gingerbread Man program in §6.1 and that for Julia sets in §5.2.

Exercise 6.4-6

If you have come to an understanding of the symmetries of the generalised Mandelbrot sets in Exercise 6.4-4, try to find something similar for the generalised Julia sets of the previous exercise. An example is shown in Figure 6.4-7. There the power p =3 and the constant

$$c = -0.5 + 0.44*i.$$

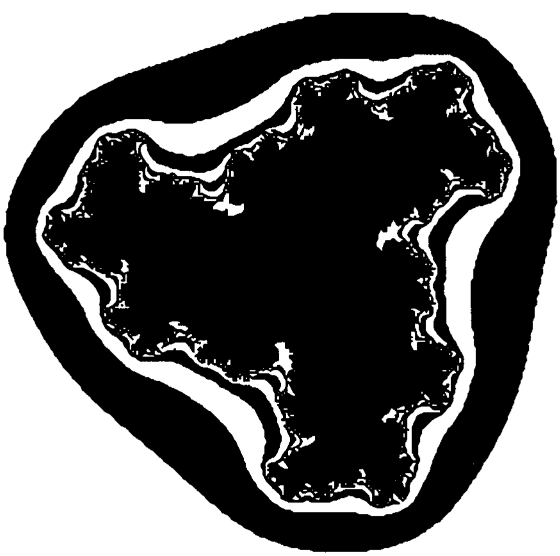


Figure 6.4-7 Generalised Julia set.