Chapter 5:

Fractals II

Goals:

- To understand why the scaling exhibited by period-doubling bifurcation sequences of one-dimensional maps is universal, and to calculate the scaling exponents.
- To examine other natural processes which give rise to fractal structures.

This chapter continues our study of self-similar structures, or fractals. First, we will pursue our investigation of the period-doubling bifurcation sequence of the logistic map and get some understanding of why the scaling exponents α and δ which characterize it are universal. We hope that this will give you some sense of how understanding fractals can give new insight into natural phenomena. Then we will survey a few other systems in nature in which fractals arise. Some of these other systems are not nearly so well understood, so perhaps you will get some idea of the scope of the question of why fractals are seen in many different situations.

A. The universality of the period-doubling bifurcation sequence. In the previous chapter and in Required Project II we investigated the scaling behavior of the period-doubling sequence of the logistic map. We saw that the sequence of r-values for the orbits of period 2^n obeys

$$\lim_{n\to\infty} \frac{r_{n-1} - r_{n-2}}{r_n - r_{n-1}} = \delta ,$$

where δ = 4.6692016...., and that the sequence of values $y_n = f^{2^{n-1}}(x = 1/2)$ satisfies

$$\lim_{n\to\infty} \frac{y_{n-1} - y_{n-2}}{y_n - y_{n-1}} = -\alpha ,$$

with α = 2.502907875..... These results mean that the bifurcation diagram looks the same when it is magnified about the point ($x = \frac{1}{2}$,

* * *

$r=r_\infty=3.569945669....$) by a factor $-\alpha$ in the x-direction and a factor δ in the r-direction. This mapping of the bifurcation diagram onto itself under certain rescaling factors is known as self-similarity (because the diagram is similar to itself). In Required Project II you show that α and δ are the same for some different choices of the map function.

In this chapter we address the question of why α and δ have the values that they do and why they are the same for many different mapping functions. We will write an equation that embodies the observation of self-similarity, and we will find that this equation determines the values of the exponents α and δ. In other words, knowing that self-similarity exists is enough information to determine what the magnification factors must be. This result explains the observation of universality (different maps having the same exponents).

We start by plotting a graph of the time series x_j versus for the logistic map for the parameter value $r = r_\infty$, starting from the point $x_{j=0} = \frac{1}{2}$.

* * *

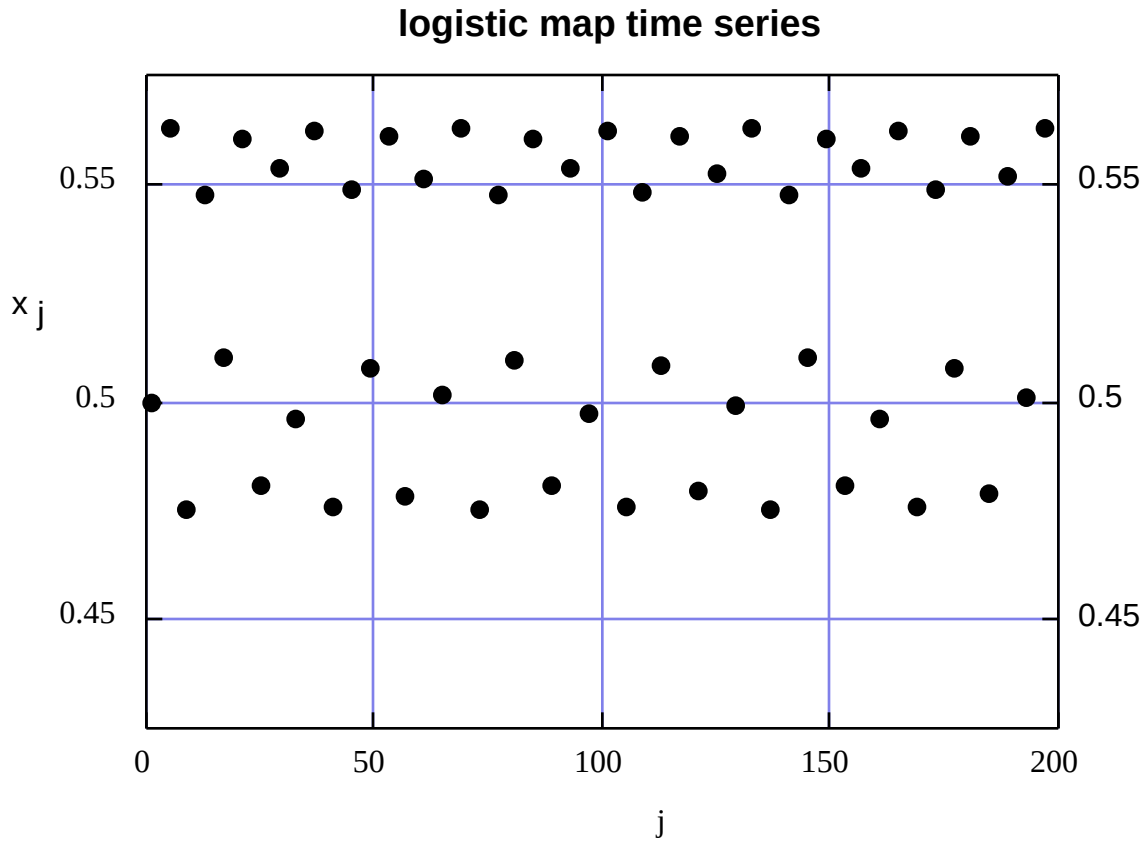

Figure 5.1 Time series of x-values from logistic map at r = r_∞, starting from $x_{j=0} = \frac{1}{2}$.

Figures 5.1 and 5.2 are two different graphs of the same time series. Notice that the axes used for figure 5.2 have different scales than those used in figure 5.1: the j-axis is scaled up by a factor of 2 and the x-axis is scaled down by a factor of roughly 2.5 (centered on x=0.5).

* * *

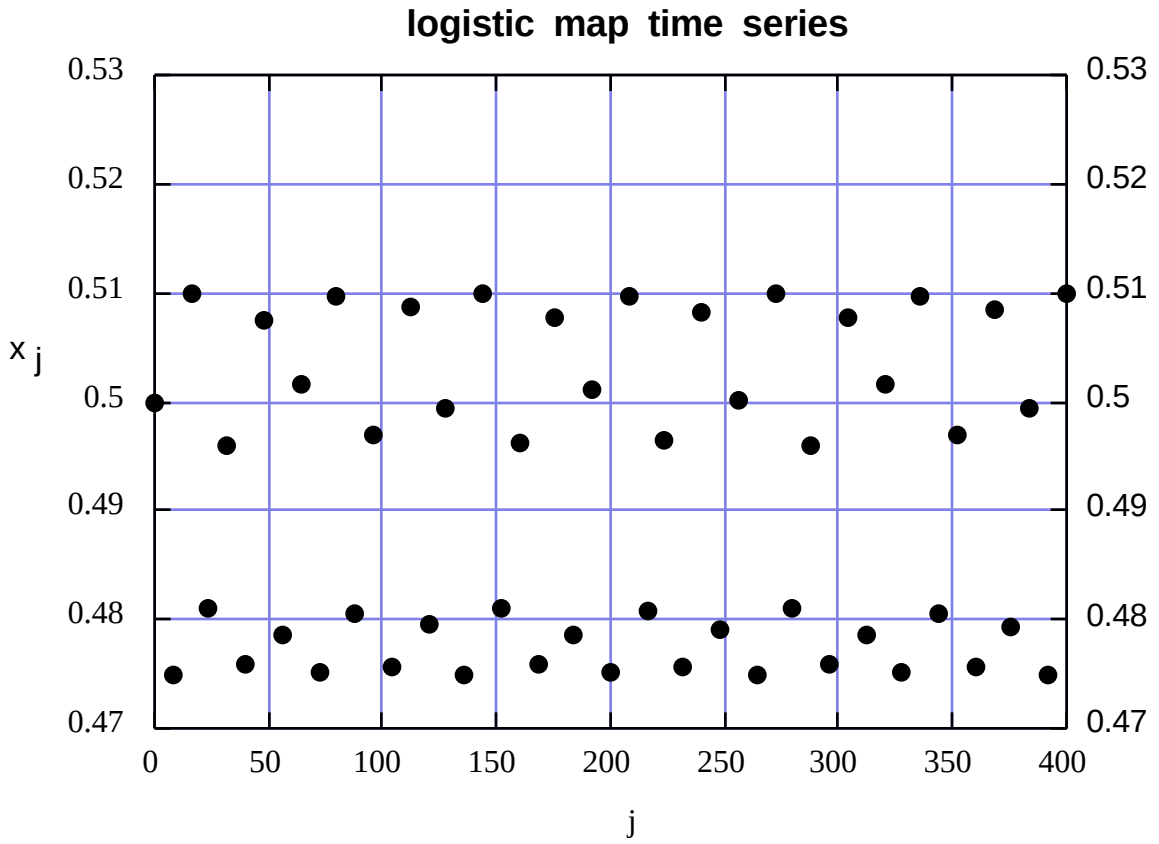

Figure 5.2 The same time series as figure 5.1, plotted with different scales.

Figure 5.2 looks a lot like figure 5.1 turned upside down. In fact, we claim (and, in Required Project II, you will check this claim) that a plot of the time series of the logistic map at $r=r_\infty$ near $x=\frac{1}{2}$ has the following property: If the j-axis of the plot is scaled by a factor of two, and if the vertical axis is inverted about $x=\frac{1}{2}$ and then scaled by the factor α, then all the points in the rescaled plot can be superimposed directly onto those in the original graph. In other words, we claim that if we label the points which appear in figure 5.1 by an index k (i.e., ignoring those iterates which are too far from $x=\frac{1}{2}$ to appear on the graph), so that the plot is of a sequence $\{x_{k=0}, x_{k=1}, x_{k=2}, \ldots\}$,[1] then every x_k in the sequence satisfies

[1]You should be able to show that k=4j in figure 5.1, and k=8j in figure 5.2.

* * *

$$-\alpha\left(x_{2k}-\frac{1}{2}\right)=x_k-\frac{1}{2}. \qquad (5.1)$$

Now we look for a function that generates a time series that has the self-similarity property embodied in equation (5.1). We define $z_k = x_k - \frac{1}{2}$, the difference between the iterate x_k and the maximum at $x=\frac{1}{2}$, and rewrite the self-similarity equation (5.1) as:

$$-\alpha z_{2k} = z_k. \qquad (5.2)$$

Equation (5.2) holds for all k.[3] In particular, it holds if we replace k with k+1:

$$-\alpha z_{2k+2} = z_{k+1} \qquad (5.3)$$

We are looking for the mapping function g that generates this sequence via $z_{k+1}=g(z_k)$. We re-write (5.3) as:

$$z_{k+1}=g(z_k)=-\alpha z_{2k+2}=-\alpha g(z_{2k+1})=-\alpha g(g(z_{2k}))$$

$$g(z_k)=-\alpha g(g(z_{2k})) \qquad (5.4)$$

Therefore, using (5.2), we have:

$$g(z_k)=-\alpha g\left(g\left(-\frac{z_k}{\alpha}\right)\right), \qquad (5.5)$$

and the function g must satisfy:

$$g(z)=-\alpha g\left(g\left(-\frac{z}{\alpha}\right)\right). \qquad (5.6)$$

Equation (5.6) is called a renormalization-group equation. It says that a plot of every other point on the graph can be rescaled so that it looks identical to the original graph of every point. Our derivation of equation (5.6) uses the fact that the time series graph looks the same when it is rescaled, but does not put in any information about the rescaling factor α other than the fact that it exists. Nonetheless, equation (5.6) determines both g and α.

---

[2]Strictly speaking, equation (5.1) holds only in the limit as x approaches 1/2. But in Required Project II you should find that it works well for x in the range shown in figure 5.1.

[3]Recall that k is not the k^{th} iterate in the map, but rather the k^{th} iterate to fall within the limits of our graph.

* * *

We see this by solving equation (5.6). We do not assume that α is known, and solve for it along with g. We do assume that the function g(z) has a quadratic maximum (which, as you recall, is true for "generic" functions) at z=0 (if the maximum is not at zero, we redefine z to be the distance from the maximum).

One way to solve equation (5.6) is to use a Taylor series expansion. We expand g(z) in a Taylor series about z = 0 and write:

$$g(z) = A_0 + A_2 z^2 + A_4 z^4 + \ldots \,, \tag{5.7}$$

where the A_i are as-yet-unknown coefficients.[4] The form (5.7) is then substituted into equation (5.6), yielding

$$A_0 + A_2 z^2 + A_4 z^4 + \ldots = -\alpha \left[A_0 + \frac{A_2}{\alpha^2}\left(A_0 + A_2 z^2 + A_4 z^4 + \ldots\right)^2 + \frac{A_4}{\alpha^4}\left(A_0 + A_2 z^2 + A_4 z^4 + \ldots\right)^4 + \ldots \right]. \tag{5.8}$$

Now we equate the coefficients of each order in z. There are an infinite number of terms in the equation, so in practice one truncates the expansion at some finite order in z. Better and better approximations to g and α are obtained by keeping more and more terms in the Taylor series. Here we keep the two lowest-order terms which are the coefficients of z^0 and z^2. Equating the coefficients of these powers yields

$$\begin{aligned} z^{(0)}: &\qquad A_0 = -\alpha A_0 - \alpha A_0^2 A_2 - \alpha A_0^4 A_4 + \ldots \\ z^{(2)}: &\qquad A_2 = -\frac{2}{\alpha} A_0 A_2^2 \ . \end{aligned} \tag{5.9}$$

Now the first equation of (5.9) is a bit unhappy, since it relates the coefficient A_0 to all the coefficients in the Taylor series. However, we can be optimistic and assume that the A_i decrease rapidly as i increases, so that it is a good first approximation to keep only A_0 and A_2. We ask you improve the approximation and to investigate whether this assumption is indeed true in a Menu Project. Once we make this assumption and ignore all terms except those which have only A_0 and A_2 in them, we find that Eq. (5.9) becomes:

---

[4]There is no linear term in the Taylor series because we are expanding about the maximum of g(z). In fact, in can be shown that there are no odd terms at all in the Taylor series; this is plausible because g(y) can be obtained by composition of a function which is even in z.

* * *

$$1 = -\alpha(1 + A_0 A_2),$$
$$\alpha = -2A_0 A_2 \ . \qquad (5.10)$$

Solving Eq. (5.10), we find $\alpha = 1+\sqrt{3}$, which is not far from the exact value $\alpha = 2.502907875.....$

Menu Project. Calculating g and α to higher accuracy.

In this project, you will determine α and g to higher accuracy than is done here in the class notes. First, please extend the Taylor-series calculation presented above to higher orders in z. How does the value of α and the form of g depend on the number of terms that are kept in the Taylor series? Does the process appear to converge? Then, we would like you to compare your results for g to those obtained when you use Feigenbaum's result that g(z) can be obtained by repeatedly iterating any function f(z) with a quadratic maximum at z=0 and rescaling appropriately. Specifically, Feigenbaum showed that g(x) can be written as the limit:

$$g(z) = \lim_{n\to\infty} (-\alpha)^n f^{2^n}\left(\frac{z}{(-\alpha)^n}\right) \qquad (5.11)$$

where the map f is iterated at the parameter value $r = r_\infty$. (We say a bit about this method in the Appendix. For more detailed expositions, see, for example, the book by Hilborn.) In practice, one plugs in the known value of α (α is determined by equation (5.8) because the limit exists only for that particular value), and then iterates some large but finite number of times to obtain an approximate result for g. How does your result for g depend on the number of iterations you perform on the map function? How does it compare to the results you obtained using the Taylor series method?

So far we have not mentioned δ. It is also determined by equation (5.6), and so therefore also is universal. Unfortunately, calculating δ is much more involved than finding α, so we have left it as a project for the intrepid among you.

* * *

Menu Project. Calculating δ. In this project, you will calculate δ, which is a measure of how many iterations are required to determine that the parameter value r is not exactly equal to r_∞. We recommend that you consult the references to see how to do this (Hilborn, section 5.7 has a heuristic discussion; the rigorous derivations are in M. J. Feigenbaum, J. Stat. Phys. 19, 25 (1978); 21, 669 (1979)); here we say just a few words to give an overview of the procedure.

To calculate δ you need to how the map function itself evolves under iteration. In particular, you need to figure out which deviations of the map function f(z) from the universal function g(z) are the ones that cause it to diverge from the universal function as it is iterated. Then, you will consider functions which are close to the fixed function and express the evolution in terms of equations which are linearized about the fixed function g(z). You will find that the deviations can be described in terms of an eigenfunction and an eigenvalue, the latter of which is δ. Since this eigenfunction grows by the factor δ each time the map is iterated, whereas all the other deviations shrink under iteration, it determines the deviations from criticality and hence the convergence of the sequence of r-values.

To summarize this section, we have seen that the observation of scaling behavior in the graph of the iterates of the logistic map turns out to be sufficient knowledge to predict the values of the magnification factors. We have shown how one can calculate the universal quantities α and g. We have shown that the scaling behavior of period-doubling bifurcation sequence is the same for any map function with a quadratic maximum.

B. Fractals elsewhere in nature.

Fractals are observed many places in nature, in systems as diverse as bacterial colonies, mountain ranges, and clouds. One of the great open questions in physics is why fractals are observed in so many different systems. Here we discuss a few other situations in nature in which fractals appear.

* * *

B.1. The Random Walk.

A random walk is very simple: start at a point (in a 2 dimensional plane, say) and take a step of size L in a randomly chosen direction. Randomly chose a new direction and take another step of size L. Continue this process, and you get a path that is called a random walk. Random walks (in 1,2, and 3 dimensions) are a very useful model for many physical processes, such as Brownian motion or diffusion of a particle.

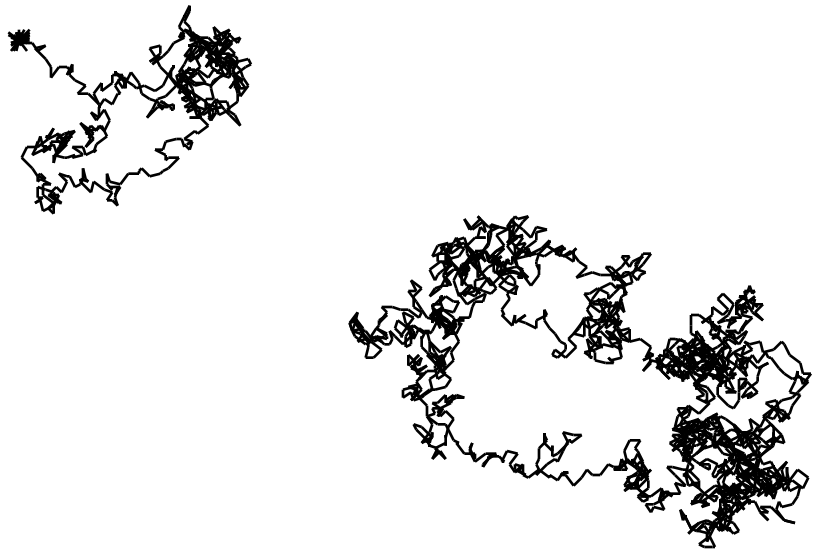

Figure 5.3 Two random walks.

Problem 5.1. The random walk. Write a program which implements the two-dimensional random walk procedure described above and which calculates how the average (mean-square) distance from the starting point varies with i) the step size L, and ii) the number of steps N. Since the process involves random numbers (which you can generate with the Math.random method), you should do a fairly large number of random walks for each L and N, and average the results. First, fix L and obtain average distances for, say, ten values of N (none smaller than about 1000). You should take data "on the fly" in the following manner. Let your particle walk 1000 steps, calculate and store the distance from the origin, let it walk another 1000 steps, calculate and store the distance from the origin, etc. The average distance, d_{av}, should vary as $d_{av} \quad N^{\gamma}$ for some exponent γ.

* * *

Now fix N, and try varying L. Again, d_{av} should vary as a power of L, $d_{av} \quad L^{\beta}$. Can you guess, a priori, what γ and β should be? Does this computer experiment verify your intuition? A random walk does not look exactly the same when it is magnified (since, after all, a different random number gets picked at every step), but there is a statistical self-similarity. How do the values of γ and β reflect this self-similarity?

Menu Project. Diffusion Limited Aggregation (DLA)
This project explores the use of random walks to model an interesting growth process of dendritic structures made up of particles that stick together when they come in contact. The particles are assumed to diffuse slowly through space until they come in contact with another particle, at which time they stick. The computer implementation is simple in concept: On a lattice (i.e. an array of points) put a particle at the origin (the "seed"). Then start another particle from far away, and allow it to execute a random walk (with step size equal to the lattice spacing). After a time, it will land on one of the points adjacent to the particle at the origin. Stop the particle there, and begin another one from far away etc. The structure that develops has a beautiful, tree-like, fractal shape. Unfortunately, it grows very slowly. This is a project that you will want to run as long as possible. Compute the fractal dimension of your clusters, defined by the relation $N \quad R^{d}$, where N is the number of particles in the cluster, R is the "radius" of the cluster (typically the radius of gyration), and d is the fractal dimension. Grow several large aggregates to get an idea of the accuracy of your estimate for d.

B.2. River networks.

Figure 5.4 (at http://www.northnet.com.au/barnesr/fractdtm.html) is an image of the topography in gorge country northeast of Armindale and 70 km east of Glen Innes in Australia, which includes the drainage basins of the Mann, Boyd, Nymboida, and Clarence rivers. The image is a false-color map of the digital elevation data of the area.

* * *

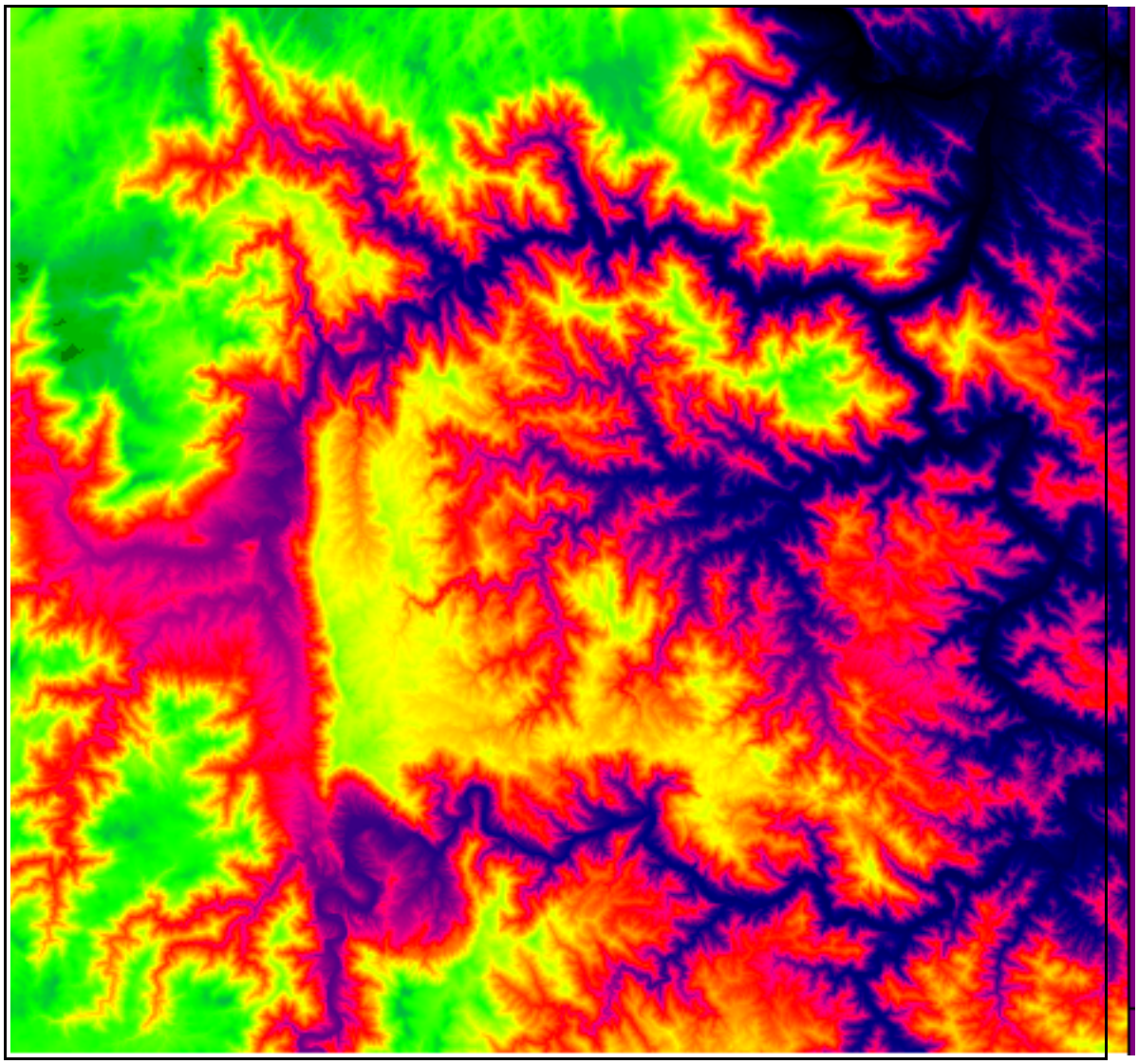

Figure 5.4 Topography of river drainage basin.

Analysis of pictures like figure 5.4 shows that these systems are statistically self-similar (e.g., fractal).

Why river basins are fractal is a matter of some debate. Several plausible models have been proposed that lead to self-similar drainage networks (some of which are in the book by Ignacio Rodriguez-Iturbe and Andrea Rinaldo, Fractal River Basins: Chance and Self-Organization, Cambridge University Press (1997)). It is an open question whether there is a fundamental unifying mechanism that causes fractals to emerge from many different models. One interesting view on this subject is presented by Per Bak in his book How Nature Works: The Science of Self-Organized Criticality, Springer-Verlag, 1996.

* * *

Menu Project. A river network model. Chapter 19 of Gould and Tobochnik (2nd edition) discusses one model of river networks (from R.L. Leheny, Phys. Rev. E 52, 5610 (1995)), in which a rectangular lattice of points describes an eroding terrain with the height of the land, h(x,y), specified at each point. The simulation begins with the landscape as a featureless incline: h(x,y) = ly. Then the following rules are implemented:

1) Precipitation lands at a random site on the lattice.

2) Water flows from this site to one of the four nearest neighbors with a probability proportional to $e^{E\Delta h}$, where Δh is the height difference between the site and the neighbor, and E is a parameter of the model. If $\Delta h<0$, this probability is set equal to zero.

3) Step 2) is repeated until the water reaches the bottom of the lattice, y=0.

4) Each point that has been visited by the flowing water has its height reduced by a constant amount D. This process represents erosion.

5) Any site at which the height difference Δh with a neighbor exceeds a threshold M is reduced in height by an amount $\Delta h/S$, where S is another parameter in the model.

Write a program that implements this model. You can get an idea of suitable parameters to use from Leheny's paper. The resulting river network is defined as follows: every lattice point receives one unit of precipitation which traces a path of steepest descent, without eroding the terrain, until it reaches the lattice edge, y=0; the river network is defined as all points through which at least R units flow. Analyze the network that is generated at different times. Does the river network appear to be fractal? How does evolving the model for longer times affect the network's properties?

Appendix: Obtaining g(z) via functional iteration.

In this chapter we solved equation (5.6) for g(z) by expanding g(z) in a Taylor series about z = 0 and equating coefficients of the different powers of z. However, the time series plotted in figures 5.1 and 5.2

was generated by iterating repeatedly the logistic map (and time series with identical scaling properties can be generated by iterating any function with a quadratic maximum). Therefore, it should be (and indeed is) possible to obtain g(z) directly from the original mapping function f. We say a bit about this other method of obtaining g(z) here.

The idea behind this method is to keep track of what happens when the map is applied repeatedly. (Applying the function f over and over is called functional iteration.) We will implement on the map function f the procedure that we used above when we rescaled the axes of the time series graph in figures 5.1 and 5.2. We again define $z_j = x_j - \frac{1}{2}$ and consider the time series generated by iterating the map f with r = r_∞, starting from $z_0 = 0$.[5]

Now we write a formula for the function that yields the points that form the time series on our rescaled graph in terms of the original map function f. Recall that each rescaling of the time series graph is by a factor of 2 on the j-axis and a factor of -α on the x-axis. To keep track of this rescaling, we define new indices n and m via $j = m2^n$, where n keeps track of the number of magnifications. The fact that after rescaling the time series graphs look the same means that there is a well-defined $n \to \infty$ limit to the sequence:

$$(-\alpha)^n z_{2^n m} = (-\alpha)^n f^{2^n m}(z=0) = (-\alpha)^n f^{2^n}\left(f^{2^n(m-1)}(z=0)\right). \qquad (5.12)$$

Now we define the variable $u_m(n)$ via:

$$u_m(n) \equiv (-\alpha)^n f^{2^n(m-1)}(z=0), \qquad (5.13)$$

so that

$$(-\alpha)^n z_{2^n m} = (-\alpha)^n f^{2^n}\left(\frac{u_m(n)}{(-\alpha)^n}\right). \qquad (5.14)$$

Recall that, for large n, $f^{2^n(m-1)}(z=0) \;\; -\frac{1}{\alpha} f^{2^{n-1}(m-1)}(z=0)$ (This follows from the rescaling pictured in figures 5.1 and 5.2, and embodied in equation (5.2).) This means that in the limit $n \to \infty$, $u_m(n)$ becomes independent of n: $u_m(n) \to u_m$. The left hand side of equation (5.14)

---

[5]The map function f for z is related in a simple fashion to the map function for x, since z is just x shifted by a constant.

* * *

also has a well-defined $n \to \infty$ limit, so we can define the function g as:

$$g(u_m) \equiv \lim_{n\to\infty} (-\alpha)^n f^{2^n}\left(\frac{u_m}{(-\alpha)^n}\right). \tag{5.15}$$

Thus the function $g(z)$ can be written as:

$$g(z) = \lim_{n\to\infty} (-\alpha)^n f^{2^n}\left(\frac{z}{(-\alpha)^n}\right). \tag{5.16}$$

* * *