20
Fractals

It is common in nature to notice objects, called fractals, *that do not have well-defined geometric shapes, but nevertheless appear regular and pleasing to the eye. These objects have dimensions that are fractions, and occur in plants, sea shells, polymers, thin films, colloids, and aerosols. We will not study the scientific theories that lead to fractal geometry, but rather will look at how some simple models and rules produce fractals. To the extent that these models generate structures that look like those in nature, it is reasonable to assume that the natural processes must be following similar rules arising from the basic physics or biology that creates the objects. As a case in point, if we look at the bifurcation plot of the logistics map, Fig. 18.2, we see a self-similarity of structure that is characteristic of fractals; in this case we know the structure arises from the equation $x_{n+1} = \mu x_n(1 - x_n)$ used for the map. Detailed applications of fractals can be found in many literature sources [34,41–43].*

20.1
Fractional Dimension (Math)

Benoit Mandelbrot, who first studied the fractional-dimension figures with the supercomputers at IBM Research, gave them the name *fractal* [44]. Some geometric objects, such as the Koch curves, are exact fractals with the same dimension for all their parts. Other objects, such as bifurcation curves, are statistical fractals in which elements of randomness occur and the dimension can be defined only locally or on the average.

Consider an abstract "object" such as the density of charge within an atom. There are an infinite number of ways to measure the "size" of this object, for example; each moment of the distribution provides a measure of the size, and there are an infinite number of moments. Likewise, when we deal with complicated objects that have fractional dimensions, there are different definitions of dimension, and each may give a somewhat different dimension. In addition, the fractal dimension is often defined by using a measuring box whose size approaches zero. In realistic applications there may be numerical or conceptual difficulties in approaching such a limit, and for this reason a precise value of the fractional dimension may not be possible.

Computationyal Physics. Problem Solving with Computers (2nd edn).
Rubin H. Landau, Manuel José Páez, Cristian C. Bordeianu

ISBN: 978-3-527-40626-5

Our first definition of fractional dimension d_f (or *Hausdorf–Besicovitch* dimension) is based on our knowledge that a line has dimension 1; a triangle, dimension 2, and a cube, dimension 3. It seems perfectly reasonable to ask if there is some mathematical formula, which agrees with our experience for regular objects, yet can also be used for determining fractional dimensions.

For simplicity, let us consider objects that have the same length L on each side, as do equilateral triangles and squares, and which have uniform density. We postulate that the dimension of an object is determined by the dependence of its total mass upon its length:

$$M(L) \propto L^{d_f} \tag{20.1}$$

where the power d_f is the *fractal dimension*. As you may verify, this rule works with the 1D, 2D, and 3D regular figures of our experience, so it is a reasonable hypothesis. When we apply (20.1) to fractal objects, we end up with fractional values for d_f. Actually, we will find it easier to determine the fractal dimension not from an object's mass, which is *extensive* (depends on size), but rather from its density, which is *intensive*. The density is defined as mass/length for a linear object, as mass/area for a planar object, and mass/volume for a solid object. That being the case, for a planar object we hypothesize that

$$\rho = \frac{M(L)}{\text{Area}} \propto \frac{L^{d_f}}{L^2} \propto L^{d_f-2} \tag{20.2}$$

20.2
The Sierpiński Gasket (Problem 1)

To generate our first fractal (Fig. 20.1), we play a game in which we pick points and place dots on them. Here are the rules (which you should try out now):

1. Draw an equilateral triangle with vertices and coordinates:

   $$\text{vertex 1}: (a_1, b_1) \qquad \text{vertex 2}: (a_2, b_2) \qquad \text{vertex 3}: (a_3, b_3)$$

2. Place a dot at an arbitrary point $P = (x_0, y_0)$ within this triangle.
3. Find the next point by selecting randomly the integer 1, 2, or 3:
   (a) If 1, place a dot halfway between P and vertex 1.
   (b) If 2, place a dot halfway between P and vertex 2.
   (c) If 3, place a dot halfway between P and vertex 3.
4. Repeat the process, using the last dot as the new P.

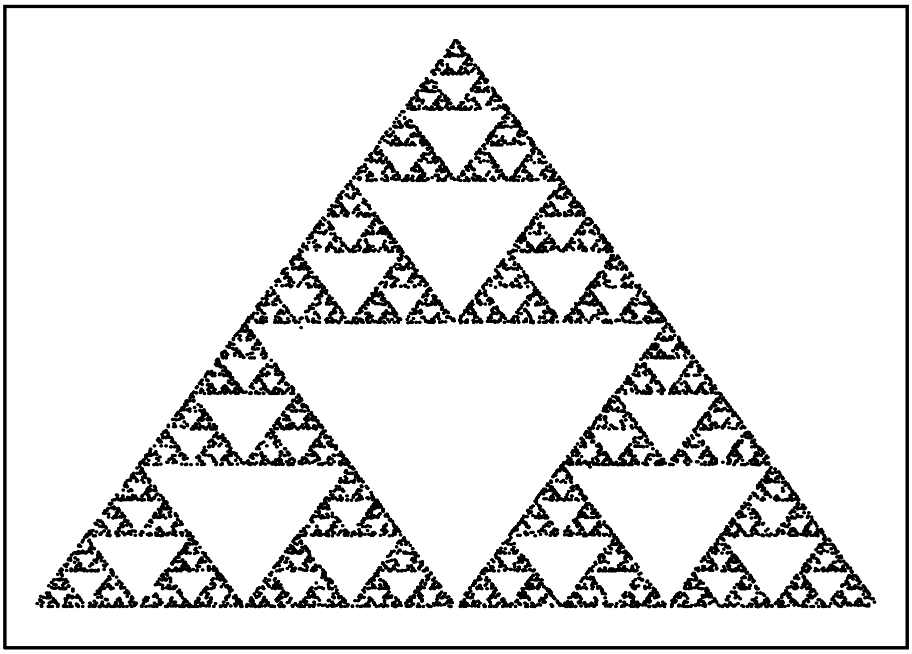

Fig. 20.1 A Sierpiński gasket containing 15,000 points constructed as a statistical fractal. Each filled part of this figure is self-similar.

Mathematically, the coordinates of successive points are given by the formula

$$(x_{k+1}, y_{k+1}) = \frac{(x_k, y_k) + (a_n, b_n)}{2} \qquad n = \text{integer}\,(1 + 3r_i) \tag{20.3}$$

where r_i is a random number between 0 and 1, and where the *Integer* function outputs the closest integer smaller than, or equal to, the argument. After 15,000 points, you should obtain a collection of dots like Fig. 20.1.

20.2.1
Sierpinsky Implementation

Write a program to produce a Sierpiński gasket. Determine empirically the fractal dimension of your figure. Assume that each dot has mass 1 and that $\rho = CL^{\alpha}$. (You can have the computer do the counting by defining an array *box* of all 0 values and then change a 0 to a 1 when a dot is placed there.)

20.2.2
Assessing Fractal Dimension

The topology of Fig. 20.1 was first analyzed by the Polish mathematician Sierpiński. Observe that there is the same structure in a small region as there is in

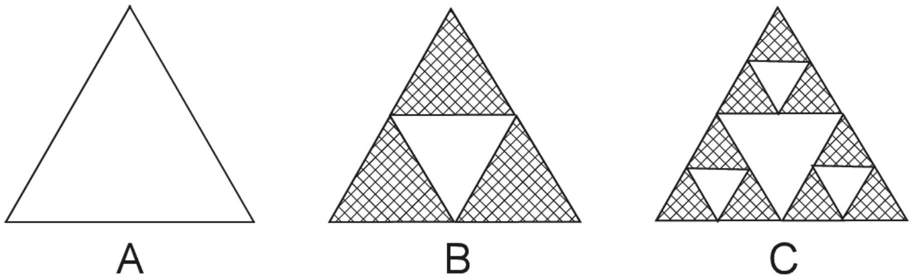

Fig. 20.2 A Sierpinsky gasket constructed by successively connecting the midpoints of the sides of each equilateral triangle. A, B, and C show the first three steps in the process.

the entire figure. In other words, if the figure had infinite resolution, any part of the figure could be scaled up in size and will be similar to the whole. This property is called *self-similarity*.

We construct a regular form of the Sierpinsky gasket by removing an inverted equilateral triangle from the center of all filled equilateral triangles to create the next figure (Fig. 20.2). We then repeat the process *ad infinitum*, scaling up the triangles so each one has side $r = 1$ after each step. To see what is unusual about this type of object, we look at how its density (mass/area) changes with size, and then apply (20.2) to determine its fractal dimension. Assume that each triangle has mass m, and assign unit density to the single triangle:

$$\rho(L = r) \propto \frac{M}{r^2} = \frac{m}{r^2} \stackrel{\text{def}}{=} \rho_0 \qquad \text{(Fig. 20.2A)}.$$

Next, for the equilateral triangle with side $L = 2$, the density

$$\rho(L = 2r) \propto \frac{(M = 3m)}{(2r)^2} = 34mr^2 = \frac{3}{4}\rho_0 \qquad \text{(Fig. 20.2B)}.$$

We see that the extra white space in Fig. 20.2B leads to a density that is $\frac{3}{4}$ that of the previous stage. For the structure in Fig. 20.2C, we obtain

$$\rho(L = 4r) \propto \frac{(M = 9m)}{(4r)^2} = (34)^2\,\frac{m}{r^2} = \left(\frac{3}{4}\right)^2 \rho_0 \qquad \text{(Fig. 20.2C)}.$$

We see that as we continue the construction process, the density of each new structure is $\frac{3}{4}$ that of the previous one. This is unusual. Yet in (20.2) we have derived that

$$\rho \propto C L^{d_f - 2} \tag{20.4}$$

Equation (20.2) implies that a plot of the logarithm of the density ρ *versus* the logarithm of the length L for successive structures, yields a straight line of slope:

$$d_f - 2 = \frac{\Delta \log \rho}{\Delta \log L} \tag{20.5}$$

As applied to our problem:

$$d_f = 2 + \frac{\Delta \log \rho(L)}{\Delta \log L} = 2 + \frac{\log 1 - \log(3/4)}{\log 1 - \log 2} \simeq 1.58496 \tag{20.6}$$

As is evident in Fig. 20.2, as the gasket gets larger and larger (and consequently more massive), it contains more and more open space. So even though its mass approaches infinity as $L \to \infty$, its density approaches zero! And since a two-dimensional figure like a solid triangle has a constant density as its length increases, a 2D figure would have a slope equal to 0. Since the Sierpiński gasket has a slope $d_f - 2 \simeq -0.41504$, it fills space to a lesser extend than a 2D object, but more than does a 1D object; it is a fractal.

20.3 Beautiful Plants (Problem 2)

It seems paradoxical that natural processes subject to chance can produce objects of high regularity and symmetry. For example, it is hard to believe that something as beautiful and symmetric as a fern (Fig. 20.3) has random elements in it. Nonetheless, there is a clue here in that much of the fern's beauty arises from the similarity of each part to the whole (self-similarity), with different ferns similar, but not identical, to each other. These are characteristics of fractals. Your **problem** is to discover if a simple algorithm including some randomness can draw regular ferns. If the algorithm produces objects that resemble ferns, then presumably you have uncovered mathematics similar to that responsible for the shape of ferns.

20.3.1 Self-Affine Connection (Theory)

In (20.3), which defines mathematically how a Sierpiński gasket is constructed, a *scaling factor* of $\frac{1}{2}$ is part of the relation of one point to the next. A more general transformation of a point $P = (x, y)$ into another point $P' = (x', y')$ via *scaling* is

$$(x', y') = s(x, y) = (sx, sy) \qquad \text{(scaling)} \tag{20.7}$$

If the scale factor $s > 0$, an amplification occurs, whereas if $s < 0$, a reduction occurs. In our definition (20.3) of the Sierpiński gasket, we also added in a

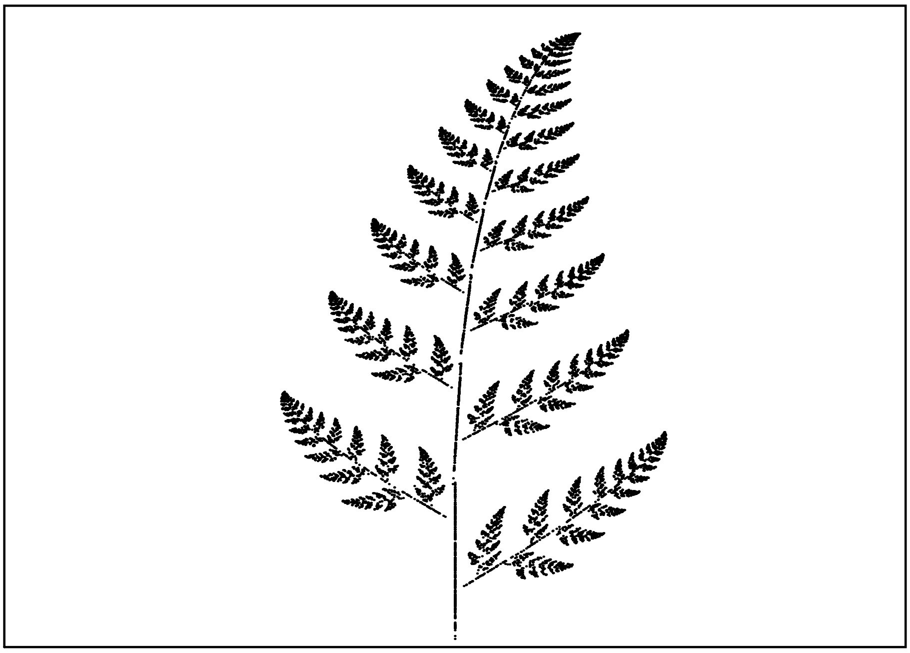

Fig. 20.3 A fern after 30,000 iterations of the algorithm (20.10). If you enlarge this, you will see that each frond has similar structure.

constant a_n. This is a *translation operation*, which has the general form

$$(x', y') = (x, y) + (a_x, a_y) \qquad \text{(translation)} \tag{20.8}$$

Another operation, not used in the Sierpiński gasket, is a *rotation* by angle θ:

$$x' = x\cos\theta - y\sin\theta \qquad y' = x\,\sin\theta + y\cos\theta \qquad \text{(rotation)} \tag{20.9}$$

The entire set of transformations, scalings, rotations, and translations, define an *affine transformation* ("affine" denotes a close relation between successive points). The transformation is still considered affine even if it is a more general linear transformation with the coefficients not all related to one θ (in that case, we can have contractions and reflections). What is important is that the object created with these rules turn out to be self-similar; each step leads to new parts of the object that bear the same relation to the ancestor parts as did the ancestors to theirs. This is what makes the object look similar at all scales.

20.3.2
Barnsley's Fern Implementation (fern.c)

We obtain a Barnsley's Fern by extending the dots game to one in which new points are selected using an affine connection with some elements of chance mixed in

$$(x, y)_{n+1} = \begin{cases} (0.5, 0.27y_n) & \text{with 2\% probability} \\ (-0.139x_n + 0.263y_n + 0.57 \\ \quad 0.246x_n + 0.224y_n - 0.036) & \text{with 15\% probability} \\ (0.17x_n - 0.215y_n + 0.408 \\ \quad 0.222x_n + 0.176y_n + 0.0893) & \text{with 13\% probability} \\ (0.781x_n + 0.034y_n + 0.1075 \\ \quad -0.032x_n + 0.739y_n + 0.27) & \text{with 70\% probability} \end{cases} \tag{20.10}$$

To select a transformation with probability $\mathcal{P}$, we select a uniform random number r in the interval $[0, 1]$ and perform the transformation if r is in a range proportional to $\mathcal{P}$:

$$\mathcal{P} = \begin{cases} 2\% & r < 0.02 \\ 15\% & 0.02 \leq r \leq 0.17 \\ 13\% & 0.17 < r \leq 0.3 \\ 70\% & 0.3 < r < 1 \end{cases} \tag{20.11}$$

The rules (20.10) and (20.11) can be combined into one:

$$(x, y)_{n+1} = \begin{cases} (0.5, 0.27y_n) & r < 0.02 \\ (-0.139x_n + 0.263y_n + 0.57 \\ \quad 0.246x_n + 0.224y_n - 0.036) & 0.02 \leq r \leq 0.17 \\ (0.17x_n - 0.215y_n + 0.408 \\ \quad 0.222x_n + 0.176y_n + 0.0893) & 0.17 < r \leq 0.3 \\ (0.781x_n + 0.034y_n + 0.1075 \\ \quad -0.032x_n + 0.739y_n + 0.27) & 0.3 < r < 1 \end{cases} \tag{20.12}$$

Although (20.10) makes the basic idea clearer, (20.12) is easier to program.

The starting point in Barnsley's fern (Fig. 20.3) is $(x_1, y_1) = (0.5, 0.0)$, and the points are generated by repeated iterations. An important property of this fern is that it is not completely self-similar, as you can see by noting how different are the stems and the fronds. Nevertheless, the stem can be viewed as a compressed copy of a frond, and the fractal obtained with (20.10) is still *self-affine*, yet with a dimension that varies in different parts of the figure.

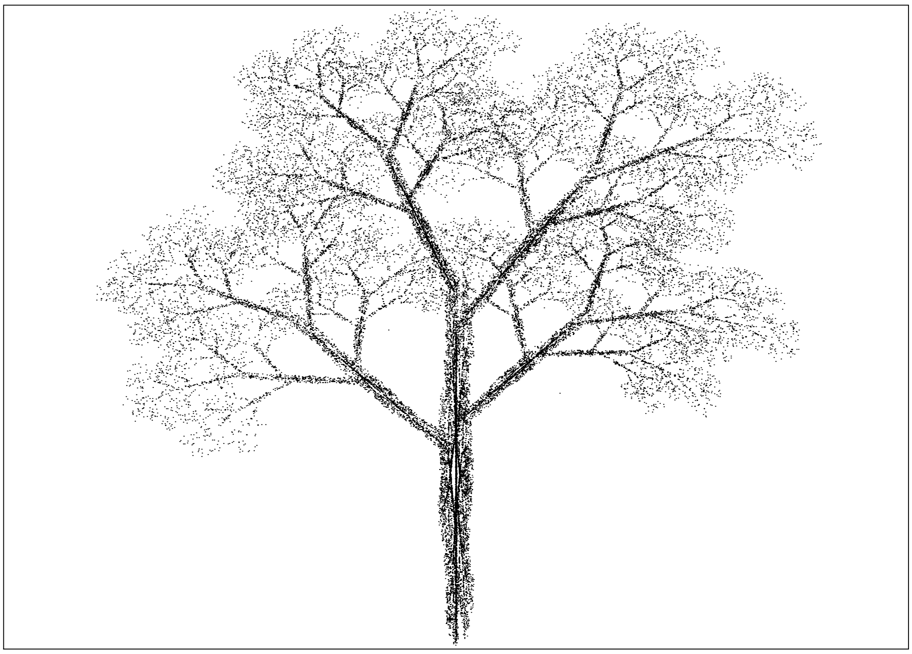

Fig. 20.4 A fractal tree created with the simple algorithm (20.13).

20.3.3
Self-Affinity in Trees Implementation (tree.c)

Now that you know how to grow ferns, look around and notice the regularity in trees (such as Fig. 20.4). Can it be that this also arises from a self-affine structure? Write a program, similar to the one for the fern, starting at $(x_1, y_1) = (0.5, 0.0)$ and iterating the following self-affine transformation:

$$(x_{n+1}, y_{n+1}) = \begin{cases} (0.05x_n, 0.6y_n) & 10\% \text{ probability} \\ (0.05x_n, -0.5y_n + 1.0) & 10\% \text{ probability} \\ (0.46x_n - 0.15y_n, 0.39x_n + 0.38y_n + 0.6) & 20\% \text{ probability} \\ (0.47x_n - 0.15y_n, 0.17x_n + 0.42y_n + 1.1) & 20\% \text{ probability} \\ (0.43x_n + 0.28y_n, -0.25x_n + 0.45y_n + 1.0) & 20\% \text{ probability} \\ (0.42x_n + 0.26y_n, -0.35x_n + 0.31y_n + 0.7) & 20\% \text{ probability} \end{cases} \tag{20.13}$$

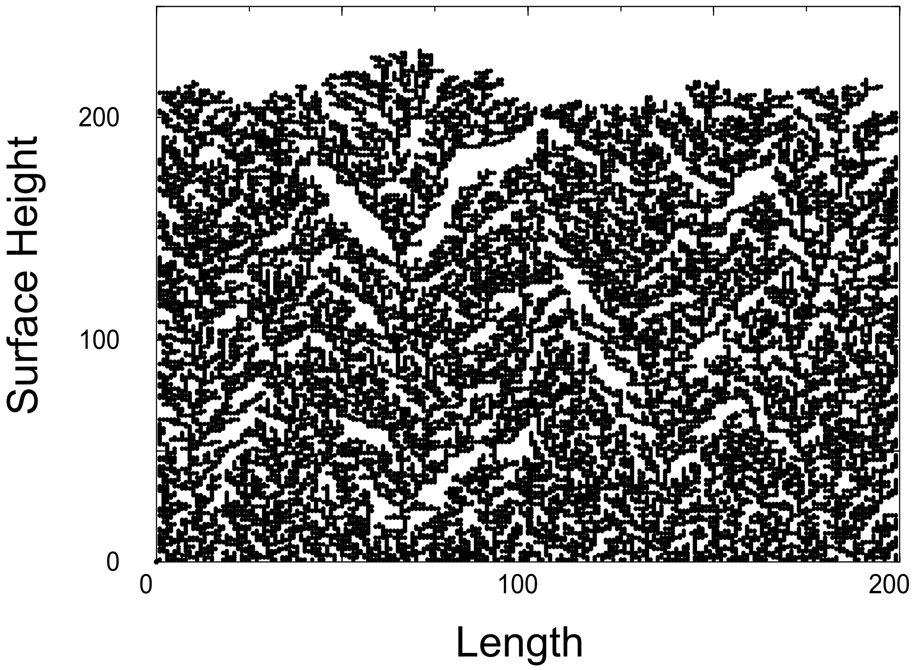

Fig. 20.5 A simulation of the ballistic deposition of 20,000 particles on a substrate of length 200. The vertical height increases with the length of deposition time so that the top is the final surface.

20.4
Ballistic Deposition (Problem 3)

There are a number of physical and manufacturing processes in which particles are deposited on a surface and form a film. Because the particles are evaporated from a hot filament, there is randomness in the emission process, yet the produced films turn out to have well-defined, regular structures. Again we suspect fractals. Your **problem** is to develop a model that simulates this growth process, and compare your produced structures to those observed.

20.4.1
Random Deposition Algorithm (film.c)

The idea of simulating random depositions began with [45], who used tables of random numbers to simulate the sedimentation of moist spheres in hydrocarbons. We shall examine a method of simulation [46] which results in the deposition shown in Fig. 20.5.

Consider particles falling onto and sticking to a horizontal line of length L composed of 200 deposition sites. All particles start from the same height, but to simulate their different velocities, we assume they start at random distances

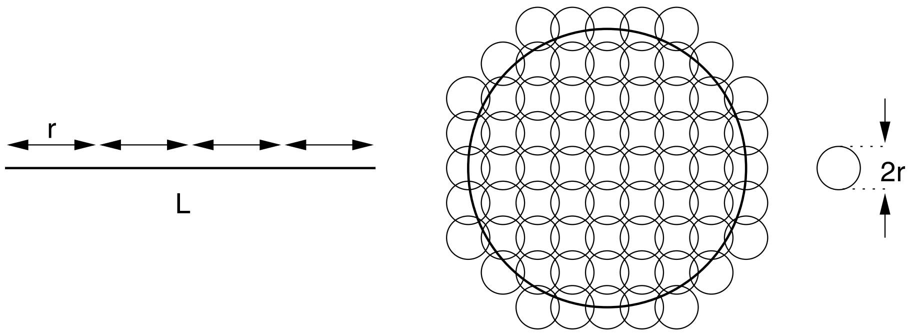

Fig. 20.6 Examples of the use of box counting to determine fractal dimension. On the left the perimeter is being covered, while on the right the entire figure is being covered.

from the left side of the line. The simulation consists of generating uniform random sites between 0 and L, and having the particle stick to the site on which it lands. Because a realistic situation may have columns of aggregates of different heights, the particle may be stopped before it makes it to the line, or may bounce around until it falls into a hole. We therefore assume that if the column height at which the particle lands is greater than that of both its neighbors, it will add to that height. If the particle lands in a hole, or if there is an adjacent hole, it will fill up the hole. We speed up the simulation by setting the height of the hole equal to the maximum of its neighbors:

1. Choose a random site r.
2. Let the array h_r be the height of the column at site r.
3. Make the decision:

$$h_r = \begin{cases} h_r + 1 & \text{if } h_r \geq h_{r-1} \;\; h_r > h_{r+1} \\ \max[h_{r-1}, h_{r+1}] & \text{if } h_r < h_{r-1} \;\; h_r < h_{r+1} \end{cases} \tag{20.14}$$

The results of this type of simulation show several empty regions scattered throughout the line (Fig. 20.5). This is an indication of the statistical nature of the process while the film is growing. Simulations by Fereydoon reproduced the experimental observation that the average height increases linearly with time, and produced fractal surfaces. (You will be asked to determine the fractal dimension of a similar surface as an exercise.)

20.5
Length of British Coastline (Problem 4)

In 1967 Mandelbrot [47] asked a classic question "How long is the Coast of Britain?" If Britain had the shape of Colorado or Wyoming, both of which have straight line boundaries, its perimeter would be a curve of dimension one with a finite length. However, coast lines are geographic not geometric curves, with each portion of the coast often statistically self-similar to the entire coast, yet at a reduced scale. In these latter cases, the perimeter is a fractal, and the length is either infinite or meaningless. Mandelbrot deduced the dimension of the west coast of Great Britain to be $d_f = 1.25$. In your **problem** we ask you to determine the dimension of the perimeter one of our fractal simulations.

20.5.1
Coastline as Fractal (Model)

The length of the coastline of an island is the perimeter of that island. While the concept of perimeter is clear for regular geometric figures, some thought is required to give the concept meaning for an object that may be infinitely self-similar. Let us assume that a map maker has a ruler of length r. If he walks along the coastline and counts the number of times N that he must place the ruler down in order to *cover* the coastline, he will obtain a value for the length L of the coast as Nr. Imagine now that the map maker keeps repeating his walk with sma ller and smaller rulers. If the coast were a geometric figure, or a *rectifiable curve*, at some point the length L would become essentially independent of r and it would approach a constant. Nonetheless, as discovered empirically by Richardson [48] for natural coast lines, such as that of South Africa and Britain, the perimeter appears to be the function of r

$$L(r) = Mr^{1-d_f} \tag{20.15}$$

where M are d_f are empirical constants. For a geometric figure, or for Colorado, $d_f = 1$, and the length approaches a constant as $r \to 0$. Yet for a fractal with $d_f > 1$, the perimeter $L \to \infty$ as $r \to 0$. This means that as a consequence of self-similarity fractals may be of finite size, but have infinite perimeter. However, at some point there may be no more details to discern as $r \to 0$ (say at the quantum), and so the limit may not be meaningful.

Our sample simulation is `Fractals/Film.java` on the CD (`FilmDim.java` on the Instructor's CD). It contains the essential loop:

```
int spot = random.nextInt(200);
if (spot == 0)  {
  if ( coast[spot] < coast[spot+1] ) coast[spot] = coast[spot+1];
    else coast[spot]++;
}
else if (spot == coast.length - 1) {
  if (coast[spot] < coast[spot-1]) coast[spot] = coast[spot-1];
    else coast[spot]++;
}
else if ( coast[spot]<coast[spot-1] && coast[spot]<coast[spot+1] ) {
  if ( coast[spot-1] > coast[spot+1] ) coast[spot] = coast[spot-1];
  else coast[spot] = coast[spot+1];
}
else coast[spot]++;
```

The results of this type of simulation show several empty regions scattered throughout the line (Fig. 20.5). This is an indication of the statistical nature of the process while the film is growing. Simulations by Fereydoon reproduced the experimental observation that the average height increases linearly with time, and produced fractal surfaces. (You will be asked to determine the fractal dimension of a similar surface as an exercise.)

20.5.2 Box Counting Algorithm

Consider a line of length L broken up into segments of length r (Fig. 20.6 left). The number of segments or "boxes" needed to cover the line is related to the size r of the box by

$$N(r) \quad = \quad \frac{L}{r} \quad \propto \quad \frac{1}{r} \tag{20.16}$$

A possible definition of fractional dimension is the power of r in this expression as $r \to 0$. In our example, it tells us that the line has dimension $d_f = 1$. If we now ask how many little circles of radius r it takes to *cover* or fill a circle of area A (Fig. 20.6, right), we would find

$$N(r) \quad = \quad \lim_{r\to 0} \frac{A}{\pi r^2} \qquad \Rightarrow \qquad d_f = 2 \tag{20.17}$$

as expected. Likewise, counting the number of little spheres or cubes that can be packed within a large sphere tells us that a sphere has dimension $d_f = 3$. In general, if it takes N "little" spheres or cubes of side $r \to 0$ to cover some object, then the fractal dimension d_f can be deduced as

$$N(r) \;\propto\; \left(\frac{1}{r}\right)^{d_f} \;\propto\; s^{d_f} \quad (\text{as } r \to 0) \qquad d_f = \lim_{r\to 0} \frac{\log N(r)}{\log(1/r)} \tag{20.18}$$

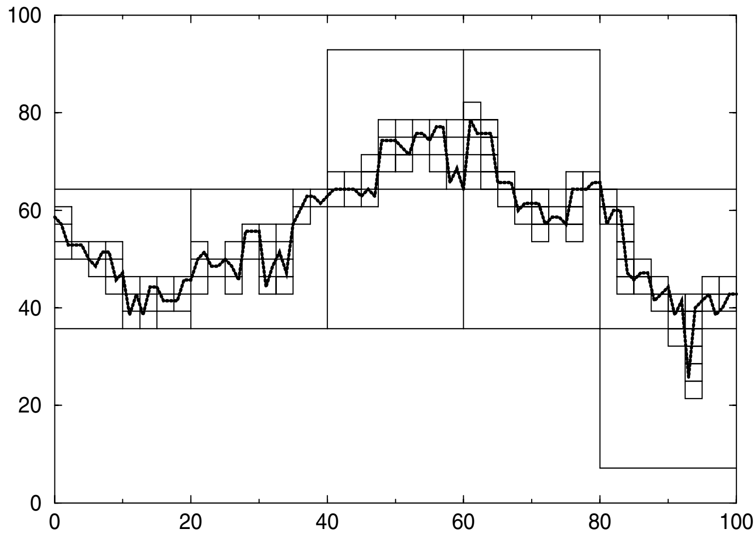

Fig. 20.7 Example of the use of box counting to determine the fractal dimension of the perimeter along a surface. Observe how the "coastline" is being covered by boxes of two different sizes (scales).

Here $s \propto 1/r$ is called the *scale* in geography, so $r \to 0$ corresponds to infinite scale. To illustrate, you may be familiar with the low scale on a map being 10,000 m to the cm, while the high scale is 100 m to the cm. If we want the map to show small details (sizes), we want a map of higher scale.

We will use box counting to determine the dimension of a perimeter, not of an entire figure. Once we have a value for d_f we can determine a value for the length of the perimeter via (20.15).

20.5.3 Coastline Implementation

Rather than ruin our eyes with a geographic map, we use a mathematical one. Specifically, with a little imagination you will see that the top portion of the graph in Fig. 20.5 looks like a natural coastline. Determine d_f by covering this figure, or one you have generated, with a semitransparent piece of graph paper.[1], and counting the number of boxes containing any part of the coastline (Fig. 20.7)

1. Print out your coastline graph with the same physical scale for the vertical and horizontal axes.

[1] Yes, we are suggesting a painfully analog technique based on the theory that trauma leaves a lasting impression. If you prefer, you can store your output as a matrix of 1 and 0 values and let the computer do the counting, but this will take more of your time!

2. The vertical height in our printout was 17 cm. This sets the scale of the graph as 1:17, or $s = 17$.

3. The largest boxes on our graph paper were 1 cm × 1 cm. We found that the coastline passed through $N = 24$ of these large boxes (i.e., 24 large boxes covered the coastline at $s = 17$).

4. The next smaller boxes on our graph paper were 0.5 cm × 0.5 cm. We found that 51 smaller boxes covered the coastline at a scale of $s = 34$.

5. The smallest boxes on our graph paper were 1 mm × 1 mm. We found that 406 smallest boxes covered the coastline at a scale of $s = 170$.

6. Equation (20.18) tells us that as the box sizes get progressively smaller, we have

$$\begin{aligned} \log N &\simeq \log A + d_f \log s \\ \Rightarrow \quad d_f &\simeq \frac{\Delta \log N}{\Delta \log s} = \frac{\log N_2 - \log N_1}{\log s_2 - \log s_1} = \frac{\log(N_2/N_1)}{\log(s_2/s_1)} \end{aligned}$$

Clearly, only the relative scales matter because the proportionality constants cancel out in the ratio. A plot of $\log N$ versus $\log s$, should yield a straight line (the third point verifies that it is a line). In our example we found a slope $d_f = 1.23$.

As given by (20.15), the perimeter of the coastline

$$L \;\propto\; s^{1.23-1} = s^{0.23} \tag{20.19}$$

If we keep making the boxes smaller and smaller so that we are looking at the coastline at higher and higher scale, *and* if the coastline is a fractal with self-similarity at all levels, then the scale s keeps getting larger and larger with no limits (or at least until we get down to some quantum limits). This means

$$L \;\propto\; \lim_{s\to\infty} s^{0.23} = \infty \tag{20.20}$$

We conclude that, in spite of being only a small island, to a mathematician the coastline of Britain is, indeed, infinite.

20.6
Problem 5: Correlated Growth, Forests, and Films

It is an empirical fact that in nature there is an increased likelihood for a plant to grow if there is another one nearby (Fig. 20.8). This *correlation* is also valid

Fig. 20.8 A scene as might be seen in the undergrowth of a forest or in a correlated ballistic deposition.

for our "growing" of surface films, as in the previous algorithm. Your **problem** is to include correlations in the surface simulation.

20.6.1 Correlated Ballistic Deposition Algorithm (column.c)

A variation of the ballistic deposition algorithm, known as *correlated ballistic deposition*, simulates mineral deposition onto substrates in which dendrites form [49]. We extend the previous algorithm to include the likelihood that a freshly deposited particle will attract another particle. We assume that the probability of sticking $\mathcal{P}$ depends on the distance d that the added particle is from the last one (Fig. 20.9):

$$\mathcal{P} = c\, d^{\eta} \tag{20.21}$$

Here η is a parameter and c is a constant that sets the probability scale.[2] For our implementation we choose $\eta = -2$, which means that there is an inverse square attraction between the particles (less probable as they get farther apart).

[2] The absolute probability, of course, must be less than one, but it is nice to choose c so that the relative probabilities produce a graph with easily seen variations.

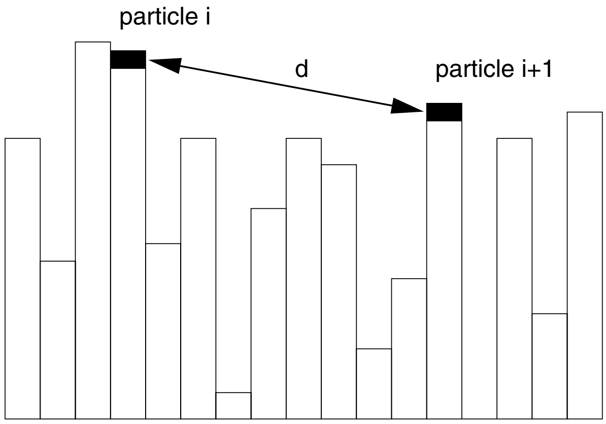

Fig. 20.9 The probability of particle $i+1$ sticking in some column depends on the distance d from the previously deposited particle i.

As in our study of uncorrelated deposition, a uniform random number in the interval $[0, L]$ determines the column in which the particle is deposited. We use the same rules about the heights as before, but now a second random number is used in conjunction with (20.21) to decide if the sticks. For instance, if the computed probability is 0.6 and if $r < 0.6$, the particle is accepted (sticks); if $r > 0.6$, the particle is rejected.

20.6.2
Globular Cluster (Problem)

Consider a bunch of grapes on an overhead vine. Your **problem** is to determine how its tantalizing shape arises. As a hint, you are told that these shapes, as well as others such as dendrites, colloids, and thin-film structure, appear to arise from an aggregation process that is limited by diffusion.

20.6.3
Diffusion-Limited Aggregation Algorithm (dla.c)

A model of diffusion-limited aggregation (DLA) has successfully explained the relation between a cluster's perimeter and mass [50]. We start with a 2D lattice containing a seed particle in the middle. We draw a circle around the particle and place another particle on the circumference of the circle at some random angle. We then release the second particle and have it execute a random walk, much like the one we studied in Chap. 11 but restricted to vertical or horizontal jumps between lattice sites. This is a type of *Brownian motion* that simulates the diffusion process. To make the model more realistic, we let the length of each step vary according to a random Gaussian distribution. If at

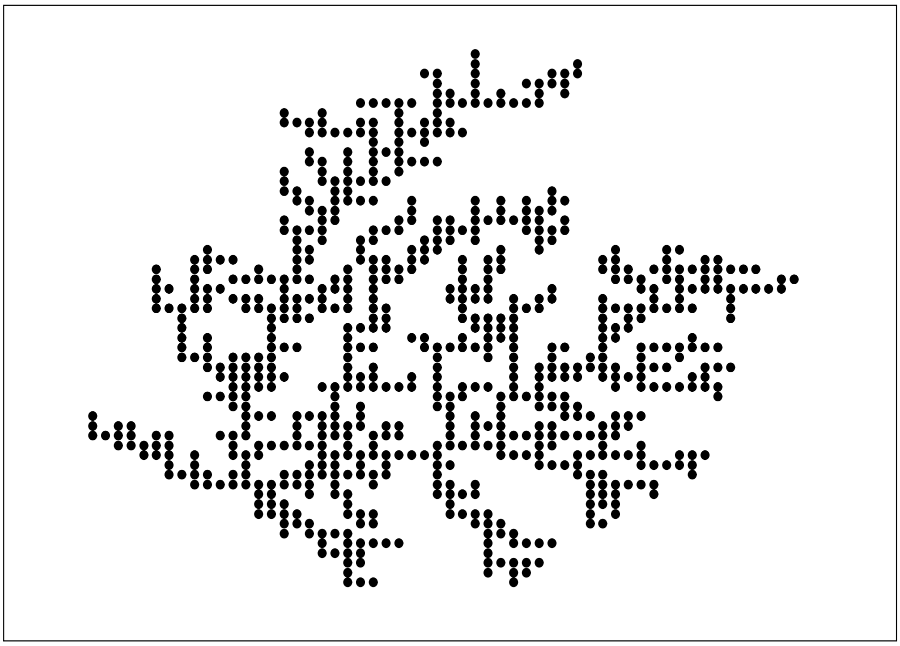

Fig. 20.10 *Left:* A globular cluster of particles of the type that might occur in a colloid. There is a seed at the center, and random walkers are started at random points around a circle, with not all reaching the center.

some point during its random walk, the particle finds another particle within one lattice spacing, they stick together and the walk terminates. If the particle passes outside the circle from which it was released, it is lost forever. The process is repeated as often as desired and results in clusters (Fig. 20.10).

1. Write a subroutine that generates random numbers with a Gaussian distribution.[3]

2. Define a 2D lattice of points represented by the array `grid[400][400]` with all elements initially zero.

3. Place the seed at the center of the lattice, that is, set `grid[199][199]=1`.

4. Imagine a circle of radius 180 lattice constants centered at `grid[199][199]`. This is the circle from which we release particles.

5. Determine the angular position of the new particle on the circle's circumference by generating a uniform random angle between 0 and 2π.

6. Compute the x and y positions of the new particle on the circle.

[3] We indicated how to do this in Section 11.7.3.

7. Determine whether the particle moves horizontally or vertically by generating a uniform random number $0 < r_{xy} < 1$ and applying the rule

$$\text{if } r_{xy} \begin{cases} < 0.5 & \text{motion is vertical} \\ > 0.5 & \text{motion is horizontal} \end{cases} \tag{20.22}$$

8. Generate a Gaussian-weighted random number in the interval $[-\infty, \infty]$. This is the size of the step, with the sign indicating direction.

9. We now know the total distance and direction in which the particle will move. It "jumps" one lattice spacing at a time until this total distance is covered.

10. Before a jump, check whether a nearest-neighbor site is occupied:
    (a) If occupied, stay at present position and the walk is over.
    (b) If unoccupied, the particle jumps one lattice spacing.
    (c) Continue the checking and jumping until the total distance is covered, until the particle sticks, or until it leaves the circle.

11. Once one random walk is over, another particle can be released and the process repeated. This is how the cluster grows.

Because many particles get "lost," you may need to generate hundreds of thousands of particles to form a cluster of several hundred particles.

20.6.4
Fractal Analysis of DLA Graph (Assessment)

A cluster generated with the DLA technique is shown in Fig. 20.10. We wish to analyze it to see if the structure is a fractal, and, if so, to determine its dimension. The analysis is a variation of the one used to determine the length of the coastline of Britain.

1. Draw a square of length L, small relative to the size of the cluster, around the seed particle. ("Small" might be seven lattice spacings to a side.)

2. Count the number of particles within the square.

3. Compute the density ρ by dividing the number of particles by the number of sites available in the box (49 in our example).

4. Repeat the procedure using larger and larger squares.

5. Stop when the cluster is covered.

6. The (box-counting) fractal dimension d_f is estimated from a log–log plot of the density ρ versus L. If the cluster is a fractal, then (20.2) tells us that $\rho \propto L^{d_f - 2}$, and the graph should be a straight line of slope $d_f - 2$.

The graph we generated had a slope of -0.36, which corresponds to a fractal dimension of 1.66. Because random numbers are involved, the graph you generate will be different, but the fractal dimension should be similar. (Actually, the structure is multifractal, and so the dimension varies with position.)

20.7 Problem 7: Fractal Structures in Bifurcation Graph

Recall the project on the logistics map where we plotted the values of the stable population numbers versus growth parameter μ. Take one of the bifurcation graphs you produced and determine the fractal dimension of different parts of the graph by using the same technique that was applied to the coastline of Britain.