EXCERPTED FROM

STEPHEN WOLFRAM

A NEW KIND OF SCIENCE

NOTES FOR CHAPTER 7:

Mechanisms in Programs and Nature

Mechanisms in Programs and Nature

Universality of Behavior

■ **History.** That very different natural and artificial systems can show similar forms has been noted for many centuries. Informal studies have been done by a whole sequence of architects interested both in codifying possible forms and in finding ways to make structures fit in with nature and with our perception of it. Beginning in the Renaissance the point has also been noted by representational and decorative artists, most often in the context of developing a theory of the types of forms to be studied by students of art. The growth of comparative anatomy in the 1800s led to attempts at more scientific treatments, with analogies between biological and physical systems being emphasized particularly by D'Arcy Thompson in 1917. Yet despite all this, the phenomenon of similarity between forms remained largely a curiosity, discussed mainly in illustrated books with no clear basis in either art or science. In a few cases (such as work by Peter Stevens in 1974) general themes were however suggested. These included for example symmetry, the golden ratio, spirals, vortices, minimal surfaces, branching patterns, and—since the 1980s—fractals. The suggestion is also sometimes made that we perceive a kind of harmony in nature because we see only a limited number of types of forms in it. And particularly in classical architecture the idea is almost universally used that structures will seem more comfortable to us if they repeat in ornament or otherwise forms with which we have become familiar from nature. Whenever a scientific model has the same character for different systems this means that the systems will tend to show similar forms. And as models like cellular automata capable of dealing with complexity have become more widespread it has been increasingly popular to show that they can capture similar forms seen in very different systems.

Three Mechanisms for Randomness

■ **Page 299 · Definition.** How randomness can be defined is discussed at length on page 552.

■ **History.** In antiquity, it was often assumed that all events must be governed by deterministic fate—with any apparent randomness being the result of arbitrariness on the part of the gods. Around 330 BC Aristotle mentioned that instead randomness might just be associated with coincidences outside whatever system one is looking at, while around 300 BC Epicurus suggested that there might be randomness continually injected into the motion of all atoms. The rise of emphasis on human free will (see page 1135) eroded belief in determinism, but did not especially address issues of randomness. By the 1700s the success of Newtonian physics seemed again to establish a form of determinism, and led to the assumption that whatever randomness was actually seen must reflect lack of knowledge on the part of the observer—or particularly in astronomy some form of error of measurement. The presence of apparent randomness in digit sequences of square roots, logarithms, numbers like π, and other mathematical constructs was presumably noticed by the 1600s (see page 911), and by the late 1800s it was being taken for granted. But the significance of this for randomness in nature was never recognized. In the late 1800s and early 1900s attempts to justify both statistical mechanics and probability theory led to ideas that perfect microscopic randomness might somehow be a fundamental feature of the physical world. And particularly with the rise of quantum mechanics it came to be thought that meaningful calculations could be done only on probabilities, not on individual random sequences. Indeed, in almost every area where quantitative methods were used, if randomness was observed, then either a different system was studied, or efforts were made to remove the randomness by averaging or some other statistical method. One case where there was occasional discussion of origins of randomness from at least

the early 1900s was fluid turbulence (see page 997). Early theories tended to concentrate on superpositions of repetitive motions, but by the 1970s ideas of chaos theory began to dominate. And in fact the widespread assumption emerged that between randomness in the environment, quantum randomness and chaos theory almost any observed randomness in nature could be accounted for. Traditional mathematical models of natural systems are often expressed in terms of probabilities, but do not normally involve anything one can explicitly consider as randomness. Models used in computer simulations, however, do very often use explicit randomness. For not knowing about the phenomenon of intrinsic randomness generation, it has normally been assumed that with the kinds of discrete elements and fairly simple rules common in such models, realistically complicated behavior can only ever be obtained if explicit randomness is continually introduced.

■ **Applications of randomness.** See page 1192.

■ **Sources of randomness.** Two simple mechanical methods for generating randomness seem to have been used in almost every civilization throughout recorded history. One is to toss an object and see which way up or where it lands; the other is to select an object from a collection mixed by shaking. The first method has been common in games of chance, with polyhedral dice already existing in 2750 BC. The second—often called drawing lots—has normally been used when there is more at stake. It is mentioned several times in the Bible, and even today remains the most common method for large lotteries. (See page 969.) Variants include methods such as drawing straws. In antiquity fortune-telling from randomness often involved looking say at growth patterns of goat entrails or sheep shoulder blades; today configurations of tea leaves are sometimes considered. In early modern times the matching of fracture patterns in broken tally sticks was used to identify counterparties in financial contracts. Horse races and other events used as a basis for gambling can be viewed as randomness sources. Children's games like musical chairs in effect generate randomness by picking arbitrary stopping times. Games of chance based on wheels seem to have existed in Roman times; roulette developed in the 1700s. Card shuffling (see page 974) has been used as a source of randomness since at least the 1300s. Pegboards (as on page 312) were used to demonstrate effects of randomness in the late 1800s. An explicit table of 40,000 random digits was created in 1927 by Leonard Tippett from details of census data. And in 1938 further tables were generated by Ronald Fisher from digits of logarithms. Several tables based on physical processes were produced, with the RAND Corporation in 1955 publishing a table of a million random digits obtained from an electronic roulette wheel. Beginning in the 1950s, however, it became increasingly common to use pseudorandom generators whenever long sequences were needed—with linear feedback shift registers being most popular in standalone electronic devices, and linear congruential generators in programs (see page 974). There nevertheless continued to be occasional work done on mechanical sources of randomness for toys and games, and on physical electronic sources for cryptography systems (see page 969).

Randomness from the Environment

■ **Page 301 · Stochastic models.** The mechanism for randomness discussed in this section is the basis for so-called stochastic models now widely used in traditional science. Typically the idea of these models is to approximate those elements of a system about which one does not know much by random variables. (See also page 588.) In the early work along these lines done by James Clerk Maxwell and others in the 1880s, analytical formulas were usually worked out for the probabilities of different outcomes. But when electronic computers became available in the 1940s, the so-called Monte Carlo method became increasingly popular, in which instead explicit simulations are performed with different choices of random variables, and then statistical averages are found. Early uses of the Monte Carlo method were mostly in physics, particularly for studies of neutron diffusion and particle shower generation in high-energy collisions. But by the 1980s the Monte Carlo method had also become common in other fields, and was routinely used in studying for example message flows in communication networks and pricing processes in financial markets. (See also page 1192.)

■ **Page 301 · Ocean surfaces.** See page 1001.

■ **Page 302 · Random walks.** See page 328.

■ **Page 302 · Electronic noise.** Three types of noise are commonly observed in typical devices:

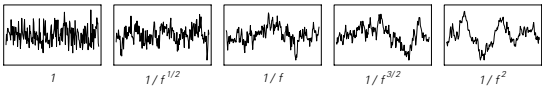
Shot noise. Electric currents are not continuous but are ultimately made up from large numbers of moving charge carriers, typically electrons. Shot noise arises from statistical fluctuations in the flow of charge carriers: if a single bit of data is represented by 10,000 electrons, the magnitude of the fluctuations will typically be about 1%. When looked at as a waveform over time, shot noise has a flat frequency spectrum.

Thermal (Johnson) noise. Even though an electric current may have a definite overall direction, the individual charge carriers within it will exhibit random motions. In a material

at nonzero temperature, the energy of these motions and thus the intensity of the thermal noise they produce is essentially proportional to temperature. (At very low temperatures, quantum mechanical fluctuations still yield random motion in most materials.) Like shot noise, thermal noise has a flat frequency spectrum.

Flicker (1/f) noise. Almost all electronic devices also exhibit a third kind of noise, whose main characteristic is that its spectrum is not flat, but instead goes roughly like $1/f$ over a wide range of frequencies. Such a spectrum implies the presence of large low-frequency fluctuations, and indeed fluctuations are often seen over timescales of many minutes or even hours. Unlike the types of noise described above, this kind of noise can be affected by details of the construction of individual devices. Although seen since the 1920s its origins remain somewhat mysterious (see below).

▪ **Power spectra.** Many random processes in nature show power spectra $Abs[Fourier[data]]^2$ with fairly simple forms. Most common are white noise uniform in frequency and $1/f^2$ noise associated with random walks. Other pure power laws $1/f^\alpha$ are also sometimes seen; the pictures below show some examples. (Note that the correlations in such data in some sense go like $t^{\alpha-1}$.) Particularly over the past few decades all sorts of examples of "$1/f$ noise" have been identified with $\alpha \simeq 1$, including flicker noise in resistors, semiconductor devices and vacuum tubes, as well as thunderstorms, earthquake and sunspot activity, heartbeat intervals, road traffic density and some DNA sequences. A pure $1/f^\alpha$ spectrum presumably reflects some form of underlying nesting or self-similarity, although exactly what has usually been difficult to determine. Mechanisms that generally seem able to give $\alpha \simeq 1$ include random walks with exponential waiting times, power-law distributions of step sizes (Lévy flights), or white noise variations of parameters, as well as random processes with exponentially distributed relaxation times (as from Boltzmann factors for uniformly distributed barrier heights), fractional integration of white noise, intermittency at transitions to chaos, and random substitution systems. (There was confusion in the late 1980s when theoretical studies of self-organized criticality failed correctly to take squares in computing power spectra.) Note that the Weierstrass function of page 918 yields a $1/f$ spectrum, and presumably suitable averages of spectra from any substitution system should also have $1/f^\alpha$ forms (compare page 586).



▪ **Page 303 · Spark chambers.** The sensitivity of sparks to microscopic details of the environment is highlighted by the several devices which essentially use them to detect the passage of individual elementary particles such as protons. Such particles leave a tiny trail of ionized gas, which becomes the path of the spark. This principle was used in Geiger counters, and later in spark chambers and wire chambers.

▪ **Physical randomness generators.** It is almost universally assumed that at some level physical processes must be the best potential sources of true randomness. But in practice their record has actually been very poor. It does not help that unlike algorithms physical devices can be affected by their environment, and can also not normally be copied identically. But in almost every case I know where detailed analysis has been done substantial deviations from perfect randomness have been found. This has however typically been attributed to engineering mistakes—or to sampling data too quickly—and not to anything more fundamental that is for example worth describing in publications.

▪ **Mechanical randomness.** It takes only small imperfections in dice or roulette wheels to get substantially non-random results (see page 971). Gaming regulations typically require dice to be perfect cubes to within one part in a few thousand; casinos normally retire dice after a few hundred rolls.

In processes like stirring and shaking it can take a long time for correlations to disappear—as in the phenomenon of long-time tails mentioned on page 999. One notable consequence were traces of insertion order among the 366 capsules used in the 1970 draft lottery in the U.S. But despite such problems mixing of objects remains by far the most common way to generate randomness when there is a desire for the public to see randomization occur. And so for example all the state lotteries in the U.S. are currently based on mixing between 10 and 54 balls. (Numbers games were instead sometimes based on digits of financial data in newspapers.)

There have been a steady stream of inventions for mechanical randomness generation. Some are essentially versions of dice. Others involve complicated cams or linkages, particularly for mechanical toys. And still others involve making objects like balls bounce around as randomly as possible in air or other fluids.

▪ **Electronic randomness.** Since the 1940s a steady stream of electronic devices for producing randomness have been invented, with no single one ever becoming widely used. An early example was the ERNIE machine from 1957 for British national lottery (premium bond) drawings, which worked by sampling shot noise from neon discharge

tubes—and perhaps because it extracted only a few digits per second no deviations from randomness in its output were found. (U.S. missiles apparently used a similar method to produce randomly spaced radar pulses for determining altitude.) Since the 1970s electronic randomness generators have typically been based on features of semiconductor devices—sometimes thermal noise, but more often breakdown, often in back-biased zener diodes. All sorts of schemes have been invented for getting unbiased output from such systems, and acceptable randomness can often be obtained at kilohertz rates, but obvious correlations almost always appear at higher rates. Macroscopic thermal diffusion undoubtedly underestimates the time for good microscopic randomization. For in addition to *1/f* noise effects, solitons and other collective lattice effects presumably lead to power-law decay of correlations. It still seems likely however that some general inequalities should exist between the rate and quality of randomness that can be extracted from a system with particular thermodynamic properties.

■ **Quantum randomness.** It is usually assumed that even if all else fails a quantum process such as radioactive decay will yield perfect randomness. But in practice the most accurate measurements show phenomena such as *1/f* noise, presumably as a result of features of the detector and perhaps of electromagnetic fields associated with decay products. Acceptable randomness has however been obtained at rates of tens of bits per second. Recent attempts have also been made to produce quantum randomness at megahertz rates by detecting paths of single photons. (See also page 1064.)

■ **Randomness in computer systems.** Most randomness needed in practical computer systems is generated purely by programs, as discussed on page 317. But to avoid having a particular program give exactly the same random sequence every time it is run, one usually starts from a seed chosen on the basis of some random feature of the environment. Until the early 1990s this seed was most often taken from the exact time of day indicated by the computer's clock at the moment when it was requested. But particularly in environments where multiple programs can start almost simultaneously other approaches became necessary. Versions of the Unix operating system, for example, began to support a virtual device (typically called /dev/random) to maintain a kind of pool of randomness based on details of the computer system. Most often this uses precise timings between interrupts generated by keys being pressed, a mouse being moved, or data being delivered from a disk, network, or other device. And to prevent the same state being reached every time a computer is rebooted, some information is permanently maintained in a file. At the end of the 1990s standard microprocessors also began to include instructions to sample thermal noise from an on-chip resistor. (Any password or encryption key made up by a human can be thought of as a source of randomness; some systems look at details of biometric data, or scribbles drawn with a mouse.)

■ **Randomness in biology.** Thermal fluctuations in chemical reactions lead to many kinds of microscopic randomness in biological systems, sometimes amplified when organisms grow. For example, small-scale randomness in embryos can affect large-scale pigmentation patterns in adult organisms, as discussed on page 1013. Random changes in single DNA molecules can have global effects on the development of an organism. Standard mitotic cell division normally produces identical copies of DNA—with random errors potentially leading for example to cancers. But in sexual reproduction genetic material is rearranged in ways normally assumed by classical genetics to be perfectly random. One reason is that which sperm fertilizes a given egg is determined by random details of sperm and fluid motion. Another reason is that egg and sperm cells get half the genetic material of an organism, somewhat at random. In most cells, say in humans, there are two versions of all 23 chromosomes—one from the father and one from the mother. But when meiosis forms egg and sperm cells they get only one version of each. There is also exchange of DNA between paternal and maternal chromosomes, typically with a few crossovers per chromosome, at positions that seem more or less randomly distributed among many possibilities (the details affect regions of repeating DNA used for example in DNA fingerprinting).

In the immune system blocks of DNA—and joins between them—are selected at random by microscopic chemical processes when antibodies are formed.

Most animal behavior is ultimately controlled by electrical activity in nerve cells—and this can be affected by details of sensory input, as well as by microscopic chemical processes in individual cells and synapses (see page 1011).

Flagellated microorganisms can show random changes in direction as a result of tumbling when their flagella counter-rotate and the filaments in them flail around.

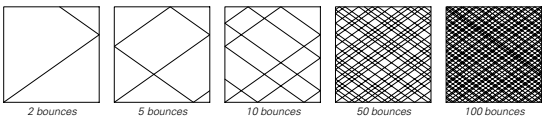
(See also page 1011.)

Chaos Theory and Randomness from Initial Conditions

■ **Page 305 · Spinning and tossing.** Starting with speed v, the speed of the ball at time t is simply $v - at$, where a is the deceleration produced by friction. The ball thus stops at time

v/a. The distance gone by the ball at a given time is $x = v\,t - a\,t^2/2$, and its orientation is $Mod[x, 2\,\pi\,r]$. For dice and coins there are some additional detailed effects associated with the shapes of these objects and the way they bounce. (Polyhedral dice have become more common since Dungeons & Dragons became popular in the late 1970s.) Note that in practice a coin tossed in the air will typically turn over between ten and twenty times while a die rolled on a table will turn over a few tens of times. A coin spun on a table can rotate several hundred times before falling over and coming to rest.

▪ **Billiards.** A somewhat related system is formed by a billiard ball bouncing around on a table. The issue of which sequence of horizontal and vertical sides the ball hits depends on the exact slope with which the ball is started (in the picture below it is $1/\sqrt{2}$). In general, it is given by the successive terms in the continued fraction form (see page 914) of this slope, and is related to substitution systems (see page 903). (See also page 1022.)



▪ **Fluttering.** If one releases a stationary piece of paper in air, then unlike a coin, it does not typically maintain the same orientation as it falls. Small pieces of paper spin in a repetitive way; but larger pieces of paper tend to flutter in a seemingly random way (as discussed, among others, by James Clerk Maxwell in 1853). A similar phenomenon can be seen if one drops a coin in water. I suspect that in these cases the randomness that occurs has an intrinsic origin, rather than being the result of sensitive dependence on initial conditions.

▪ **History of chaos theory.** The idea that small causes can sometimes have large effects has been noted by historians and others since antiquity, and captured for example in "for want of a nail … a kingdom was lost". In 1860 James Clerk Maxwell discussed how collisions between hard sphere molecules could lead to progressive amplification of small changes and yield microscopic randomness in gases. In the 1870s Maxwell also suggested that mechanical instability and amplification of infinitely small changes at occasional critical points might explain apparent free will (see page 1135). (It was already fairly well understood that for example small changes could determine which way a beam would buckle.) In 1890 Henri Poincaré found sensitive dependence on initial conditions in a particular case of the three-body problem (see below), and later proposed that such phenomena could be common, say in meteorology. In 1898 Jacques Hadamard noted general divergence of trajectories in spaces of negative curvature, and Pierre Duhem discussed the possible general significance of this in 1908. In the 1800s there had been work on nonlinear oscillators—particularly in connection with models of musical instruments—and in 1927 Balthazar van der Pol noted occasional "noisy" behavior in a vacuum tube oscillator circuit presumably governed by a simple nonlinear differential equation. By the 1930s the field of dynamical systems theory had begun to provide characterizations of possible forms of behavior in differential equations. And in the early 1940s Mary Cartwright and John Littlewood noted that van der Pol's equation could exhibit solutions somehow sensitive to all digits in its initial conditions. The iterated map $x \to 4\,x\,(1 - x)$ was also known to have a similar property (see page 918). But most investigations centered on simple and usually repetitive behavior—with any strange behavior implicitly assumed to be infinitely unlikely. In 1962, however, Edward Lorenz did a computer simulation of a set of simplified differential equations for fluid convection (see page 998) in which he saw complicated behavior that seemed to depend sensitively on initial conditions—in a way that he suggested was like the map $x \to FractionalPart[2\,x]$. In the mid-1960s, notably through the work of Steve Smale, proofs were given that there could be differential equations in which such sensitivity is generic. In the late 1960s there began to be all sorts of simulations of differential equations with complicated behavior, first mainly on analog computers, and later on digital computers. Then in the mid-1970s, particularly following discussion by Robert May, studies of iterated maps with sensitive dependence on initial conditions became common. Work by Robert Shaw in the late 1970s clarified connections between information content of initial conditions and apparent randomness of behavior. The term "chaos" had been used since antiquity to describe various forms of randomness, but in the late 1970s it became specifically tied to the phenomenon of sensitive dependence on initial conditions. By the early 1980s at least indirect signs of chaos in this sense (see note below) had been seen in all sorts of mechanical, electrical, fluid and other systems, and there emerged a widespread conviction that such chaos must be the source of all important randomness in nature. So in 1985 when I raised the possibility that intrinsic randomness might instead be a key phenomenon this was greeted with much hostility by some younger proponents of chaos theory. Insofar as what they had to say was of a scientific nature, their main point was that somehow what I

had seen in cellular automata must be specific to discrete systems, and would not occur in the continuous systems assumed to be relevant in nature. But from many results in this book it is now clear that this is not correct. (Note that James Gleick's 1987 popular book *Chaos* covers somewhat more than is usually considered chaos theory—including some of my results on cellular automata from the early 1980s.)

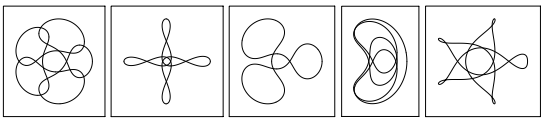
■ **Information content of initial conditions.** See page 920.

■ **Recognizing chaos.** Any system that depends sensitively on digits in its initial conditions must necessarily be able to show behavior that is not purely repetitive (compare page 955). And when it is said that chaos has been found in a particular system in nature what this most often actually means is just that behavior with no specific repetition frequency has been seen (compare page 586). To give evidence that this is not merely a reflection of continual injection of randomness from the environment what is normally done is to show that at least some aspect of the behavior of the system can be fit by a definite simple iterated map or differential equation. But inevitably the fit will only be approximate, so there will always be room for effects from randomness in the environment. And in general this kind of approach can never establish that sensitive dependence on initial conditions is actually the dominant source of randomness in a given system—say as opposed to intrinsic randomness generation. (Attempts are sometimes made to detect sensitive dependence directly by watching whether a system can do different things after it appears to return to almost exactly the same state. But the problem is that it is hard to be sure that the system really is in the same state—and that there are not all sorts of large differences that do not happen to have been observed.)

■ **Instability.** Sensitive dependence on initial conditions is associated with a kind of uniform instability in systems. But vastly more common in practice is instability only at specific critical points—say bifurcation points—combined with either intrinsic randomness generation or randomness from the environment. (Note that despite its widespread use in discussions of chaos theory, this is also what usually seems to happen with the weather; see page 1177.)

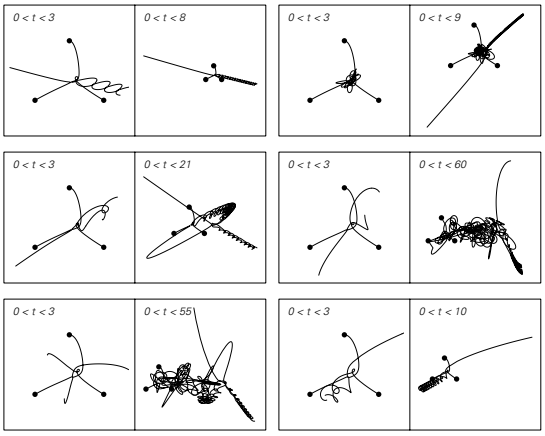
■ **Page 313 · Three-body problem.** The two-body problem was analyzed by Johannes Kepler in 1609 and solved by Isaac Newton in 1687. The three-body problem was a central topic in mathematical physics from the mid-1700s until the early 1900s. Various exact results were obtained—notably the existence of stable equilateral triangle configurations corresponding to so-called Lagrange points. Many approximate practical calculations, particularly on the Earth-Moon-Sun system, were done using series expansions involving thousands of algebraic terms. (It is now possible to get most results just by direct numerical computation using for example *NDSolve*.) From its basic setup the three-body system conserves standard mechanical quantities like energy and angular momentum. But it was thought it might also conserve other quantities (or so-called integrals of the motion). In 1887, however, Heinrich Bruns showed that there could be no such quantities expressible as algebraic functions of the positions and velocities of the bodies (in standard Cartesian coordinates). In the mid-1890s Henri Poincaré then showed that there could also be no such quantities analytic in positions, velocities and mass ratios. And from these results the conclusion was drawn that the three-body problem could not be solved in terms of algebraic formulas and integrals. In 1912 Karl Sundman did however find an infinite series that could in principle be summed to give the solution—but which converges exceptionally slowly. And even now it remains conceivable that the three-body problem could be solved in terms of more sophisticated standard mathematical functions. But I strongly suspect that in fact nothing like this will ever be possible and that instead the three-body problem will turn out to show the phenomenon of computational irreducibility discussed in Chapter 12 (and that for example three-body systems are universal and in effect able to perform any computation). (See also page 1132.)

In Henri Poincaré's study of the collection of possible trajectories for three-body systems he identified sensitive dependence on initial conditions (see above), noted the general complexity of what could happen (particularly in connection with so-called homoclinic tangles), and developed topology to provide a simpler overall description. With appropriate initial conditions one can get various forms of simple behavior. The pictures below show some of the possible repetitive orbits of an idealized planet moving in the plane of a pair of stars that are in a perfect elliptical orbit.



The pictures below show results for a fairly typical sequence of initial conditions where all three bodies interact. (The two bodies at the bottom are initially at rest; the body at the top is given progressively larger rightward velocities.) What generically happens is that one of the bodies escapes from the other two (like t or sometimes $t^{2/3}$). Often this happens quickly, but sometimes all three bodies show complex and

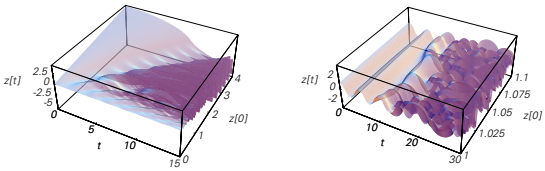
apparently random behavior for quite a while. (The delay before escaping is reminiscent of resonant scattering.)



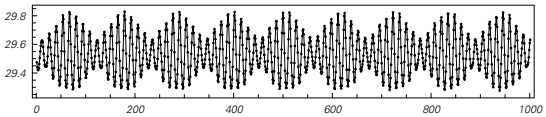
▪ **Page 314 · Simple case.** The position of the idealized planet in the case shown satisfies the differential equation

$$\partial_{tt} z[t] == -z[t]/(z[t]^2 + (1/2\,(1 + e\,\mathrm{Sin}[2\,\pi\,t]))^2)^{3/2}$$

where e is the eccentricity of the elliptical orbit of the stars ($e = 0.1$ in the picture). (Note that the physical situation is unstable: if the planet is perturbed so that there is a difference between its distance to each star, this will tend to increase.) Except when $e = 0$, the equation has no solution in terms of standard mathematical functions. It can be solved numerically in *Mathematica* using *NDSolve*, although a working precision of 40 decimal digits was used to obtain the results shown. Following work by Kirill Sitnikov in 1960 and by Vladimir Alekseev in 1968, it was established that with suitably chosen initial conditions, the equation yields any sequence *Floor[t[i + 1] - t[i]]* of successive zero-crossing times *t[i]*. The pictures below show the dependence of *z[t]* on *t* and *z[0]*. As *t* increases, *z[t]* typically begins to vary more rapidly with *z[0]*—reflecting sensitive dependence on initial conditions.



▪ **Page 314 · Randomness in the solar system.** Most motion observed in the solar system on human timescales is highly regular—though sometimes intricate, as in the sequence of numbers of days between successive new moons shown below. In the mid-1980s, however, work by Jack Wisdom and others established that randomness associated with sensitive dependence on initial conditions could occur in certain current situations in the solar system, notably in the orbits of asteroids. Various calculations suggest that there should also be sensitive dependence on initial conditions in the orbits of planets in the solar system—with effects doubling every few million years. But there are so far no observational signs of randomness resulting from this, and indeed the planets—at least now—mostly just seem to have orbits that are within a few percent of circles. If a planet moved in too random a way then it would tend to collide or escape from the solar system. And indeed it seems quite likely that in the past there may have been significantly more planets in our solar system—with only those that maintained regular orbits now being left. (See also page 1021.)



The Intrinsic Generation of Randomness

▪ **Autoplectic processes.** In the 1985 paper where I introduced intrinsic randomness generation I called processes that show this autoplectic, while I called processes that transcribe randomness from outside homoplectic.

▪ **Page 316 · Algorithmic randomness.** The idea of there being no simple procedure that can generate a particular sequence can be stated more precisely by saying that there is no program shorter than the sequence itself which can be used to generate the sequence, as discussed in more detail on page 1067.

▪ **Page 317 · Randomness in *Mathematica*.** *SeedRandom[n]* is the function that sets up the initial conditions for the cellular automaton. The idea of using this kind of system in general and this system in particular as a source of randomness was described in my 1987 U.S. patent number 4,691,291.

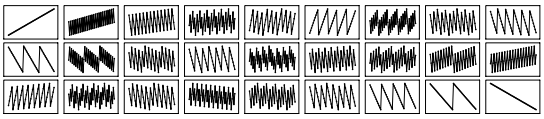
▪ **Page 321 · Cellular automata.** From the discussion here it should not be thought that in general there is necessarily anything better about generating randomness with cellular automata than with systems based on numbers. But the point is that the specific method used for making practical linear congruential generators does not yield particularly good randomness and has led to some incorrect intuition about the generation of randomness. If one goes beyond the specifics of linear congruential generators, then one can find many features of systems based on numbers that seem to be

perfectly random, as discussed in Chapter 4. In addition, one should recognize that while the complete evolution of the cellular automaton may effectively generate perfect randomness, there may be deviations from randomness introduced when one constructs a practical random number generator with a limited number of cells. Nevertheless, no such deviations have so far been found except when one looks at sequences whose lengths are close to the repetition period. (See however page 603.)

▪ **Page 321 · Card shuffling.** Another rather poor example of intrinsic randomness generation is perfect card shuffling. In a typical case, one splits the deck of cards in two, then carefully riffles the cards so as to make alternate cards come from each part of the deck. Surprisingly enough, this simple procedure, which can be represented by the function

```
s[list_] := Flatten[
    Transpose[Reverse[Partition[list, Length[list]/2]]]]
```

with or without the *Reverse*, is able to produce orderings which at least in some respects seem quite random. But by doing *Nest[s, Range[52], 26]* one ends up with a simple reversal of the original deck, as in the pictures below.



▪ **Random number generators.** A fairly small number of different types of random number generators have been used in practice, so it is possible to describe all the major ones here.

Linear congruential generators. The original suggestion made by Derrick Lehmer in 1948 was to take a number n and at each step to replace it by *Mod[a n, m]*. Lehmer used $a = 23$ and $m = 10^8 + 1$. Most subsequent implementations have used $m = 2^j$, often with $j = 31$. Such choices are particularly convenient on computers where machine integers are represented by 32 binary digits. The behavior of the linear congruential generator depends greatly on the exact choice of a. Starting with the so-called RANDU generator used on mainframe computers in the 1960s, a common choice made was $a = 65539$. But as shown in the main text, this choice leads to embarrassingly obvious regularities. Starting in the mid-1970s, another common choice was $a = 69069$. This was also found to lead to regularities, but only in six or more dimensions. (Small values of a also lead to an excess of runs of identical digits, as mentioned on page 903.)

The repetition period for a generator with rule $n \to$ *Mod[a n, m]* is given (for a and m relatively prime) by *MultiplicativeOrder[a, m]*. If m is of the form 2^j, this implies a maximum period for any a of $m/4$, achieved when *MemberQ[{3, 5}, Mod[a, 8]]*. In general the maximum period is *CarmichaelLambda[m]*, where the value $m - 1$ can be achieved for prime m.

As illustrated in the main text, when $m = 2^j$ the right-hand base 2 digits in numbers produced by linear congruential generators repeat with short periods; a digit k positions from the right will typically repeat with period no more than 2^k. When $m = 2^j - 1$ is prime, however, even the rightmost digit repeats only with period $m - 1$ for many values of a.

More general linear congruential generators use the basic rule $n \to$ *Mod[a n + b, m]*, and in this case, $n = 0$ is no longer special, and a repetition period of exactly m can be achieved with appropriate choices of a, b and m. Note that if the period is equal to its absolute maximum of m, then every possible n is always visited, whatever n one starts from. Page 962 showed diagrams that represent the evolution for all possible starting values of n.

Each point in the 2D plots in the main text has coordinates of the form *{n[i], n[i + 1]}* where *n[i + 1] = Mod[a n[i], m]*. If one could ignore the *Mod*, then the coordinates would simply be *{n[i], a n[i]}*, so the points would lie on a single straight line with slope a. But the presence of the *Mod* takes the points off this line whenever $a\,n[i] \geq m$. Nevertheless, if a is small, there are long runs of *n[i]* for which the *Mod* is never important. And that is why in the case $a = 3$ the points in the plot fall on obvious lines.

In the case $a = 65539$, the points lie on planes in 3D. The reason for this is that

```
n[i + 2] == Mod[65539^2 n[i], 2^31] ==
  Mod[6 n[i + 1] - 9 n[i], 2^31]
```

so that in computing *n[i + 2]* from *n[i + 1]* and *n[i]* only small coefficients are involved.

It is a general result related to finding short vectors in lattices that for some d the quantity *n[i + d]* can always be written in terms of the *n[i + k]*'s; $k < d$ using only small coefficients. And as a consequence, the points produced by any linear congruential generator must lie on regular hyperplanes in some number of dimensions.

(For cryptanalysis of linear congruential generators see page 1089.)

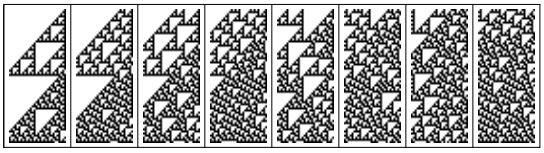
Linear feedback shift registers. Used since the 1950s, particularly in special-purpose electronic devices, these systems are effectively based on running additive cellular automata such as rule 60 in registers with a limited number

of cells and with a certain type of spiral boundary conditions. In a typical case, each cell is updated using

```
LFSRStep[list_] :=
  Append[Rest[list], Mod[list[[1]] + list[[2]], 2]]
```

with a step of cellular automaton evolution corresponding to the result of updating all cells in the register. As with additive cellular automata, the behavior obtained depends greatly on the length n of the register. The maximal repetition period of $2^n - 1$ can be achieved only if `Factor[1 + x + x^n, Modulus → 2]` finds no factors. (For $n < 512$, this is true when $n = 1, 2, 3, 4, 6, 7, 9, 15, 22, 28, 30, 46, 60, 63, 127, 153, 172, 303$ or 471. Maximal period is assured when in addition `PrimeQ[2^n - 1]`.) The pictures below show the evolution obtained for $n = 30$ with

```
NestList[Nest[LFSRStep, #, n] &,
  Append[Table[0, {n - 1}], 1], t]
```



Like additive cellular automata as discussed on page 951, states in a linear feedback shift register can be represented by a polynomial `FromDigits[list, x]`. Starting from a single 1, the state after t steps is then given by

```
PolynomialMod[x^t, {1 + x + x^n, 2}]
```

This result illustrates the analogy with linear congruential generators. And if the distribution of points generated is studied with the Cantor set geometry, the same kind of problems occur as in the linear congruential case (compare page 1094).

In general, linear feedback shift registers can have "taps" at any list of positions on the register, so that their evolution is given by

```
LFSRStep[taps_List, list_] :=
  Append[Rest[list], Mod[Apply[Plus, list[[taps]]], 2]]
```

(With taps specified by the positions of 1's in a vector of 0's, the inside of the `Mod` can be replaced by `vec . list` as on page 1087.) For a register of size n the maximal period of $2^n - 1$ is obtained whenever $x^n + $ `Apply[Plus, x^(taps-1)]` is one of the `EulerPhi[2^n - 1]/n` primitive polynomials that appear in `Factor[Cyclotomic[2^n - 1, x], Modulus → 2]`. (See pages 963 and 1084.)

One can also consider nonlinear feedback shift registers, as discussed on page 1088.

Generalized Fibonacci generators. It was suggested in the late 1950s that the Fibonacci sequence `f[n_] := f[n - 1] + f[n - 2]` modulo 2^k might be used with different choices of `f[0]` and `f[1]` as a random number generator (see page 891). This particular idea did not work well, but generalizations based on the recurrence `f[n_] := Mod[f[n - p] + f[n - q], 2^k]` have been studied extensively, for example with $p = 24$, $q = 55$. Such generators are directly related to linear feedback shift registers, since with a list of length q, each step is simply

```
Append[Rest[list], Mod[list[[1]] + list[[q - p + 1]], 2^k]]
```

Cryptographic generators. As discussed on page 598, so-called stream cipher cryptographic systems work essentially by generating a repeatable random sequence. Practical stream cipher systems can thus be used as random number generators. Starting in the 1980s, the most common example has been the Data Encryption Standard (DES) introduced by the U.S. government (see page 1085). Unless special-purpose hardware is used, however, this method has not usually been efficient enough for practical random number generation applications.

Quadratic congruential generators. Several generalizations of linear congruential generators have been considered in which nonlinear functions of n are used at each step. In fact, the first known generator for digital computers was John von Neumann's "middle square method"

```
n → FromDigits[Take[IntegerDigits[n^2, 10, 20], {5, 15}], 10]
```

In practice this generator has too short a repetition period to be useful. But in the early 1980s studies of public key cryptographic systems based on number theoretical problems led to some reinvestigation of quadratic congruential generators. The simplest example uses the rule

```
n → Mod[n^2, m]
```

It was shown that for $m = p\,q$ with p and q prime the sequence `Mod[n, 2]` was in a sense as difficult to predict as the number m is to factor (see page 1090). But in practice, the period of the generator in such cases is usually too short to be useful. In addition, there has been the practical problem that if n is stored on a computer as a 32-bit number, then n^2 can be 64 bits long, and so cannot be stored in the same way. In general, the period divides `CarmichaelLambda[CarmichaelLambda[m]]`. When m is a prime, this implies that the period can then be as long as $(m - 3)/2$. The largest m less than 2^{16} for which this is true is 65063, and the sequence generated in this case appears to be fairly random.

Cellular automaton generators. I invented the rule 30 cellular automaton random number generator in 1985. Since that time the generator has become quite widely used for a variety of applications. Essentially all the other generators discussed here have certain linearity properties which

allow for fairly complete analysis using traditional mathematical methods. Rule 30 has no such properties. Empirical studies, however, suggest that the repetition period, for example, is about $2^{0.63\,n}$, where n is the number of cells (see page 260). Note that rule 45 can be used as an alternative to rule 30. It has a somewhat longer period, but does not mix up nearby initial conditions as quickly as rule 30. (See also page 603.)

■ **Unequal probabilities.** Given a sequence a of n equally probable 0's and 1's, the following generates a single 0 or 1 with probabilities approximating $\{1-p, p\}$ to n digits:

```
Fold[{BitAnd, BitOr}[[1 + First[#2]]][#1, Last[#2]] &, 0,
  Reverse[Transpose[{First[RealDigits[p, 2, n, -1]], a}]]]
```

This can be generalized to allow a whole sequence to be generated with as little as an average of two input digits being used for each output digit.

■ **Page 323 · Sources of repeatable randomness.** In using repeatability to test for intrinsic randomness generation, one must avoid systems in which there is essentially some kind of static randomness in the environment. Sources of this include the profile of a rough solid surface, or the detailed patterns of grains inside a solid.

■ **Page 324 · Probabilistic rules.** There appears to be a discrete transition as a function of the size of the perturbations, similar to phase transitions seen in the phenomenon of directed percolation. Note that if one just uses the original cellular automata rules, then with any nonzero probability of reversing the colors of cells, the patterns will be essentially destroyed. With more complicated cellular automaton rules, one can get behavior closer to the continuous cellular automata shown here. (See also page 591.)

■ **Page 325 · Noisy cellular automata.** In correspondence with electronics, the continuous cellular automata used here can be thought of as analog models for digital cellular automata. The specific form of the continuous generalization of the modulo 2 function used is

```
λ[x_] := Exp[-10 (x - 1)^2] + Exp[-10 (x - 3)^2]
```

Each cell in the system is then updated according to $\lambda[a + c]$ for rule 90, and $\lambda[a + b + c + b\,c]$ for rule 30. Perturbations of size δ are then added using $v + \text{Sign}[v - 1/2]\,\text{Random}[]\,\delta$.

Note that the basic approach used here can be extended to allow discrete cellular automata to be approximated by partial differential equations where not only color but also space and time are continuous. (Compare page 464.)

■ **Page 326 · Repeatably random experiments.** Over the years, I have asked many experimental scientists about repeatability in seemingly random data, and in almost all cases they have told me that they have never looked for such a thing. But in a few cases they say that in fact on thinking about it they remember various forms of repeatability.

Examples where I have seen evidence of repeatable randomness as a function of time in published experimental data include temperature differences in thermal convection in closed cells of liquid helium, reaction rates in oxidation of carbon monoxide on catalytic surfaces, and output voltages from firings of excited single nerve cells. Typically there are quite long periods of time where the behavior is rather accurately repeatable—even though it may wiggle tens or hundreds in a seemingly random way—interspersed with jumps of some kind. In most cases the only credible models seem to be ones based on intrinsic randomness generation. But insofar as there is any definite model, it is inevitable that looking in sufficient detail at sufficiently many components of the system will reveal regularities associated with the underlying mechanism.

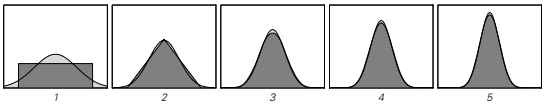
The Phenomenon of Continuity

■ **Discreteness in computer programs.** The reason for discreteness in computer programs is that the only real way we know how to construct such programs is using discrete logical structures. The data that is manipulated by programs can be continuous, as can the elements of their rules. But at some level one always gives discrete symbolic descriptions of the logical structure of programs. And it is then certainly more consistent to make both data and programs involve only discrete elements. In Chapter 12 I will argue that this approach is not only convenient, but also necessary if we are to represent our computations using processes that can actually occur in nature.

■ **Central Limit Theorem.** Averages of large collections of random numbers tend to follow a Gaussian or normal distribution in which the probability of getting value x is

```
Exp[-(x - μ)^2/(2 σ^2)]/(Sqrt[2 π] σ)
```

The mean μ and standard deviation σ are determined by properties of the random numbers, but the form of the distribution is always the same. The only conditions are that the random numbers should be statistically independent, and that their distribution should have bounded variance, so that, for example, the probability for very large numbers is rapidly damped. (The limit of an infinite collection of numbers gives $\sigma \to 0$ in accordance with the law of large numbers.) The pictures at the top of the next page show how averages of successively larger collections of uniformly distributed numbers converge to a Gaussian distribution.



The Central Limit Theorem leads to a self-similarity property for the Gaussian distribution: if one takes n numbers that follow Gaussian distributions, then their average should also follow a Gaussian distribution, though with a standard deviation that is $1/\sqrt{n}$ times smaller.

▪ **History.** That averages of random numbers follow bell-shaped distributions was known in the late 1600s. The formula for the Gaussian distribution was derived by Abraham de Moivre around 1733 in connection with theoretical studies of gambling. In the late 1700s Pierre-Simon Laplace did this again to predict the distribution of comet orbits, and showed that the same results would be obtained for other underlying distributions. Carl Friedrich Gauss made connections to the distribution of observational errors, and the relevance of the Gaussian distribution to biological and social systems was noted. Progressively more general proofs of the Central Limit Theorem were given from the early 1800s to the 1930s. Many natural systems were found to exhibit Gaussian distributions—a typical example being height distributions for humans. (Weight distributions are however closer to lognormal; compare page 1003.) And when statistical methods such as analysis of variance became established in the early 1900s it became increasingly common to assume underlying Gaussian distributions. (Gaussian distributions were also found in statistical mechanics in the late 1800s.)

▪ **Related results.** Gaussian distributions arise when large numbers of random variables get added together. If instead such variables (say probabilities) get multiplied together what arises is the lognormal distribution

```
Exp[-(Log[x] - μ)^2/(2 σ^2)]/(Sqrt[2 π] x σ)
```

For a wide range of underlying distributions the extreme values in large collections of random variables follow the Fisher-Tippett distribution

```
Exp[(x - μ)/β] Exp[-Exp[(x - μ)/β]]/β
```

related to the Weibull distribution used in reliability analysis.

For large symmetric matrices with random entries following a distribution with mean 0 and bounded variance the density of normalized eigenvalues tends to Wigner's semicircle law

```
2 Sqrt[1 - x^2] UnitStep[1 - x^2]/π
```

while the distribution of spacings between tends to

```
1/2 (π x) Exp[1/4 (-π) x^2]
```

The distribution of largest eigenvalues can often be expressed in terms of Painlevé functions.

(See also $1/f$ noise on page 969.)

▪ **Page 328 · Random walks.** In one dimension, a random walk with t steps of length 1 starting at position 0 can be generated from

```
NestList[# + (-1)^Random[Integer] &, 0, t]
```

or equivalently

```
FoldList[Plus, 0, Table[(-1)^Random[Integer], {t}]]
```

A generalization to d dimensions is then

```
FoldList[Plus, Table[0, {d}], Table[RotateLeft[PadLeft[
  {(-1)^Random[Integer]}, d], Random[Integer, d - 1]], {t}]]
```

A fundamental property of random walks is that after t steps the root mean square displacement from the starting position is proportional to $\sqrt{t}$. In general, the probability distribution for the displacement of a particle that executes a random walk is

```
With[{σ = 1}, (d/(2 π σ t))^(d/2) Exp[-d r^2/(2 σ t)]]
```

The same results are obtained, with a different value of σ, for other random microscopic rules, so long as the variance of the distribution of step lengths is bounded (as in the Central Limit Theorem).

As mentioned on page 1082, the frequency spectrum `Abs[Fourier[list]]^2` for a 1D random walk goes like $1/\omega^2$.

The character of random walks changes somewhat in different numbers of dimensions. For example, in 1D and 2D, there is probability 1 that a particle will eventually return to its starting point. But in 3D, this probability (on a simple cubic lattice) drops to about 0.341, and in d dimensions the probability falls roughly like `1/(2 d)`. After a large number of steps t, the number of distinct positions visited will be proportional to t, at least above 2 dimensions (in 2D, it is proportional to `t/Log[t]` and in 1D $\sqrt{t}$). Note that the outer boundaries of patterns like those on page 330 formed by n random walks tend to become rougher when t is much larger than `Log[n]`.

To make a random walk on a lattice with k directions in two dimensions, one can set up

```
e = Table[{Cos[2 π s/k], Sin[2 π s/k]}, {s, 0, k - 1}]
```

then use

```
FoldList[Plus, {0, 0}, Table[e[[Random[Integer, {1, k}]]], {t}]]
```

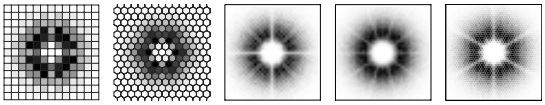
It turns out that on any regular lattice, in any number of dimensions, the average behavior of a random walk is always isotropic. As discussed in the note below, this can be viewed as a consequence of the fact that the probability distribution in a random walk depends only on

```
Sum[Outer[Times, e[[s]], e[[s]]], {s, Length[e]}]
```

and not on products of more of the `e[[s]]`.

There are nevertheless some properties of random walks that are not isotropic. The picture below, for example, shows the

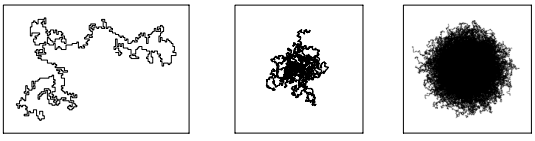
so-called extreme value distribution of positions furthest from the origin reached after 10 steps and 100 steps by random walks on various lattices.



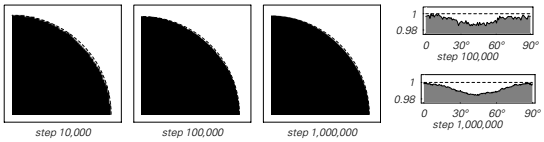
In the pictures in the main text, all particles start out at a particular position, and progressively spread out from there. But in general, one can consider sources that emit new particles every step, or absorbers and reflectors of particles. The average distribution of particles is given in general by the diffusion equation shown on page 163. The solutions to this equation are always smooth and continuous.

A physical example of an approximation to a random walk is the spreading of ink on blotting paper.

■ **Self-avoiding walks.** Any walk where the probabilities for a given step depend only on a fixed number of preceding steps gives the same kind of limiting Gaussian distribution. But imposing the constraint that a walk must always avoid anywhere it has been before (as for example in an idealized polymer molecule) leads to correlations over arbitrary times. If one adds individual steps at random then in 2D one typically gets stuck after perhaps a few tens of steps. But tricks are known for generating long self-avoiding walks by combining shorter walks or successively pivoting pieces starting with a simple line. The pictures below show some 1000-step examples. They look in many ways similar to ordinary random walks, but their limiting distribution is no longer strictly Gaussian, and their root mean square displacement after t steps varies like $t^{3/4}$. (In $d \le 4$ dimensions the exponent is close to the Flory mean field theory value $3/(2+d)$; for $d > 4$ the results are the same as without self-avoidance.)



■ **Page 331 · Basic aggregation model.** This model appears to have first been described by Murray Eden in 1961 as a way of studying biological growth, and was simulated by him on a computer for clusters up to about 32,000 cells. By the mid-1980s clusters with a billion cells had been grown, and a very surprising slight anisotropy had been observed. The pictures below show which cells occur in more than 10% of 1000 randomly grown clusters. There is a 2% or so anisotropy that appears to remain essentially fixed for clusters above perhaps a million cells, tucking them in along the diagonal directions. The width of the region of roughness on the surface of each cluster varies with the radius of the cluster approximately like $r^{1/3}$. The most extensive use of the model in practice has been for studying tumor growth: currently a typical tumor at detection contains about a billion cells, and it is important to predict what protrusions there will be that can break off and form additional tumors elsewhere.



■ **Implementation.** One way to represent a cluster is by giving a list of the coordinates at which each black cell occurs. Then starting with a single black cell at the origin, represented by *{{0, 0}}*, the cluster can be grown for t steps as follows:

```
AEvolve[t_] := Nest[AStep, {{0, 0}}, t]

AStep[c_] := (If[! MemberQ[c, #], Append[c, #],
     AStep[c]] &)[f[c] + f[{{1, 0}, {0, 1}, {-1, 0}, {0, -1}}]]
f[a_] := a[[Random[Integer, {1, Length[a]}]]]
```

This implementation can easily be extended to any type of lattice and any number of dimensions. Even with various additional optimizations, it is remarkable how much slower it is to grow a cluster with a model that requires external random input than to generate similar patterns with models such as cellular automata that intrinsically generate their own randomness.

The implementation above is a so-called type B Eden model in which one first selects a cell in the cluster, then randomly selects one of its neighbors. One gets extremely similar results with a type A Eden model in which one just randomly selects a cell from all the ones adjacent to the cluster. With a grid of cells set up in advance, each step in this type of Eden model can be achieved with

```
AStep[a_] := ReplacePart[a, 1, (#[[Random[
     Integer, {1, Length[#]}]]] &)[Position[(1 - a) Sign[
     ListConvolve[{{0, 1, 0}, {1, 0, 1}, {0, 1, 0}}, a, {2, 2}]], 1]]]
```

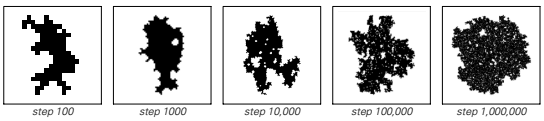
This implementation can readily be extended to generalized aggregation models (see below).

■ **Page 332 · Generalized aggregation models.** One can in general have rules in which new cells can be added only at positions whose neighborhoods match specific templates (compare page 213). There are 32 possible symmetric such rules with just 4 immediate neighbors—of which 16 lead to growth (from any seed), and all seem to yield at least

approximately circular clusters (of varying densities). Without symmetry, all sorts of shapes can be obtained, as in the pictures below. (The rule numbers here follow the scheme on page 927 with offsets *{{-1, 0}, {0, -1}, {0, 1}, {1, 0}}*). Note that even though the underlying rule involves randomness definite geometrical shapes can be produced. An extreme case is rule 2, where only a single neighborhood with a single black cell is allowed, so that growth occurs along a single line.

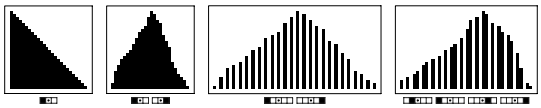
If one puts conditions on where cells can be added one can in principle get clusters where no further growth is possible. This does not seem to happen for rules that involve 4 neighbors, but with 8 neighbors there are cases in which clusters can get fairly large, but end up having no sites where further cells can be added. The pictures below show examples for a rule that allows growth except when there are exactly 1, 3 or 4 neighbors (totalistic constraint 242).

The question of what ultimate forms of behavior can occur with any sequence of random choices, starting from a given configuration with a given rule, is presumably in general undecidable. (It has some immediate relations to tiling problems and to halting problems for non-deterministic Turing machines.) With the rule illustrated above, however, those clusters that do successfully grow exhibit complicated and irregular shapes, but nevertheless eventually seem to take on a roughly circular shape, as in the pictures below.

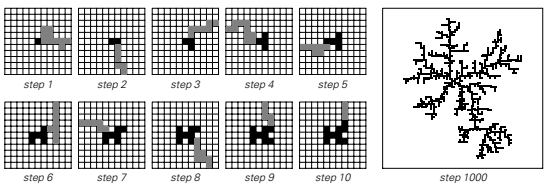


At some level the basic aggregation model of page 331 has a deterministic outcome: after sufficiently many steps every cell will be black. But most generalized aggregation models do not have this property: instead, the form of their internal patterns depends on the sequence of random choices made. Particularly with more than two colors it is however possible to arrange that the internal pattern always ends up being the same, or at least has patches that are the same—essentially by using rules with the confluence property discussed on page 1036.

The pictures below show 1D generalized aggregation systems with various templates. The second one is the analog of the system from page 331.

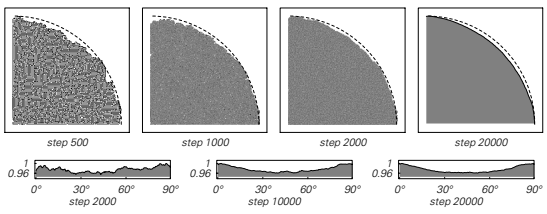


▪ **Page 333 · Diffusion-limited aggregation (DLA).** While many 2D cellular automata produce intricate nested shapes, the aggregation models shown here seem to tend to simple limiting shapes. Most likely there are some generalized aggregation models for which this is not the case. And indeed this phenomenon has been seen in other systems with randomness in their underlying rules. An example studied extensively in the 1980s is diffusion-limited aggregation (DLA). The idea of this model is to add cells to a cluster one at a time, and to determine where a cell will be added by seeing where a random walk that starts far from the cluster first lands on a square adjacent to the cluster. An example of the behavior obtained in this model is shown below:

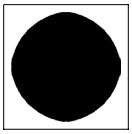


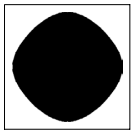
The lack of smooth overall behavior in this case can perhaps be attributed to the global probing of the cluster that is effectively done by each incoming random walk. (See also page 994.)

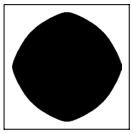
▪ **Page 334 · Code 746.** Much as in the aggregation model above, the pictures below show that there is a slight deviation from perfect circular growth, with an anisotropy that appears to remain roughly fixed at perhaps 4% above a few thousand steps (corresponding to patterns with a few million cells).



■ **Other rules.** The pictures below show patterns generated after 10,000 steps with several rules, starting respectively from rows of 7, 6, 7 and 11 cells (compare pages 177 and 181). The outer boundaries are somewhat smooth, though definitely not circular. In the second rule shown, the interior of the pattern always continues to change; in the others it remains essentially fixed.


code 746


code 29408


code 175850


code 174826

■ **Isotropy.** Any pattern grown from a single cell according to rules that do not distinguish different directions on a lattice must show the same symmetry as the lattice. But we have seen that in fact many rules actually yield almost circular patterns with much higher symmetry. One can characterize the symmetry of a pattern by taking the list v of positions of cells it contains, and looking at tensors of successive ranks n:

```
Apply[Plus,
  Map[Apply[Outer[Times, ##] &, Table[#, {n}]] &, v]]
```

For circular or spherical patterns that are perfectly isotropic in d dimensions these tensors must all be proportional to

```
(d - 2)!! Array[Apply[Times, Map[(1 - Mod[#, 2]) (# - 1)!! &,
  Table[Count[{##}, i], {i, d}]]] &, Table[d, {n}]]/(d + n - 2)!!
```

For odd n this is inevitably true for any lattice with mirror symmetry. But for even n it can fail. For a square lattice, it still nevertheless always holds up to $n = 2$ (so that the analogs of moments of inertia satisfy $\mathcal{I}_{xx} == \mathcal{I}_{yy}$, $\mathcal{I}_{xy} == \mathcal{I}_{yx} == 0$). And for a hexagonal lattice it holds up to $n = 4$. But when $n = 4$ isotropy requires the $\{1, 1, 1, 1\}$ and $\{1, 1, 2, 2\}$ tensor components to have ratio $\beta = 3$—while square symmetry allows these components to have any ratio. (In general there will be more than one component unless the representation of the lattice symmetry group carried by the rank n tensor is irreducible.) In 3D no regular lattice forces isotropy beyond $n = 2$, while in 4D the SO(8) lattice works up to $n = 4$, in 8D the E_8 lattice up to $n = 6$, and in 24D the Leech lattice up to $n = 10$. (Lattices that give dense sphere packings tend to show more isotropy.) Note that isotropy can also be characterized using analogs of multipole moments, obtained in 2D by summing $r_i \, \text{Exp}[i \, n \, \theta_i]$, and in higher dimensions by summing appropriate *SphericalHarmonicY* or *GegenbauerC* functions. For isotropy, only the $n = 0$ moment can be nonzero. On a 2D lattice with m directions, all moments are forced to be zero except when m divides n. (Sums of squares of moments of given order in general provide rotationally invariant measures of anisotropy—equal to pair correlations weighted with *LegendreP* or *GegenbauerC* functions.)

Even though it is not inevitable from lattice symmetry, one might think that if there is some kind of effective randomness in the underlying rules then sufficiently large patterns would still often show some sort of average isotropy. And at least in the case of ordinary random walks, they do, so that for example, the ratio averaged over all possible walks of $n = 4$ tensor components after t steps on a square lattice is $\beta = 3 + 2/(t - 1)$, converging to the isotropic value 3, and the ratio of $n = 6$ components is $5 - 4/(t - 1) + 32/(3\,t - 4)$. For the aggregation model of page 331, β also decreases with t, reaching 4 around $t = 10$, but now its asymptotic value is around 3.07.

In continuous systems such as partial differential equations, isotropy requires that coordinates in effect appear only in ∇. In most finite difference approximations, there is presumably isotropy in the end, but the rates of convergence are almost inevitably rather different in different directions relative to the lattice.

■ **Page 336 · Domains.** Some of the effective rules for interfaces between black and white domains are easy to state. Given a flat interface, the layer of cells immediately on either side of this interface behaves like the rule 150 1D cellular automaton. On an infinitely long interface, protrusions of cells with one color into a domain of the opposite color get progressively smaller, eventually leaving only a certain pattern of cells in the layer immediately on one side of the interface. 90° corners in an otherwise flat interface effectively act like reflective boundary conditions for the layer of cells on top of the interface.

The phenomenon of domains illustrated here is also found in various 2D cellular automata with 4-neighbor rather than 8-neighbor rules. One example is totalistic code 52, which is a direct analog in the 4-neighbor case of the rule illustrated here. Other examples are outer totalistic codes 111, 293, 295 and 920. The domain boundaries in these cases, however, are not as clear as for the 8-neighbor totalistic rule with code 976 that is shown here.

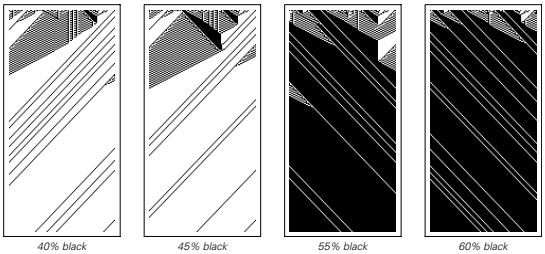
■ **Spinodal decomposition.** The separation into progressively larger black and white regions seen in the cellular automata shown here is reminiscent of the phenomena that occur for example in the separation of randomly mixed oil and water. Various continuous models of such processes have been proposed, notably the Cahn-Hilliard equation from 1958. One feature often found is that the average radius of "droplets" increases with time roughly like $t^{1/3}$.

Origins of Discreteness

■ **Page 339 · 1D transitions.** There are no examples of the phenomenon shown here among the 256 rules with two possible colors and depending only on nearest neighbors. Among the 4,294,967,296 rules that depend on next-nearest neighbors, there are a handful of examples, including rules with numbers 4196304428, 4262364716, 4268278316 and 4266296876. The behavior obtained with the first of these rules is shown below. An example that depends on three neighbors on each side was discovered by Peter Gacs, Georgii Kurdyumov and Leonid Levin in 1978, following work on how reliable electronic circuits can be built from unreliable components by Andrei Toom:

```
{a1_, a2_, a3_, a4_, a5_, a6_, a7_} →
  If[If[a4 == 1, a1 + a3 + a4, a4 + a5 + a7] ≥ 2, 1, 0]
```

The 4-color rule shown in the text is probably the clearest example available in one dimension. It has rule number 2948697645239957498148900977948124938 24.



■ **Page 340 · 2D transitions.** The simplest symmetrical rules (such as 4-neighbor totalistic code 56) which make the new color of a cell be the same as the majority of the cells in its neighborhood do not exhibit the discrete transition phenomenon, but instead lead to fixed regions of black and white. The 4-neighbor rule with totalistic code 52 can be used as an alternative to the second rule shown here. A probabilistic version of the first rule shown here was discussed by Andrei Toom in 1980.

■ **Phase transitions.** The discrete transitions shown in cellular automata in this section are examples of general phenomena known in physics as phase transitions. A phase transition can be defined as any discontinuous change that occurs in a system with a large number of components when a parameter associated with that system is varied. (Some physicists might argue for a somewhat narrower definition that allows only discontinuities in the so-called partition function of equilibrium statistical mechanics, but for many of the most interesting applications, the definition I use is the appropriate one.) Standard examples of phase transitions include boiling, melting, sublimation (solids such as dry ice turning into gases), loss of magnetization when a ferromagnet is heated, alignment of molecules in liquid crystals above a certain electric field (the basis for liquid crystal displays), and the onset of superconductivity and superfluidity at low temperatures.

It is conventional to distinguish two kinds of phase transitions, often called first-order and higher-order. First-order transitions occur when a system has two possible states, such as liquid and gas, and as a parameter is varied, which of these states is the stable one changes. Boiling and melting are both examples of first-order transitions, as is the phenomenon shown in the cellular automaton in the main text. Note that one feature of first-order transitions is that as soon as the transition is passed, the whole system always switches completely from one state to the other.

Higher-order transitions are in a sense more gradual. On one side of the transition, a system is typically completely disordered. But when the transition is passed, the system does not immediately become completely ordered. Instead, its order increases gradually from zero as the parameter is varied. Typically the presence of order is signalled by the breaking of some kind of symmetry—say of rotational symmetry by the spontaneous selection of a preferred direction.

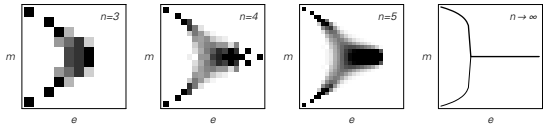
■ **The Ising model.** The 2D Ising model is a prototypical example of a system with a higher-order phase transition. Introduced by Wilhelm Lenz in 1920 as an idealization of ferromagnetic materials (and studied by Ernst Ising) it involves a square array *s* of spins, each either up or down (+1 or -1), corresponding to two orientations for magnetic moments of atoms. The magnetic energy of the system is taken to be

```
e[s_] := -1/2 Apply[Plus, s ListConvolve[
          {{0, 1, 0}, {1, 0, 1}, {0, 1, 0}}, s, 2], {0, 1}]
```

so that each pair of adjacent spins contributes -1 when they are parallel and +1 when they are not. The overall magnetization of the system is given by `m[s_] := Apply[Plus, s, {0, 1}]`.

In physical ferromagnetic materials what is observed is that at high temperature, corresponding to high internal energy, there is no overall magnetization. But when the temperature goes below a critical value, spins tend to line up, and an overall magnetization spontaneously develops. In the context of the 2D Ising model this phenomenon is associated with the fact that those configurations of a large array of spins that have high total energy are overwhelmingly likely to have near zero overall magnetization, while those that have low

total energy are overwhelmingly likely to have nonzero overall magnetization. For an $n \times n$ array s of spins there are a total of 2^{n^2} possible configurations. The pictures below show the results of picking all configurations with a given energy $e[s]$ (cyclic boundary conditions are assumed) and then working out their distribution of magnetization values $m[s]$. Even for small n the pictures demonstrate that for large $e[s]$ the magnetization $m[s]$ is likely to be close to zero, but for smaller $e[s]$ two branches approaching +1 and -1 appear. In the limit $n \to \infty$ the distribution of magnetization values becomes sharp, and a definite discontinuous phase transition is observed.



Following the work of Lars Onsager around 1944, it turns out that an exact solution in terms of traditional mathematical functions can be found in this case. (This seems to be true only in 2D, and not in 3D or higher.) Almost all spin configurations with $e[s] > -\sqrt{2}$ (where here and below all quantities are divided by the total number of spins, so that $-2 \le e[s] \le 2$ and $-1 \le m[s] \le +1$) yield $m[s] == 0$. But for smaller $e[s]$ one can show that

```
Abs[m[s]] == (1 - Sinh[2 β]^-4)^1/8
```

where β can be deduced from

```
e[s] == -(Coth[2 β] (1 + 2 EllipticK[4 Sech[2 β]^2 Tanh[2 β]^2] (-1 + 2 Tanh[2 β]^2))/π))
```

This implies that just below the critical point $e_0 = -\sqrt{2}$ (which corresponds to $\beta = Log[1 + \sqrt{2}]/2$) $Abs[m] \sim (e_0 - e)^{1/8}$, where here 1/8 is a so-called critical exponent. (Another analytical result is that for $e \sim e_0$ correlations between pairs of spins can be expressed in terms of Painlevé functions.)

Despite its directness, the approach above of considering sets of configurations with specific energies $e[s]$ is not how the Ising model has usually been studied. Instead, what has normally been done is to take the array of spins to be in thermal equilibrium with a heat bath, so that, following standard statistical mechanics, each possible spin configuration occurs with probability $Exp[-\beta e[s]]$, where β is inverse temperature. It nevertheless turns out that in the limit $n \to \infty$ this so-called canonical ensemble approach yields the same results for most quantities as the microcanonical approach that I have used; β simply appears as a parameter, as in the formulas above.

About actual spin systems evolving in time the Ising model itself does not make any statement. But whenever the evolution is ergodic, so that all states of a given energy are visited with equal frequency, the average behavior obtained will at least eventually correspond to the average over all states discussed above.

In Monte Carlo studies of the Ising model one normally tries to sample states with appropriate probabilities by randomly flipping spins according to a procedure that can be thought of as emulating interaction with a heat bath. But in most actual physical spin systems it seems unlikely that there will be so much continual interaction with the environment. And from my discussion of intrinsic randomness generation it should come as no surprise that even a completely deterministic rule for the evolution of spins can make the system visit possible states in an effectively random way.

Among the simplest possible types of rules all those that conserve the energy $e[s]$ turn out to have behavior that is too simple and regular. And indeed, of the 4096 symmetric 5-neighbor rules, only identity and complement conserve $e[s]$. Of the 2^{32} general 5-neighbor rules 34 conserve $e[s]$—but all have only very simple behavior. (Compositions of several such rules can nevertheless yield complex behavior. Note that as indicated on page 1022, 34 of the 256 elementary 1D rules conserve the analog of $e[s]$.) Of the 262,144 9-neighbor outer totalistic rules the only ones that conserve $e[s]$ are identity and complement. But among all 2^{512} 9-neighbor rules, there are undoubtedly examples that show effectively random behavior. One marginally more complicated case effectively involving 13 neighbors is

```
IsingEvolve[list_, t_Integer] :=
  First[Nest[IsingStep, {list, Mask[list]}, t]]
IsingStep[{a_, mask_}] := {MapThread[
    If[#2 == 2 && #3 == 1, 1 - #1, #1] &, {a, ListConvolve[
      {{0, 1, 0}, {1, 0, 1}, {0, 1, 0}}, a, 2], mask}, 2], 1 - mask}
```

where

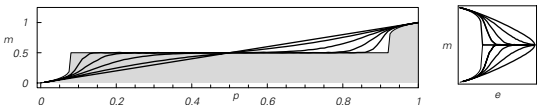
```
Mask[list_] := Array[Mod[#1 + #2, 2] &, Dimensions[list]]
```

is set up so that alternating checkerboards of cells are updated on successive steps.

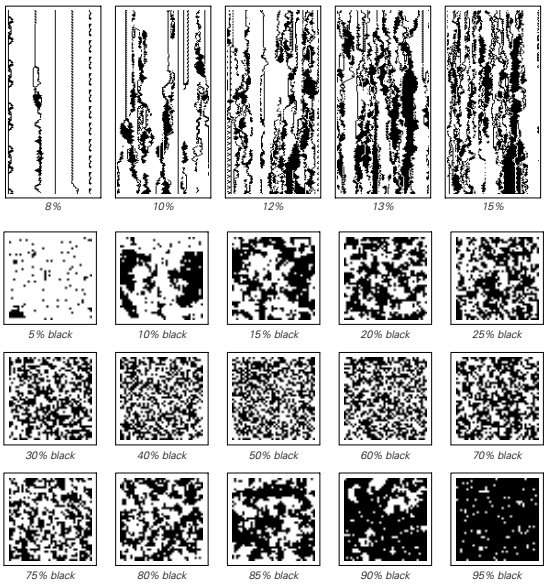
One can see a phase transition in this system by looking at the dependence of behavior on conserved total energy $e[s]$. If there are no correlations between spins, and a fraction p of them are +1, then $m[s] == p$ and $e[s] == -2(1 - 2p)^2$. And since the evolution conserves $e[s]$ changing the initial value of p allows one to sample different total energies. But since the evolution does not conserve $m[s]$ the average of this after many steps can be expected to be typical of all possible states of given $e[s]$.

The pictures at the top of the next page show the values of $m[s]$ (densities of +1 cells) after 0, 10, 100 and 1000 steps for a 500×500 system as a function of the initial values of $m[s]$ and $e[s]$. Also shown is the result expected for an infinite system at infinite time. (The slow approach to this limit can

be viewed as being a consequence of smallness of finite size scaling exponents in Ising-like systems.)



The phase transition in the Ising model is associated with a lack of smoothness in the dependence of the final m value on e or the initial value p of m in limiting cases of the pictures above. The transition occurs at $e = -\sqrt{2}$, corresponding to $p = (1 \pm 2^{-1/4})/2$. The pictures show typical configurations generated after 1000 steps from various initial densities, as well as slices through their evolution.



And what one sees at least roughly is that right around the phase transition there are patches of black and white of all sizes, forming an approximately nested random pattern. (See also pages 989 and 1149.)

▪ **General features of phase transitions.** To reproduce the Ising model, a cellular automaton must have several special properties. In addition to conserving energy, its evolution must be reversible in the sense discussed on page 435. And with the constraint of reversibility, it turns out that it is impossible to get a non-trivial phase transition in any 1D system with the kind of short-range interactions that exist in a cellular automaton. But in systems whose evolution is not reversible, it is possible for phase transitions to occur in 1D, as the examples in the main text show.

One point to notice is that the sharp change which characterizes any phase transition can only be a true discontinuity in the limit of an infinitely large system. In the case of the system on page 339, for example, it is possible to find special configurations with a finite total number of cells which lead to behavior opposite to what one expects purely on the basis of their initial density of black cells. When the total number of cells increases, however, the fraction of such configurations rapidly decreases, and in the infinite size limit, there are no such configurations, and a truly discontinuous transition occurs exactly at density 1/2.

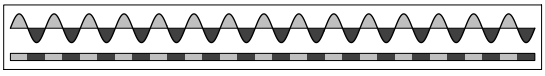
The discrete nature of phase transitions was at one time often explained as a consequence of changes in the symmetry of a system. The idea is that symmetry is either present or absent, and there is no continuous variation of level of symmetry possible. Thus, for example, above the transition, the Ising model treats up and down spins exactly the same. But below the transition, it effectively makes a choice of one spin direction or the other. Similarly, when a liquid freezes into a crystalline solid, it effectively makes a choice about the alignment of the crystal in space. But in boiling, as well as in a number of model examples, there is no obvious change of symmetry. And from studying phase transitions in cellular automata, it does not seem that an interpretation in terms of symmetry is particularly useful.

A common feature of phase transitions is that right at the transition point, there is competition between both phases, and some kind of nested structure is typically formed, as discussed on page 273 and above. The overall form and fractal dimension of this nested structure is typically independent of small-scale features of the system, making it fairly universal, and amenable to analysis using the renormalization group approach (see page 955).

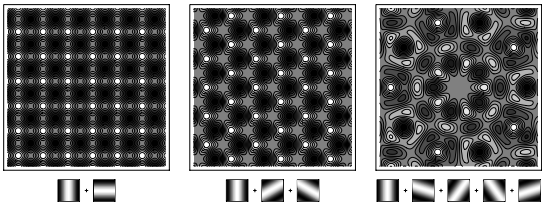
▪ **Percolation.** A simple example of a phase transition studied extensively since the 1950s involves taking a square lattice and filling in at random a certain density of black cells. In the limit of infinite size, there is a discrete transition at a density of about 0.592746, with zero probability below the transition to find a connected "percolating" cluster of black cells spanning the lattice, and unit probability above. (For a triangular lattice the critical density is exactly 1/2.) One can also study directed percolation in which one takes account of the connectivity of cells only in one direction on the lattice. (Compare the probabilistic cellular automata on pages 325 and 591. Note that the evolution of such systems is also analogous to the process of applying transfer matrices in studies of spin systems like Ising models.)

■ **Page 341 · Rate equations.** In standard chemical kinetics one assumes that molecules are uniformly distributed in space, so that the rates for particular reactions are proportional to the products of the densities of the molecules that react in them. Conditions for equilibrium where rates balance thus tend to be polynomial equations for densities—with discontinuous jumps in solutions sometimes occurring as parameters are changed. Analogous equations arise in probabilistic approximations to systems like cellular automata, as on page 953. But here—as well as in fast chemical reactions—correlations in spatial arrangements of elements tend to be important, invalidating simple probabilistic approaches. (For the cellular automaton on page 339 the simple condition for equilibrium is $p == p^2 (3 - 2 p)$, which correctly implies that 0, 1/2 and 1 are possible equilibrium densities.)

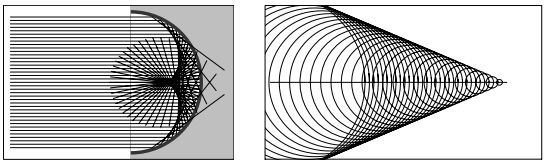
■ **Discreteness in space.** Many systems with continuous underlying rules generate discrete cellular structures in space. One common mechanism is for a wave of a definite wavelength to form (see page 988), and then for some feature of each cycle of this wave to be picked out, as in the picture below. In Chladni figures of sand on vibrating plates and in cloud streets in the atmosphere what happens is that material collects at points of zero displacement. And when a stream of water breaks up into discrete drops what happens is that oscillation minima yield necks that break.



Superpositions of waves at different angles can lead to various 2D cellular structures, as in the pictures below (compare page 1078).



Various forms of focusing and accumulation can also lead to discreteness in continuous systems. The first picture below shows a caustic or catastrophe in which a continuous distribution of light rays are focused by a circular reflector onto a discrete line with a cusp. The second picture shows a shock wave produced by an accumulation of circular waves emanating from a moving object—as seen in wakes of ships, sonic booms from supersonic aircraft, and Cerenkov light from fast-moving charged particles.



The Problem of Satisfying Constraints

■ **Rules versus constraints.** See page 940.

■ **NP completeness.** Finding 2D patterns that satisfy the constraints in the previous section is in general a so-called NP-complete problem. And this means that no known algorithm can be expected to solve this problem exactly for a size n array (say with given boundaries) in much less than 2^n steps (see page 1145). The same is true even if one allows a small fraction of squares to violate the constraints. However, the 1D version of the problem is not NP-complete, and in fact there is a specific rather efficient algorithm described on page 954 for solving it. Nevertheless, the procedures discussed in this section do not manage to make use of such specific algorithms, and in fact typically show little difference between problems that are and are not formally NP-complete.

■ **Page 343 · Distribution.** The distribution shown here rapidly approaches a Gaussian. (Note that in a 5×5 array, there are 10 interior squares that are subject to the constraints, while in a 10×10 array there are 65.) Very similar results seem to be obtained for constraints in a wide range of discrete systems.

■ **Page 346 · Implementation.** The number of squares violating the constraint used here is given by

```
Cost[list_] := Apply[Plus, Abs[list - RotateLeft[list]]]
```

When applied to all possible patterns, this function yields a distribution with Gaussian tails, but with a sharp point in the middle. Successive steps in the iterative procedure used on this page are given by

```
Move[list_] := (If[Cost[#] < Cost[list], #, list] &)[
    MapAt[1 - # &, list, Random[Integer, {1, Length[list]}]]]
```

while those in the procedure on page 347 have $\leq$ in place of $<$. The third curve shown on page 346 is obtained from

```
Table[Cost[IntegerDigits[i, 2, n]], {i, 0, 2^n - 1}]
```

There is no single ordering that makes all states which can be reached by changing a single square be adjacent. However, the ordering defined by *GrayCode* from page 901 does do this for one particular sequence of single square changes. The resulting curve is very similar to what is already shown.

■ **Page 347 · Iterative improvement.** The borders of the regions of black and white in the picture shown here essentially

follow random walks and annihilate in pairs so that their number decreases with time like $1/\sqrt{t}$. In 2D the regions are more complicated and there is no such simple behavior. Indeed starting from a particular state it is for example not clear whether it is ever possible to reach all other states.

■ **Gradient descent.** A standard method for finding a minimum in a smooth function `f[x]` is to use

```
FixedPoint[# - a f'[#] &, x0]
```

If there are local minima, then which one is reached will depend on the starting point x_0. It will not necessarily be the one closest to x_0 because of potentially complicated overshooting effects associated with the step size a. Newton's method for finding zeros of `f[x]` is related and is given by

```
FixedPoint[# - f[#]/f'[#] &, x0]
```

■ **Combinatorial optimization.** The problem of coming as close as possible to satisfying constraints in an arrangement of black and white squares is a simple example of a combinatorial optimization problem. In general, such problems involve minimization of a quantity that is determined by the arrangement of some set of discrete elements. A typical example is finding a placement of components in a 2D circuit so that the total length of wire necessary to the connect these components is minimized (related to the so-called travelling salesman problem). In using iterative procedures to solve combinatorial optimization problems, one issue is what kind of changes should be made at each step. In the main text we considered changing just one square at a time. But one can also change larger numbers of squares, or, for example, interchange whole blocks of squares. In general, the larger the changes made, the faster one can potentially approach a minimum, but the greater the chance is of overshooting. In the main text, we assumed that at each step we should always move closer to the minimum, or at least not get further away. But in trying to get over the kind of bumps shown in the third curve on page 346 it is sometimes better also to allow some probability of moving away from the minimum at a particular step. One approach is simulated annealing, in which one starts with this probability being large, and progressively decreases it. The notion is that at the beginning, one wants to move easily over the coarse features of a jagged curve, but then later home in on details. If the curve has a nested form, which appears to be the case in some combinatorial optimization problems, then this scheme can be expected to be at least somewhat effective. For the problems considered in the main text, simulated annealing provides some improvement but not much.

■ **Biologically motivated schemes.** The process of biological evolution by natural selection can be thought of as an iterative procedure for optimization. Usually, however, what is being optimized is some aspect of the form or behavior of an organism, which represents a very complicated constraint on the underlying genetic material. (It is as if one is defining constraints on the initial conditions for a cellular automaton by looking at the pattern generated by the cellular automaton after a long time.) But the strategies of biological evolution can also be used in trying to satisfy simpler constraints. Two of the most important strategies are maintaining a whole population of individuals, not just the single best result so far, and using sex to produce large-scale mixing. But once again, while these strategies may in some cases lead to greater efficiency, they do not usually lead to qualitative differences. (See also page 1105.)

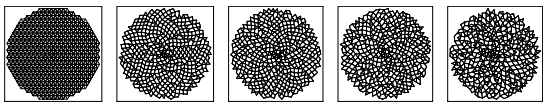
■ **History.** Work on combinatorial optimization started in earnest in the late 1950s, but by the time NP completeness was discovered in 1971 (see page 1143) it had become clear that finding exact solutions would be very difficult. Approximate methods tended to be constructed for specific problems. But in the early 1980s, simulated annealing was suggested by Scott Kirkpatrick and others as one of the first potentially general approaches. And starting in the mid-1980s, extensive work was done on biologically motivated so-called genetic algorithms, which had been advocated by John Holland since the 1960s. Progress in combinatorial optimization is however often difficult to recognize, because there are almost no general results, and results that are quoted are often sensitive to details of the problems studied and the computer implementations used.

■ **Page 349 · 2D cellular automata.** The rule numbers are specified as on page 927.

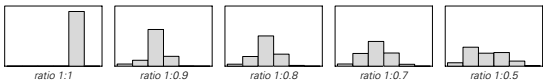
■ **Page 349 · Circle packings.** Hexagonal packing of equal circles has been known since early antiquity (e.g. the fourth picture on page 43). It fills a fraction $\pi/\sqrt{12} \simeq 0.91$ of area—which was proved maximal for periodic packings by Carl Friedrich Gauss in 1831 and for any packing by Axel Thue in 1910 and László Fejes Tóth in 1940. Much has been done to study densest packings of limited numbers of circles into various shapes, as well as onto surfaces of spheres (as in golf balls, pollen grains or radiolarians). Typically it has been found that with enough circles, patches of hexagonal packing always tend to form. (See page 987.)

For circles of unequal sizes rather little has been done. A procedure analogous to the one on page 350 was introduced by Charles Bennett in 1971 for 3D spheres (relevant for binary alloys). The picture below shows the

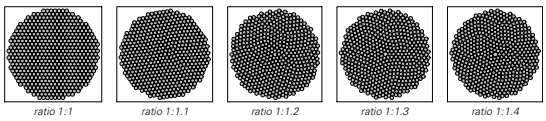
network of contacts between circles in the cases from page 350. Note that with the procedure used, each new circle added must immediately touch two existing ones, though subsequently it may get touched by varying numbers of other circles.



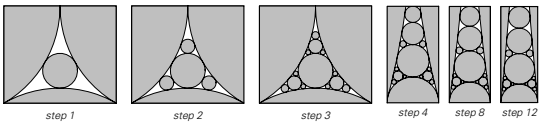
The distribution of numbers of circles that touch a given circle changes with the ratio of circle sizes, as in the picture below. The total filling fraction seems to vary fairly smoothly with this ratio, though I would not be surprised if some small-scale jumps were present.



Note that even a single circle of different size in the center can have a large-scale effect on the results of the procedure, as illustrated in the pictures below.



Finding densest packings of n circles is in general like solving quadratic programming problems with about n^2 constraints. But at least for many size ratios I suspect that the final result will simply involve each kind of circle forming a separated hexagonally-packed region. This will not happen, however, for size ratios $\leq 2/\sqrt{3} - 1 \simeq 0.15$, since then the small circles can fit into the interstices of an ordinary hexagonal pattern, yielding a filling fraction $1/18\,(17\sqrt{13} - 24)\,\pi \simeq 0.95$. The picture below shows what happens if one repeatedly inserts circles to form a so-called Apollonian packing derived from the problem studied by Apollonius of finding a circle that touches three others. At step t, 3^{t-1} circles are added for each original circle, and the network of tangencies among circles is exactly example (a) from page 509. Most of the circles added at a given step are not the same size, however, making the overall geometry not straightforwardly nested. (The total numbers of different sizes of circles for the first few steps are $\{2, 3, 5, 10, 24, 63, 178, 521\}$. At step 3, for example, the new circles have radii $(25 - 12\sqrt{3})/193$ and $(19 - 6\sqrt{3})/253$. In general, the radius of a circle inscribed between three other touching circles that have radii p, q, r is $p\,q\,r/(p\,q + p\,r + q\,r + 2\,\mathrm{Sqrt}[p\,q\,r\,(p + q + r)])$.) In the limit of an infinite number of steps the filling fraction tends to 1, while the region left unfilled has a fractal dimension of about 1.3057.



To achieve filling fraction 1 requires arbitrarily small circles, but there are many different arrangements of circles that will work, some not even close to nested. When actual granular materials are formed by crushing, there is probably some tendency to generate smaller pieces by following essentially substitution system rules, and the result may be a nested distribution of sizes that allows an Apollonian-like packing.

Apollonian packings turn out to correspond to limit sets invariant under groups of rational transformations in the complex plane. Note that as on page 1007 packings can be constructed in which the sizes of circles vary smoothly with position according to a harmonic function.

▪ **Sphere packings.** The 3D face-centered cubic (fcc) packing shown in the main text has presumably been known since antiquity, and has been used extensively for packing fruit, cannon balls, etc. It fills space with a density $\pi/\sqrt{18} \simeq 0.74$, which Johannes Kepler suggested in 1609 might be the maximum possible. This was proved for periodic packings by Carl Friedrich Gauss in 1831, and for any packing by Thomas Hales in 1998. (By offsetting successive layers hexagonal close packing (hcp) can be obtained; this has the same density as fcc, but has a trapezoid-rhombic dodecahedron Voronoi diagram—see note below and page 929—rather than an ordinary rhombic dodecahedron.)

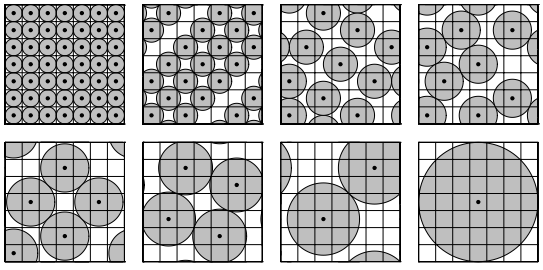
Random packings of spheres typically have densities around 0.64 (compared to 0.74 for fcc). Many of their large pores appear to be associated with poor packing of tetrahedral clusters of 4 spheres. (Note that individual such clusters—as well as for example 13-sphere approximate icosahedra—represent locally dense packings.)

It is common for shaking to cause granular materials (such as coffee or sand grains) to settle and pack at least a few percent better. Larger objects normally come to the top (as with mixed nuts, popcorn or pebbles and sand), essentially because the smaller ones more easily fall through interstices.

▪ **Higher dimensions.** In no dimension above 3 is it known for certain what configuration of spheres yields the densest packing. Cases in which spheres are arranged on repetitive lattices are related to error-correcting codes and groups. Up to 8D, the densest packings of this type are known to be ones obtained by successively adding layers individually

optimized in each dimension. And in fact up to 26D (with the exception of 11 through 13) all the densest packings known so far are lattices that work like this. In 8D and 24D these lattices are known to be ones in which each sphere touches the maximal number of others (240 and 196560 respectively). (In 8D the lattice also corresponds to the root vectors of the Lie group E_8; in 24D it is the Leech lattice derived from a Golay code, and related to the Monster Group). In various dimensions above 10 packings in which successive layers are shifted give slightly higher densities than known lattices. In all examples found so far the densest packings can always be repetitive; most can also be highly symmetrical—though in high dimensions random lattices often do not yield much worse results.

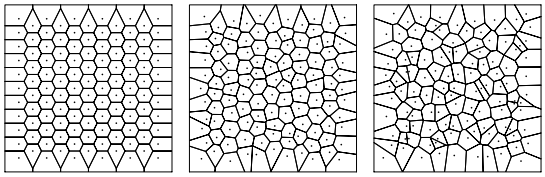
▪ **Discrete packings.** The pictures below show a discrete analog of circle packing in which one arranges as many circles as possible with a given diameter on a grid. (The grid is assumed to wrap around.)



The pictures show all the distinct maximal cases that exist for a 7×7 grid, corresponding to possible circles with diameters $Sqrt[m^2 + n^2]$. Already some of these are difficult to find. And in fact in general finding such packings is an NP-complete problem: it is equivalent to the problem of finding the maximum clique (completely connected set) in the graph whose vertices are joined whenever they correspond to grid points on which non-overlapping circles could be centered.

On large grids, optimal packings seem to approach rational approximations to hexagonal packings. But what happens if one generalizes to allow circles of different sizes is not clear.

▪ **Voronoi diagrams.** The Voronoi diagram for a set of points shows the region around each point in which one is closer to that point than to any other. (The edges of the regions are thus like watersheds.) The pictures below show a few examples. In 2D the regions in a Voronoi diagram are always polygons, and in 3D polyhedra. If all the points lie on a repetitive lattice each region will always be the same, and is often known as a Wigner-Seitz cell or a Dirichlet domain. For a simple cubic lattice the regions are cubes with 6 faces. For an fcc lattice they are rhombic dodecahedra with 12 faces and for a bcc lattice they are truncated octahedra (tetradecahedra) with 14 faces. (Compare page 929.)



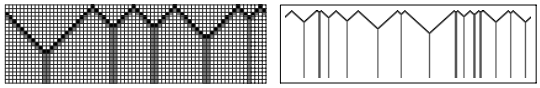
Voronoi diagrams for irregularly distributed points have found many applications. In 2D they are used in studies of animal territories, retail store utilization and municipal districting. In 3D they are used as simple models of foams, grains in solids, assemblies of biological cells and self-gravitating regions in primordial galaxy formation. Voronoi diagrams are relevant whenever there is growth in all directions at an identical speed from a collection of seed points. (In high dimensions they also appear immediately in studying error-correcting codes.)

Modern computational geometry has provided efficient algorithms for constructing Voronoi diagrams, and has allowed them to be used in mesh generation, point location, cluster analysis, machining plans and many other computational tasks.

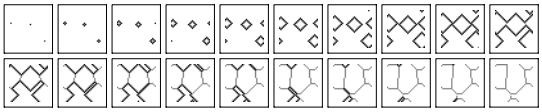
▪ **Discrete Voronoi diagrams.** The $k = 3$, $r = 1$ cellular automaton

$$\{\{0 \mid 1, n : (0 \mid 1), 0 \mid 1\} \to n, \{\_, 0, \_\} \to 2, \{\_, n\_, \_\} \to n - 1\}$$

is an example of a system that generates discrete 1D Voronoi diagrams by having regions that grow from every initial black cell, but stop whenever they meet, as shown below.



Analogous behavior can also be obtained in 2D, as shown for a 2D cellular automaton in the pictures below.



▪ **Brillouin zones.** A region in an ordinary Voronoi diagram shows where a given point is closest. One can also consider higher-order Voronoi diagrams in which each region shows where a given point is the k^{th} closest. The total area of each region is the same for every k, but some complexity in shape is seen, though for large k they always in a sense

approximate circles. 3D versions of such regions have been encountered in studies of quantum mechanical properties of crystals since the 1930s.

■ **Packing deformable objects.** If one pushes together identical deformable objects in 2D they tend to arrange themselves in a regular hexagonal array—and this configuration is known to minimize total boundary length. In 3D the arrangement one gets is typically not very regular—although as noted at various times since the 1600s individual objects often have pentagonal faces suggestive of dodecahedra. (The average number of faces for each object depends on the details of the random process used to pack them, but is typically around 14. Note that for a 3D Voronoi diagram with randomly placed points, the average number of faces for each region is $2 + 48\,\pi^2/35 \simeq 15.5$.) It was suggested by William Thomson (Kelvin) in 1887 that an array of 14-faced tetradecahedra on a bcc lattice might yield minimum total face area. But in 1993 Denis Weaire and Robert Phelan discovered a layered repetitive arrangement of 12- and 14-faced polyhedra (average 13.5) that yields 0.003 times less total area. It seems likely that there are polyhedra which fill space in a less regular way and yield still smaller total area. (Note that if the surfaces minimize area like soap films they are slightly curved in all these cases. See also pages 1007 and 1039.)

■ **Page 351 · Protein folding.** When the molecular structure of proteins was first studied in the 1950s it was assumed that given their amino acid sequences pure minimization of energy would determine their often elaborate overall shapes. But by the 1990s it was fairly clear that in fact many details of the actual processes by which proteins are assembled can greatly affect their specific pattern of folding. (Examples include effects of chaperone molecules and prions.) (See pages 1003 and 1184.)

Origins of Simple Behavior

■ **Previous approaches.** Before the discoveries in this book, nested and sometimes even repetitive behavior were quite often considered complex, and it was assumed that elaborate theories were necessary to explain them. Most of the theories that have been proposed are ultimately equivalent to what I discuss in this section, though they are usually presented in vastly more complicated ways.

■ **Uniformity in frequency.** As shown on page 587, a completely random sequence of cells yields a spectrum that is essentially uniform in frequency. Such uniformity in frequency is implied by standard quantum theory to exist in the idealized zero-point fluctuations of a free quantum field—with direct consequences for such semiclassical phenomena as the Casimir effect and Hawking radiation. (See page 1062.)

■ **Repetition in numbers.** A common source of repetition in systems involving numbers is the almost trivial fact that in a sequence of successive integers there is a repetitive pattern of cases at which a particular divisor occurs. Other examples include the repetitive structure of digits in rational numbers (see page 138) and continued fraction terms in square roots (see page 144).

■ **Repetition in continuous systems.** A standard approach to partial differential equations (PDEs) used for more than a century is so-called linear stability analysis, in which one assumes that small fluctuations around some kind of basic solution can be treated as a superposition of waves of the form $Exp[i\,k\,x]\,Exp[i\,\omega\,t]$. And at least in a linear approximation any given PDE then typically implies that ω is connected to the wavenumber k by a so-called dispersion relation, which often has a simple algebraic form. For some k this yields a value of ω that is real—corresponding to an ordinary wave that maintains the same amplitude. But for some k one often finds that ω has an imaginary part. The most common case $Im[\omega] > 0$ yields exponential damping. But particularly when the original PDE is nonlinear one often finds that $Im[\omega] < 0$ for some range of k—implying an instability which causes modes with certain spatial wavelengths to grow. The mode with the most negative $Im[\omega]$ will grow fastest, potentially leading to repetitive behavior that shows a particular dominant spatial wavelength. Repetitive patterns with this type of origin are seen in a number of situations, especially in fluids (and notably in connection with Kelvin-Helmholtz, Rayleigh-Taylor and other well-studied instabilities). Examples are ripples and swell on an ocean (compare page 1001), Bénard convection cells, cloud streets and splash coronas. Note that modes that grow exponentially inevitably soon become too large for a linear approximation—and when this approximation breaks down more complicated behavior with no sign of simple repetitive patterns is often seen.

■ **Examples of nesting.** Examples in which a single element splits into others include branching in plants, particle showers, genealogical trees, river deltas and crushing of rocks. Examples in which elements merge include river tributaries and some cracking phenomena.

■ **Page 358 · Nesting in numbers.** Chapter 4 contains several examples of systems based on numbers that exhibit nested behavior. Ultimately these examples can usually be traced to

nesting in the pattern of digits of successive integers, but significant translation is often required.

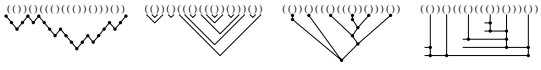
▪ **Nested lists.** One can think of structures that annihilate in pairs as being like parentheses or other delimiters that come in pairs, as in the picture below.

A string of balanced parentheses is analogous to a nested *Mathematica* list such as *{{{}, {{}}}, {}}*. The *Mathematica* expression tree for this list then has a structure analogous to the nested pattern in the picture.

The set of possible strings of balanced parentheses forms a context-free language, as discussed on page 939. The number of such strings containing $2n$ characters is the n^{th} Catalan number *Binomial[2 n, n]/(n + 1)* (as obtained from the generating function *(1 - Sqrt[1 - 4 x])/(2 x)*). The number of strings of depth d (and thus taking d steps to annihilate completely) is given by *c[{n, n}, d] - c[{n, n}, d - 1]* where

```
c[{_, _}, -1] = 0; c[{0, 0}, _] = 1; c[{m_, n_}, _] := 0 /; n > m;
c[{m_, n_}, d_] :=
  Sum[c[{i, j}, d], {i, 0, m - 1}, {j, m - d, n - 1}]
```

Several types of structures are equivalent to strings of balanced parentheses, as illustrated below.

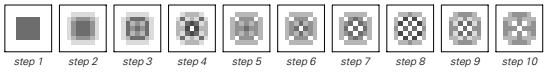


▪ **Phase transitions.** Nesting in systems like rule 184 (see page 273) is closely related to the phenomenon of scaling studied in phase transitions and critical phenomena since the 1960s. As discussed on page 983 ordinary equilibrium statistical mechanics effectively samples configurations of systems like rule 184 after large numbers of steps of evolution. But the point is that when the initial number of black and white cells is exactly equal—corresponding to a phase transition point—a typical configuration of rule 184 will contain domains with a nested distribution of sizes. The properties of such configurations can be studied by considering invariance under rescalings of the kind discussed on page 955, in analogy to renormalization group methods. A typical result is that correlations between colors of different cells fall off like a power of distance—with the specific power depending only on general features of the nested patterns formed, and not on most details of the system.

▪ **Self-organized criticality.** The fact that in traditional statistical mechanics nesting had been encountered only at the precise locations of phase transitions led in the 1980s to the notion that despite its ubiquity in nature nesting must somehow require fine tuning of parameters. Already in the early 1980s, however, my studies on simple additive and other cellular automata (see page 26) had for example made it rather clear that this is not the case. But in the late 1980s it became popular to think that in many systems nesting (as well as the largely unrelated phenomenon of $1/f$ noise) might be the result of fine tuning of parameters achieved through some automatic process of self-regulation. Computer experiments on various cellular automata and related systems were given as examples of how this might work. But in most of these experiments mistakes and misinterpretations were found, and in the end little of value was learned about the origins of nesting (or $1/f$ noise). Nevertheless, a number of interesting systems did emerge, the best known being the idealized sandpile model from the 1987 work of Per Bak, Chao Tang and Kurt Wiesenfeld. This is a $k = 8$ 2D cellular automaton in which toppling of sand above a critical slope is captured by updating an array of relative sand heights s according to the rule

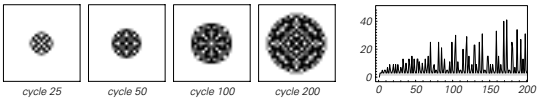
```
SandStep[s_] := s + ListConvolve[
    {{0, 1, 0}, {1, -4, 1}, {0, 1, 0}}, UnitStep[s - 4], 2, 0]
```

Starting from any initial condition, the rule eventually yields a fixed configuration with all values less than 4, as in the picture below. (With an $n \times n$ initial block of 4's, stabilization typically takes about $0.4\, n^2$ steps.).



To model the pouring of sand into a pile one can consider a series of cycles, in which at each cycle one first adds 4 to the value of the center cell, then repeatedly applies the rule until a new fixed configuration *FixedPoint[SandStep, s]* is obtained. (The more usual version of the model adds to a random cell.) The picture below shows slices through the evolution at several successive cycles. Avalanches of different sizes occur, yielding activity that lasts for varying numbers of steps.

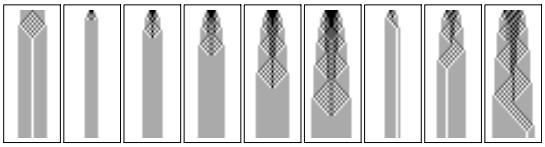
The pictures at the top of the next page show some of the final fixed configurations, together with the number of steps needed to reach them. (The total value of s at cycle t is $4t$; the radius of the nonzero region is about $0.74\sqrt{t}$.) The behavior one sees is fairly complicated—a fact which in the past resulted in much confusion and some bizarre claims, but which in the light of the discoveries in this book no longer seems surprising.



The system can be generalized to d dimensions as a $k = 4\,d$ cellular automaton with $2\,d$ final values. The total value of s is always conserved. In 1D, the update rule is simply

```
SandStep[s_] :=
  s + ListConvolve[{1, -2, 1}, UnitStep[s - 2], 2, 0]
```

In this case the evolution obtained if one repeatedly adds to the center cell (as in the first picture below) is always quite simple. But as the pictures below illustrate, evolution from typical initial conditions yields behavior that often looks a little like rule 184. With a total initial s value of m, the number of steps before a fixed point is reached seems to increase roughly like m^2.



When $d > 1$, more complicated behavior is seen for evolution from at least some initial conditions, as indicated above.

▪ **Random walks.** It is a consequence of the Central Limit Theorem that the pattern of any random walk with steps of bounded length (see page 977) must have a certain nested or self-similar structure, in the sense that rescaled averages of different numbers of steps will always yield patterns that look qualitatively the same. As emphasized by Benoit Mandelbrot in connection with a variety of systems in nature, the same is also true for random walks whose step lengths follow a power-law distribution, but are unbounded. (Compare page 969.)

▪ **Structure of algorithms.** The two most common overall frameworks that have traditionally been used in algorithms in computer science are iteration and recursion—and these correspond quite directly to having operations performed respectively in repetitive and nested ways. But while iteration is generally viewed as being quite easy to understand, until recently even recursion was usually considered rather difficult. No doubt the methods of this book will in the future lead to all sorts of algorithms based on much more complex patterns of behavior. (See page 1142.)

▪ **Origins of localized structures.** Much as with other features of behavior, one can identify several mechanisms that can lead to localized structures. In 1D, localized structures sometimes arise as defects in largely repetitive behavior, or more generally as boundaries between states with different properties—such as the different phases of the repetitive background in rule 110. In higher dimensions a common source—especially in systems that show some level of continuity—are point, line or other topological defects (see page 1045), of which vortices are a typical example.