Chapter 5

Chaos

5.1 Chaotic Behavior in Continuous and Discrete Time

We began our study of dynamics by looking at equilibrium behavior, modeled by stable equilibrium points, or as we learned to call them, point attractors. We then argued that these concepts are inadequate to describe an important scientific phenomenon: robust and stable oscillations in systems. Therefore, we extended our thinking to embrace the concept of oscillation, as modeled mathematically by limit cycle attractors.

It is reasonable to ask, is this all there is? Are equilibrium behavior and oscillatory behavior the only forms of behavior that a system can display? To put it mathematically, are point attractors and limit cycle attractors the only kinds of attractors that can occur in dynamical systems?

Interestingly, the answer is no.

Think about fluid turbulence. Picture yourself in a boat on a river that's in white-water turbulent flow. What is this? It certainly isn't exhibiting static equilibrium behavior, but neither is it periodic. Is it random? Not really: there are large-scale structures such as vortices. Then what is it?

It is now clear that a large number of phenomena, ranging from fluid turbulence to the flapping of a flag in the breeze to cardiac arrhythmias, are examples of a third kind of behavior, which has come to be called **chaos**. Chaotic behavior is represented mathematically by attractors of a third kind, called chaotic attractors.

We will now study this behavior in various kinds of dynamical systems.

Continuous Chaos

We have been studying predator–prey models since the start of this course, but those models typically had only two species. Real ecosystems have many more species than that, which allows for behavior that is more complex than what is seen in two-variable models. In this section, we will develop a three-species model and study the surprising dynamics that emerge.

Imagine a food chain consisting of three species or groupings of species—say plants, rabbits, and foxes, or algae, microscopic invertebrates, and fish (Hastings et al. 1993; Hastings and Powell 1991). We will call the plant mass X, the number of herbivores Y, and the number of carnivores Z. As in the Holling–Tanner two-species model, we assume that in the absence of herbivores, plants would exhibit logistic growth. Also, we assume that the per herbivore consumption of

A. Garfinkel et al., *Modeling Life*,
DOI 10.1007/978-3-319-59731-7_5

plants saturates with increasing plant density, following the function

$$F_1(X) = \frac{a_1 X}{1 + b_1 X}$$

The overall equation is then

$$\textbf{plants} \qquad X' = rX(1 - \frac{X}{K}) - \frac{a_1 X}{1 + b_1 X} Y$$

For the herbivore and predator, we assume that the per capita birth rate is proportional to the amount of food consumed and that the per capita death rate is a constant (d_1 for herbivores and d_2 for predators). The rate at which the predator consumes the herbivore is a saturating function of herbivore density, as in the Holling–Tanner model. The overall equations are

$$\textbf{herbivores} \qquad Y' = c_1 \frac{a_1 X}{1 + b_1 X} Y - d_1 Y - \frac{a_2 Y}{1 + b_2 Y} Z$$

$$\textbf{carnivores} \qquad Z' = c_2 \frac{a_2 Y}{1 + b_2 Y} Z - d_2 Z$$

To simplify our analysis of these equations, we can get rid of the parameters r, K, c_1, and c_2 by setting them equal to 1. The resulting system of equations is

$$\begin{aligned} X' &= X(1 - X) - \frac{a_1 X}{1 + b_1 X} Y \\ Y' &= \frac{a_1 X}{1 + b_1 X} Y - d_1 Y - \frac{a_2 Y}{1 + b_2 Y} Z \\ Z' &= \frac{a_2 Y}{1 + b_2 Y} Z - d_2 Z \end{aligned} \tag{5.1}$$

If we simulate this model, we see something unusual (Figure 5.1).

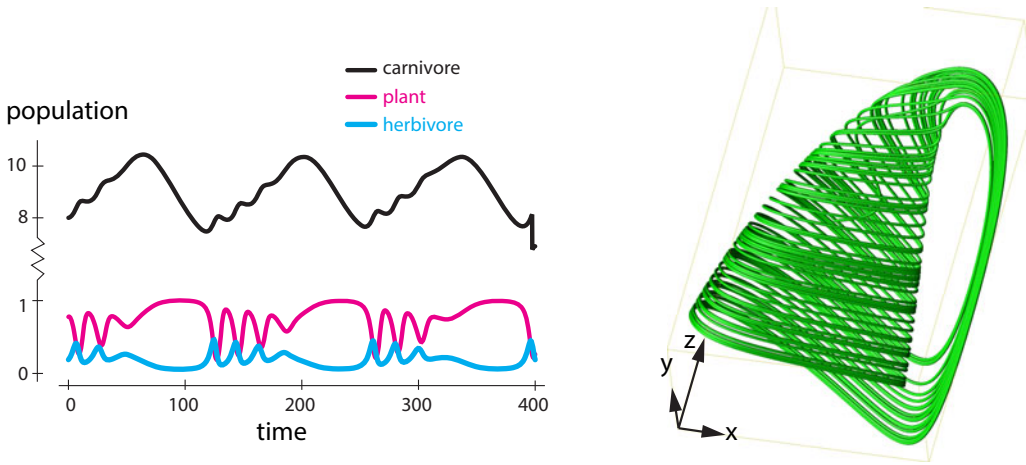


Figure 5.1: Left: a simulation of the three-species food chain model described in the text with $a_1 = 5$, $b_1 = 3$, $a_2 = 0.1$, $b_2 = 2$, $d_1 = 0.4$, and $d_2 = 0.01$. Right: a typical trajectory of the three-species model.

At first glance, the output appears to oscillate. However, a closer look reveals that each cycle is slightly different from the previous one. The number of small ups and downs in each large cycle

is different from one cycle to the next, as is the exact shape of each cycle. So the output is not really periodic. It appears to be somewhat periodic, but also to have some kind of randomness.

If we consider a 3D trajectory generated by the model, we see a complex shape that does not resemble the simple points and loops we've seen before. It looks like an upside-down jug. The path of a typical state point begins in the jug part and then spirals inward mostly in the X-Y plane, while slowly rising along the Z axis. Finally, the state point gets thrown into the handle of the jug, where it plummets down to begin another cycle. The nonrepeating time series and associated complex trajectories are hallmarks of the dynamical behavior known as chaos (Figure 5.1).

Exercise 5.1.1 Simulate the food chain model in SageMath for at least two sets of initial conditions. Plot the results as both time series and trajectories. (For the latter, you can use zip to combine three lists of values before plotting them with list_plot .) Use the parameter values given in the caption of Figure 5.1.

In order to understand chaotic behavior, we will first introduce a different kind of dynamical model, one in which time advances in discrete steps. After learning what we need to there, we will come back to differential equations and continuous time.

Discrete-Time Dynamical Systems

In a *discrete-time model*, time advances in discrete steps. In differential equations, time is the continuous variable t. But in a discrete-time system, time comes in discrete intervals 0, 1, 2, 3, . . . , with no values between them. Therefore, we represent "time" by the integer-valued variable N, so $N = 0, 1, 2, 3, \ldots$.

Such models work well for organisms with well-defined breeding seasons or in other situations in which the data come at discrete times. For example, heartbeats are discrete; there are the Nth heartbeat and the $(N + 1)$st heartbeat, but nothing in between.

Consider a deer population growing at 5% a year. If population size is denoted by the variable X, its value at time N is written $X(N)$ or X_N (pronounced "X of N"). Then, we can write an equation that gives us X_{N+1} as a function of X_N,

$$X_{N+1} = X_N + 0.05X_N = 1.05X_N \tag{5.2}$$

This kind of equation, which gives the population size at time $N + 1$ in terms of that at time N, is called a *difference equation*. The value 1.05 in equation (5.2) is typically represented by the parameter r, so the general difference equation is

$$X_{N+1} = rX_N \tag{5.3}$$

When $r > 1$, the population is growing, and when $r < 1$, it is shrinking. If r is exactly 1, the population stays the same size, but this is essentially impossible in nature (Figure 5.2).

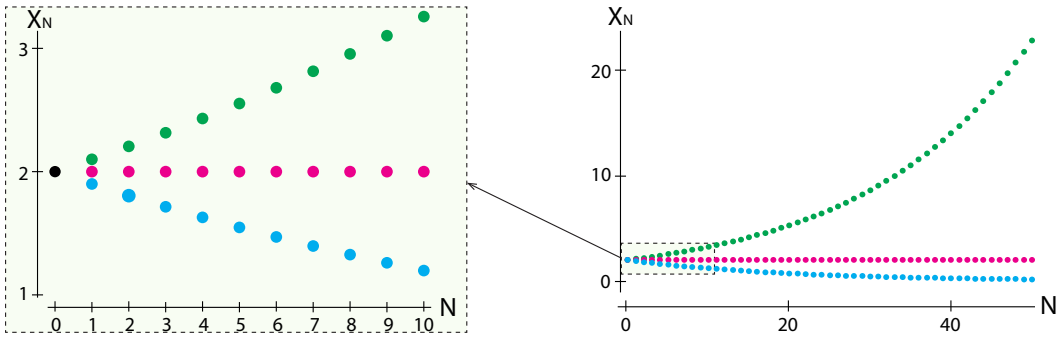


Figure 5.2: Exponential growth in discrete time for three different values of $r = 1.05$, 1.0, and 0.95, with initial condition $X(0) = 2$. Left: short time. Right: long time.

Exercise 5.1.2 Choose an initial value for X and use values of r that are less than, greater than, and equal to one to test the above statements by computing two values of X for each value of r.

As you've probably noticed, exponential growth starts off slowly and then gets faster and faster, with each increment of growth being larger than the previous one. How much larger? To find out, we subtract X_{N+1} from X_{N+2}:

$$X_{N+2} - X_{N+1} = rX_{N+1} - rX_N = r(X_{N+1} - X_N)$$

If we use the symbol Δ_{N+1} to represent the growth increment $(X_{N+2} - X_{N+1})$, then

$$\Delta_{N+1} = r\Delta_N$$

In other words, the growth increment increases at the same rate r as the population itself, although the per capita growth rate remains constant. When the population is small, it grows slowly, but as the population increases, so does its growth rate. This allows exponential growth to sneak up on you, as the following exercise illustrates.

Exercise 5.1.3 An inedible alga is growing on a pond in a city park. Only a small part of the pond is now covered by the algae, but the area covered is doubling each day. The city decides to remove the algae once it covers half the pond. If the pond will be completely overgrown in thirty days, on what day will it be half covered? *(Hint: Try working backward.)*

If we start with an initial condition X_0, then

$$\begin{aligned} X_1 &= rX_0 \\ X_2 &= rX_1 \quad = r^2X_0 \\ &\vdots \qquad\quad \vdots \\ X_N &= rX_{N-1} = r^2X_{N-2} = \cdots = r^NX_0 \end{aligned}$$

Exponential growth in discrete time is represented by the difference equation

$$X_{N+1} = r \cdot X_N$$

It has a solution, namely,

$$X_N = r^N \cdot X_0$$

Exercise 5.1.4 A rabbit population is growing at 10% a year. If there are 10 rabbits this year and time is discrete, how many will there be in 10 years? Use a loop in SageMath to check your answer.

Exercise 5.1.5 While we have been working with $r > 1$, representing growth, r can be less than 1, representing a quantity that decreases over time. The *half-life* of a radioactive element is the amount of time needed for half the element to decay. What fraction of the initial amount of such an element will remain after ten half-lives?

Exercise 5.1.6 When money in a bank account accrues *compound interest*, the interest earned in one time period is added to the principal, and then the sum is used as the base for the next time period.

a) If you start off with \$1000 and earn 2% interest that is compounded annually, how much money will you have in 5 years? In 10 years? In 20 years?

b) How long will it take you to accumulate \$10,000?

The Discrete-Time Logistic Model

We will now develop and examine an important discrete-time model, the discrete logistic equation (May 1976).

Consider a population of insects that live one year, lay eggs, and then die. The insect population in year $N+1$ is a function of the population in year N. If we call the population X, then $X_{N+1} = f(X_N)$.

Suppose there are enough resources to support a maximum of K insects. If the current population is X_N, then the current population is using only $\frac{X_N}{K}$ of the total resources available. But that means that the fraction of total resources that are **un**used is $1 - \frac{X_N}{K}$. It is these unused resources that are available to support new births. Therefore, just as in the continuous-time logistic equation model, we will assume that the per capita insect birth rate is proportional to the available resources, with proportionality constant r. This gives us

$$\text{per capita birth rate}_N = r(1 - \frac{X_N}{K})$$

As always, the per capita birth rate must be multiplied by the population size X_N to get the total birth rate. Then the population as a whole lays $rX_N(1 - \frac{X_N}{K})$ eggs in year N, and since no adults survive from one year to the next, and assuming that each egg laid leads to a mature adult that reproduces,

$$X_{N+1} = rX_N(1 - \frac{X_N}{K})$$

This equation has two parameters, r and K. To simplify our analysis of its behavior, we can set $K = 1$, so the numbers X_N can be interpreted as fractions of the carrying capacity. We then

have the equation

$$X_{N+1} = rX_N(1 - X_N) \tag{5.4}$$

Exercise 5.1.7 If $r = 1.2$ and $X_0 = 0.42$, what is X_1? X_2?

Equation (5.4) is called the *discrete logistic equation* or the *discrete logistic model*. As usual, "discrete" refers to time (Figure 5.3). The state of the system can be any number between 0 and 1. The discrete logistic model is just as deterministic as all the other models we've studied. There is no randomness in equation (5.4). Also, we specified that the maximum possible insect population is 1, and as long as $r \leq 4$, the population will indeed stay between 0 and 1. Thus, the dynamics of this system are bounded. Simulating this model (using iteration) gives us a surprisingly irregular time series (Figure 5.4).

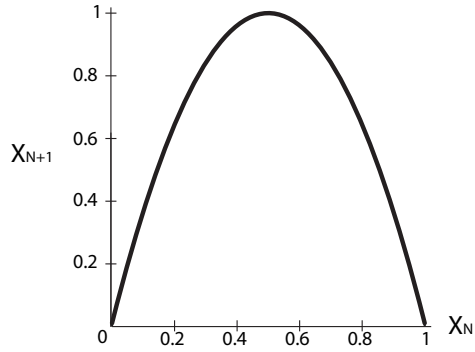


Figure 5.3: Graph of the function $X_{N+1} = rX_N(1 - X_N)$ for $r = 4$.

While this time series is irregular, there are also some predictable aspects to the behavior. Note, for example, that when the state variable takes values close to zero, the subsequent changes are small. Similarly, when the state variable takes values near 0.75, the subsequent changes are also small (look around $N = 10$ and $N = 27$). We will see why shortly.

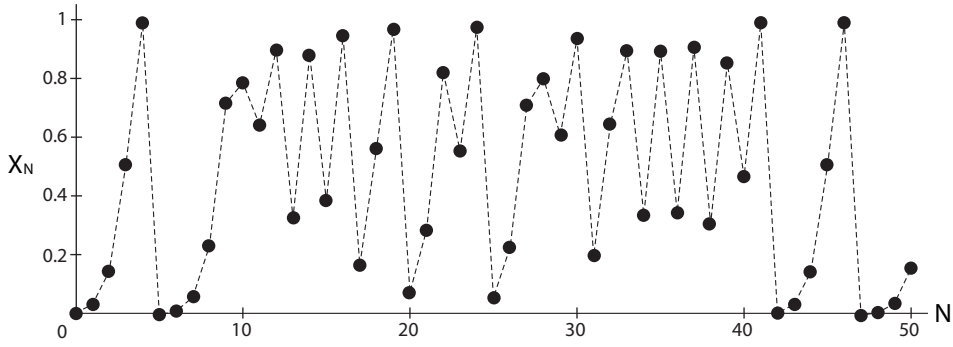


Figure 5.4: A simulation of the discrete logistic model with $r = 4$ and $X_0 = 0.01$.

Exercise 5.1.8 Recreate Figure 5.4 in SageMath. *(Hint: You may need to review iteration.)*

Exercise 5.1.9 Run another simulation of the discrete logistic model with a different initial value and value of r. (Recall that r has to be between 0 and 4.)

Dynamics from a Discrete-Time Model: Cobwebbing

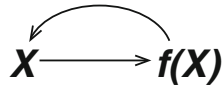
Consider a discrete-time model

$$X_{n+1} = f(X_n)$$

We get dynamics from this model by realizing that if we start at $X = X_0$, then

$$\begin{array}{lll} X_1 = f(X_0) & & \\ X_2 = f(X_1) & = f(f(X_0)) & = f^2(X_0) \\ \quad\vdots & \quad\vdots & \quad\vdots \\ X_n = f(X_{n-1}) & = f(f(\ldots f(X_0))) & = f^n(X_0) \end{array}$$

So the successive values $X_1, X_2, X_3, \ldots, X_n$ are produced by applying f over and over. This is called *iterating* the function, and this subject is sometimes called *iterated function dynamics*.



Of course, we can generate these values by pressing the "f" button over and over, and indeed that's how we generated Figure 5.4. But there is another, geometric, way to look at this process.

As our example, let's use the discrete-time logistic function. Suppose we start with an X_0. Then the graph tells us the value of $X_1 = f(X_0)$: simply shoot up from X_0 to the function f and look to the left to see its value (Figure 5.5).

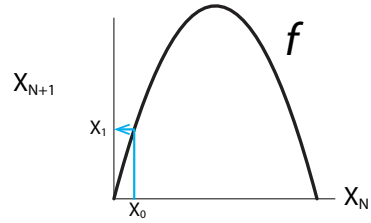


Figure 5.5: The first step of the iteration of f from the initial condition X_0, finding $X_1 = f(X_0)$.

Now we would like to find $X_2 = f(X_1)$. But we have a problem: we have X_1 on the vertical axis, but we need it on the horizontal axis in order to shoot it up to the function. The problem is solved with a simple piece of geometry. Let's draw a construction line of slope 1 (the gray line in Figure 5.6). Then, to find the value on the horizontal axis corresponding to any value on the vertical axis, just draw a horizontal line from the value on the vertical axis to the line of slope 1, and then drop a vertical line down to the horizontal axis. Because the construction line has slope 1, the resulting object is a square, and we have now found X_1 on the horizontal axis (Figure 5.6).

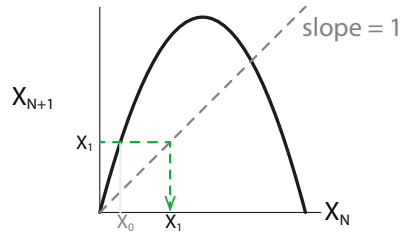


Figure 5.6: Using the slope = 1 projection line, X_1 is located on the horizontal axis.

Now we simply shoot up from X_1 to the function to read off $X_2 = f(X_1)$ (Figure 5.7).

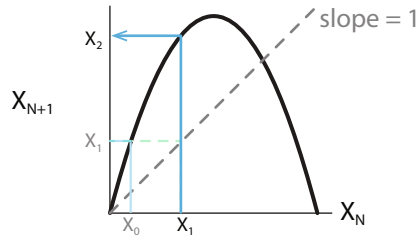


Figure 5.7: The second step of the iteration finds $X_2 = f(X_1)$.

Staring at the diagram, we realize that part of this process involves unnecessary back-and-forth motions. After we have found the point on the graph of the function corresponding to X_1, it is not necessary to go left to the vertical axis and then back again to X_1 on the function. We could just have gone directly from X_1 on the function and headed to the right to find the intersection with the line of slope 1. Similarly, once we have found the point on the 1-1 line, it is not necessary to go down to the horizontal axis and then go back up again to the same point: we could just go up from the point on the 1-1 line to the function to get the next value (Figure 5.8). Repeating this process of reflecting alternately between the function and the 1-1 line, we generate a process called *cobwebbing*. Cobwebbing is a general procedure for obtaining time dynamics from a discrete-time function.

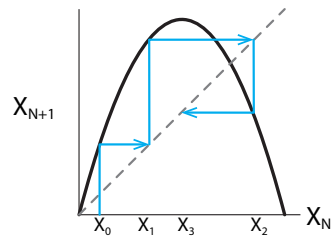


Figure 5.8: Repeated iterations result in the cobwebbing process.

Cobwebbing, the process of reflecting alternately between the function f and the 1-1 reference line, is the geometric realization of the process of iterating the function f over and over.

If we carry out this cobwebbing for the discrete logistic function, we see that the process never closes (Figure 5.9).

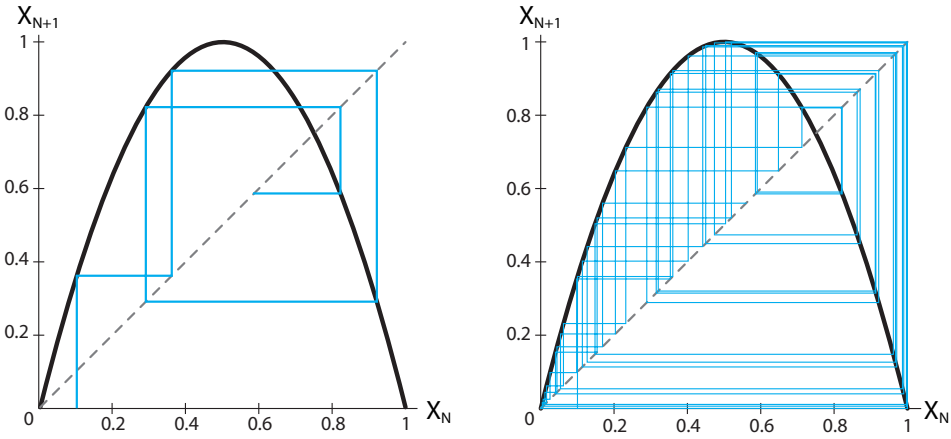


Figure 5.9: Left: 10 steps of the cobwebbing process starting from initial condition $X_0 = 0.1$. Right: 50 steps.

Exercise 5.1.10 Use cobwebbing to determine the dynamics for the linear discrete-time systems $X_{N+1} = rX_N$ for values of r in the following ranges:

a) $0 < r < 1$ b) $r > 1$ c) $-1 < r < 0$ d) $r < -1$

Further Exercises 5.1

1. The Beverton–Holt model

$$X_{N+1} = \frac{rX_n}{1 + X_n/m}$$

is a discrete-time population model sometimes used in fisheries research.

a) With $m = 20$ and $r = 3$, use cobwebbing to iterate the model for four steps.

b) Numerically simulate the model with the same parameter values for 20 time steps and plot your results. Describe the model's behavior.

c) Experiment with different values of r and m. What kinds of behavior can you generate?

2. The Ricker model

$$X_{N+1} = X_N e^{r(1-\frac{X_n}{k})}$$

is another discrete-time model used in ecology and fisheries.

a) With $k = 20$ and $r = 3$, use cobwebbing to iterate the model for four steps.

b) Numerically simulate the model with the same parameter values for 100 time steps and plot your results. Describe the model's behavior.

c) Experiment with different values of r and m. What kinds of behavior can you generate?

5.2 Characteristics of Chaos

Chaos is dynamical behavior that is deterministic, bounded in state space, irregular, and, most intriguingly, extremely sensitive to initial conditions. We will discuss each of the defining characteristics of chaos in turn.

Linguistic Caveats

"Chaos" is one of the rare mathematical terms to have penetrated popular culture. However, the term is a misleading one, although we are stuck with it for historical reasons. Chaotic behavior can look erratic, but it embodies a complex order that we will study in this section. Also, we will sometimes speak of "chaotic systems," but chaos is a type of behavior, not a type of system. A chaotic system is one that is behaving chaotically, just as an oscillating system is one that is behaving in an oscillatory manner. Unfortunately, "chaosing" is not a word.

Determinism

To say that a system is deterministic means that each state is completely determined by the previous state.

In the food chain model, just as in all the other models we have studied, there are no unmodeled outside influences or chance events.

If we allow outside chance events, it is easy to produce an irregular time series by, say, flipping a coin, but there's nothing like this in the food chain model or the discrete logistic model. The system is deterministically producing its own irregular behavior without any randomness.

Boundedness

Another characteristic of chaotic behavior is boundedness. Boundedness means that the system does not go off to infinity. Rather, as Figure 5.1 illustrates, it stays within a certain region of state space. In other words, we could draw a box in state space and the system would stay within that box. And in the discrete logistic model, as long as our initial condition is within the interval $(0, 1)$, the result will always be in that interval; the state point will not escape to higher values.

Exercise 5.2.1 Give an example of a system whose dynamics are not bounded.

Irregularity

We are now ready to start discussing the characteristics of chaos that make it different from the types of dynamical behavior we've encountered before. The first of these is that chaotic behavior is irregular, or aperiodic. Aperiodic behavior *never* exactly repeats. If a trajectory ever *exactly* repeated, that is, returned to the very same mathematical state point, it would have to be periodic, because determinism would require that it return again and again. All closed orbits are periodic trajectories, hence limit cycle attractors are closed periodic orbits.[1]

Systems with point or limit cycle attractors have initial transients but then settle down into repetitive behavior. Chaotic behavior, on the other hand, starts out irregular and remains irregular. In some systems, it may look like the behavior repeats and it can come very close to previous state values, but it never exactly repeats.

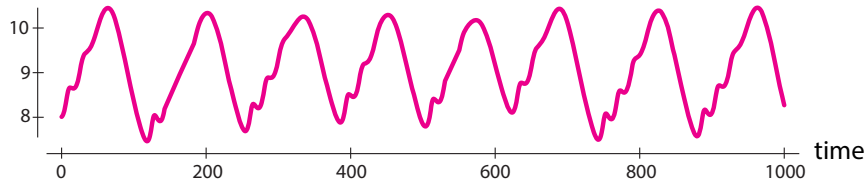


Figure 5.10: Time series of the carnivore population from a typical simulation of the three species food chain model.

Let's take a closer look at the carnivore populations in a simulation of the food chain model (Figure 5.10). At first glance, it seems that the populations oscillate, albeit in a somewhat complex way. However, a closer look reveals differences. The second large oscillation contains one bump before the peak, while the third oscillation has at least two. Moreover, each peak has a somewhat different shape. This is aperiodicity. Despite a general qualitative similarity, the behavior of the system *never* repeats and *never* approaches repetition. A similar statement is true about the output of the discrete-time logistic model. It may look as though some shapes repeat themselves, but if we look closely, we see that the sequence in fact never repeats.

Exercise 5.2.2 In Figure 5.4 on page 228, the last eight or so points look about the same as the first eight. Run this simulation in SageMath for 100 time steps.

a) Are the points actually the same? *(Hint: Look at the numerical output of your simulation.)*

b) After $N = 50$, does the simulation continue to act as it did at the beginning?

Sensitive Dependence on Initial Conditions

The most intriguing and famous characteristic of chaos is *sensitive dependence on initial conditions*. This term refers to the fact that in a chaotic system, two time series that start very close together will eventually diverge to the point where their behavior is completely uncorrelated.

[1] It also never *approaches* repetitive (periodic) behavior. The last part is critical. Strictly speaking, trajectories approaching a stable equilibrium point or limit cycle don't repeat, either, because trajectories cannot cross. However, as time goes on, they get closer and closer to completely repetitive behavior, so it makes sense to call them periodic. Technically, they are asymptotically periodic.

Let's consider two simulations of the food chain model, with two closely spaced initial conditions. The two simulations are at first indistinguishable; they then diverge slowly from each other (first and second panels). But then toward the end (third panel), they become completely uncorrelated (Figure 5.11).

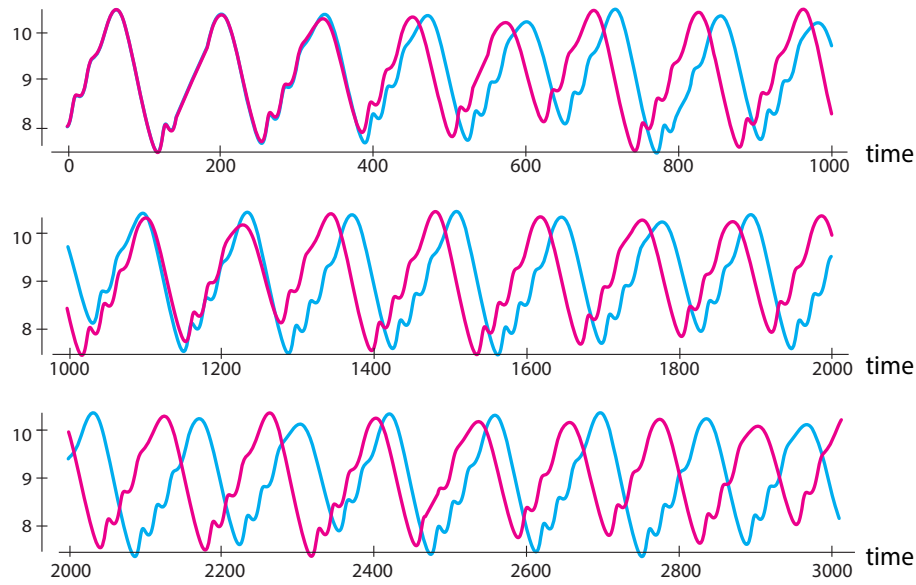


Figure 5.11: Time series of the carnivore population from two simulations of the Hastings food chain model for two different initial conditions, 8.0 and 8.01.

Sensitive dependence on initial conditions is also a property of discrete-time chaotic systems such as the logistic system. Two simulations of the discrete logistic model with $r = 4$, one for $X_0 = 0.01$ and one for $X_0 = 0.011$, show initial agreement but quickly diverge (Figure 5.12).

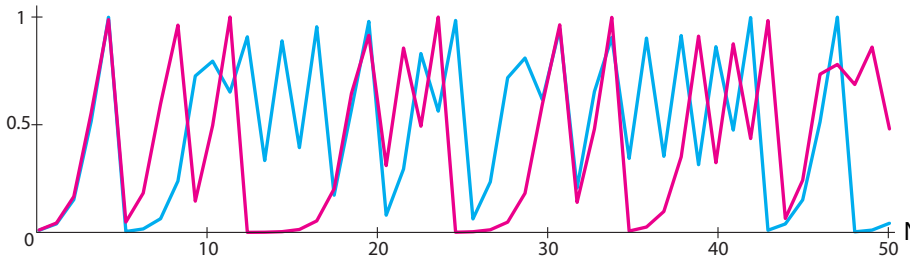


Figure 5.12: Two simulations of the discrete logistic model with $r = 4$.

The concept of "sensitive dependence" can be given a precise definition. Suppose N_0 and M_0 are two different initial conditions for the food chain model. Let's define $d(M_0, N_0)$ as the distance between M_0 and N_0. Since M_0 and N_0 are points in 3-dimensional (X, Y, Z) space, the distance between them is the Euclidean distance

$$d(M, N) = \sqrt{(X_M - X_N)^2 + (Y_M - Y_N)^2 + (Z_M - Z_N)^2}$$

After a time t, the two points M_0 and N_0 have evolved to M_t and N_t. Sensitive dependence says that the distance $d(M_t, N_t)$ grows exponentially with time for some λ (Figure 5.13):

$$d(M_t - N_t) \quad = \quad e^{\lambda \cdot t} \cdot d(M_0 - N_0)$$

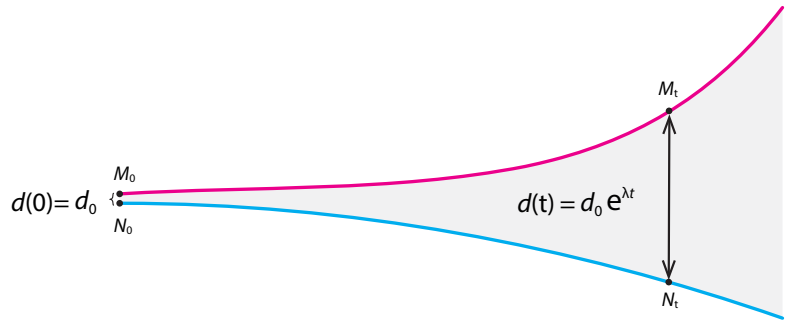


Figure 5.13: Exponential divergence over time of nearby trajectories. This is characteristic of chaotic systems.

In general, for multivariable systems, in both discrete and continuous time, sensitive dependence means exponential divergence of nearby trajectories: there is a number λ (greek letter lambda), called the Lyapunov characteristic exponent, such that

$$d(M_t - N_t) \quad = \quad e^{\lambda \cdot t} \cdot d(M_0 - N_0)$$

Exercise 5.2.3 Derive the expression for exponential divergence in the case of the discrete-time logistic system. *(Hint: Here, because the logistic system has a single state variable, the distance between two points X and Y is just the absolute value of their difference, $|X - Y|$.)*

But of course, this exponentially fast divergence cannot continue forever, because the whole behavior is contained in a box. Therefore, two nearby trajectories will start by diverging from each other exponentially fast, but then they will ultimately be folded back into the box by the dynamics. This tension between "wanting to diverge" and "staying in the box" creates many of the key properties of chaos.

Unpredictability

Edward Lorenz, the meteorologist and mathematician who helped discover chaos, gave a talk at the annual meeting of the American Association for the Advancement of Science in 1972, called "Predictability: does the flap of a butterfly's wings in Brazil set off a tornado in Texas?" He posed a question: Consider two planets that are absolutely identical, down to the clothes you are wearing today, every tree, every detail, except that in world A there is one more butterfly in Brazil. What will happen to the weather systems of the two planets? Common sense says there will be no difference from such an infinitesimal change, but common sense is wrong. In

fact, after not much time, the weather systems will diverge completely, so that there is, say, a tornado in Texas in world A but not in world B!

Sensitive dependence on initial conditions helps explain a fundamental property of chaotic behavior, its unpredictability. Look at Figure 5.12. We took two initial conditions, $X_0 = 0.01$ and $X_0 = 0.011$, that differ by 1 part in 1000, or a tenth of a percent. But the X_{50} corresponding to these two initial conditions is completely different: look at $N = 49$ and $N = 50$ and note that the pink tracing is very high, while the blue tracing is very low.

Let's see how accurate your initial condition would have to be to correctly predict X_{50}. Note that there is a squaring inside the function: in order to calculate the next X you have to, among other things, square the previous X. But the square of 0.01 (which has two decimal places) is 0.0001, which has four. And the square of that has eight decimal places. So by the time you are calculating X_{50}, you need a number whose length has doubled 50 times. But as we saw in Chapter 2, 2^{50} is around 10^{15}, so you would need an initial condition that has a thousand trillion decimal places. Good luck getting your computer to handle that!

This same property has another surprising consequence.

Exercise 5.2.4 In this model, let $r = 4$, and assume that the initial value is $X_0 = 0.6$. Simulate the model in two different ways:

a) Use SageMath to print the values of X_0, X_1, X_2, ..., up to at least X_{20}.

b) With the help of a simple pocket calculator (or a calculator app on your phone), create the same list on paper. In this case, continue until the numbers that you are calculating on paper look completely different from the numbers that SageMath gave you.

The result of the above exercise should be surprising. We simulated exactly the same deterministic system with exactly the same initial condition and got completely different results. What's going on?

As we saw, the true length, the number of significant digits of the number X_N, doubles with each $N = 1, 2, 3, \ldots$, and X_{50} has a staggering length. But computers have a finite amount of memory, and it would quickly run out. Therefore, computers and calculators are designed to round off decimals to a certain number of places. Exactly when and how this rounding is done depends on the particular combination of hardware and software. It's very unlikely that the SageMath system you are using rounds numbers in exactly the same way as your calculator. Most of the time, the rounding error described here (say in the 16th decimal place) has undetectably tiny effects. However, chaotic systems' sensitivity to exact state values amplifies these tiny errors beyond all expectation. After a certain period of time (how long depends on the system), this magnified error overwhelms our knowledge of the system, and quantitative prediction becomes impossible.

The lesson here is very profound. It actually makes us rethink the question, "what is the purpose of science?" If the answer was "to make detailed numerical predictions of the exact future state of systems," then chaos means that this is often impossible. So we have to redefine the purpose so that predicting and understanding the *qualitative* behaviors of systems is a legitimate (and important) goal of science.

Chaotic Attractors

In the previous chapter, we defined an attractor A of a dynamical system

$$V : X \longrightarrow T(X)$$

as a subset A of X that has the property that for a large set of initial conditions, every trajectory tends to A as $t \to \infty$.

We have already seen two major kinds of attractor:

1) *Point attractors*, or stable equilibrium points, represent behavior that is either static or approaching it.
2) *Limit cycle attractors*, or stable closed orbits, represent behavior that is periodic (or approaching it).

For a long time, scientists and mathematicians thought that those were the only two kinds of attractor that could exist, either mathematically or physically. It turns out they were wrong, both mathematically and in real systems (Hilborn 2000).

Look at the chaotic trajectory of the three-species food chain model (Figure 5.1 on page 224). The simulation you are looking at has been run for a long enough time that you are looking at the long-term behavior. This system has gone to an attractor. But what can that attractor be?

The answer is: it's certainly not a point, and it's not a closed orbit either. It's a third kind of attractor, called a "strange attractor" or chaotic attractor. It's a very complicated geometry, but it exists, and it satisfies the definition of attractor.

Chaotic attractors represent a third kind of motion, other than equilibrium behavior and oscillatory behavior. Motion on a chaotic attractor is irregular and unpredictable. Nevertheless, there is an overall form to the behavior.

Behavior	Mathematical model
equilibrium	stable equilibrium point ("point attractor")
oscillation	limit cycle attractor
chaos	chaotic attractor

To get a better sense of what the attractor is, consider two simulations allowed to run for a long time. Two simulations of a chaotic system that start a tiny distance apart will eventually become completely uncorrelated, but they can still retain a certain qualitative similarity (Figure 5.11).

This situation becomes even clearer when we consider continuous-time systems. Consider the trajectories of the two food chain simulations whose time series are shown in Figure 5.11. The two initial conditions lead to very similarly shaped trajectories (Figure 5.14).

However, look at the superposition of the two trajectories, shown in the right-hand figure. Note that the red trajectory and the blue trajectory *have zero points in common*. This would be true even if the two trajectories were extended to infinity. As we have seen before, the two trajectories have identical well-defined shapes. This shape is an example of a *chaotic attractor* or *strange attractor*.

Notice that although the red and blue trajectories always remain distinct, they have the same shape. The behavior of a chaotic system is governed by an attractor, just as equilibrium and oscillating systems are. The attractor has a more complex shape, but it is still an attractor. If

we plot the two state space trajectories corresponding to these two initial conditions, we see an important fact (Figure 5.11): the behaviors of the two simulations are qualitatively similar, but carrying out one simulation does not allow you to predict the behavior of the other in quantitative detail.

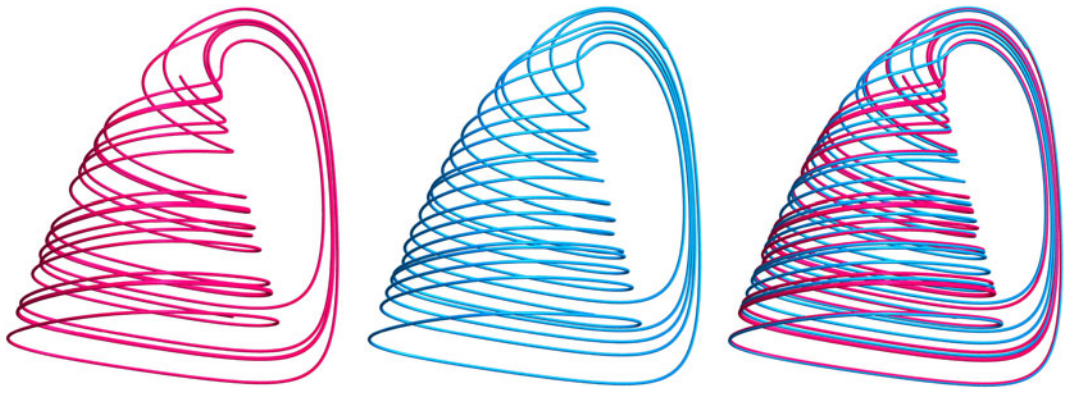


Figure 5.14: Left and middle: two simulations of the food chain model with different initial conditions. Right: superposition of these two simulations shows the same form, yet completely different details.

Exercise 5.2.5 Although the two trajectories in Figure 5.14 look very similar, they do not share any points. Why does this have to be true?

Further Exercises 5.2

1. Make a table showing which of the four defining characteristics of chaotic behavior are shared by exponential growth, equilibrium behavior, and oscillation.

2. As in our previous Romeo–Julet models, let R and J be Romeo's and Juliet's love for each other. Tybalt is a sworn foe of Romeo's family, so let T be Tybalt's *hatred* for Romeo. (This model is normally called the Rössler model and is a well-known mathematical example of chaos.)
   - Juliet's love is fueled by Romeo's love for her, and to a small extent (0.1) by her own love for him. Therefore, $J' = R + 0.1J$.
   - Romeo has a fear of commitment, so the more Juliet loves him, the faster his love decreases. Also, since Tybalt is Juliet's cousin, Tybalt's hatred of Romeo drives Romeo away from Juliet. Thus, $R' = -J - T$.
   - Finally, Tybalt's hatred of Romeo grows naturally at a constant rate (0.1), minus some multiple (c) of itself, to prevent it from growing without bound. But when Tybalt finds out that Romeo is in love with his cousin, his hatred of Romeo fuels itself at a rate equal to Romeo's love. Therefore, $T' = 0.1 - cT + RT$.

a) Simulate the Romeo, Juliet, and Tybalt model for $c = 14$ and the initial conditions $R(0) = 5$, $J(0) = 5$, and $T(0) = 1$. Plot the results as both a time series and a 3D trajectory. *(Hint: You'll probably want to use* `plotjoined=True`*.)*

b) Pick an initial condition close to the original one and create both trajectories and time series for it. Overlay the plots with those from part (a). What do you observe?

c) It can also be useful to look at the variables two at a time. Plot the trajectories of Romeo and Juliet, Romeo and Tybalt, and Juliet and Tybalt.

d) In part (c), it sometimes looked as if the trajectory was crossing itself. Why is this impossible? What's really going on?

3. The characters are Romeo, Juliet, and Juliet's nurse. Since the nurse is involved, it makes sense to model the characters' happiness rather than their attraction to each other. With that, we have the following assumptions.

   – Juliet loves her nurse and wants to be exactly as happy as the nurse herself is. So $J' = s(N - J)$.

   – The nurse is similarly attached to Juliet and her happiness grows in proportion to Juliet's. However, she's a bit of a worrywart, and her happiness causes itself to decline. She's also worried about Juliet's developing relationship with Romeo, so whenever their emotions are in sync, her happiness drops precipitously. Thus, $N' = rJ - N - RJ$.

   – Finally, Romeo just wants his life to be simple. When Juliet and the nurse have different emotions, he finds it hard to deal with them, but he doesn't actually care whether they're happy or not. Also, his happiness causes itself to decline, just as in the case of the nurse. So $R' = JN - bR$.

   This model, which usually has a meteorological rather than a literary motivation, is called the Lorenz model; it played a key role in the discovery of chaos.

   a) Simulate this model using the parameter values $s = 10$, $b = \frac{8}{3}$, and $r = 28$, with the initial conditions $J(0) = 0.1$, $N(0) = -6$, and $R(0) = 0.01$. Use a step size of 0.01 to get better plotting. Plot a time series and 3D trajectory using the plotting option `plotjoined=True`. (The masklike trajectory you are seeing is called the Lorenz attractor or, more descriptively, the Lorenz butterfly or Lorenz mask.) Interpret these plots in terms of the system being modeled. You can focus on just Romeo and Juliet.

   b) Pick an initial point close to the original one and plot a trajectory and time series as in the previous part. Overlay the plots for the two initial conditions. What do you observe?

5.3 Routes to Chaos

We mentioned that chaos is not a kind of system; it's a kind of behavior, which a system may or may not exhibit. Whether a system displays chaos generally depends on the value of some critical parameter. For some values, the system's behavior will be chaotic, but other values can result in equilibrium or oscillatory behavior.

Let's look at the logistic equation

$$X_{N+1} = rX_N(1 - X_N)$$

and consider the parameter r. We saw chaotic behavior for $r = 4$, and it exists for almost all values of r greater than 3.57, but what about lower values of r? It turns out that there are many parameter regimes for which the behavior is not chaotic.

For example, if $r = 2.9$, the behavior is a point attractor or stable equilibrium point (Figure 5.15). The cobweb converges to a stable equilibrium point, and the time series also confirms this convergence.

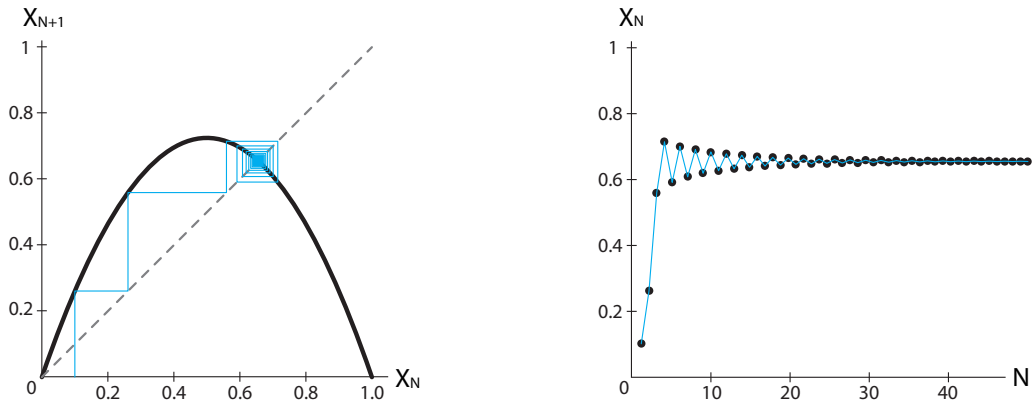


Figure 5.15: Discrete logistic model with $r = 2.9$. Each time the cobweb process touches the graph of the function (left figure), it creates a new data point (black dots in the right-hand figure).

But if we increase r to values above 3.0, the equilibrium point becomes unstable. It is easy to show why this happens.

First, let's calculate the value of the equilibrium point for arbitrary r. Since the definition of an equilibrium point in a discrete-time system is

$$X_{N+1} = X_N$$

we can say that an equilibrium point of the discrete logistic system is a point where

$$X_{N+1} = f(X_N) = rX_N(1 - X_N) = X_N$$

Dividing both sides by X_N gives

$$r(1 - X_N) = 1$$

The equilibrium point is therefore

$$X_{eq} = \frac{r - 1}{r}$$

This holds for all values of r; for example, when $r = 3$, the equilibrium point is $X = \frac{3-1}{3}$, or $X = \frac{2}{3}$.

Exercise 5.3.1 Find the equilibrium point for discrete-time exponential growth, $X_{N+1} = rX_N$.

Exercise 5.3.2 The Ricker model,

$$X_{N+1} = X_N e^{r(1-\frac{X_N}{k})}$$

is another discrete-time population model in which population growth is limited by crowding. Find this model's equilibria.

Next, we need to determine the stability of this equilibrium point. When we were studying single-variable differential equations, we developed the *principle of linearization* (the Hartman–Grobman theorem), which says that near an equilibrium point, a differential equation has the same behavior as its linear approximation. A similar principle holds for discrete-time dynamical systems: near an equilibrium point, the system has the same behavior as its linear approximation. But what is this linearization? In Chapter 2, we wrote this linear approximation as

$$\Delta Y = \frac{df}{dX}\Big|_{X_{eq}} \Delta X$$

Here, "Y" $= X_{N+1}$ and "X" $= X_N$, so the linear approximation in discrete time translates to

$$X_{N+1} - X_{eq} = \frac{df}{dX}\Big|_{X_{eq}} \cdot (X_N - X_{eq})$$

Note what this implies: if the absolute value of the slope $\left|\frac{df}{dX}\right|_{X_{eq}}$ is less than 1, then $X_{N+1} - X_{eq}$ is less than $X_N - X_{eq}$, which is another way of saying that X_{N+1} is closer than X_N to the equilibrium point. In other words, perturbations die out. This, in turn, implies that the equilibrium point is stable (Figure 5.16).

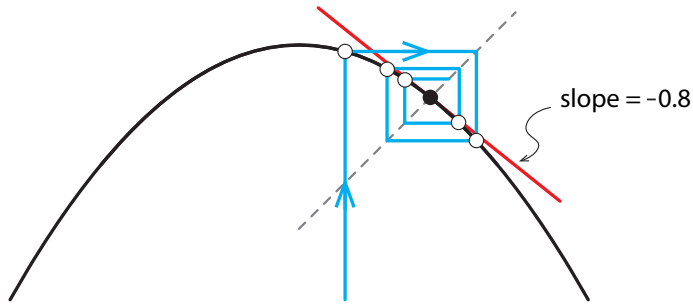


Figure 5.16: When the slope of the tangent line (red) at the equilibrium point has absolute value less than 1, the cobweb process near the equilibrium point converges, producing a stable equilibrium.

Another way to look at this is to see that the linearization is just the linear discrete-time equation $X_{N+1} = rX_N$. This equation has only one equilibrium point, at $X = 0$, and this is stable if and only if $r < 1$.

Exercise 5.3.3 Why is this true?

Similarly, the equilibrium point is unstable exactly when the slope of f at that equilibrium point has absolute value greater than 1 (Figure 5.17):

$$\left|\frac{df}{dX}\right|_{X_{eq}} > 1$$

Now let's calculate the stability of the equilibrium point of the discrete logistic equation. In this case, the slope of f is

$$\begin{aligned}\frac{df}{dX} &= \frac{d}{dX}(rX - rX^2)\\ &= r - 2rX\end{aligned}$$

Plugging in the equilibrium point value X_{eq} gives

$$\begin{aligned}\frac{df}{dX}\Big|_{X_{eq}} &= r - 2r(\frac{r-1}{r})\\ &= r - 2(r-1)\\ &= 2 - r\end{aligned}$$

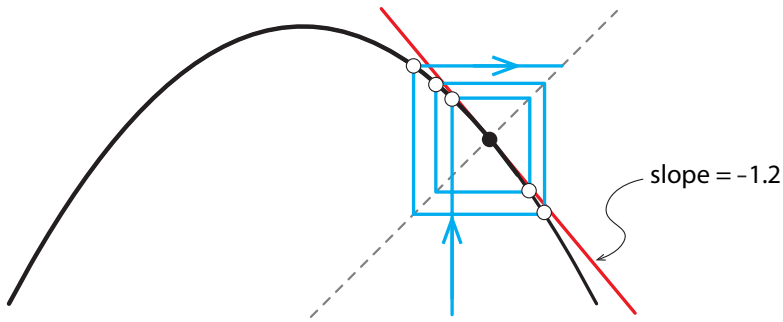


Figure 5.17: When the slope of the tangent line (red) at the equilibrium point has absolute value greater than 1, the cobweb process diverges, and the equilibrium point is unstable.

So our criterion for *in*stability becomes

$$\left|\frac{df}{dX}\right|_{X_{eq}} = |2 - r| > 1$$

Let's deal separately with the two cases that are included in the notion of absolute value. First of all, if $r < 2$, then $|2 - r| = 2 - r$, which makes our instability criterion

$$2 - r > 1 \quad \text{or} \quad r < 1$$

but if $r < 1$, then we know that the equilibrium point has to be stable, so this is absurd.

In the other case, if $r \geq 2$, then $|2 - r| = r - 2$, which makes our instability criterion

$$r - 2 > 1 \quad \text{or} \quad r > 3$$

and so we have answered our question: the equilibrium point becomes unstable when $r > 3$.

Exercise 5.3.4 In Exercise 5.3.2, you found the equilibria of the Ricker model. At what value of r does the equilibrium point at $X = k$ become unstable?

Let's look at what happens when we increase r in the discrete logistic model. When, for example, $r = 3.35$, a simple 2-point periodic attractor arises. The system's attractor consists of two points, and the final behavior is an oscillation between these two points: it goes $A, B, A, B, \ldots$ (Figure 5.18).

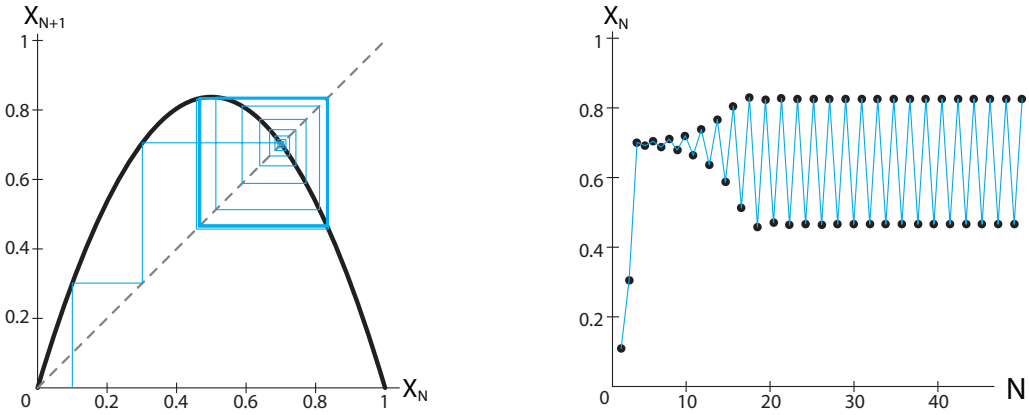


Figure 5.18: Discrete logistic model with $r = 3.35$.

If we raise r further, for example, $r = 3.53$, the 2-point oscillation is lost and is replaced by a 4-point oscillation $A, B, C, D, A, B, C, D, \ldots$ (Figure 5.19).

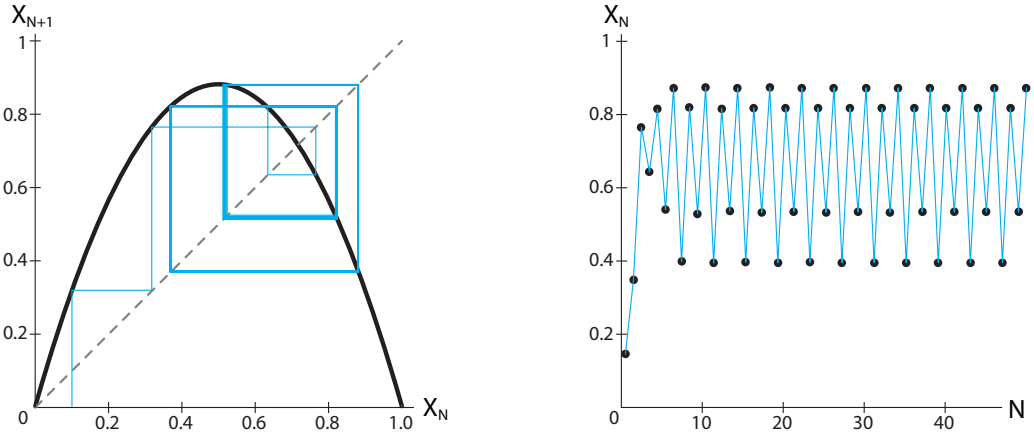


Figure 5.19: Discrete logistic model with $r = 3.53$.

As r increases, the 4-point oscillation gives way to an 8-point oscillation, and these kinds of bifurcations, called *period-doubling bifurcations*, occur faster and faster, until a limit point is reached and the system behavior becomes truly chaotic (Figure 5.20). Consider the time series.

For most values of r between 3.57 and 4, it is typical. The graph certainly looks irregular. For these values of r, the dynamics of the discrete logistic model are aperiodic.

Nevertheless, as we already observed, there is some structure to this time series; it isn't completely random. For example, note the point where the graph of the function intersects the graph of $X_{N+1} = X_N$, which is the (unstable) equilibrium point. We noted earlier that when X goes near 0.75 the changes are small. Now we can explain why: $X = 0.75$ is the equilibrium point of this system. Changes *at* the equilibrium point are 0, by definition, and the derivative is continuous, and therefore, changes *near* the equilibrium point will be near 0.

This sequence of bifurcations is called the *period-doubling route to chaos*. We can make a bifurcation diagram representing the period-doubling route. We will use a technique similar to the one we used in Chapter 3 to construct bifurcation diagrams. In that situation, we stacked up 1D state spaces, one for each parameter value r, and showed the location of the equilibrium points (Figure 5.21). Now we will do the same thing, but plot **all** the state values the system visits after an initial transient.

This diagram is read as follows: each value of r on the X axis represents one model. On the vertical line corresponding to that r-value, we plot all the points of the behavior for large N. Thus, for values of r less than 3, there is only one point per r value, indicating that the system

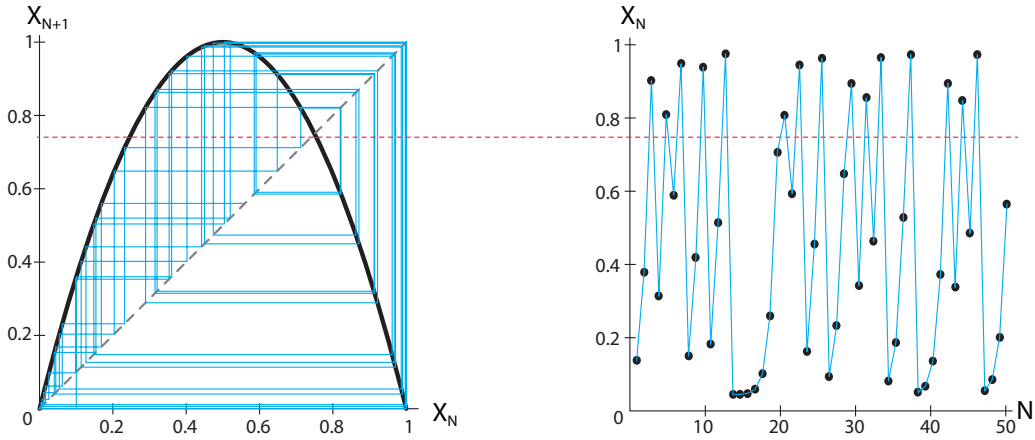


Figure 5.20: Discrete logistic model with $r = 4$.

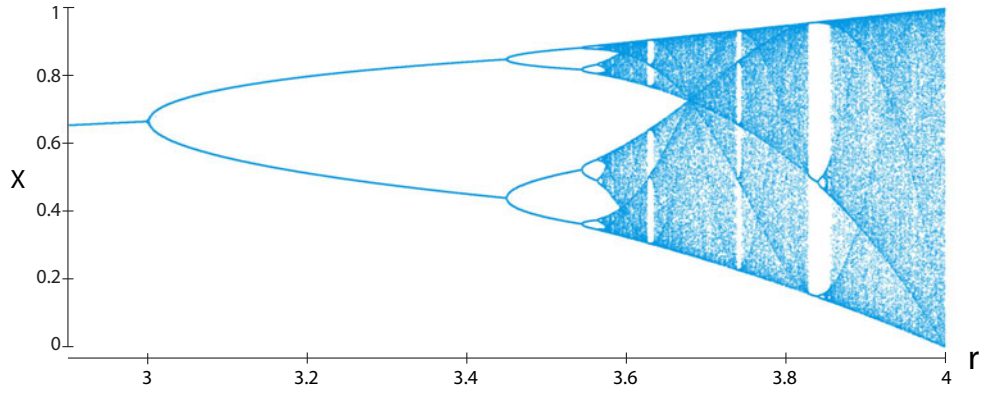


Figure 5.21: Bifurcation diagram for the discrete logistic model.

has a stable equilibrium point at that value of N (so the location of the stable equilibrium point increases slightly with r). But at $r = 3.0$, a bifurcation happens, and the presence of two points per r value indicates that the system now has a period-2 attractor, and that it cycles between the two points. Then for r greater than 3.4 another period-doubling occurs, and we now have four points for each r value. Finally, the presence of many (actually infinitely many) values for most $r > 3.6$ indicates the presence of chaos.

If we mark the four examples above on this bifurcation diagram, we see that it correctly predicts the behavior of the logistic model (Figure 5.22).

Exercise 5.3.5 Use Figure 5.21 to describe how the discrete logistic model will behave for $r = 3.1$, $r = 3.5$, and $r = 3.7$.

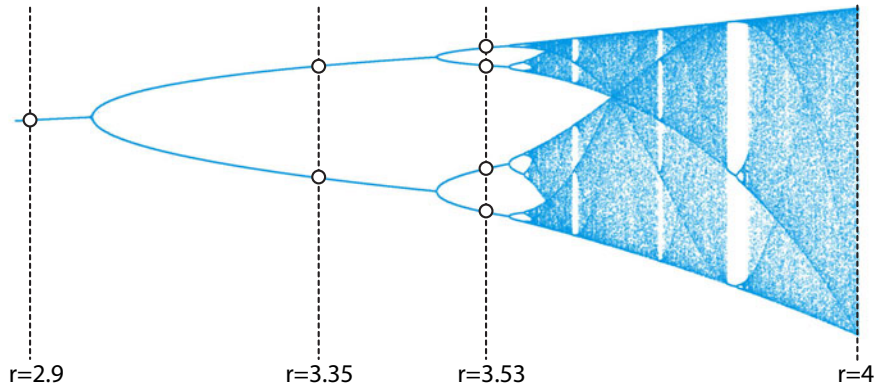


Figure 5.22: Different r values in the bifurcation diagram for the discrete logistic model.

Other routes to chaos, other sequences of bifurcations leading to a chaotic attractor, have been identified both mathematically and in physical systems. In most of them, complex oscillations are way stations on the route to chaos. That is,

$$\text{equilibrium} \longrightarrow \text{oscillation} \longrightarrow \text{complex oscillation} \longrightarrow \text{chaos}$$

is a frequent scenario.

A Period-Doubling Route to Chaos in the Three-Species Food Chain Model

The three-species food chain model offers an excellent example of an important route to chaos. Consider the parameter b_1 in the model (equation (5.1) on page 224). It controls the level of plants that the herbivores can consume. If b_1 is low, the herbivores can consume a large fraction of the plants, and the consumption therefore saturates quickly.

But if b_1 is increased, the herbivores can consume more and more of the plant mass. If we increase b_1 from 2 to 3, we see a sequence of changes. For $b_1 = 2$, the model has a stable equilibrium point (Figure 5.23). As we raise b_1, we see first a Hopf bifurcation (Figure 5.24). Now the equilibrium point is unstable, and a stable limit cycle attractor is born. Then, as b_1 is increased, another bifurcation occurs, from the simple oscillation to a more complex one with twice the period. Now the oscillations have an alternating $A, B, A, B, \ldots$ pattern. This is therefore a period-doubling bifurcation (Figure 5.25).

As b_1 increases further, there is another period-doubling bifurcation to a period-4 rhythm (Figure 5.26), and finally, further increases in b_1 produce chaos (Figure 5.27).

Figure 5.23: For low values of b_1, the system has a stable equilibrium point of spiral type.

Figure 5.24: For slightly higher values of b_1, the system exhibits a stable oscillation.

Figure 5.25: As b_1 further increases, a period-doubling bifurcation occurs, and the rhythm becomes more complex.

Figure 5.26: Still further increases in b_1 cause a second period-doubling bifurcation, to an even more complex periodic rhythm.

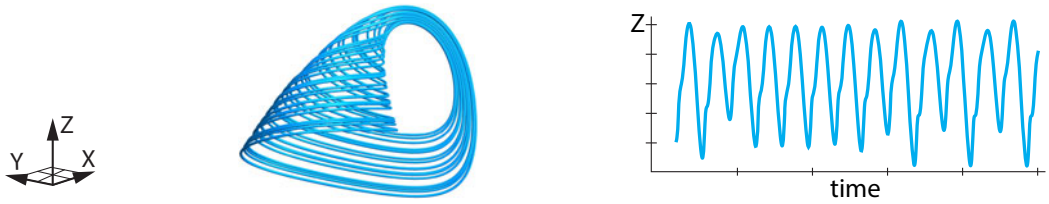


Figure 5.27: Increasing b_1 even more produces a bifurcation to a chaotic attractor.

Further Exercise 5.3

1. In Exercise 5.2.3 on page 235, you learned about the Romeo, Juliet, and Juliet's nurse model (Lorenz model),

$$J' = s(N - J)$$
$$N' = rJ - N - RJ$$
$$R' = JN - bR$$

   a) Simulate this model with the parameter values $s = 10$, $b = 8/3$, and $r = 1$, and the initial conditions $J(0) = 0.1$, $N(0) = -6$, and $R(0) = 0.01$. Use a step size of 0.01. Plot trajectories and time series of your simulation and describe the system's behavior.

   b) Find out what kinds of behavior you can generate by manipulating r. *(Hint: Try making an interactive.)*

5.4 Stretching and Folding: The Mechanism of Chaos

Chaos has typical kinds of causes. Let's use the discrete logistic equation as an example. The inverted parabola shape of the function suggests that there are two main processes in this model (Figure 5.28):

1) a *growth* process represented by the left-hand part of the curve, above the 1-1 reference line. In this region, X_{N+1} is greater than X_N, and
2) a *crowding* or shrinking process represented by the right-hand part of the curve, below the 1-1 reference line. In this region, X_{N+1} is less than X_N.

Suppose the system begins with an initial condition on the left-hand side (small X_0). Then it is in growth mode, and the population will begin by growing exponentially. As the value nears the center ($X = 0.5$), we see the crowding term beginning to flatten out the curve and turn it downward. As long as the point is to the left of the equilibrium point, X_{N+1} is larger than X_N, but as the population grows, it eventually reaches the right side. Then X_{N+1} is less than X_N, so the population is shrinking. This takes the state point back to the left-hand side, and another growth–decline cycle begins. The state point is like a tennis ball being batted back and forth between the growth mode and the shrinking mode.

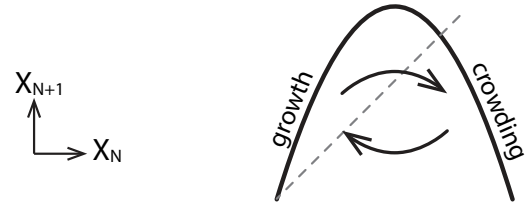


Figure 5.28: Schematic illustration of the dynamics of the parabola function. The part of the curve above the 1-1 reference line causes population growth, while the part below the 1-1 reference line causes the population to shrink.

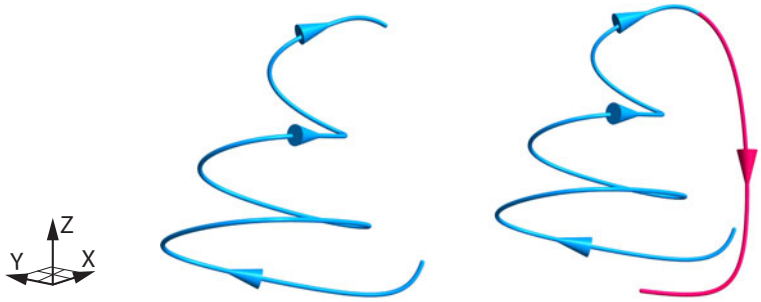


Figure 5.29: Two stages of a typical trajectory in the three species food chain model. Starting from an initial condition at the bottom right, at first the state point spirals inward in the X-Y plane while heading upward in the direction of increasing Z (blue segment). Then, at high Z, the Z population crashes (red segment), which releases the predation pressure on Y and X, allowing them to return to high values.

It is typical of chaotic systems that a careful look at their attractors will reveal an interesting causal mechanism: the picture tells a story. For example, let's look at the three-species food chain model (Figure 5.29). If we follow the state point around the attractor, we see that there are basically two modes:

1) If we start anywhere in the jug itself, the dynamics of X (= plant) and Y (= herbivore) oscillate like a shark–tuna model in the X-Y plane, but with slowly diminishing amplitude (like the spring with friction), while Z (= carnivore) grows slowly.
2) Finally, Z grows so large that the state point is sent into the handle, where it plummets downward rapidly. The crash in the Z population (which is caused by the decrease in Y, its food) then takes the pressure off Y, and the cycle begins again.

Thus, the ecosystem can be seen as two interacting oscillatory systems. The X-Y oscillator is the plant/herbivore oscillator; it is similar to the Holling–Tanner model we developed earlier, which displayed stable limit cycle attractors.

But now this X-Y oscillation is coupled to another oscillatory process, the Z-Y oscillation, in which the carnivore preys on the herbivore in a second cyclic process. Chaos as a result of the interaction of two coupled cyclic processes is a frequent scenario.

Regardless of the shape of the attractor and whether it is a discrete-time or a continuous system, there is a universal mechanism at work in all chaotic attractors that accounts for most of their interesting behaviors: **stretching and folding**.

The purest example illustrating the stretching and folding process is called the baker's transformation (Figure 5.30). Imagine a piece of dough that is repeatedly rolled flat and folded over. First, the rolling pin spreads out the dough twice as wide (steps 1 and 2), then the spread-out dough is folded over to recreate a two-layered structure that has exactly the same dimensions as the original (steps 3 and 4). If we repeat this process a second time (steps 5 through 8), we get a four-layered structure of the same dimensions as the original. If this stretching and folding process is repeated N times, the result is to create a layered structure with 2^N layers exactly occupying the original volume.

Consider the fate of two points that are initially extremely close, say in the left eyebrow. Note that in steps 6 and 7, the left eyebrow was divided into two pieces that were assigned to different layers in step 8. As this process continues over many iterations, the left eyebrow will become completely fragmented, and the fate of the two closely spaced points will diverge, so that knowing the location of one does not help us find the other.

Mathematically, the baker's transformation can be written as a two-variable discrete-time system. Assume that the dimensions of the square are 1×1. If (X_N, Y_N) is the location of a point at time N, that point is transformed to

$$(X_{N+1}, Y_{N+1}) = \begin{cases} (2X_N, 0.5Y_N) & \text{if} \quad X_N < 0.5 \\ (2 - 2X_N, 1 - 0.5Y_N) & \text{if} \quad X_N \geq 0.5 \end{cases}$$

Exercise 5.4.1 Pick two neighboring points and follow them through the baker's transformation for five steps. Where does each point end up? What happens to the distance between them?

A Recipe for Chaos

"... start pulling with your fingertips, allowing a spread of about 18 inches between your hands. Then fold it back on itself. Repeat the motion rhythmically."

— *The Joy of Cooking*
I.S. Rombauer and M.R. Becker
(citation from Ian Stewart, *Does God Play Dice?* (Stewart 1997))

The stretching and folding process is at work in every chaotic attractor. For example, let's look at the discrete logistic model. Let's begin with a small section between 0.1 and 0.2, represented by the horizontal red bar on the X axis (Figure 5.31, top left). (Figure 5.31, top right).

This piece of the interval, when it is shot up to the graph of the function, produces a larger interval as its output. This is the red bar on the Y axis.

Thus, the original bar of initial conditions has been stretched by the function, because the slope of the function in this region is greater than 1.

Then we take the stretched bar and reflect it back down onto the X axis.

When now we shoot this bar up to the function and project it onto the Y axis, we see that the projection is not 1-to-1: two points in the bar on the X axis are sent to the same output value on the Y axis. This is the folding process.

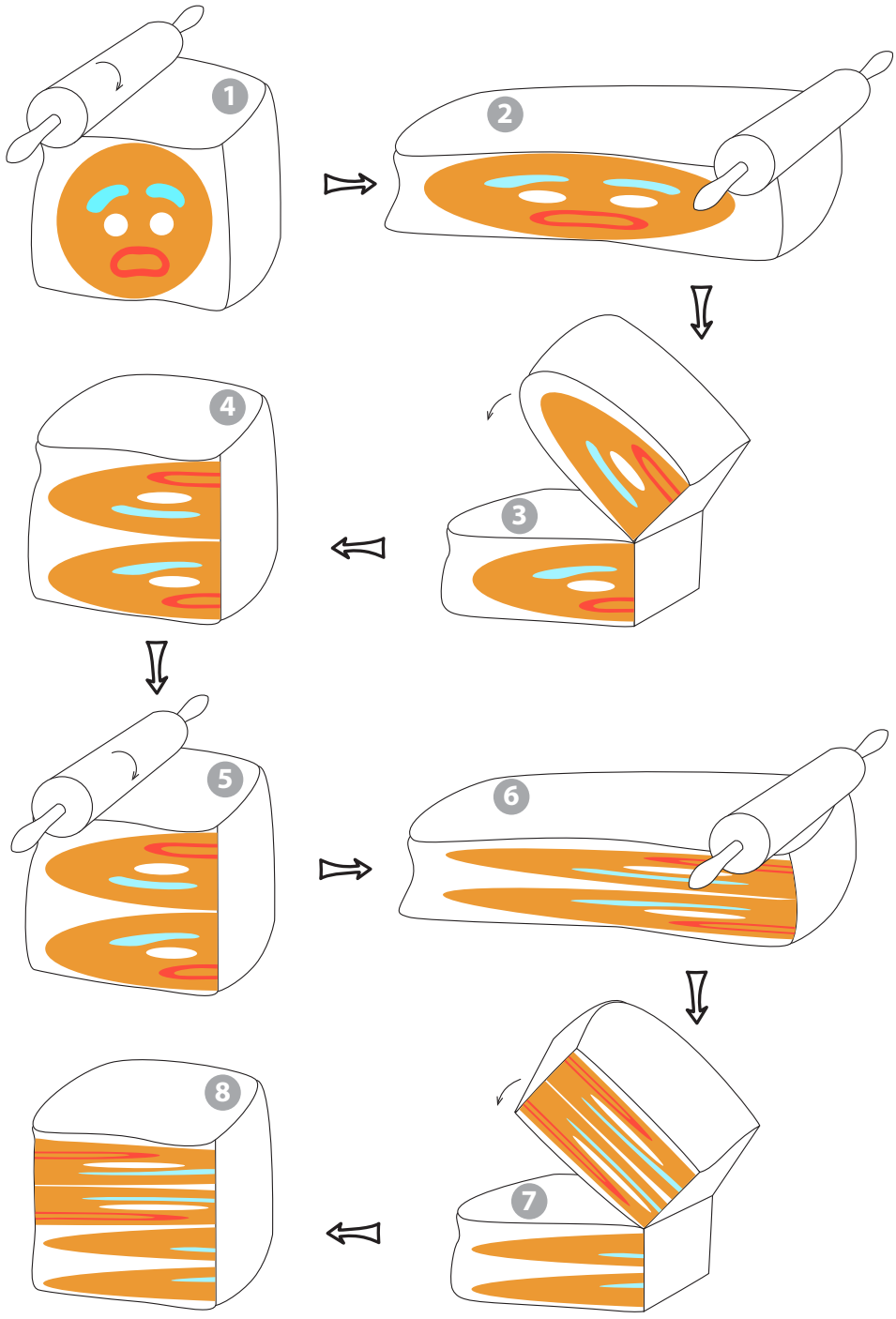


Figure 5.30: The baker's transformation. From a simple initial condition, repeated stretching and folding produces an intricate layering. Note that parts of the eyebrows can now be seen in all of the layers.

Then, to begin the next cycle, the red bar at the bottom right is shot up to the function again and becomes stretched again.

Thus, there is a "Joy of Cooking" stretching and folding process within the discrete logistic function.

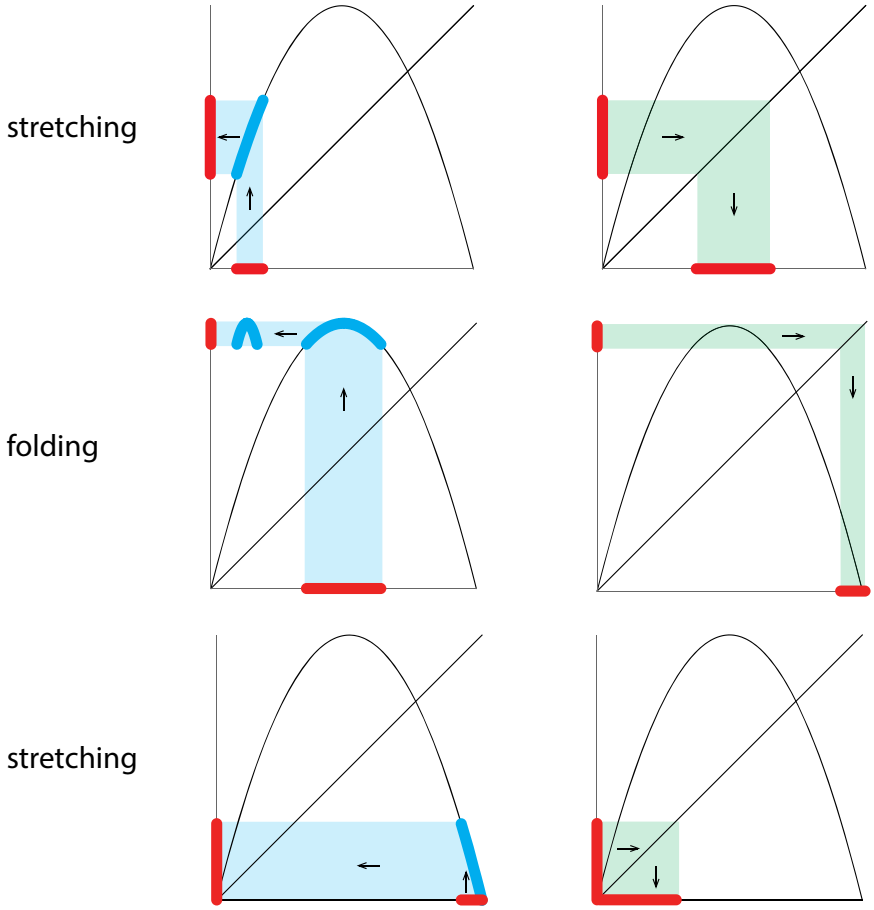


Figure 5.31: Stretching and folding in the discrete logistic function. Upper left: a small interval of initial conditions on the horizontal axis (red bar) is stretched by applying the function f to all the points (larger red interval on the vertical axis). Upper right: we reflect the larger interval back onto the horizontal axis using the 1-1 reference line. Middle left: Applying the function f to the new stretched interval results in a folded interval, because two different values of X_N are assigned the same value of X_{N+1}. Middle right: the newly folded interval is projected back onto the horizontal axis. Lower left: applying the function f to the folded interval produces a stretched version of the folded interval. Lower right: the new interval is projected back onto the horizontal axis.

The same thing is true of the three-species food chain model. We began with a point cloud of 10,000 initial conditions in a very small region (top left, black arrow in Figure 5.32).

As the point cloud went around the attractor, it was stretched. By the third time around the attractor, it had elongated into a stringlike structure. Then, the fourth time around, $t = 400$, the

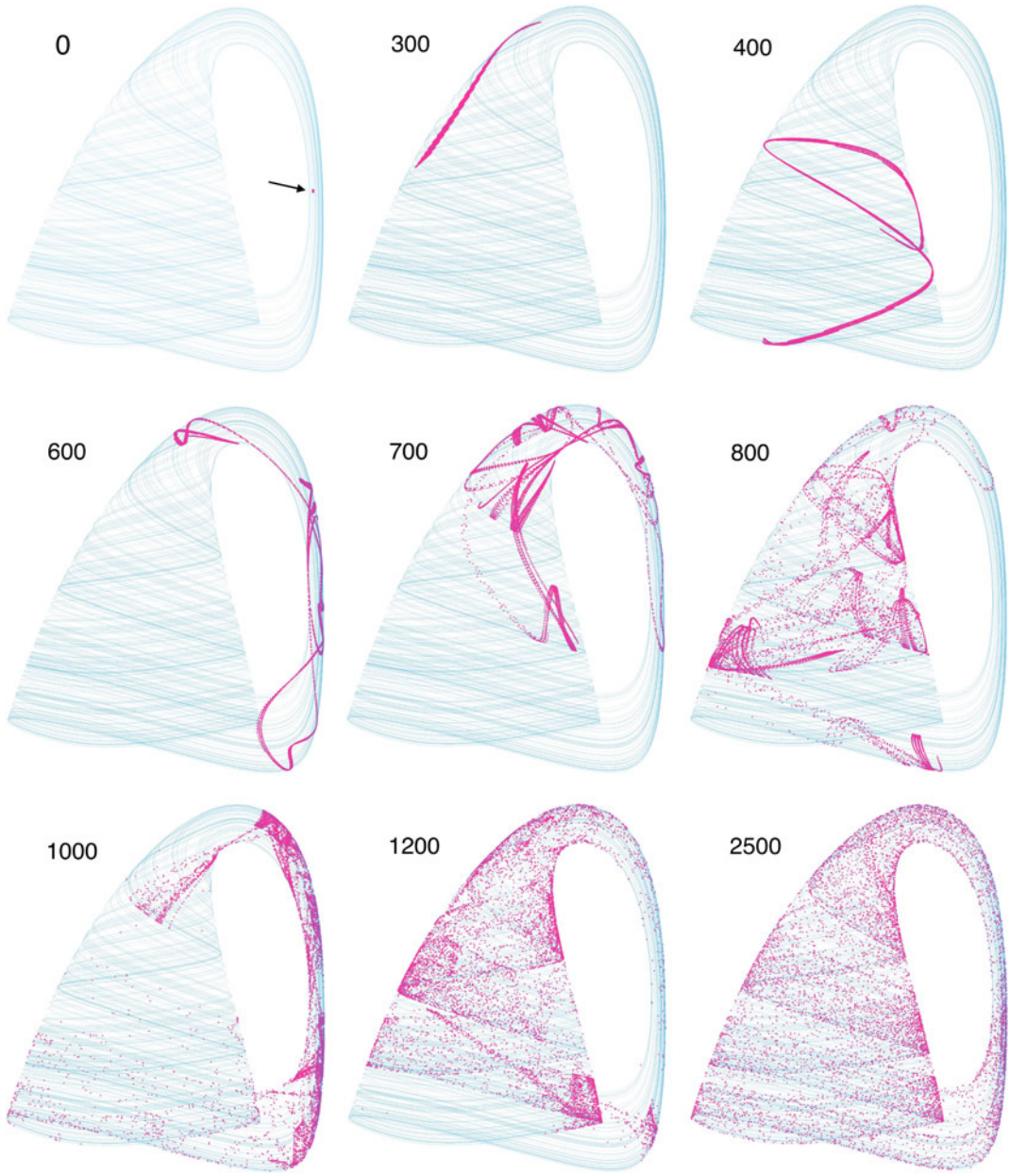


Figure 5.32: Chaos by stretching and folding on the food chain attractor. At $t = 0$, 10,000 initial conditions, packed very closely (black arrow, top left), were evolved forward in time by the three-species Hastings model. At $t = 300$, three times around the attractor, the initial dot has stretched into a long filament. By $t = 400$, the filament has stretched further and folded over. Further evolution shows repeated stretching and folding, with a resulting fragmentation of the filament. Sensitive dependence on initial conditions ends up spreading the initial 10,000 points across the whole attractor.

string was further elongated and was folded in half. Further trips around the attractor continue the stretching and folding process.

At $t = 2500$, the resulting fragmentation has distributed points from the original initial conditions broadly across the attractor, so that points that were initially very close together are now far-flung across the attractor.

Prediction of the detailed fate of a point has obviously become impossible.

The fact that the original tiny ball of initial conditions is now spread out across the attractor by the repeated stretching and folding is called the "mixing property" of chaotic systems. The mixing property can also be seen in the baker's transformation: after many iterations, pieces of the left eyebrow are found in every layer.

It is interesting that this mixing process is actually used to mix things! A thousand years ago, Japanese swordsmiths needed to mix two metals to make their best sword.

But the metals could not be simply melted and stirred, because melting and cooling would destroy their desirable properties. So they needed a way to cold mix two metals. They hit on the idea of placing sheets of the two metals on top of each other, hammering them down into a thinner sheet, and folding the result over. This process would be repeated, and in a manner identical to the baker's transformation, they were able to mix the two metals effectively at room temperature.

Further Exercise 5.4

1. In Further Exercise 5.1.1 and Further Exercise 5.1.2, you learned about the Beverton–Holt ($X_{N+1} = \frac{rX_n}{1+X_n/m}$) and Ricker ($X_{N+1} = X_N e^{r(1-\frac{X_n}{k})}$) population models. The Ricker model can generate chaos, while the Beverton–Holt model cannot. Use what you learned in this section to generate a hypothesis about why this is the case. *(Hint: Plot the functions.)*

5.5 Chaos in Nature: Dripping Faucets, Cardiac Arrhythmias, and the Beer Game

It's easy to diagnose chaos in a differential equation or a discrete-time dynamical system.

(1) Is it behaving irregularly? Yes.
(2) Is there any random input into the system? No.
(3) Then it is chaotic.

But what about real systems? Can we observe erratic behavior in nature and determine whether it is random or chaotic? Frequently, the answer is yes. We will illustrate this with a discrete-time example, but there are similar methods available for continuous-time differential equations.

The idea is this: deterministic chaos means that the future is determined by the past. In order to tell whether a system is deterministic (hence chaotic and not random), let's make a picture from the data of how X_{N+1} relates to X_N by plotting that relationship graphically.

For example, suppose we gathered some data from a natural system (Figure 5.33). We have no idea what the dynamics that produced it were. But if we took the data points $X_1, X_2, X_3, \ldots,$

and plotted them as X_{N+1} against X_N, in what is called a *Poincaré plot*, we would get Figure 5.34. The dots would lie exactly on the curve $X_{N+1} = 4X_N(1 - X_N)$, but they fill in that curve in what looks like a random manner, though of course it isn't.

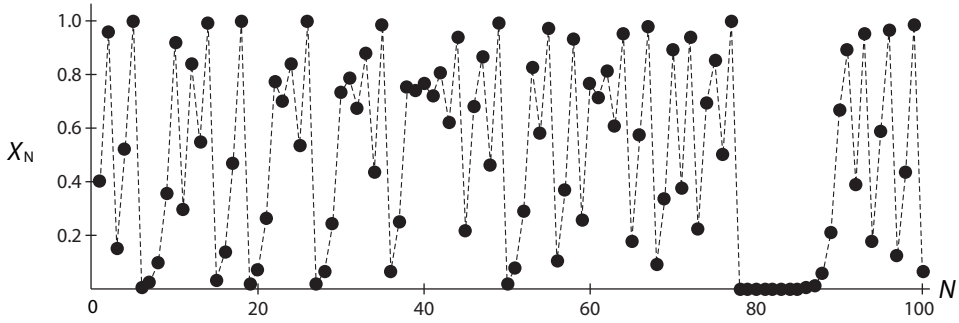


Figure 5.33: Time series of a typical output from the discrete logistic function in its chaotic regime.

Exercise 5.5.1 Why isn't it random?

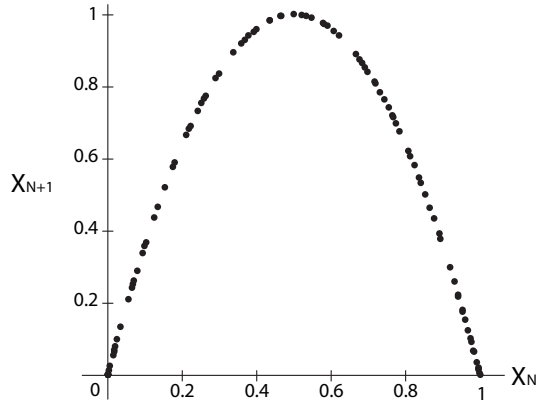


Figure 5.34: Poincaré plot of data from the previous figure.

If the Poincaré plot has some simple shape, with internal structure, then the behavior is not random, but chaotic.

If it is an oval blob-shaped cloud of points, we cannot rule out randomness.

Exercise 5.5.2 By hand, make a Poincaré plot of the values 4, 3, 10, 5, 12.

Dripping Faucet

Scientists have done such data analysis for some basic examples in natural systems and gotten surprising results.

One beautiful example is the study, led by Rob Shaw at UC Santa Cruz, of the behavior of a dripping faucet (Martien et al. 1985). We all know that faucets can sometimes drip regularly, with a periodic drip-drip-drip. And we also know that if we turn the handle all the way, and open the faucet sufficiently, we can get full-on continuous flow. But what about intermediate values of the handle position, between regular dripping and continuous flow? It is easy to produce irregular dripping. You can try this with a sink at home.[2]

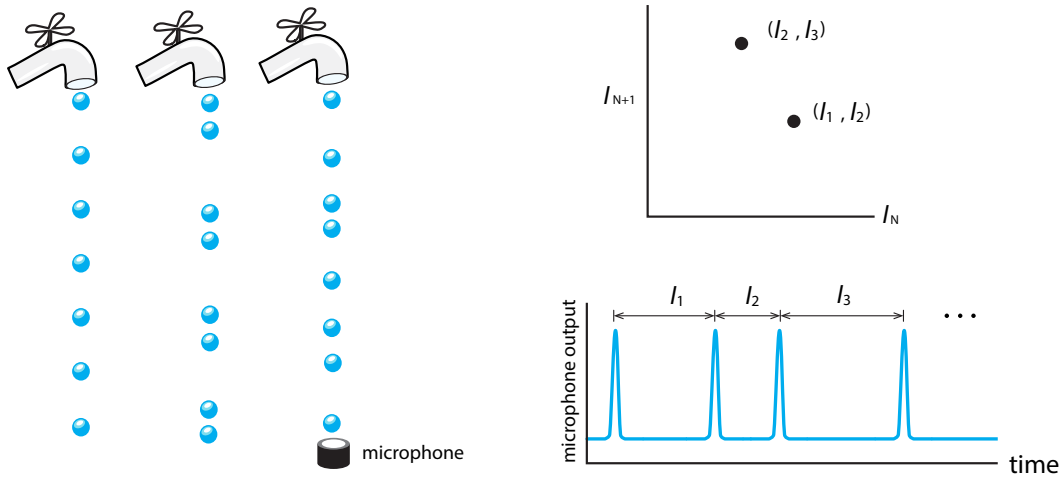


Figure 5.35: Apparatus of the dripping faucet experiment. The falling drops land on a microphone, and recording of the sound is used to calculate the interdrip intervals $I_1, I_2, I_3, \ldots$. Then a Poincaré plot is made, plotting I_{N+1} against I_N (adapted from (Crutchfield et al., 1986)).

Is this irregular dripping random or chaotic? The Santa Cruz group set up a faucet in a lab and began making measurements (Figure 5.35). The group took precise measurements of the time intervals between drips. Calling the Nth interdrip interval I_N, they made Poincaré plots of I_{N+1} against I_N (Figure 5.36).

Note the clear indication of shapes (not blobs) in the Poincaré plot. Several features of these plots are worth noting. The specific shapes that these data form suggest functions that are known to produce chaos as dynamical systems $X_{N+1} = f(X_N)$. For example, all "functions" have apparent equilibrium points, points where the data cross the imaginary diagonal 1-1 line. Also, if we draw that line and look at the intersection point, the slope of the "function" at the intersection is steeper than -1, which is the requirement for an unstable equilibrium point.

We can conclude that the behavior of the dripping faucet is chaotic, not random.

The process underlying the dripping faucet chaos is worth examining in detail, because it reveals a simple and general mechanism for generating chaos. When a drop forms at the mouth of the faucet, it begins to balloon outward and downward due to its growing mass. The descending droplet pinches in, and then the neck separates and the detached drop falls downward. But the process isn't over. There is a final step that most people don't notice: when the drop separates,

[2] Faucets without aerators work better than those so equipped.

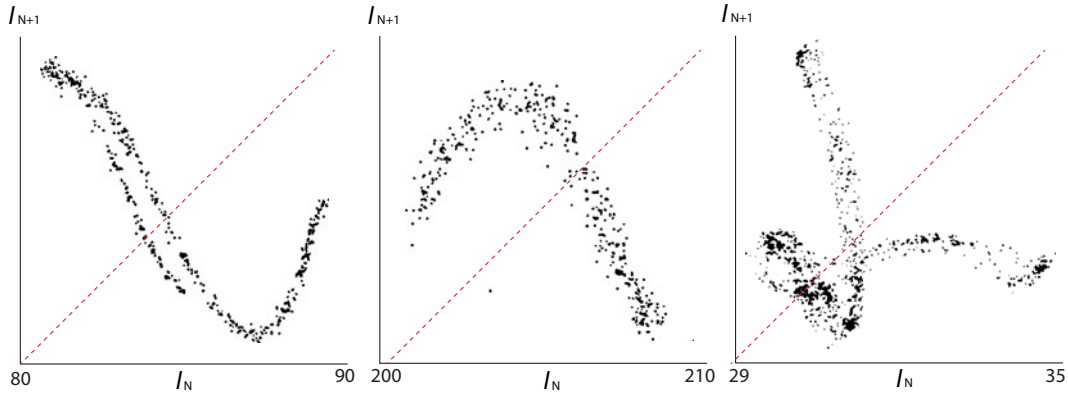


Figure 5.36: Three examples of Poincaré plots of interdrip intervals in the dripping faucet. Reprinted from *Physics Letters A*, 110(7), P. Martien, S. Pope, P. Scott, and R. Shaw, "The chaotic behavior of the leaky faucet," pp. 399–404, copyright 1985, with permission from Elsevier.

there is a small undropped part that snaps back (due to surface tension) and gives a small elastic oscillation as it retracts (Figure 5.37).

We therefore have a process that has two characteristic time intervals in it:

(1) the drop formation process has a characteristic time interval that is set by the flow rate, controlled by the handle. For slow dripping, this is ≈ 1 sec.

(2) the snap-back of the unseparated part of the drop is faster, at ≈ 0.1 sec.

For low flow rates, which produce slow dripping, the two processes do not interact due to the wide separation in their characteristic times; by the time the next drop forms, the snapback from the previous drop is completed, and the system has fully recovered from the previous drop.

But at higher flow rates and hence faster dripping, when the $(N+1)$st drop begins to separate, the system has not fully recovered from the Nth drop.

Now the exact state of the recovery from the Nth drop affects the timing of the $(N+1)$st drop.

In particular, if the little oscillation in the undropped recoiling part is in its downward phase when the next drop is near separation, that slightly retards the separation, whereas if it is in its upward movement, separation comes faster.

Thus, for high rates of dripping, the timing of the $(N+1)$st drop depends on the precise state of the recovery from the previous drop. *This is a recipe for chaos*: if a process consists

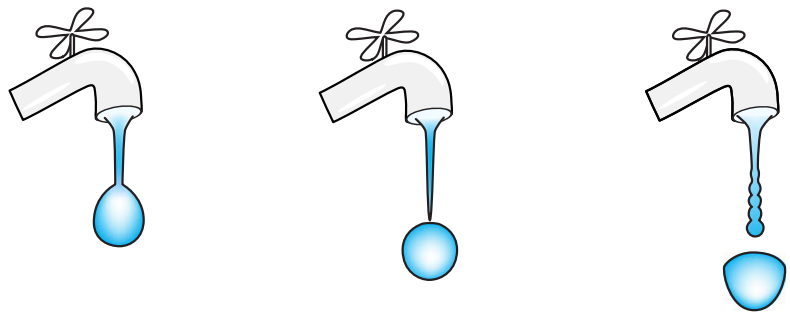


Figure 5.37: The process of drop formation and separation in a dripping faucet.

of an action phase followed by a recovery phase, and the process is pushed to high rates, then the formation of the next action phase depends on the state of the recovery from the previous action.

This is the mechanism that produces irregular dripping.

Cardiac Arrhythmia

The same mechanism of chaos can be identified in a very different subject: cardiac arrhythmia.

Cardiac researchers at UCLA studied a cardiac arrhythmia induced by drugs in a piece of heart tissue (Garfinkel et al. 1992). At lower doses of the drug, the tissue beat periodically, but as the drug dose was increased, irregular beating set in. The researchers made Poincaré plots of the interbeat intervals, which revealed several properties that are diagnostic for chaos.

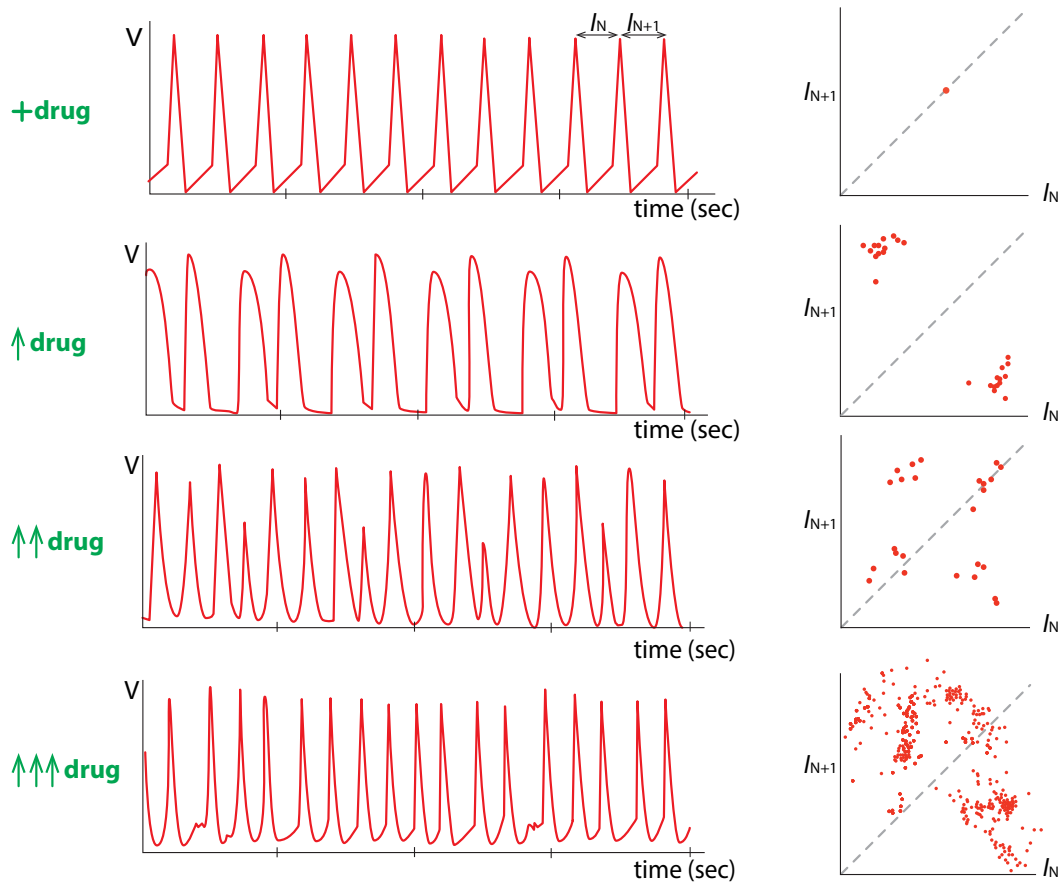


Figure 5.38: Times series and Poincaré plots of interbeat intervals from a drug-induced cardiac arrhythmia. From top to bottom, the drug dose is increasing. Left: time series of voltage recordings from the tissue. Right: Poincaré plots of interbeat intervals. Top row: low doses of the drug produce periodic beating. Second row: higher drug levels induce a period-2 rhythm. Third row: still higher drug levels produce a period-4 rhythm. Bottom row: at very high levels, the drug produces irregular beating, a cardiac arrhythmia. Redrawn from " Controlling cardiac chaos," by A. Garfinkel, M. Spano, W. Ditto, and J.N. Weiss, 1992, *Science* 257:1230–1235. Reprinted with permission from AAAS.

First of all, a clear route to chaos was seen as the drug dose was increased. As the drug dose was increased, the first changed after the periodic rhythm was a period-2 rhythm, consisting of an *A*-shape and a *B*-shape, alternating with each other. A further increase in drug produced a change to a period-4 rhythm, and then a still further increase produced a transition to irregular beating (Figure 5.38).

Notice also that the shape of the Poincaré plot suggests a function. In fact, it resembles the parabolic form of the discrete logistic function. If you drew a function that had that shape and iterated it, the result would be chaotic. Note that it has an unstable equilibrium point, at which its slope is steeper than -1.

The mechanism of this arrhythmia is, surprisingly, similar to the mechanism of chaos in the dripping faucet. In order to see this, let's briefly review cardiac physiology.

The heart is a muscle, and as in any muscle, muscular contraction is created by an electrical activation exactly like the action potential of the neuron.

In the cardiac cell, ions such as sodium (Na^+), potassium (K^+), and calcium (Ca^{2+}) are maintained at steady-state levels by the cell machinery. When a contraction is called for by an electrical stimulus (from the heart's natural pacemaker or experimentally by an external stimulus), Na^+ ions rush into the cell and elevate its voltage, and then K^+ ions rush out of the cell to restore the cell's voltage to the steady state. But the process isn't over: pumps now go to work, pumping the sodium out of the cell and the potassium back into it. This is the recovery phase.

The electrical activation and return to baseline voltage is called the **cardiac action potential**. The cardiac action potential and the following ionic restoration phase make up the cardiac cycle (Figure 5.39).

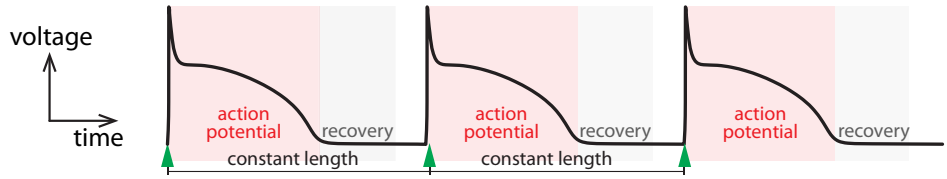


Figure 5.39: Phases of the cardiac cell. At slow periodic pacing (green triangles), the next stimulus occurs after the end of the recovery phase from the previous action potential.

If the heart is paced (stimulated) slowly, the recovery from the Nth action potential is fully completed by the time the $(N+1)$st action potential forms.

But for faster pacing, the formation of the $(N+1)$st action potential is affected by the precise state of the recovery from the Nth action potential, and chaos ensues, a process identical to that of the dripping faucet.

We can reproduce this phenomenon in a simulation experiment. The cardiac action potential has been intensely modeled using differential equations. One of the early cell models was the Luo–Rudy model (Luo and Rudy 1991). We will use this as our experimental cell model and pace it at varying rates.

When the periodic pacing stimulus is at a slow rate, such as every 400 milliseconds, the result is perfectly periodic beating (Figure 5.40).

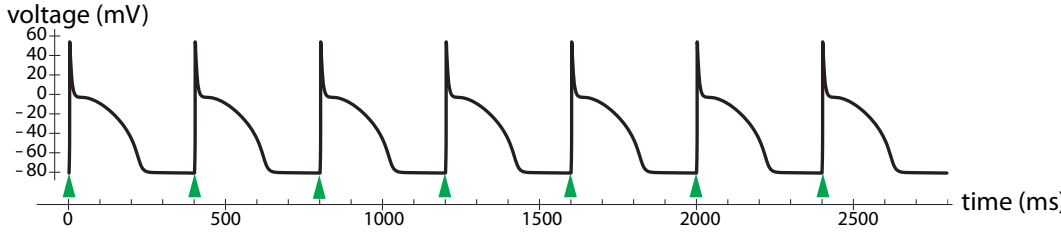


Figure 5.40: Slow pacing (every 400 ms) produces a periodic train of action potentials.

But if we pace the cell much faster, such as every 100 milliseconds, the next stimulus occurs during the recovery phase of the previous action potential, resulting in chaos (Figure 5.41).

Of course, we can conclude that this irregular beating is chaos, because this irregularity is being produced by a differential equation with no random input. However, we can further demonstrate that this is mathematical chaos by first turning it into a discrete time system, and then making a Poincaré plot.

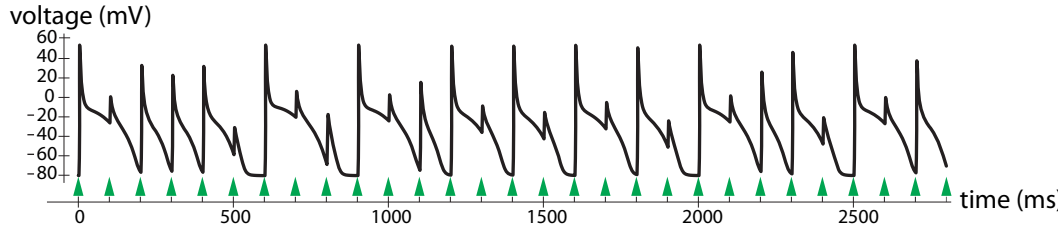


Figure 5.41: Fast pacing (every 100 ms) produces a chaotic response.

We will use the device of plotting the duration of the $(N+1)$st action potential against the duration of the Nth. We will draw a horizontal line at $V = -60$mV and count as the action potential duration the time spent above this line. If we do this for a long train of stimulated action potentials, we get a striking picture: a two-piece function with steep negative slopes. This is another example of a known chaos-generating function when considered as a discrete-time dynamical system (Figure 5.42 and Exercise 5.5.3).

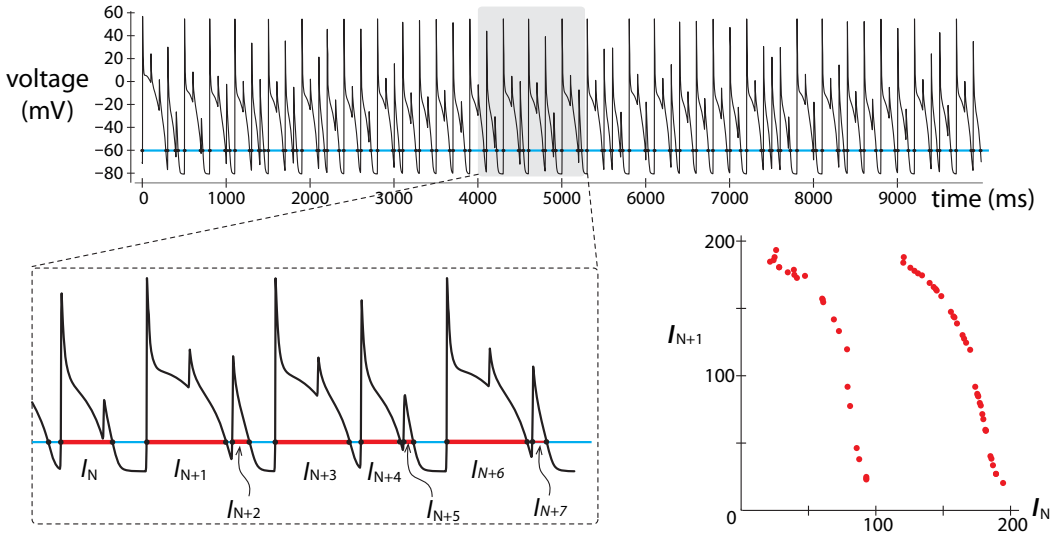


Figure 5.42: Top: pacing the cardiac cell model at a rapid rate (every 100 milliseconds) produces an irregular train of action potentials. Lower left: if we measure the action potential duration as the time spent above $V = -60$ mV (blue line), we can record the sequence $I_1, I_2, I_3, \ldots$ of action potential durations (red segments). Lower right: plotting I_{N+1} against I_N produces a known chaos-generating function.

Exercise 5.5.3 The Poincaré plot in Figure 5.42 is drawn from simulation data. The I_N/I_{N+1} plot looks like a function. Stylize that function as

$$X_{N+1} = \begin{cases} 1 - 2X & if \quad 0 \leq X < 0.5 \\ 2 - 1.99999X & if \quad 0.5 \leq X \leq 1 \end{cases}$$

a) Plot this function. Does it resemble the Poincaré plot?
b) Use the function as a discrete-time dynamical system and iterate it for 100 steps. What behavior do you see?

Neural Chaos

The idea that rapid periodic pacing of an excitable system can result in a chaotic output is very general, and such results can be observed in many systems. We just saw it using a seven-variable cardiac cell model. But it is easy to produce the same phenomenon even in a very simple model of an excitable system.

In Chapter 4, we developed the FitzHugh–Nagumo (FHN) model of the neuron. It consisted of a fast inward phase and a slow recovery phase, summarized in a two-variable differential equation:

$$V' = \frac{1}{\epsilon}\Big(- w + f(V) + I_{ext}\Big)$$
$$w' = V - gw$$

Here we will use as our I_{ext} a periodic train of square-wave pulses. The pulses all have the same duration and amplitude, and we will vary the period of the stimulation. We see that for slow pacing (long period), the neuron responds in a one-to-one fashion with a periodic train of action potentials (Figure 5.43).

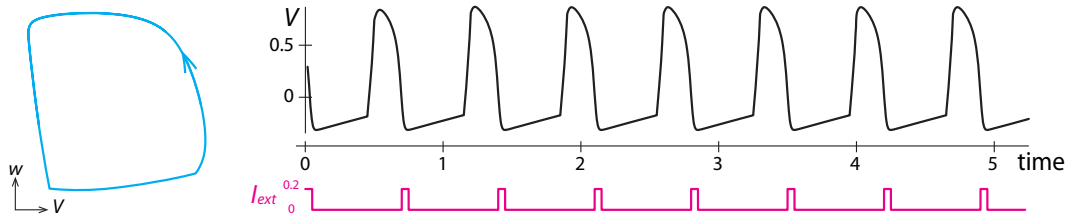


Figure 5.43: Period = 0.7.

If we increase the pacing rate, we see a sequence of bifurcations. First, we see a period-2 rhythm (Figure 5.44), then a period-4 rhythm (Figure 5.45).

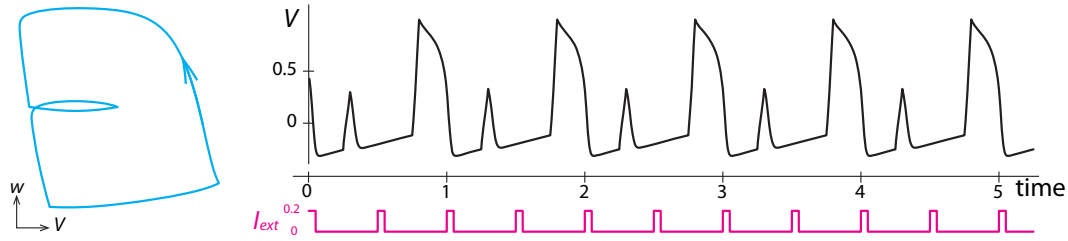


Figure 5.44: Period = 0.5.

Finally, when the pacing is rapid, we get a chaotic response (Figure 5.46).

These experiments with mathematical models can be confirmed by experiments in real neurons.

Hayashi, Aihara, and their colleagues did a number of studies in which they paced real neurons taken from animals from mollusks to mammals (Aihara et al. 1985; Hayashi et al. 1982). They found that under rapid pacing (they used sinusoidal stimuli instead of our square-wave pacing), the response of the neuron was chaotic (Figure 5.47). They confirmed that the response was chaos by constructing Poincaré plots of voltage, using a technique that is slightly different from

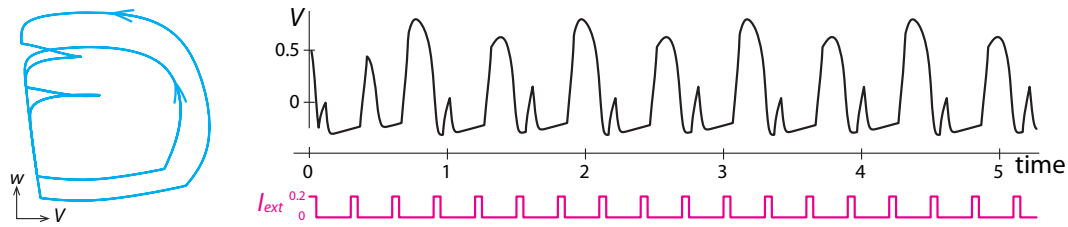


Figure 5.45: Period = 0.3.

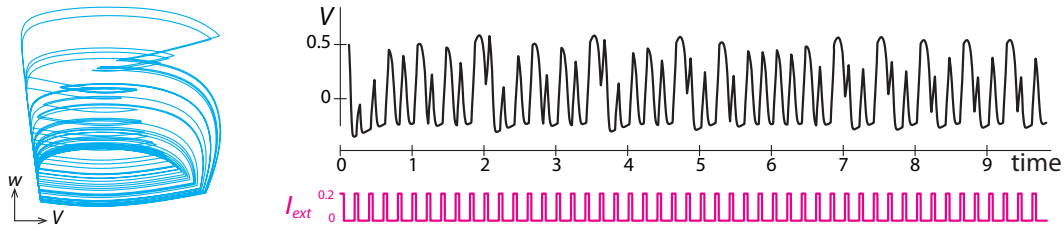


Figure 5.46: Period = 0.2.

the one we used, but equivalent. It is another way of constructing a discrete-time series from a continuous one. They took the continuous voltage record and made a "stroboscopic plot," in which a snapshot is taken of the voltage once each pacing cycle, at the same point in the cycle. This gives us a sequence of voltage snapshots $V_1, V_2, \ldots$. Then they plotted V_{N+1} against V_N for these values.

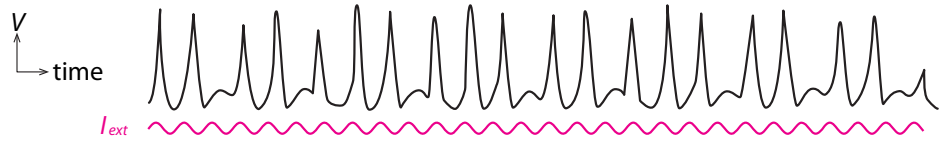


Figure 5.47: Chaotic response of a neuron to sinusoidal stimulation. Redrawn from *Physics Letters A* 111(5), K. Aihara, G. Matsumoto, and M. Ichikawa, 1985, "An alternating periodic-chaotic sequence observed in neural oscillators," pp. 251–255, Copyright 1985, with permission from Elsevier.

Their results are quite striking: unexpectedly simple figures, including plots that are functions, $V_{N+1} = f(V_N)$, that look like a cusp (Figure 5.48 right). The cusp-shaped function is akin to the parabola we studied earlier and is a function known to generate chaos when iterated.

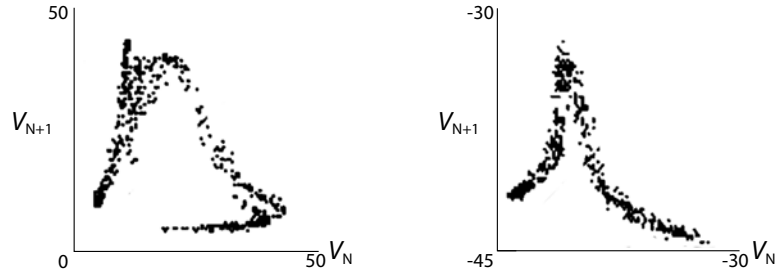


Figure 5.48: Poincaré plots of stroboscopic data from a neuron experiment. Here V is in millivolts. Reprinted from *Physics Letters A* 88(8), Hatsuo Hayashi, Satoru Ishizuka, Masahiro Ohta, and Kazuyoshi Hirakawa, "Chaotic behavior in the *Onchidium* giant neuron under sinusoidal stimulation," pp. 435–438, Copyright 1982, with permission from Elsevier.

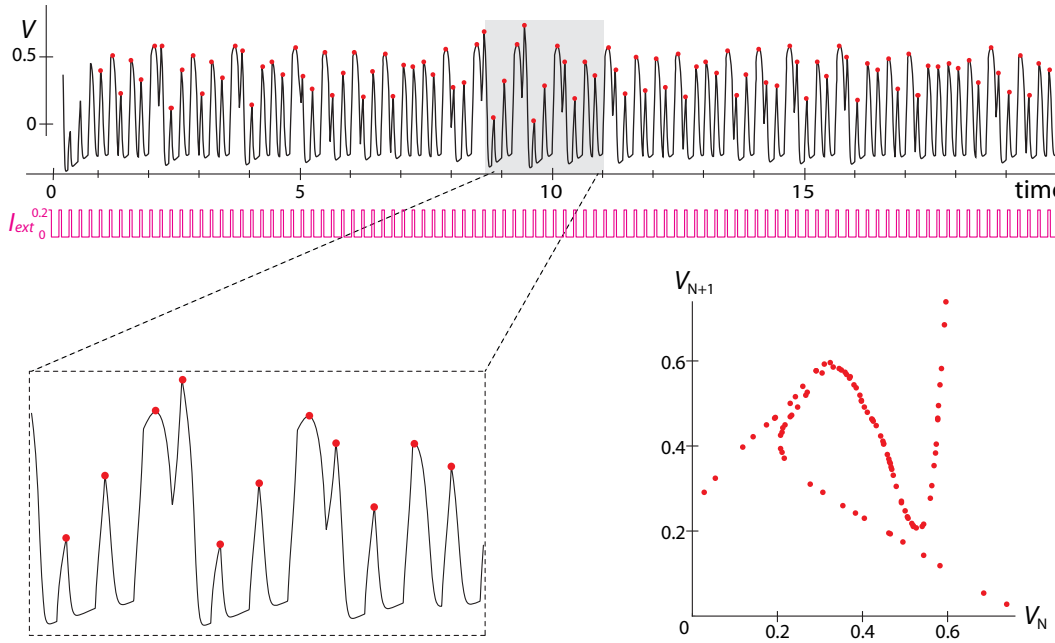


Figure 5.49: Upper: Time series of a periodically stimulated neural (FHN) model in a chaotic regime. Stimulus is shown below. Lower left: inset showing successive local maxima (red dots). The amplitude of each peak is recorded as V_1, V_2, V_3, ... Lower right: Poincaré plot of V_{N+1} against V_N for this time series.

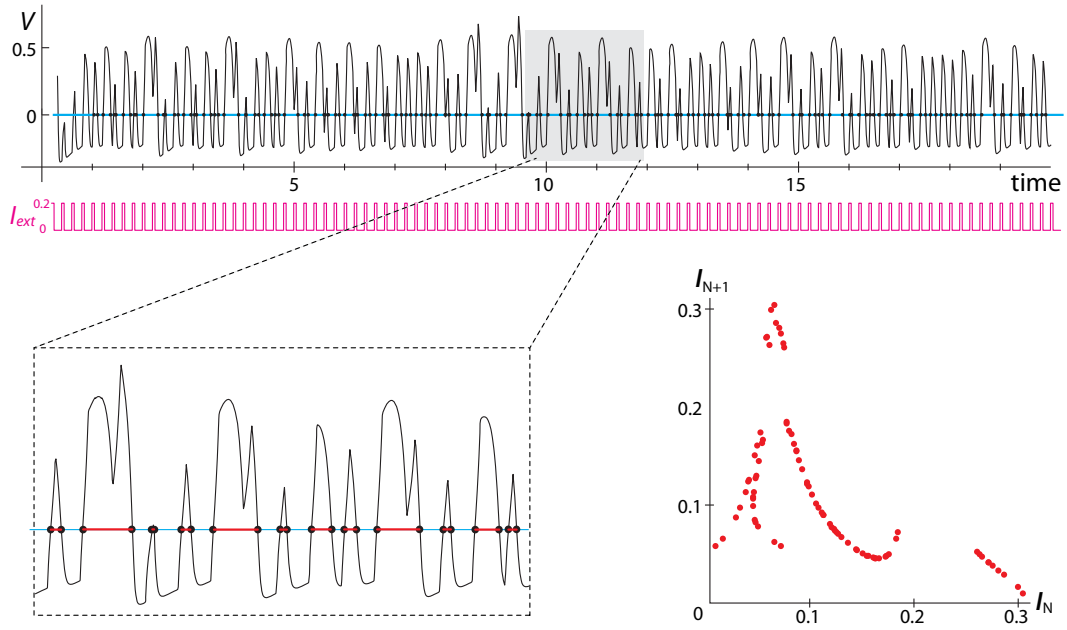


Figure 5.50: Alternative method for Poincaré Plot. Here, the time series is turned into a discrete time series by drawing a line at $V = 0$, and recording the time intervals (red segments) spent above that line. These peak durations are recorded as I_1, I_2, I_3, ... Lower right: Plotting I_{N+1} against I_N gives another version of the Poincaré plot for the same time series.

It is particularly interesting to compare this to Poincaré plots from our experiments with the FHN model (Figure 5.49 and Figure 5.50). The similarity in the Poincaré plots suggests that there is a common mechanism underlying neural chaos in these kinds of preparations.

We have now seen three different ways to diagnose chaos in a continuous-time series, by extracting a discrete-time series $X_1, X_2, X_3, \ldots$ from the continuous data and then plotting X_{N+1} against X_N in a Poincaré plot.

They are as follows:

(1) plotting the duration of the $(N+1)$st active phase against the duration of the Nth active phase (Figure 5.50).

(2) plotting the maximum amplitude of the $(N+1)$st phase against the amplitude of the Nth phase (Figure 5.49).

(3) stroboscopic plot in which we take the value of the variable at times $t = 1, 2, 3, \ldots$ and then plot the value at time $t+1$ against the value at time t (Figure 5.48).

The Beer Game: Chaos in a Supply Chain

The role of steep slopes and time delays in destabilizing systems is beautifully illustrated by a supply chain model developed at MIT's Sloan School of Management. It's called the beer game, and it is a model of a beer distribution chain, including consumers, retailers, wholesalers, distributors, and a brewery (Laugesen and Mosekilde 2006; Mosekilde and Laugesen 2007; Sterman 1989).

The basic idea is that orders for beer go one level up the supply line from the consumer to the retailer, from the retailer to the wholesaler, from the wholesaler to the distributor, and then to the brewery. Then cases of beer come one level down the supply line, from the brewery to the distributor, from the distributor to the wholesaler, from the wholesaler to the retailer, and finally to the consumer. Naturally, there are time delays associated with each of these steps (Figure 5.51).

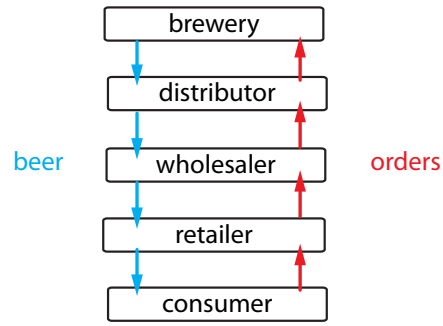


Figure 5.51: Structure of the beer game.

At each level of the game, there is a model of the manager at that level (Figure 5.52). Managers keep track of their inventory, ship beer according to demand from the level below, and generate orders for beer that ultimately result in beer being shipped to them from the level above. The manager makes choices: how much inventory to keep on hand, how far ahead to plan, and especially, how sensitive his or her order placement policy will be to changes in incoming demand. At one extreme, a manager can say "When my demand increases by X, I will increase

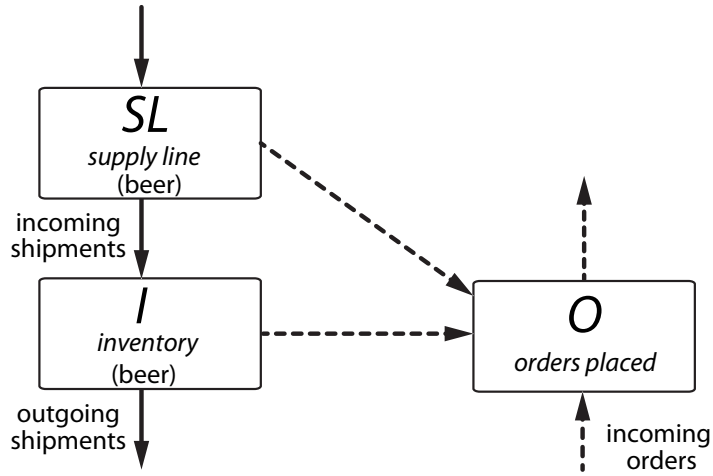


Figure 5.52: Schematic illustrating the managerial decision-making at each level. Solid lines denote the flow of beer, and dashed lines denote the flow of information.

my orders by, say, $2X$." This is a highly sensitive reaction. At the other extreme, the manager can have zero flexibility, and say, "No matter how the demand on me changes, I will not change my order placement policy."

At each level, the managers make their decisions based on the data available to them. The overall problem faced by the manager is modeled by two kinds of variables, representing beer and orders for beer.

The basic form of the equations is

$$\textbf{beer} \qquad I_{N+1} = I_N + \begin{matrix}\text{incoming}\\ \text{beer}\end{matrix} - \begin{matrix}\text{outgoing}\\ \text{beer}\end{matrix}$$

and

$$\textbf{orders} \qquad O_{N+1} = O_N + f\left(\begin{matrix}\text{expected}\\ \text{demand}\end{matrix}, \begin{matrix}\text{desired}\\ \text{inventory}\end{matrix}, \begin{matrix}\text{supply}\\ \text{line}\end{matrix} \right)$$

At each level, managers decide how to respond to demand from below and how to respond to potential shortages of inventory. In making their ordering decisions, managers need to estimate expected demand. This week's expected demand will be last week's expected demand updated with the new demand from below. This can be represented by a weighted sum of last week's expected demand and the new demand from below with a weighting factor θ (theta), which denotes the sensitivity of the update process to the new demand. When $\theta = 0$, demand never changes. When $\theta = 1$, this week's expected demand depends only on the new orders:

$$\textbf{expected demand} \qquad ED_{N+1} = \theta \cdot \begin{matrix}\text{demand}\\ \text{from}\\ \text{below}\end{matrix} + (1-\theta) \cdot ED_N$$

Note that θ is playing the role of a sensitivity parameter. It is really the slope of a function, namely,

$$\theta = \frac{\Delta(\text{outgoing orders})}{\Delta(\text{incoming orders})}$$

The other managerial decision is how much to care about inventory shortages. The manager has a desired inventory Q, which here we assume to be constant. The manager looks at the

quantity $Q-I-\beta SL$, where I is the current inventory and SL is the quantity of beer in the supply line;[3] $Q-I$ is the discrepancy between desired inventory and current inventory, a discrepancy that is lessened by the incoming supply line SL. So the quantity $Q-I-\beta SL$ is the total estimated inventory shortage. The parameter β measures how much weight the manager wants to give to the supply line.

The key equation for the manager is the equation for outgoing orders, or orders placed (OP):

$$\textbf{orders placed} \qquad OP_{N+1} = max\{0, ED_{N+1} + \alpha(Q - I_{N+1} - \beta SL_{N+1})\}$$

Note the parameter α. It represents the decision regarding how much to care about inventory shortages. If α is large, then the estimated inventory shortage plays a big role in the decision of how many orders to place.

In the full beer game model, there are four sectors in the supply chain. We keep track of the following seven state variables for each sector.

I	inventory
B	backlog orders of beer
IS	incoming shipments
OS	outgoing shipments
IO	incoming orders
ED	expected demand
OP	orders placed

In general, I is the current inventory of beer on hand, B is the backlog of orders from the level below that you have not yet filled, IS is the amount of beer that is incoming to you from the level above, and OS is the amount of beer you are shipping to the level below. IO is the amount of orders that have come in from the level below, and ED is expected demand. Managers then combine these into an equation to determine their key output, orders placed (OP) (Figure 5.53).

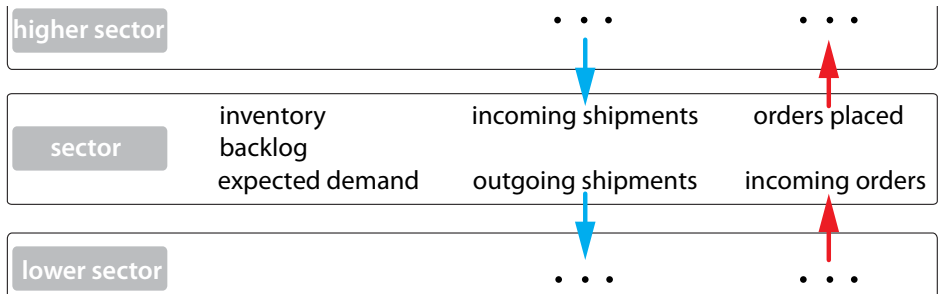


Figure 5.53: Schematic of the decision process of the manager at each level in the beer game. The manager of each sector must control ordering based on various factors.

[3] In describing this model, we will adopt a practice more common in programming than in math and use multiletter variable names; SL is a single variable, not a product.

A One-Sector Beer Game Model

First we will study a simple one-sector model, consisting of just the consumer and a factory. The consumer demand, called the consumer order rate (COR), is assumed to be constant. The factory manager's state variables are

I	inventory	FI	factory's inventory
B	backlog orders of beer	FB	factory's backlog orders
IS	incoming shipments	$FPD2$	factory's production delay
		$FPD1$	factory's production delay
		FPR	factory's production request
OS	outgoing shipments	N/A	N/A
IO	incoming orders	$FIO(=COR)$	factory's incoming orders
ED	expected demand	FED	factory's expected demand
OP	orders placed	N/A	N/A

The factory's inventory FI is straightforward. It is the amount of beer on hand. The quantity FB is the amount of beer that has been ordered by the consumer but not yet shipped. While IS would ordinarily be the incoming shipment from the level above, the factory has no level above: it fills its own orders by a production schedule. When the factory makes a production request (FPR), it is delayed by one week ($FPD1$) and then there is again a one-week delay to produce $FPD2$, which is the amount of beer actually produced.

We do not keep track of outgoing shipments (FOS), because in this model, consumer demand does not change, so the outgoing shipments do not affect anything. The factory's incoming orders (FIO) is just the consumer order rate (COR).

From these quantities, the manager calculates the factory's expected demand (FED). The variable "orders placed" (FOP) does not apply in this one-sector model.

The overall equations for the factory's manager are

$$\begin{aligned}
FI_{N+1} &= max\{0, FI_N + FPD2_N - FB_N - COR\} \\
FPD2_{N+1} &= FPD1_N \\
FPD1_{N+1} &= FPR_N \\
FPR_{N+1} &= max\{0, FED_{N+1} + \alpha(Q - FI_{N+1} + FB_{N+1} - \beta \cdot FSL_{N+1})\} \\
FB_{N+1} &= max\{0, FB_N + COR - FI_N - FPD2_N\} \\
FED_{N+1} &= \theta \cdot COR + (1-\theta) \cdot FED_N
\end{aligned}$$

where the factory's supply line at time $N+1$ is given by

$$FSL_{N+1} = FPD1_{N+1} + FPD2_{N+1}$$

Researchers at MIT ran many sessions in which managers actually tried playing the game by hand. The researchers were therefore able to see where real-life managers set their parameters (Sterman 1989).

The most interesting parameters are

θ = sensitivity of the manager to changing demand

α = sensitivity of the manager to inventory maintenance

β = degree of manager's awareness of his or her future production

Q = desired inventory

In the real-life games, these parameters are chosen by the players, who are trying to maximize revenue, not stability. For example, it would be possible to achieve total stability by maintaining a large desired inventory Q, but that would involve high storage costs. It turns out that the real-life choices that the managers make are often in the realm of instability (Laugesen and Mosekilde 2006; Mosekilde and Laugesen 2007; Sterman 1989).

For example, the parameter β plays an important role in the stability of the system. The values chosen here are all fairly low, indicating a fairly dim awareness by the manager of his or her outstanding production requests. But Sterman reports that real-life players often choose low values of β.

Laugesen and Mosekilde provide an excellent analysis of this one-sector model. They show that the model undergoes a Hopf bifurcation when α is sufficiently high and β is sufficiently low. We can illustrate this by choosing appropriate parameter values. In each case, we will simulate the system's response to a step change in consumer demand (COR). For the first four weeks, $COR = 4$, and then starting at week five, it changes to $COR = 8$ and maintains that level from then on.

When we choose a fairly low $\alpha = 0.7$ and a β value of 0.2, the system goes to a stable equilibrium point (Figure 5.54). Even after the step change (red arrow), the system is able to return to equilibrium, although only after many weeks. Note, however, that the stable equilibrium that was achieved was not the desired inventory $Q = 17$. In this case, stability has been achieved only at the cost of suboptimal performance.

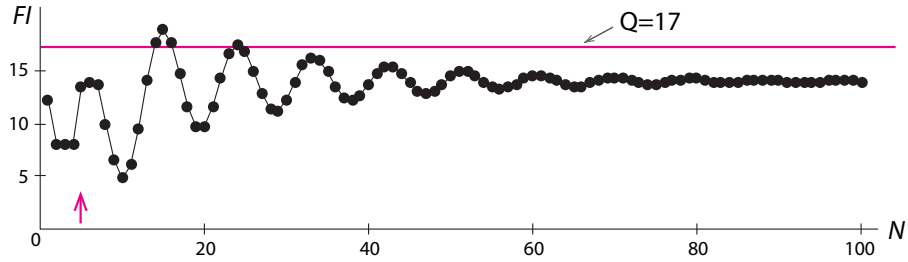


Figure 5.54: A stable equilibrium point in the one-sector beer game model. The factory inventory achieves a stable equilibrium even after a change in COR (red arrow). The value of N is in weeks, $\alpha = 0.7$, and $\beta = 0.2$.

Now if we increase the manager's sensitivity to inventory shortages to $\alpha = 0.9$ and decrease the manager's awareness of the supply line to $\beta = 0.05$, then the system goes to sustained oscillation (Figure 5.55). The presence of oscillation does not depend on the step change in the consumer demand COR, and it is seen even when consumer demand is constant throughout the simulation.

The one-sector beer game model illustrates the fundamental lesson of Chapter 4: in a system with built-in time delays, an increase in the sensitivity of negative feedbacks causes a Hopf bifurcation and a consequent change from stable equilibrium to oscillation.

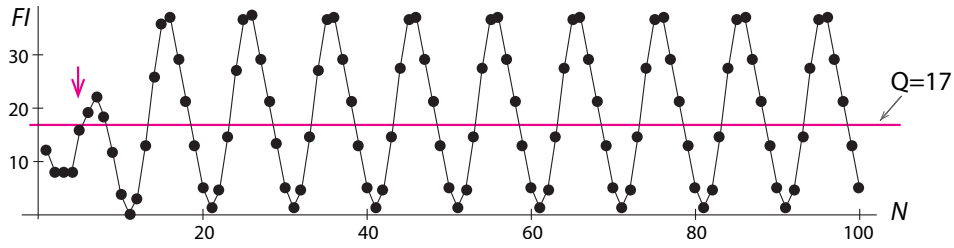


Figure 5.55: Sustained oscillations in the one-sector beer game model for $\alpha = 0.9$ and $\beta = 0.05$.

Chaos in the Two-Sector Beer Game Model

Many of the classic papers on the subject treat the full four-sector model described above. They report a variety of chaotic phenomena. Here we will study a simpler two-sector model that is capable of displaying chaotic dynamics (Laugesen and Mosekilde 2006; Mosekilde and Laugesen 2007).

In this model, the consumer orders beer from a retailer, who then orders from the factory.

The state variables for the factory are

I	inventory	FI	factory's inventory
B	backlog orders of beer	FB	factory's backlog orders
		$FPD2$	factory's production delay
IS	incoming shipments	$FPD1$	factory's production delay
		FPR	factory's production request
OS	outgoing shipments	FOS	factory's outgoing shipping
IO	incoming orders	FIO	factory's incoming orders
ED	expected demand	FED	factory's expected demand
OP	order placed	N/A	N/A

The major changes from the one-sector model are that now the factory has outgoing shipments that go to the retailer, and the factory's incoming orders now come from the retailer, not the consumer.

The state variables for the retailer are

I	inventory	RI	retailer's inventory
B	backlog orders of beer	RB	retailer's backlog orders
IS	incoming shipments	RIS	retailer's incoming shipments from the factory
OS	outgoing shipments	N/A	N/A
IO	incoming order	$RIO(=COR)$	retailer's incoming order from consumer
ED	expected demand	RED	retailer's expected demand
OP	orders placed	ROP	orders placed by retailer's to the factory

The retailer has its own inventory (RI) and its backlog of consumer orders (RB); the quantity of the retailer's incoming shipments from the factory is RIS. The retailer's incoming orders come from the consumer ($RIO = COR$), and the retailer must calculate an expected demand (RED) and make a decision to arrive at an outgoing order (ROP).

The factory's equations are

$$\begin{aligned}
FI_{N+1} &= max\{0, FI_N + FPD2_N - FB_N - FIO_N\} \\
FB_{N+1} &= max\{0, FB_N + FIO_N - FI_N - FPD2_N\} \\
FPD2_{N+1} &= FPD1_N \\
FPD1_{N+1} &= FPR_N \\
FPR_{N+1} &= max\{0, FED_{N+1} + \alpha(Q - FI_{N+1} + FB_{N+1} - \beta \cdot FSL_{N+1})\} \\
FOS_{N+1} &= min\{FI_N + FPD2_N, FB_N + FIO_N\} \\
FIO_{N+1} &= ROP_N \\
FED_{N+1} &= \theta \cdot FIO_N + (1-\theta) \cdot FED_N
\end{aligned}$$

where the factory's supply line is

$$FSL_{N+1} = FPD1_{N+1} + FPD2_{N+1}$$

The retailer's equations are

$$\begin{aligned}
RI_{N+1} &= max\{0, RI_N + RIS_N - RB_N - COR\} \\
RB_{N+1} &= max\{0, RB_N + COR - RI_N - RIS_N\} \\
RIS_{N+1} &= FOS_N \\
RED_{N+1} &= \theta \cdot COR_N + (1-\theta) \cdot RED_N \\
ROP_{N+1} &= max\{0, RED_{N+1} + \alpha(Q - RI_{N+1} + RB_{N+1} - \beta(RSL_{N+1}))\}
\end{aligned}$$

where the retailer's supply line is

$$RSL_{N+1} = RIS_{N+1} + FIO_{N+1} + FB_{N+1} + FOS_{N+1}$$

In this model, choosing a high α value and a low β value leads to chaos in the supply chain, for example, if we choose $\alpha = 0.9$ and $\beta = 0.25$ (Figure 5.56).

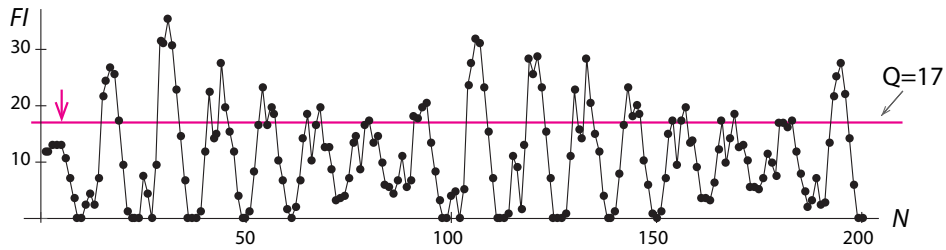


Figure 5.56: Chaotic behavior in the two-sector beer game model for $\alpha = 0.9$ and $\beta = 0.25$.

Is Chaos Necessarily Bad?

The term "chaos" certainly suggests something that is undesirable, something to avoid or prevent. But this may not be necessarily true. We already saw a functional role for chaos: early Japanese swordsmiths used the stretching and folding process to mix two metals together effectively.

It may well be that chaos has other virtues. The electroencephalogram (EEG) is a record of the electrical activity of the brain, as recorded by electrodes on the scalp. Consider the following two human EEGs: the first is regular and periodic (Figure 5.57, top), while the second is random-looking and irregular (Figure 5.57, bottom). Which one would you rather have?

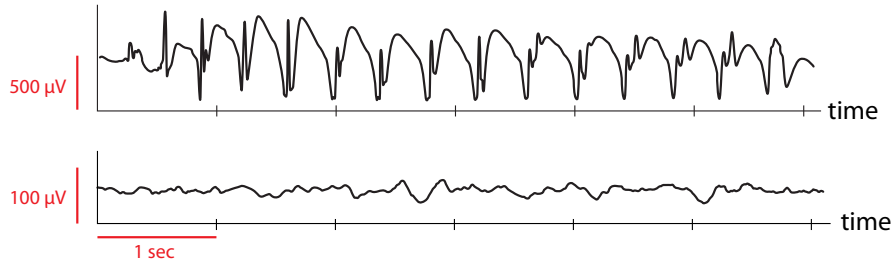


Figure 5.57: Top: EEG during during a seizure in childhood absence epilepsy. Redrawn from F. Marten, S. Rodrigues, O. Benjamin, M.P. Richardson, and J.R. Terry, 2009, "Onset of polyspike complexes in a mean-field model of human electroencephalography and its application to absence epilepsy," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 367(1891):1145–1161, by permission of the Royal Society. Bottom: Normal, eyes-open human EEG.

Be careful, because the irregular and ragged-looking one is a normal human, eyes-open EEG, and the beautiful periodic one is an epileptic seizure!

Although it is controversial whether the irregularity of normal brain waves is an instance of true chaos, it is certainly true that order, or periodicity, is pathological in the brain.

This is especially clear if we view the onset of a seizure out of normal background EEG activity (Figure 5.58). It is very tempting to speculate that a bifurcation has occurred in the sharp onset of the seizure activity, which is the periodic signal in the middle of the record.

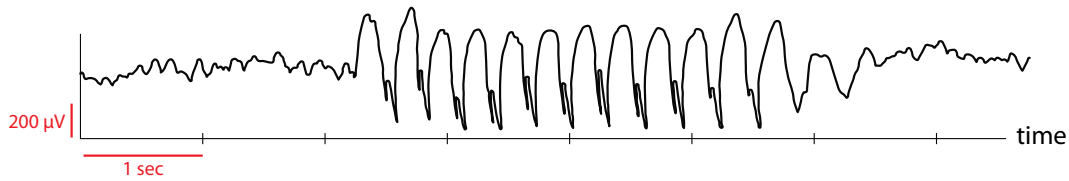


Figure 5.58: Onset of a seizure. Two seconds of normal EEG are followed by the abrupt onset of a spike-wave complex seizure.

Further Exercise 5.5

1. a) Simulate the discrete logistic equation for $r = 4$ for 20 time units and make a Poincaré plot of the results.

   b) Run the simulation with a slightly different initial value and make a Poincaré plot of the results. Overlay the two plots. In what ways are they similar and different?