

last revised: 8 March 2011

## 11 One-Dimensional Nonlinear Models

### 11.1 From linear to nonlinear models

The first half of this book focused on linear models, which can be written as

$$\mathbf{x}_{t+1} = A\mathbf{x}_t$$

where  $\mathbf{x}_t$  is a  $(n \times 1)$  vector and  $A$  is an  $(n \times n)$  matrix of (fixed) parameters. Such systems are “linear” in the sense that each element  $i$  of the  $\mathbf{x}_{t+1}$  vector is a linear combination of the elements of the  $\mathbf{x}_t$  vector. More precisely, unpacking the preceding matrix equation, we obtain the  $n$ -equation system

$$\begin{aligned}\mathbf{x}_{t+1}(1) &= A(1,1) \mathbf{x}_t(1) + A(1,2) \mathbf{x}_t(2) + \dots + A(1,n) \mathbf{x}_t(n) \\ \mathbf{x}_{t+1}(2) &= A(2,1) \mathbf{x}_t(1) + A(2,2) \mathbf{x}_t(2) + \dots + A(2,n) \mathbf{x}_t(n) \\ &\vdots \\ \mathbf{x}_{t+1}(n) &= A(n,1) \mathbf{x}_t(1) + A(n,2) \mathbf{x}_t(2) + \dots + A(n,n) \mathbf{x}_t(n)\end{aligned}$$

By this point, you’ve hopefully been impressed by the wide range of social science applications of linear models. However, many other topics require the use of nonlinear models. The remainder of the book is thus devoted to this class of models.

Generically, nonlinear models can be written as

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t)$$

where  $f$  is called a *generator function*. The simplicity of this equation may be deceiving because the generator function maps  $n$  inputs (the elements of the  $\mathbf{x}_t$  vector) into  $n$  outputs (the elements of the  $\mathbf{x}_{t+1}$  vector). An alternative specification given by the system of equations

$$\begin{aligned}\mathbf{x}_{t+1}(1) &= f_1(\mathbf{x}_t(1), \mathbf{x}_t(2), \dots, \mathbf{x}_t(n)) \\ \mathbf{x}_{t+1}(2) &= f_2(\mathbf{x}_t(1), \mathbf{x}_t(2), \dots, \mathbf{x}_t(n)) \\ &\vdots \\ \mathbf{x}_{t+1}(n) &= f_n(\mathbf{x}_t(1), \mathbf{x}_t(2), \dots, \mathbf{x}_t(n))\end{aligned}$$

reveals that the generator function  $f$  is actually a set of functions  $\{f_1, \dots, f_n\}$ , with each function  $f_i$  mapping  $n$  inputs into the output  $\mathbf{x}_{t+1}(i)$ . Comparing the systems of equations for linear and nonlinear systems, we see that linear models are a special case of nonlinear models; the latter permit much more freedom in specifying functional forms within the generator system. For example, the  $f_i$ ’s might include “interaction

terms” (say the product  $\mathbf{x}_t(1)\mathbf{x}_t(2)$ ) and nonlinear transformations (e.g., we could assume  $f_1 = (\alpha\mathbf{x}_t(1)\mathbf{x}_t(2) + \beta\mathbf{x}_t(3))^2$  where  $\alpha$  and  $\beta$  are parameters).

While linear models have many useful applications, nonlinear models clearly permit much greater flexibility in modeling social processes. However, this flexibility comes at a price. Analysis of linear models is trivial when  $n = 1$ . For instance, letting the scalar  $x_t$  denote a population level (or bank balance) at time  $t$ , and letting  $r$  denote the population growth rate (or interest rate), we could simply note that

$$x_{t+1} = (1 + r) x_t$$

and obtain

$$x_t = (1 + r)^t x_0$$

by induction. In order to make things more interesting, we thus assumed  $n \geq 2$  throughout the examples in the first half of the book. In contrast, “one-dimensional” nonlinear models (with  $n = 1$ ) can be more interesting and consequently harder to analyze. Indeed, the next several chapters are devoted entirely to these one-dimensional models and some social science applications. Once we have developed some appreciation for the one-dimensional case, subsequent chapters will address higher dimensional nonlinear models (with  $n \geq 2$ ) and their applications.

## 11.2 The discrete logistic model

Our first example of a one-dimensional nonlinear system is the (discrete-time) logistic model – a model of population growth that is simple yet mathematically interesting. Letting the scalar  $p_t$  denote population size in period  $t$ , this model assumes that population dynamics are given by the equation

$$p_{t+1} = \left[1 + r \left(1 - \frac{p_t}{k}\right)\right] p_t$$

where  $r$  and  $k$  are parameters. To help motivate this equation, we can rewrite it as

$$p_{t+1} - p_t = r \left(1 - \frac{p_t}{k}\right) p_t$$

and hence

$$\Delta p = r \left(1 - \frac{p}{k}\right) p$$

where  $\Delta p$  denotes the change in  $p$  over one period, and the time subscript on  $p$  is now implicit.<sup>1</sup> Dividing both sides of this equation by  $p$ , we obtain

$$\frac{\Delta p}{p} = r \left(1 - \frac{p}{k}\right)$$

---

<sup>1</sup>You should become comfortable with this “delta” notation, which we’ll use frequently to simply the specification of nonlinear models.

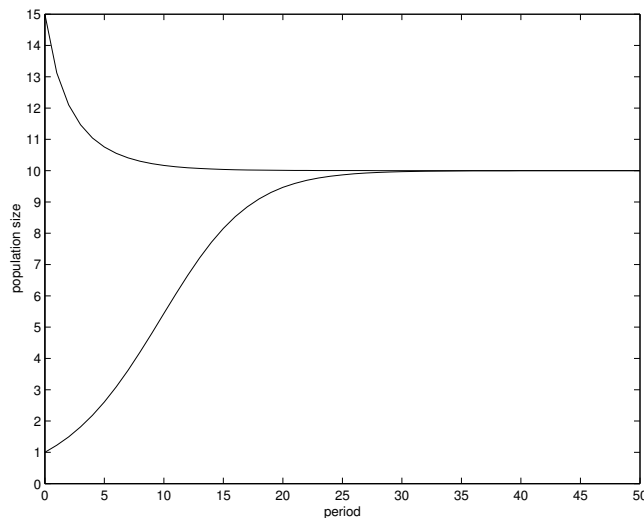
and the left-hand side can be interpreted as the population growth rate. Thus, when the population is small relative to the “carrying capacity” of the environment (given by the parameter  $k$ ), the population grows at approximately rate  $r$ . However, as the population becomes large relative to the carrying capacity (so that  $p/k$  approaches 1), the growth rate slows to zero.

We can begin to understand the behavior of this system through some numerical examples. Setting  $r = 0.25$  and  $k = 10$ , we first use Matlab to compute the time path  $(p_0, p_1, \dots, p_{50})$  given the initial population sizes  $p_0 = 1$  and  $p_0 = 15$ .

```
>> r = .25; k = 10; % set parameter values
>> p = 1; % initial condition p = 1
>> y1 = p;
for t = 1:50;
    p = (1+r*(1-p/k))*p;
    y1 = [y1; p];
end
>> p = 15; % initial condition p = 15
>> y2 = p;
for t = 1:50;
    p = (1+r*(1-p/k))*p;
    y2 = [y2; p];
end
```

Note our use of the `for` loop. For each time period  $t$  (from 1 to 50), we use the old value  $p_{t-1}$  to obtain the new value  $p_t$ ; these values are then collected in the time-path vector `y1` (for our first example where  $p_0 = 1$ ) and `y2` (for our second example where  $p_0 = 15$ ). To see the results, we plot both time paths together.

```
>> plot(0:50,y1,0:50,y2)
>> % time paths in logistic model with r = .25 and k = 10
```



As already discussed above, the population begins to grow geometrically (at rate  $r = 0.25$ ) when it is initially small relative to its carrying capacity ( $k = 10$ ). But as  $p$  rises toward  $k$ , the growth rate decreases. Graphically, the time-path curve is initially convex but becomes concave as  $t$  rises. In contrast, if the population initially exceeds its carrying capacity, the population size begins to fall rapidly. In both cases, the population size converges to the carrying capacity in the long run.

### 11.3 Equilibrium

Based on these numerical examples, we might suspect – but have not demonstrated – that population size  $p$  will converge to the carrying capacity  $k$  for any initial condition  $p_0$ . Generically, the dynamics for a one-dimensional nonlinear system is given by the equation

$$x_{t+1} = f(x_t)$$

where  $f$  is a generator function mapping the (scalar) input  $x_t$  into the (scalar) output  $x_{t+1}$ . The system is in equilibrium when  $x_{t+1} = x_t = x^*$  where

$$x^* = f(x^*)$$

Equivalently, the equilibrium condition may be written as

$$f(x^*) - x^* = \Delta x^* = 0$$

Researchers describe this outcome using a variety of terms (which, for our purposes, may be used interchangeably). We may say that  $x^*$  is an *equilibrium* or *steady state* of the system. Alternatively,  $x^*$  is a *fixed point* of the generator function.

For the discrete logistic model, the equilibria are determined by the condition

$$\Delta p^* = r \left( 1 - \frac{p^*}{k} \right) p^* = 0$$

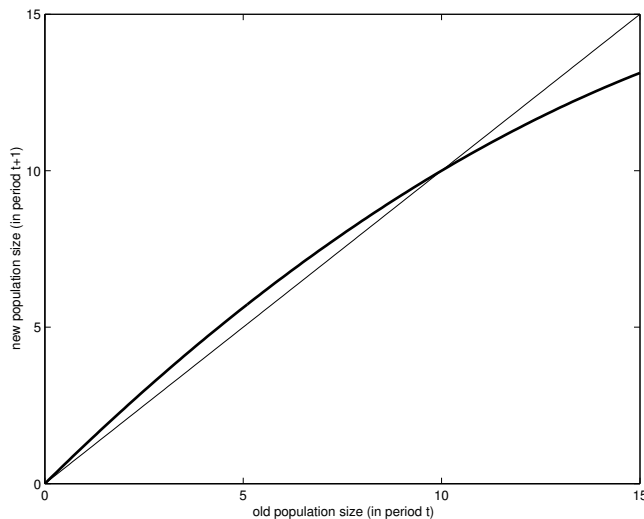
which implies  $p^* = k$  or  $p^* = 0$ . Thus, while our population projections suggested a unique equilibrium, we find there are actually two. Intuitively, if population size is initially zero, the population size will remain zero forever.

### 11.4 The cobweb diagram

To better understand the dynamics of one-dimensional models, it is helpful analyze them graphically using a *cobweb diagram*. On this diagram, we plot both the generator function  $f(x)$  and  $x$  itself. Of course, plotting  $x$  as a function of itself, we simply obtain a 45-degree line. Thus, the cobweb diagram is essentially a graph of the generator function superimposed on a 45-degree line. Any intersection of these curves indicates an equilibrium, since the intersection corresponds to some  $x^*$  such that  $x^* = f(x^*)$ .

To illustrate, we first solve numerically for the generator function in our preceding examples (maintaining the parameter values  $r = 0.25$  and  $k = 10$ ).<sup>2</sup> We then plot this function (thicker curve) against the 45-degree line (thinner curve). Inspection of this diagram confirms the existence of two equilibria:  $p^* = 0$  and  $p^* = 10$ .

```
>> p = 0:.1:15; f = (1+r*(1-p/k)).*p;
>> plot(p,f,p,p) % cobweb diagram for logistic model with r = .25 and k = 10
```



Having shown how the cobweb diagram can be used to determine the existence of an equilibrium (or equilibria), we can also use this diagram to depict the time path from an initial condition to an equilibrium. Choosing a particular initial condition (say  $p_0 = 1$ ), the population size in period 1 is given by

$$p_1 = f(p_0) = f(1) = 1.225,$$

the population size in period 2 is given by

$$p_2 = f(p_1) = f(1.225) = 1.4937,$$

the population size in period 3 is given by

$$p_3 = f(p_2) = f(1.4937) = 1.8114,$$

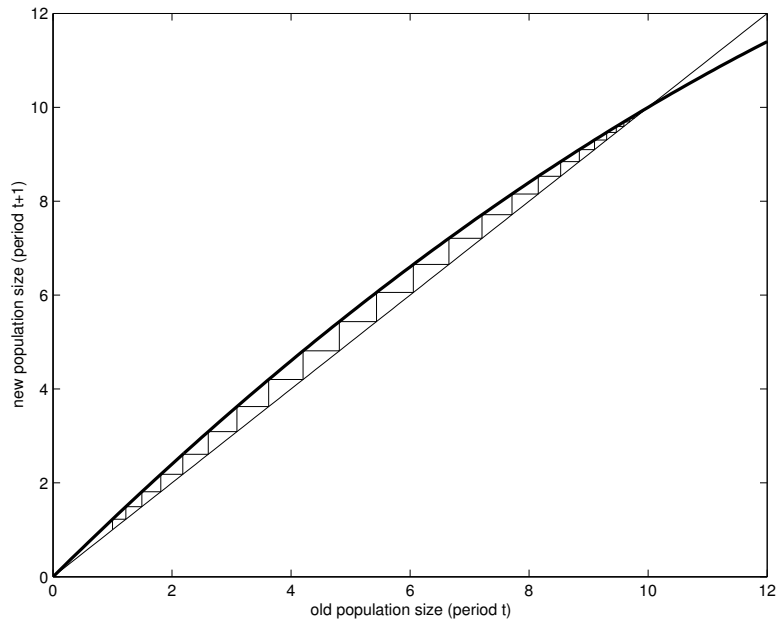
---

<sup>2</sup>Here, we make use of one of Matlab's strengths by "vectorizing" the computation. The step size (of 0.1) is arbitrary, but small enough to ensure that the generator function appears smooth when plotted. Note that the same result could be obtained in a less elegant fashion using a `for` loop (iterating  $p$  from 0 to 15 by steps of 0.1). Alternatively, the generator function could have been plotted directly using the `ezplot` command. Readers are encouraged to study Matlab documentation for more information.

and so on. To depict this sequence graphically, we might start by plotting a line from point  $(p_0, p_0)$  to point  $(p_0, f(p_0))$  to indicate the change in the population over the first period. To indicate that the old population size (for period 0) becomes the new population size (for period 1), we then plot a line from  $(p_0, f(p_0))$  to  $(f(p_0), f(p_0))$ . Of course, this latter point is simply  $(p_1, p_1)$ . Repeating this procedure, we then plot a line from  $(p_1, p_1)$  to  $(p_1, f(p_1))$  to indicate change in population size over the second period, and then another line from  $(p_1, f(p_1))$  to  $(f(p_1), f(p_1))$  to indicate that the old population size (for period 1) becomes the new population size (for period 2). Continuing in a similar manner for subsequent periods, we obtain a staircase curve that looks something like a “cobweb” (hence the name of the diagram).

In the code below, we first compute the cobweb (using the `y1` vector computed earlier) and then superimpose the cobweb on the other curves shown earlier.

```
>> cobweb = [ ];
for t = 1:50;
    cobweb = [cobweb; y1(t) y1(t); y1(t) y1(t+1)];
end
>> plot(p,f,p,p,cobweb(:,1),cobweb(:,2))
>> % cobweb diagram for initial condition p = 1
```



In a similar way, you could plot the cobweb for the second time path (from the initial condition  $p_0 = 15$ ) to show how this sequence converges to the equilibrium  $p^* = 10$  from above. (We’ll leave this task for the reader, since it merely involves substituting the `y2` vector for the `y1` vector in the computations above.)

## 11.5 Stability

Having identified the equilibria of the model (by finding fixed points of the generator function), we can now assess the stability of each equilibrium. Conceptually, an equilibrium is stable when the system returns to this outcome after a small shock (often referred to as a “perturbation”). For our present example, it is easy to see from the cobweb diagram that the upper equilibrium  $p^* = 10$  is stable while the lower equilibrium  $p^* = 0$  is unstable. Starting from the upper equilibrium, a small drop in the population level (below 10) would cause the population to grow, while a small increase in the population level (above 10) would cause the population to shrink. In contrast, starting from the lower equilibrium, a small increase in the population level (above 0) would cause the population to grow. Thus, following a small shock, the system would return to the upper equilibrium but diverge from the lower equilibrium.

More formally, stability depends on the slope of the generator function. Consider a generic one-dimensional nonlinear system where

$$x_{t+1} = f(x_t)$$

Borrowing notation from calculus, let  $f'(x)$  denote the slope of the generator function  $f$  evaluated at  $x$ . An equilibrium  $x^*$  is *stable* if

$$|f'(x^*)| < 1$$

and is *unstable* if

$$|f'(x^*)| > 1$$

where the vertical bars denote absolute value.<sup>3</sup> While the need for the absolute value signs may not be immediately apparent, examples in section xx below will help demonstrate their relevance.

Returning to our present example, we can easily assess stability from the cobweb diagram even without computing the precise slopes  $f'(10)$  and  $f'(0)$ . Recognizing that 45-degree line has slope equal to 1, the generator function must have a slope less than 1 when it is rising and crosses the 45-degree line from above. Hence  $f'(10) < 1$ . Conversely, the generator function must have a slope greater than 1 when it is rising and crosses the 45-degree line from below. Hence  $f'(0) > 1$ . Thus, consistent with our informal argument above, we find that the upper equilibrium is stable while the lower equilibrium is unstable.

## 11.6 Further analysis of stability

For some applications, qualitative analysis of stability may be sufficient – we can learn everything we wish to know from the cobweb diagram. But for other applications,

---

<sup>3</sup>In the borderline case where  $|f'(x^*)| = 1$ , assessment of stability requires additional information about the curvature (i.e., higher order derivatives) of the generator function.

we may wish to assess stability numerically. For readers who know calculus, the task is straightforward: differentiate the generator function and evaluate this derivative at each equilibrium. For our present example, the generator function is

$$f(p) = \left[1 + r \left(1 - \frac{p}{k}\right)\right] p = (1 + r) p - \left(\frac{r}{k}\right) p^2$$

and thus differentiation yields

$$f'(p) = 1 + r - 2 \left(\frac{r}{k}\right) p$$

Evaluating this derivative at each equilibrium, we obtain

$$|f'(k)| = \left|1 + r - 2 \left(\frac{r}{k}\right) k\right| = |1 - r|$$

$$|f'(0)| = \left|1 + r - 2 \left(\frac{r}{k}\right) 0\right| = |1 + r|$$

Thus, given the parameter value  $r = 0.25$ , we verify that the equilibrium at  $k$  is stable while the equilibrium at 0 is unstable.

For readers who do not know calculus, we can tackle this problem in a different way. (Even readers who know calculus might benefit from working through this approach, which will be generalized for two-dimensional models in chapter xx.) To begin, we suppose that the system was in equilibrium at  $p^*$  but becomes disturbed by a small shock in period  $t$ . We may thus write

$$p_t = p^* + \epsilon_t$$

where  $\epsilon_t$  reflects the (small) perturbation. Even with no further shocks in subsequent periods, the system will not return immediately to equilibrium.<sup>4</sup> For period  $t + 1$ , we may thus write

$$p_{t+1} = p^* + \epsilon_{t+1}$$

where  $\epsilon_{t+1}$  reflects the deviation from the initial equilibrium. Conceptually, the equilibrium  $p^*$  is stable if the deviation from equilibrium falls over time, so that

$$|\epsilon_{t+1}| < |\epsilon_t|$$

To determine whether this condition holds, note first that

$$\begin{aligned} p_{t+1} &= f(p_t) \\ p^* + \epsilon_{t+1} &= f(p^* + \epsilon_t) \\ &= (1 + r)(p^* + \epsilon_t) - \left(\frac{r}{k}\right) (p^* + \epsilon_t)^2 \\ &= (1 + r)(p^* + \epsilon_t) - \left(\frac{r}{k}\right) (p^{*2} + 2p^*\epsilon_t + \epsilon_t^2) \end{aligned}$$

---

<sup>4</sup>In contrast to Young's evolutionary framework (which we studied in Chapter 5), we assume a one-time disturbance to the system, not a new shock in every period.



Recognizing that  $p^*$  is a fixed point of  $f(p)$ , we know that

$$p^* = (1+r)p^* - \left(\frac{r}{k}\right)p^{*2}$$

and thus the preceding equation simplifies to

$$\epsilon_{t+1} = (1+r)\epsilon_t - \left(\frac{r}{k}\right)(2p^*\epsilon_t + \epsilon_t^2)$$

Further, because we have assumed that the shock  $\epsilon_t$  is small, the term  $\epsilon_t^2$  is very small and can be ignored. Thus, the preceding equation simplifies further to

$$\epsilon_{t+1} \approx \left[ (1+r) - \left(\frac{r}{k}\right)2p^* \right] \epsilon_t$$

From this equation, it is apparent that the equilibrium  $p^*$  is stable if

$$\left| (1+r) - \left(\frac{r}{k}\right)2p^* \right| < 1$$

In particular, the equilibrium  $p^* = k$  is stable if

$$|1-r| < 1$$

and the equilibrium  $p^* = 0$  is stable if

$$|1+r| < 1$$

Thus, given the parameter value  $r = 0.25$ , we verify (yet again) that the equilibrium at  $k$  is stable while the equilibrium at 0 is unstable.

Of course, there is a close connection between these two approaches to assessing stability. Thinking graphically about the second approach, the slope of the generator function at  $p^*$  is well approximated by the slope of a chord from point  $(p^*, p^*)$  to point  $(p^* + \epsilon_t, p^* + \epsilon_{t+1})$ . Because the slope of the chord is determined by its rise ( $\epsilon_{t+1}$ ) over its run ( $\epsilon_t$ ), we obtain

$$f'(p^*) \approx \frac{\epsilon_{t+1}}{\epsilon_t}$$

Alternatively, using the formal definition of a derivative from calculus, we may write

$$f'(p^*) = \lim_{\epsilon \rightarrow 0} \frac{f(p^* + \epsilon) - f(p^*)}{\epsilon}$$

and it becomes apparent that our “non-calculus” approach essentially uses this definition to compute the slope of the generator function. While we will attempt to minimize the use of calculus throughout the remainder of the book, utilizing graphical methods whenever possible, this example shows how (even a little) knowledge of calculus can greatly simplify the analysis of nonlinear systems. For more complicated problems (in which the generator function incorporates less “convenient” functional forms), calculus becomes an indispensable tool.

## 11.7 Oscillations, cycles, and chaos in the logistic model

Throughout the analysis of our present example, we have fixed the parameters  $r = 0.25$  and  $k = 10$ . Perhaps you've wondered what happens for other parameter values. It turns out that a change in the capacity constraint  $k$  is not very interesting, as this change merely rescales the size of the population. In contrast, a change in the parameter  $r$  may result in a qualitative change in the dynamics of the model.

To illustrate, we first consider the case where  $r = 1.5$ . From the stability conditions derived in the preceding section, we already know that the upper equilibrium ( $p^* = k$ ) will be stable (because  $|1 - r| < 1$ ), while the lower equilibrium ( $p^* = 0$ ) will be unstable (because  $|1 + r| > 1$ ). However, as we can see from the diagrams in Figure 1, the dynamics now differ qualitatively from those shown in Section 11.3. In particular, the population “overshoots” the long-run population size, oscillating as it converges to this equilibrium. This example helps explain the need for the absolute value signs in the stability definition. Given an initial equilibrium at  $p^* = k$ , a small positive shock to population size in period  $t$  would actually cause the population level to fall *below*  $p^*$  in period  $t + 1$ . That is, a positive deviation  $\epsilon_t$  induces a negative deviation  $\epsilon_{t+1}$ . However, as long as the absolute value of these deviations are decreasing, the oscillations dampen over time and the system returns to  $p^*$ .

Now consider the case where  $r = 2.2$ . From the stability conditions, we know that both equilibria are unstable (because both  $|1 - r|$  and  $|1 + r|$  exceed 1). The dynamics are depicted in the diagrams in Figure 2. Starting from an initial condition close to the (unstable) upper equilibrium, we see that system converges not to a steady state but rather to a *cycle* in which population size alternates every other period between 7.4625 and 11.6284. The reader may verify that this same cycle is reached from other initial conditions (at least for  $p_0 \in (0, 14.5454)$ ). Despite the obvious qualitative differences between this outcome and the (steady state) equilibria we have already encountered, it may also be called a (*cyclical*) *equilibrium*. To help motivate this terminology, note that history repeats itself every two periods in this equilibrium. Thus, the (2-cycle) equilibrium condition can be written as

$$p = f(f(p))$$

and the existence of a 2-cycle may be verified by noting that

$$f(f(7.4625)) = f(11.6284) = 7.4625$$

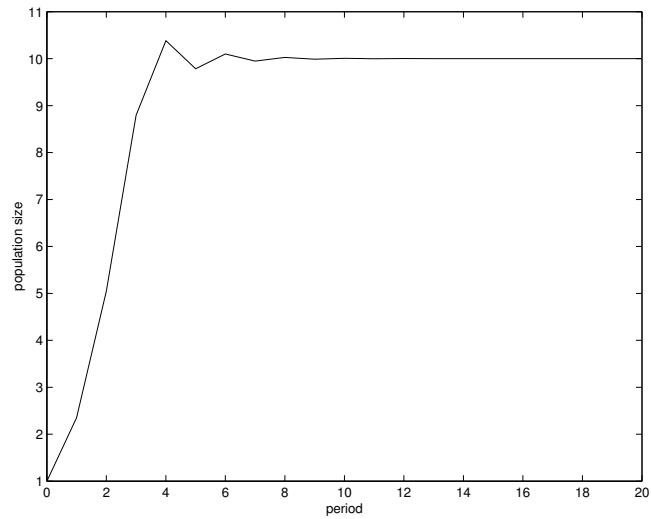
$$f(f(11.6284)) = f(7.4625) = 11.6284$$

Of course, either of the steady states ( $p = k$  or  $p = 0$ ) would also satisfy this 2-cycle equilibrium condition. But stability analysis (using the iterated generator function  $f(f(p))$  in place of the original function  $f(p)$ ) would demonstrate that the 2-cycle depicted in Figure 2 is the only stable equilibrium.

Finally, consider the case where  $r = 2.8$ . Both equilibria are unstable, and the dynamics are depicted in the diagrams in Figure 3. Unlike all of the previous cases,

**Figure 1: Oscillations in the logistic model ( $r = 1.5$ )**

```
>> r = 1.5; k = 10;  
>> p = 1; y = p; for t = 1:20; p = (1+r*(1-p/k))*p; y = [y p]; end  
>> plot(0:20,y) % time path for logistic model with r = 1.5
```



```
>> cobweb = [ ]; for t = 1:20; cobweb = [cobweb; y(t) y(t); y(t) y(t+1)]; end  
>> p = 0:.1:12; f = (1+r*(1-p/k)).*p;  
>> plot(p,p,p,f,cobweb(:,1),cobweb(:,2)) % cobweb diagram
```

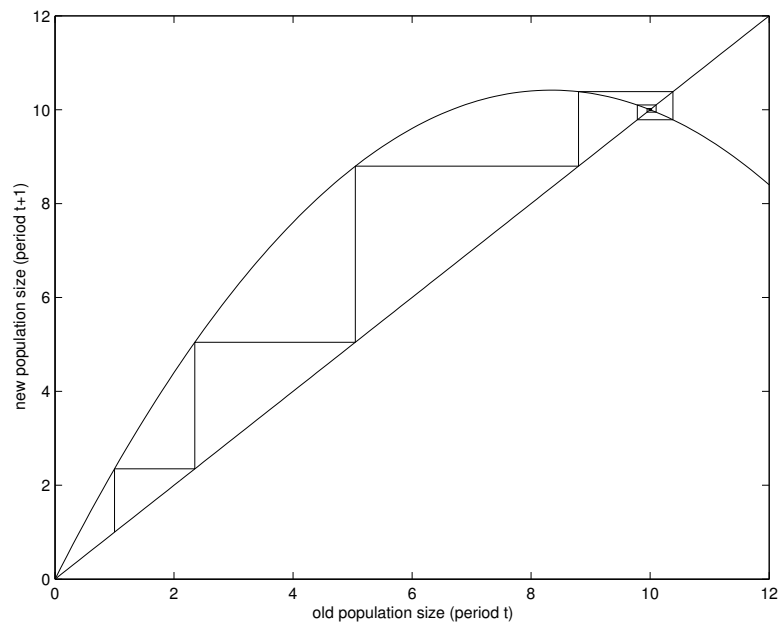
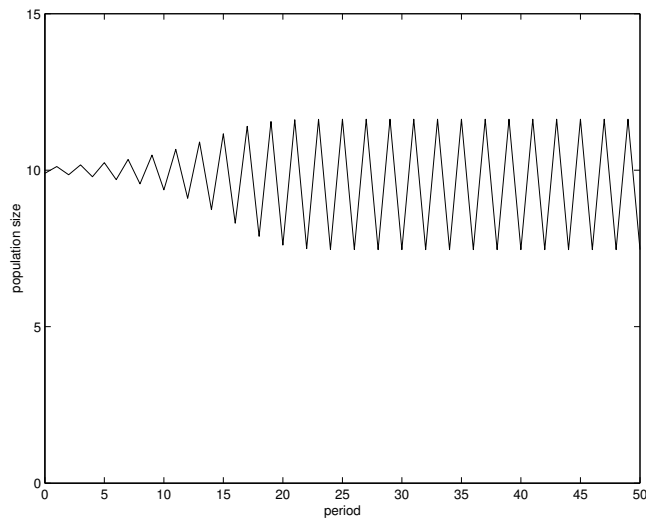
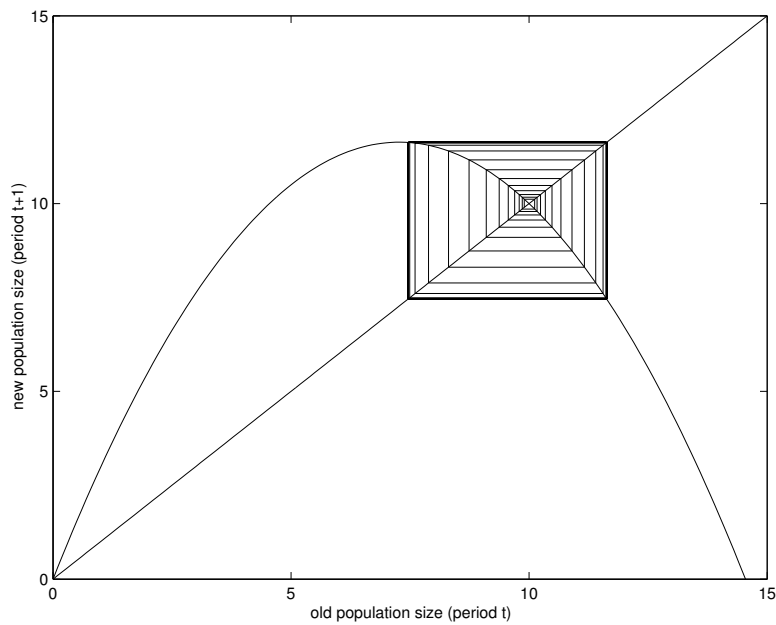


Figure 2: A 2-cycle in the logistic model ( $r = 2.2$ )

```
>> r = 2.2; k = 10;  
>> p = 9.9; y = p; for t = 1:50; p = (1+r*(1-p/k))*p; y = [y p]; end  
>> plot(0:50,y) % time path for logistic model with r = 2.2
```

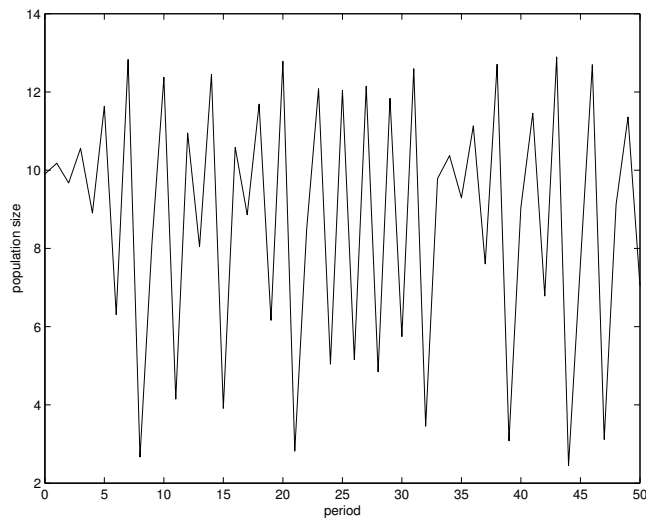


```
>> cobweb = []; for t = 1:50; cobweb = [cobweb; y(t) y(t); y(t) y(t+1)]; end  
>> p = 0:.1:15; f = (1+r*(1-p/k)).*p;  
>> plot(p,p,p,f,cobweb(:,1),cobweb(:,2)) % cobweb diagram
```

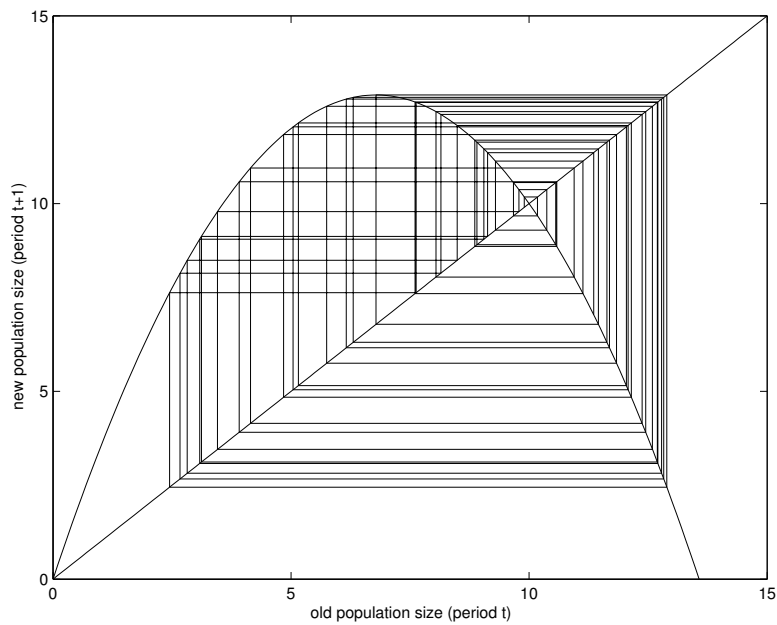


**Figure 3: Chaos in the logistic model ( $r = 2.8$ )**

```
>> r = 2.8; k = 10;  
>> p = 9.9; y = p; for t = 1:50; p = (1+r*(1-p/k))*p; y = [y p]; end  
>> plot(0:50,y) % time path for logistic model with r = 2.8
```



```
>> cobweb = []; for t = 1:50; cobweb = [cobweb; y(t) y(t); y(t) y(t+1)]; end  
>> p = 0:.1:15; f = (1+r*(1-p/k)).*p;  
>> plot(p,p,p,f,cobweb(:,1),cobweb(:,2)) % cobweb diagram
```



we see that the dynamics never converge to an equilibrium.<sup>5</sup> Rather, adopting terminology from the dynamical systems literature, we are observing *chaotic* dynamics. It is intriguing that a model as simple as the discrete logistic model can produce such complicated dynamics. This model has consequently received much attention from mathematicians, who have learned much about its behavior. For instance, increasing the parameter  $r$  more slowly from 2.2 to 2.8, we would find that the equilibrium changes from a 2-cycle to a 4-cycle to an 8-cycle (and so on) before becoming chaotic (illustrating the “period-doubling” route to chaos). Readers interested in more details are encouraged to consult the further readings at the end of this chapter.

## 11.8 Approximating continuous-time models

From a mathematical perspective, the examples reported in the previous section are fascinating, and certainly warrant further discussion. But given the applications we will be considering, it is appropriate to question their relevance. In particular, we might worry that these sorts of behaviors – oscillations, cycles, chaos – are artifacts of the discrete-time specification of the logistic growth model. To see this, consider again the case where  $r = 2.2$ . In the 2-cycle equilibrium, the population level of 7.46 induces a large positive growth rate (of 56%) which pushes the population well over capacity (to a level of 11.63); this new population level then induces a large negative growth rate (of  $-35\%$ ) which pushes the population well below capacity (to 7.46). However, these discontinuous jumps in the growth rate depend crucially on the discrete-time specification of the model. In reality, population size changes through continuous flows of births and deaths. As the population begins to approach its capacity constraint from below, the growth rate would not remain constant, but would begin to fall toward zero, thereby inhibiting the “overshooting” observed above. Conversely, as the population approaches the capacity constraint from above, the growth rate would become less negative, falling to zero as population size falls. Thus, in reality, we might expect not to observe oscillations (much less cycles or chaos).

More formally, we are describing the behavior of the continuous-time version of the logistic growth model, given by the differential equation

$$\frac{dp}{dt} = r \left(1 - \frac{p}{k}\right) p$$

where time ( $t$ ) is a continuous variable and  $dp/dt$  denotes the derivative of  $p$  with respect to  $t$ . For the benefit of readers who do not know calculus, we remain within the discrete-time framework, rewriting this equation as

$$\frac{\Delta p}{\Delta t} = r \left(1 - \frac{p}{k}\right) p$$

---

<sup>5</sup>This conclusion is somewhat premature based on merely on the first 50 periods shown in Figure 3. But the reader can verify (by altering the number of periods in the `for` loop) that the model never converges to an equilibrium even after thousands of periods.

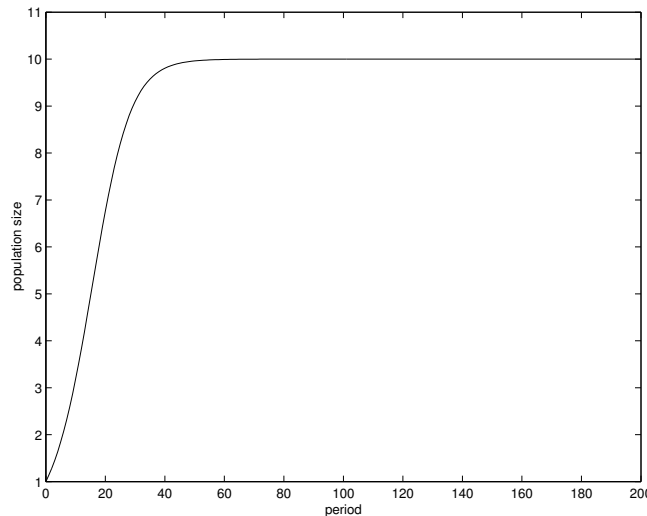
which implies

$$\Delta p = r \left(1 - \frac{p}{k}\right) p \Delta t = r \left(1 - \frac{p}{k}\right) p h$$

where  $h$  is a new parameter that denotes period length. To this point, we have implicitly set  $h = 1$ . Note that period length  $h$  does not affect the steady states of the system – the equation  $\Delta p^* = 0$  continues to have the solutions  $p^* = k$  and  $p^* = 0$  for any choice of  $h$  – but merely the disequilibrium dynamics. By setting  $h$  sufficiently small, we can approximate the behavior of the continuous-time model while remaining within the discrete-time framework (and avoiding the mathematics of differential equations).

To illustrate, consider the time path of the logistic model given  $r = 1.5$  and  $h = 0.1$ .

```
>> r = 1.5; k = 10; h = .1;
>> p = 1; y = p; for t = 1:200; dp = r*(1-p/k)*p*h; p = p+dp; y = [y p]; end
>> plot(0:200,y) % time path given r = 1.5 and h = 0.1
```



In contrast to the time-path diagram shown in Figure 1 (which assumed  $r = 1.5$  but  $h = 1$ ), we see that the population size does not oscillate, but instead rises monotonically as it converges to equilibrium. It is also important to note the change in the horizontal scale. In decreasing  $h$  from 1 to 0.1, the number of periods has increased by a factor of 10. More concretely, if period length had been conceptualized as a year (given  $h = 1$ ), we now interpret each period as 1/10 of a year (given  $h = 0.1$ ). Thus, while the two time-path diagrams illustrate dynamics over the same time frame (20 years), the diagram above subdivides each year into 10 periods.

Partly to address the concerns we have just raised, researchers often use continuous-time (differential equation) models rather than discrete-time (difference equation) models for empirical applications. However, for the remainder of this book, we will

continue to recast continuous-time as discrete-time models, setting period length as small as necessary to approximate the continuous-time dynamics. Moreover, even for models originally specified in discrete time, we will sometimes incorporate a period-length parameter when it appears that the dynamics are an artifact of the discrete-time specification. On the other hand, it is important to realize that discrete-time specifications may actually be more realistic for some types of empirical applications, and that the potential for cycles or chaos may be genuine rather than artifactual. Mathematical modelers must inevitably exercise judgement about the most appropriate model for a given application. Further, in moving to higher dimensional models, even continuous-time models admit cycles (for  $n \geq 2$ ) and chaos (for  $n \geq 3$ ). We'll return to the topic of cycles when we address predator-prey models in Chapter xx.

## 11.9 Further reading

More advanced treatments of nonlinear dynamical systems include P G Drazin, *Nonlinear Systems* (Cambridge 1992) and Steven H Strogatz, *Nonlinear Dynamics and Chaos* (Perseus 1994). A graphical introduction to nonlinear systems is provided by Ralph H Abraham and Christopher D Shaw, *Dynamics: The Geometry of Behavior* (Addison-Wesley 1992). Our treatment of the discrete logistic model draws on the presentation by Elizabeth S Allman and John A Rhodes, *Mathematical Models in Biology* (Cambridge 2004). This model began to receive much attention from mathematicians following the classic article by R M May (1978) "Simple Mathematical Models with Very Complicated Dynamics," *Nature* 261:459-567. For more extensive treatment of this class of models, see Robert L Devaney, *A First Course in Chaotic Dynamical Systems* (Addison-Wesley 1992). For students interested in learning about differential equations, there are many textbooks including Paul Blanchard, Robert L Devaney, Glen R Hall, *Differential Equations* (Thompson Brooks/Cole 2006).