Chapter 15

Predators and Prey

Models of population growth.

The simplest model for the growth, or decay, of a population says that the growth rate, or the decay rate, is proportional to the size of the population itself. Increasing or decreasing the size of a the population results a proportional increase or decrease in the number of births and deaths. Mathematically, this is described by the differential equation

$$\dot{y} = ky$$

The proportionality constant k relates the size of the population, $y(t)$, to its rate of growth, $\dot{y}(t)$. If k is positive, the population increases; if k is negative, the population decreases.

As we know, the solution to this equation is a function $y(t)$ that is proportional to the exponential function

$$y(t) = \eta e^{kt}$$

where $\eta = y(0)$.

This simple model is appropriate in the initial stages of growth when there are no restrictions or constraints on the population. A small sample of bacteria in a large Petri dish, for example. But in more realistic situations there are limits to growth, such as finite space or food supply. A more realistic model says that the population competes with itself. As the population increases, its growth rate decreases linearly. The differential equation is sometimes called the *logistic* equation.

$$\dot{y} = k(1 - \frac{y}{\mu})y$$

April 6, 2008

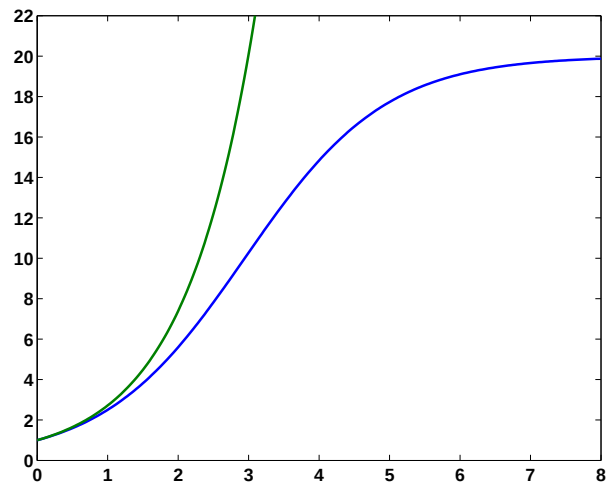

Figure 15.1. *Exponential growth and logistic growth.*

The new parameter μ is the *carrying capacity*. As $y(t)$ approaches μ the growth rate approaches zero and the growth ultimately stops. It turns out that the solution is

$$y(t) = \frac{\mu\eta e^{kt}}{\eta e^{kt} + \mu - \eta}$$

You can easily verify for yourself that as t approaches zero, $y(t)$ approaches η and that as t approaches infinity, $y(t)$ approaches μ. If you know calculus, then with quite a bit more effort, you can verify that $y(t)$ actually satisfies the logistic equation.

Figure 15.1 shows the two solutions when both η and k are equal to one. The exponential function

$$y(t) = e^t$$

gives the rapidly growing green curve. With carrying capacity $\mu = 20$, the logistic function

$$y(t) = \frac{20e^t}{e^t + 19}$$

gives the more slowly growing blue curve. Both curves have the same initial value and initial slope. The exponential function grows exponentially, while the logistic function approaches, but never exceeds, its carrying capacity.

Figure 15.1 was generated with the following code.

```
k = 1
eta = 1
mu = 20
t = 0:1/32:8;
```

```
y = mu*eta*exp(k*t)./(eta*exp(k*t) + mu - eta);
plot(t,[y; exp(t)])
axis([0 8 0 22])
```

If you don't have the formula for the solution to the logistic equation handy, you can compute a numerical solution with `ode45`, one of the MATLAB ordinary differential equation solvers. Try running the following code. It will automatically produce a plot something like the blue curve in figure 15.1.

```
k = 1
eta = 1
mu = 20
ydot = @(t,y) k*(1-y/mu)*y
ode45(ydot,[0 8],eta)
```

The `@` sign and `@(t,y)` specify that you are defining a function of `t` and `y`. The `t` is necessary even though it doesn't explicitly appear in this particular differential equation.

The logistic equation and its solution occur in many different fields. The logistic function is also known as the *sigmoid* function and its graph is known as the *S-curve*.

Populations do not live in isolatation. Everybody has a few enemies here and there. The Lotka-Volterra predator-prey model is the simplest description of competion between two species. Think of rabbits and foxes, or zebras and lions, or little fish and big fish.

The idea is that, if left to themselves with an infinite food supply, the rabbits or zebras would live happily and experience exponential population growth. On the other hand, if the foxes or lions were left with no prey to eat, they would die faster than they could reproduce, and would experience exponential population decline.

The predator-prey model is a pair of differential equations involving a pair of competing populations, $y_1(t)$ and $y_2(t)$. The growth rate for y_1 is a linear function of y_2 and vice versa.

$$\begin{aligned} \dot{y}_1 &= (1 - \frac{y_2}{\mu_2})y_1 \\ \dot{y}_2 &= -(1 - \frac{y_1}{\mu_1})y_2 \end{aligned}$$

We are using notation $y_1(t)$ and $y_2(t)$ instead of, say, $r(t)$ for rabbits and $f(t)$ for foxes, because our MATLAB program uses a two-component vector `y`.

The extra minus sign in the second equation distinguishes the predators from the prey. Note that if y_1 ever becomes zero, then

$$\dot{y}_2 = -y_2$$

and the predators are in trouble. But if y_2 ever becomes zero, then

$$\dot{y}_1 = y_1$$

and the prey population grows exponentially.

We have a formula for the solution of the single species logistic model. However it is not possible to express the solution to this predator-prey model in terms of exponential, trigonmetric, or any other elementary functions. It is necessary, but easy, to compute numerical solutions.

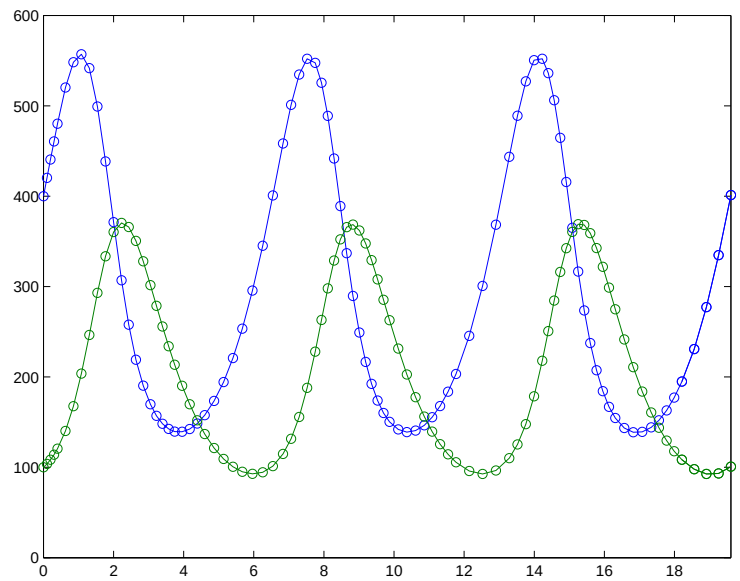

Figure 15.2. *A typical solution of the predator-prey equations.*

There are four parameters, the two constants μ_1 and μ_2, and the two initial conditions,

$$\begin{aligned} \eta_1 &= y_1(0) \\ \eta_2 &= y_2(0) \end{aligned}$$

If we happen to start with $\eta_1 = \mu_1$ and $\eta_2 = \mu_2$, then both $\dot{y}_1$ and $\dot{y}_2$ are zero and the populations remain constant at their initial values. In other words, the point (μ_1, μ_2) is an *equilibrium* point. The origin, $(0, 0)$ is another *equilibrium* point, but not a very interesting one.

The following code uses `ode45` to automatically plot the typical solution shown in figure 15.2.

```
mu = [300 200]'
eta = [400 100]'
sig = [1 -1]'
ppode = @(t,y) sig.*flipud(1-y./mu).*y
pit = 6.5357
ode45(ppode,[0 3*pit],eta)
```

There are two tricky parts of this code. MATLAB vector operations are used to define `ppode`, the predator-prey differential equations, in one line. And, the calculation that generates figure 15.3 provides the value assigned to `pit`. This value specifies a value of t when the populations return to their initial values given by `eta`. The code integrates over three of these time intervals, and so at the end we get back to where we started.

The circles superimposed on the plots in figure 15.2 show the points where `ode45` computes the solution. The plots look something like trig functions, but they're not. Notice that the curves near the minima are broader, and require more steps to compute, then the curves near the maxima. The plot of $\sin t$ would look the same at the top as the bottom.

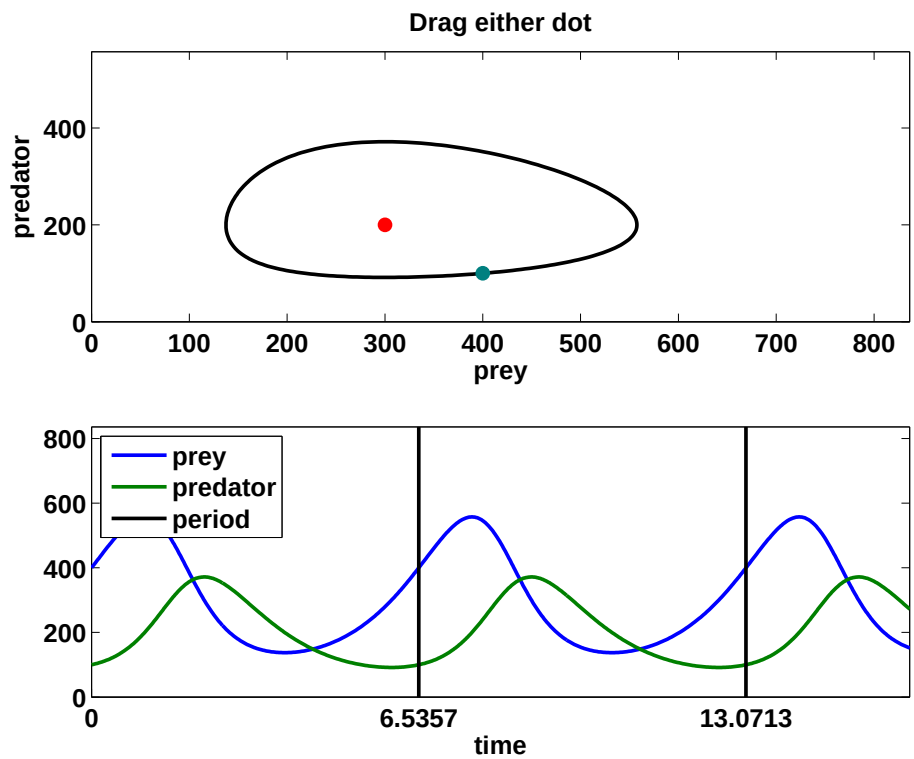

Figure 15.3. *The predprey experiment.*

Our MATLAB program `exm/predprey` shows a red dot at the equilibrium point, (μ_1, μ_2), and a blue-green dot at the initial point, (η_1, η_2). When you drag either dot with the mouse, the solution is recomputing by `ode45` and plotted. Figure 15.3 shows that two plots are produced — a *phase plane* plot of $y_2(t)$ versus $y_1(t)$ and a *time series* plot of $y_1(t)$ and $y_2(t)$ versus t. Figures 15.2 and 15.3 have the same parameters, and consequently show the same solution, but with different scaling of

the axes.

The remarkable property of the Lotka-Volterra model is that the solutions are always periodic. The populations always return to their initial values and repeat the cycle. This property is not obvious and not easy to prove. It is rare for nonlinear models to have periodic solutions. The `predprey` program uses a feature of the MATLAB ODE solvers called "event handling" to compute the length of a period.

If the initial values (η_1, η_2) are close to the equilibrium point (μ_1, μ_2), then the length of the period is close to a familar value. An exercise asks you to discover that value experimentally.

Exercises

15.1 *Plot.* Make a more few plots like figures 15.1 and 15.2, but with different values of the parameters k, η, and μ.

15.2 *Decay.* Compare exponential and logistic decay. Make a plot like figure 15.1 with negative k.

15.3 *Differentiate.* Verify that our formula for $y(t)$ actually satisfies the logistic differential equations.

15.4 *Easy as pie.* In `predprey`, if the red and blue-green dots are close to each other, then the length of the period is close to a familar value. What is that value? Does that value depend upon the actual location of the dots, or just their relative closeness?

15.5 *Period.* In `predprey`, if the red and blue-green dots are far apart, does the length of the period get longer or shorter? Is it possible to make the period shorter than the value it has near equilibrium?

15.6 *Phase.* If the initial value is near the equilibrium point, the graphs of the predator and prey populations are nearly sinusoidal, with a phase shift. In other words, after the prey population reaches a maximum or minimum, the predator population reaches a maximum or minamum some fraction of the period later. What is that fraction?

15.7 *Pitstop.* The `predprey` subfunction `pitstop` is involved in the "event handling" that `ode45` uses to compute the period. `pitstop`, in turn, uses `atan2` to compute angles `theta0` and `theta1`. What is the difference between the two MATLAB functions `atan2`, which takes two arguments, and `atan`, which takes only one? What happens if `atan2(v,u)` is replaced by `atan(v/u)` in `predprey`?
Draw a sketch showing the angles `theta0` and `theta1`.

15.8 *tfinal.* The call to `ode45` in `predprey` specifies a time interval of [0 100]. What is the significance of the value 100? What happens if you change it?

15.9 *Limit growth.* Modify `predprey` to include a growth limiting term for the prey, similar to one in the logistic equation. Avoid another parameter by making the carrying capacity twice the initial value. The equations become

$$\begin{aligned} \dot{y}_1 &= (1 - \frac{y_1}{2\,\eta_1})(1 - \frac{y_2}{\mu_2})y_1 \\ \dot{y}_2 &= -(1 - \frac{y_1}{\mu_1})y_2 \end{aligned}$$

What happens to the shape of the solution curves? Are the solutions still periodic? What happens to the length of the period?