

Differential Equation Model.

Let $x(t)$ be the fox population size and

$y(t)$ be the rabbit population size.

Modify the birth rate minus the death rate for the fox populations to be $-d + ey$. The e is a positive constant because the rabbits are a source of food for the foxes.

Thus, the differential equation for the foxes is

$$x' = (-d + ey)x \text{ and } x(0) = x_0. \quad \text{Fox Eq.}$$

A similar term must be inserted into the birth rate minus the death rate for the rabbits. But, in this case the rabbit population should decrease, and so, we will insert $-cx$.

$$y' = (b - cx)y \text{ and } y(0) = y_0. \quad \text{Rabbit Eq.}$$

These two differential equations are coupled, and they are known as the **Lotka-Volterra**:

$$x' = (-d + ey)x \text{ and } x(0) = x_0. \quad \text{Fox Eq.}$$

$$y' = (b - cx)y \text{ and } y(0) = y_0. \quad \text{Rabbit Eq.}$$

If the rabbit population is very large, then there will be plenty of food for the fox population and it should increase. As the fox population increases, more of the rabbits are consumed. So, the food supply for the fox population decreases, and the fox population will have to decrease.

One expects there is a perfect balance of the populations where neither population changes. This means both x' and y' must be zero. This is called the **steady state or equilibrium solution**. There is one positive steady state solution

$$x = b/c \text{ from the rabbit equation, and } y = d/e$$

from the fox equation.

Method of Solution.

Matlab's ode45 can be used to find numerical solution of systems of ODEs. The numerical method is a variation of the Runge-Kutta 45 variable time step method.

Matlab Implementation.

Use the m-files yprf.m and rf.m.

Note the column vectors yprf in yprf.m and yo in rf.m

Note the two types of graphs generated in rf.m

```
function yprf =yprf(t,y)
    yprf(1) =-.5*y(1) + .01*y(1)*y(2);
    yprf(2) = .5*y(2) -.01*y(1)*y(2);
    yprf = [yprf(1) yprf(2)]';
```

```
%your name, your student number,  
lesson number  
clear;  
to = 0;  
tf =50;  
yo = [80 100];  
[t y] = ode45('yprf',[to tf],yo);  
plot(y(:,1),y(:,2))  
title('your name, your student  
number, lesson number')  
ylabel('rabbits')  
xlabel('fox')  
%plot(t,y(:,1),t,y(:,2))  
%xlabel('time')  
%ylabel('rabbits and fox')
```

The nonzero steady state solutions are

$$x = 50 \text{ and } y = 50.$$

Our numerical solution should orbit about the steady state solution. The first graph below does not quite do this. This is because some of the error options for ode45 need to be made smaller. The default value for relative error is .001, and this is what was used in the first graph below.

The Matlab command `odeset('RelTol', 1e-5)` was used to "tune" the ode45 method. Adjust the rf.m file by replacing line 6, the line with ode45, by the two following lines:

```
options = odeset('RelTol', 1e-5);  
[t y] = ode45('yprf',  
             [to tf],yo,options);
```

Here more time steps will be required; use the `whos` command to find the number of time steps $n = 133$ with $\text{RelTol} = 1e-3$, and $n = 289$ with $\text{RelTol} = 1e-5$.

