APPM 2360 Project 1: A Mathematical Investigation of Populations and Predator-Prey Dynamics

Lilac Intrater (Section 251), David Orozco (Section 266), Jack Ohlmeier (Section 231)

February 28, 2019

Introduction

During the 1980's in the Sierra Nevada, researchers working for the National Forest Service conducted a study on the native mule deer population in the area. They noticed that the population of deer had decreased dramatically since the 1950 census on the population, which they attributed to over-population of the habitat. However, the population continued to fall even beyond a level which the researches deemed sustainable within the habitat. As the researches continued their study, they discovered that the population of the native mountain lion, a natural predator of mule deer, had steadily increased as the population of deer decreased. The interactions between these two species, deer and mountain lions, were studied by the researchers to deduce reasons why the deer population remained low even after its initial decline. In this project, we will use various models to mathematically analyze the interaction between predators and prey.

Modeling Individual Populations Using the Logistic Equation

One way to analyze the population of deer and mountain lions is to model each population individually and see how they change over time.

If we assume the mountain lions are protected from hunting, have no natural predators, and assume the lions' food source (the deer population) is finite, we can model the population of mountain lions using the logistic equation:

$$\frac{dx}{dt} = r\left(1 - \frac{x}{L}\right)x$$

$x(t)$ is the population of mountain lions in dozens of animals at any given time t in days. The parameters r and L are both greater than zero, with r and L being the initial growth rate of the population and carrying capacity of the population respectively. The units on r are dozens of animals per day and the units on L are dozens of animals.

The equilibrium solutions to the equation above are $x(t) = 0$ and $x(t) = L$. Solving the equation by separation of variables yields the following non-equilibrium solution (Derivation D1):

$$x(t) = \frac{x_0 e^{rt} L}{L - x_0 + x_0 e^{rt}}$$

where $x_0 = x(0)$,or the population at time zero.

If we acquire the values $r = 0.65$ and $L_M = 5.4$ for our parameters and assume that the initial population of the mountain lions is 6, we can use Euler's method over the time interval from 0 days to 25 days to solve the logistic equation with varying step size h.

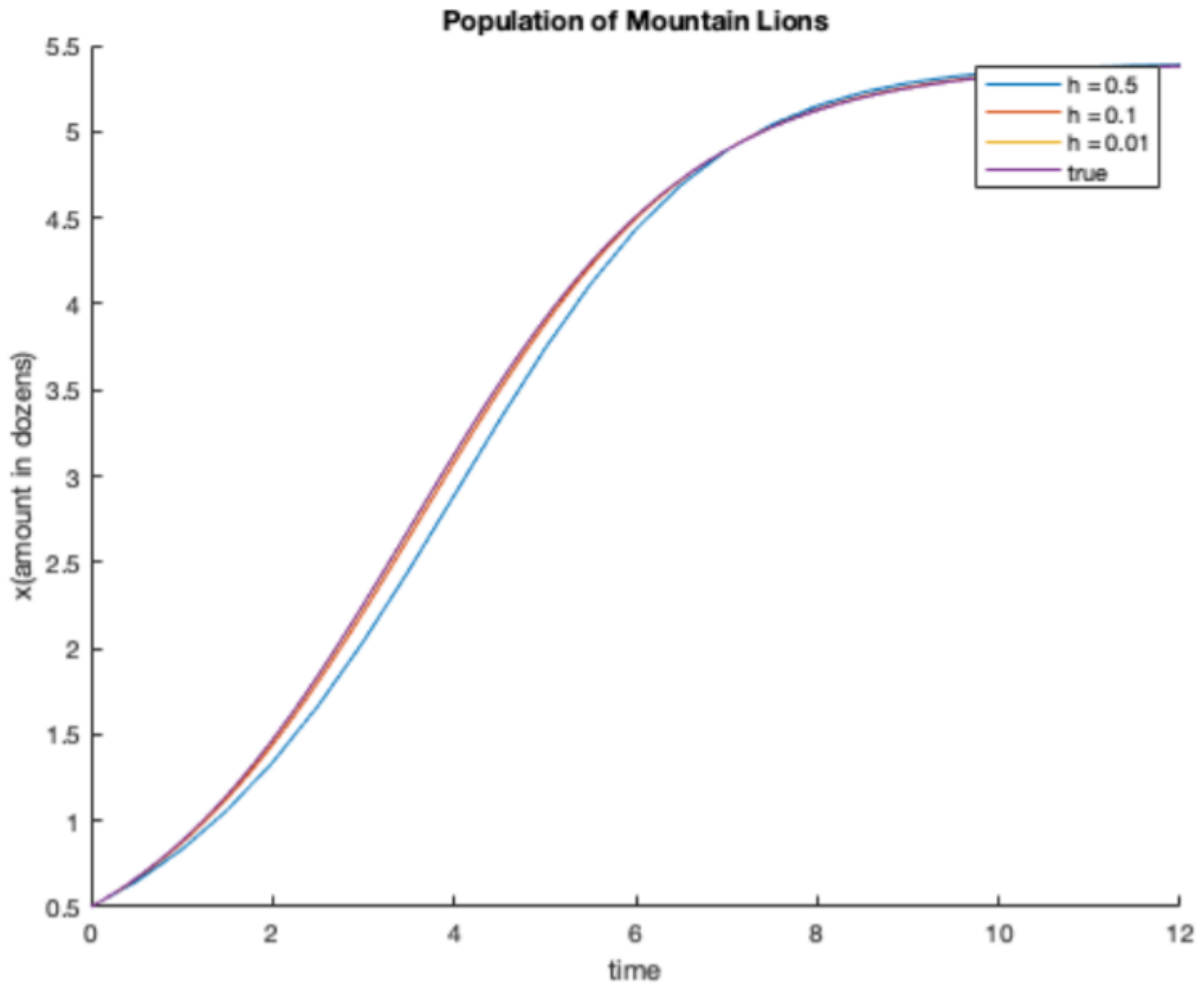


Figure 1: Mountain lion population as a function of time. Approximate solutions using Euler's Method with varying step size h and the real solution are plotted.

As seen in Figure 1, decreasing the step size yields more accurate approximate solutions. To get a better picture of how step size affects error, we can plot absolute errors of each approximation together in Figure 2.

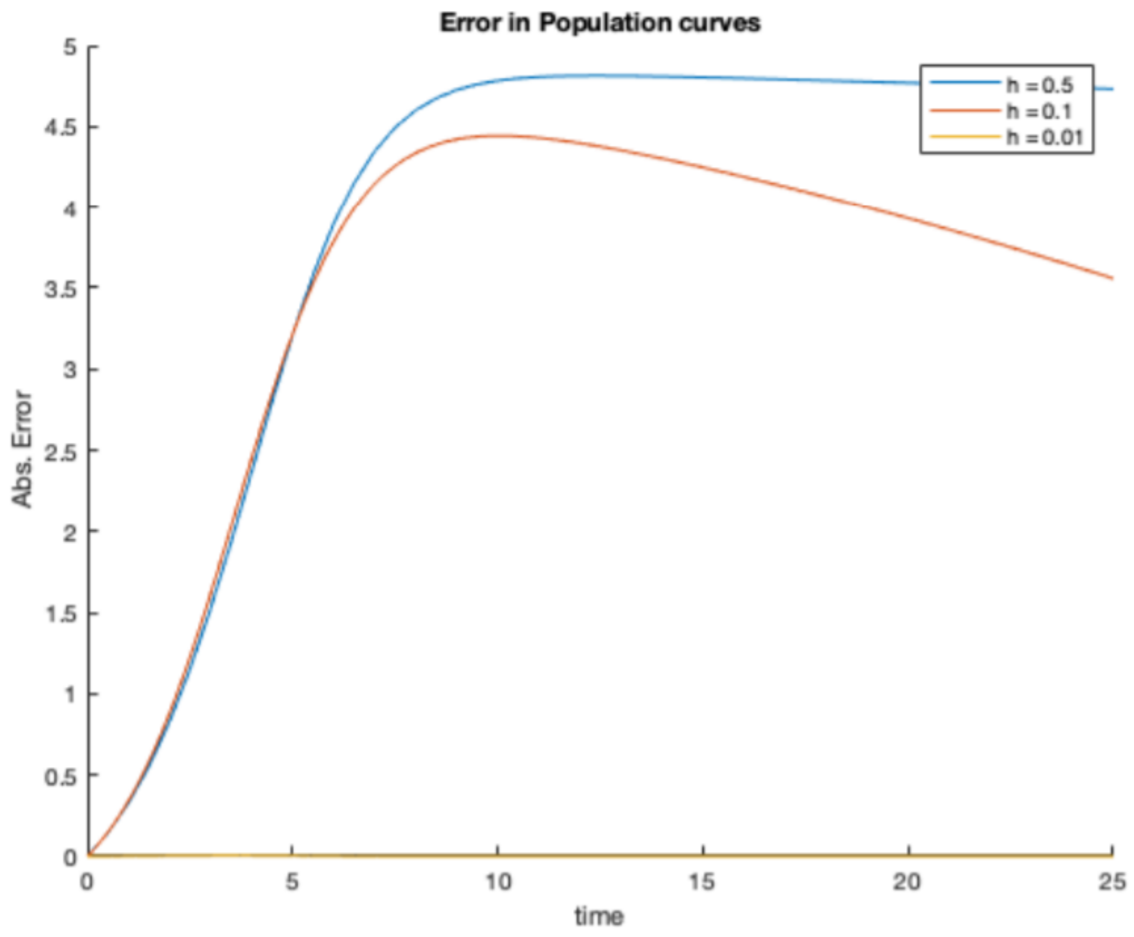


Figure 2: Error in approximate solutions.

As we can see, for all step sizes (it is hard to see for step size 0.01, but it is visible when the graph is enlarged), error rises rapidly before peaking or leveling off at about $t = 7$. This “spike” is most likely due to the fact that the approximation curves are an underestimate of the actual solution before time 7 and an overestimate after time 7.

Looking at Figures 1 and 2, we can see that a step size of 0.01 produced an extremely accurate approximation of the real solution and the computational efficiency of calculating an approximate solution was not much worse than using larger step sizes. From this, we conclude that using a step size of 0.01 was the best balance of numerical accuracy and efficiency.

We can model the deer population in a similar way to how we modeled the mountain lion population. This time including a harvesting function $H(x)$ to represent the number of deer killed by mountain lions. By including this new term we arrive at the model:

$$\frac{dx}{dt} = r\left(1 - \frac{x}{L}\right)x - H(x)$$

Where all parameters are the same as the model for the mountain lion population. This differential equation is nonlinear, 1st degree, autonomous, and constant coefficient. In a physical sense, the fact that this equation is autonomous means that the rate at which deer population is changing depends only on the current deer population and the amount of deer that are being killed. It does not depend on time. If population is constant, the rate at which the population grows will not change just because time has passed.

A reasonable harvesting function imposed on the population of deer could be:

$$H(x) = \frac{px^2}{q + x^2}$$

Where p and q represent how skilled the mountain lions are at catching deer.

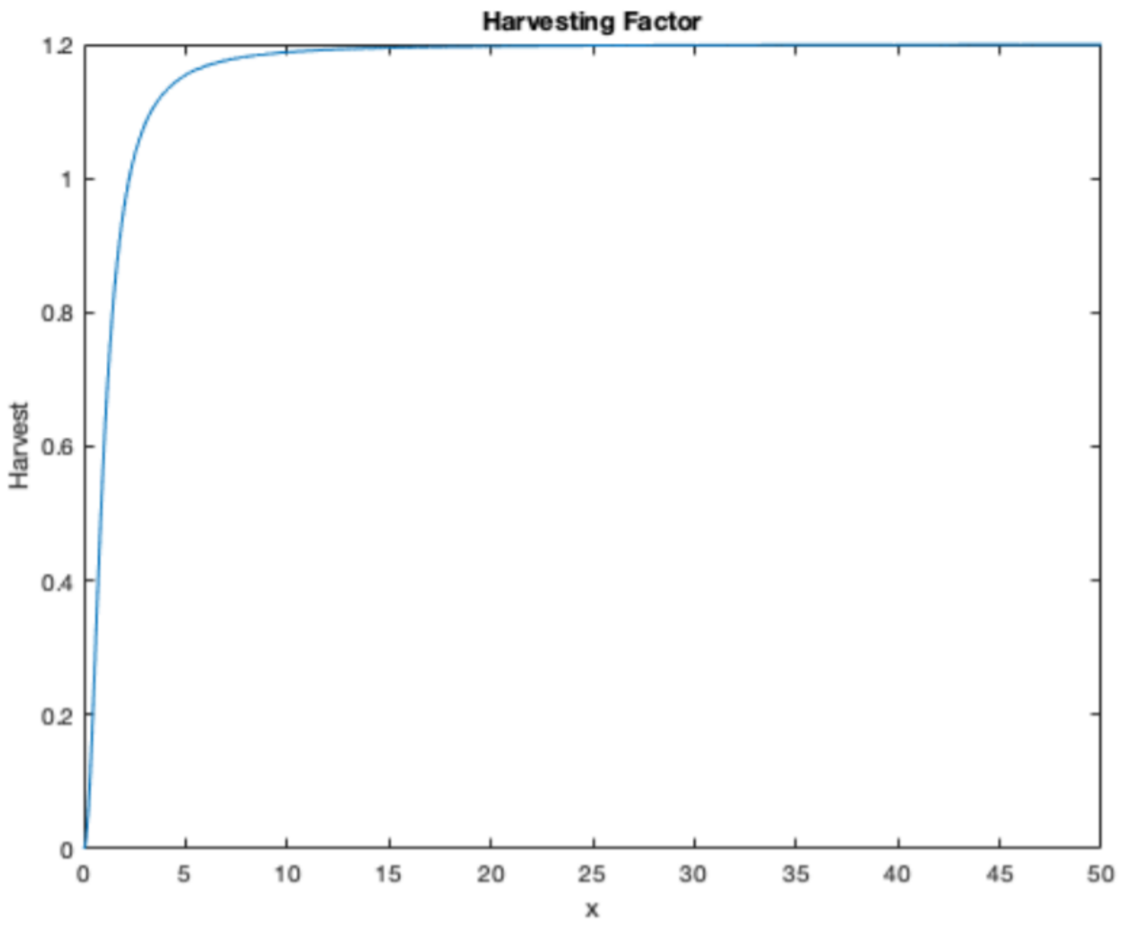


Figure 3: Harvesting factor as a function of deer population. Parameters are arbitrarily chosen.

As we can see, if the deer population is very large, the harvesting factor is correspondingly large, and if the deer population is zero, no deer can be hunted. Therefore, the harvesting factor is zero. This makes sense physically as a small deer population means that it is hard for mountain lions to find deer to kill. If deer population is large, there is an abundance of deer for mountain lions to hunt.

If we assume parameter values $r = 0.65, L_D = 8.1, p = 1.2$ and $q = 1$, and assume the initial deer population is 24, we can solve the logistic equation with harvesting numerically using Euler's method with step size 0.1. Figure 4 plots approximate solutions from various initial conditions.

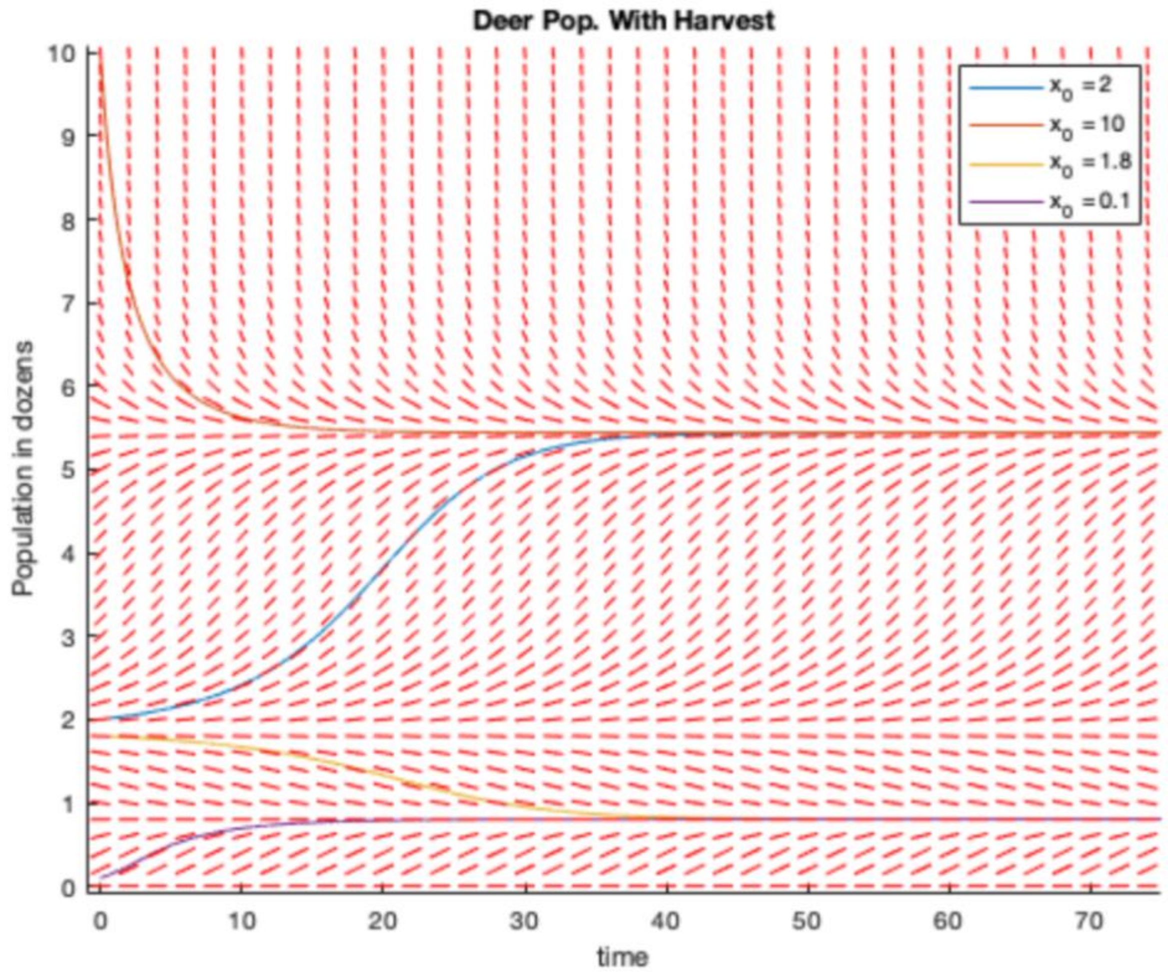


Figure 4: Approximate solutions obtained using Euler's method with step size 0.1 for various initial conditions.

From Figure 4, we can see that the solution changes with the initial condition. It appears in this case that in the long run, the deer population can't exceed about 5.5 dozen. In addition, it appears that there is another long run equilibrium population of about 0.9 dozen. The solutions seem to go towards the smaller equilibrium if the initial population is below about 2 dozen and go towards the larger equilibrium if the initial population is above about 2 dozen. This is because there appears to be an unstable equilibrium at a population of about 2 dozen deer, which splits possible solutions into either going towards the larger equilibrium or the smaller equilibrium.

Modeling Population Interactions Using the Lotka-Volterra System

While modeling populations using the logistic equation is a robust method for individual populations, there is no way to directly represent the interaction between two species. One of the simplest ways to model the interaction between a predator species and prey species, such as mountain lions and mule deer, is through the Lotka-Volterra system. The system reflects the Balance Law, which states the net rate of change of a population is equal to the rate-in of members minus the rate-out members. When we use x_1 and x_2 to represent the mountain lion and deer populations respectively, the model appears as follows.

$$\frac{dx_1}{dt} = -ax_1 + bx_1x_2$$

$$\frac{dx_2}{dt} = cx_2 - dx_1x_2$$

The parameters represent the following: a is the predator mortality rate, b is the predator attack rate/ conversion efficiency, c is the prey growth rate, and d is the prey mortality rate/predator attack rate. The second term in each equation represent the interaction between the two species, showing that interaction grows the population of mountain lions and shrinks the population of deer. The Lotka-Volterra system is a system of first order, autonomous, linear differential equations. Nullclines can be found analytically, which when done gives nullclines of $x_1 = 0, x_2 = 0, x_1 = \frac{c}{d}$, and $x_2 = \frac{a}{b}$. Equilibrium solutions, given in the form (x_1, x_2) are $(0,0)$ and $\left(\frac{c}{d}, \frac{a}{b}\right)$ (Derivation D2). If we assign parameter values $a = 1.5, b = 1.1, c = 2.5$ and $d = 1.4$, we can create a phase portrait of the system (Figure 5).

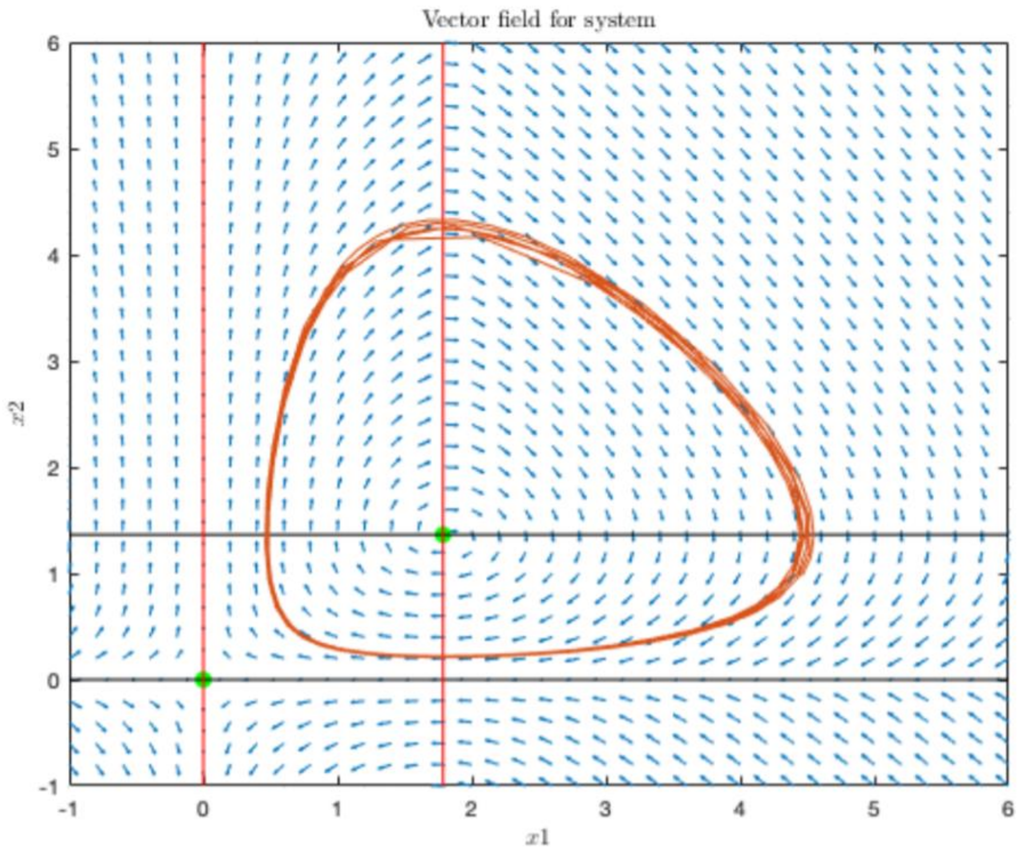


Figure 5: Phase portrait for the Lotka-Volterra System. Nullclines from the x_1' equation are plotted in red, and nullclines from the x_2' equation are plotted in black. Green dots are equilibrium solutions. The orange curve is a simulated solution starting from the initial condition $x_1(0) = 0.5, x_2(0) = 1$.

Looking at Figure 5, we can see that the equilibrium solution at $(0,0)$ is unstable, as any solution starting near it would diverge away from the origin. Looking at the other equilibrium, it appears that solutions will swirl around it. The simulated solution curve on the graph behaves as expected, moving cyclically, as there are no stable equilibria for the populations to converge to.

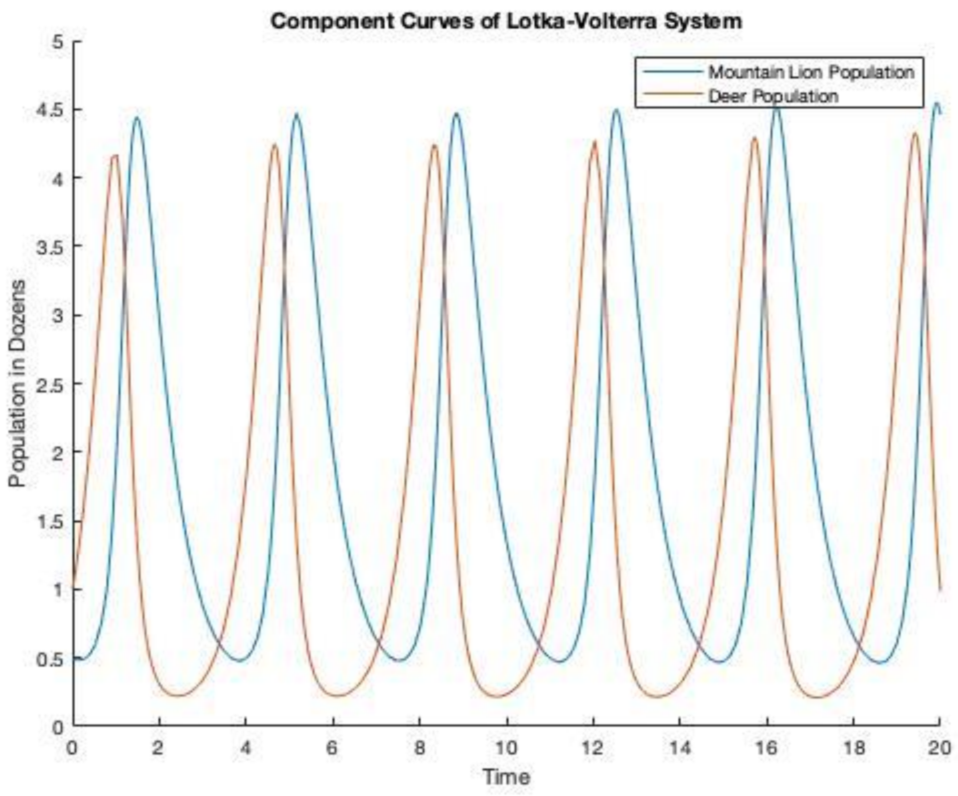


Figure 6: Component curves for a sample solution of the Lotka-Volterra system.

Component curves for the system, plotted in Figure 6, can help us better visualize the sample solution generated. As the deer population increases, the mountain lions have more food and their population grows accordingly. But soon the mountain lion population gets too high, and their hunting starts to shrink the deer population, and the mountain lion population falls from lack of food. Once the mountain lion population falls, the deer population can grow again and the cycle repeats itself. The curves are out of phase, reflecting that interaction of the predator and prey depend on the population of the other.

The Logistic Predator-Prey Model

One of the underlying assumptions of the Lotka-Volterra system is that in the absence of inter-species interaction, both species will exhibit exponential growth, ignoring the natural limits imposed on a prey population by its environment. The Logistic Predator-Prey system accounts for these limitations and is as follows:

$$\frac{dx_1}{dt} = -ax_1 + bx_1x_2$$

$$\frac{dx_2}{dt} = c(1 - kx_2)x_2 - dx_1x_2$$

All parameters have the same meaning as in the Lotka-Volterra system. Now, in the absence of a predator, prey will exhibit logistic behavior, and in absence of any prey, the predator population will exhibit exponential decay. Nullclines for the system are $x_1 = 0, x_2 = 0, x_1 = \frac{1}{k} - \frac{d}{ck}x_1$, and $x_2 = \frac{a}{b}$, and the equilibrium solutions given in the form (x_1, x_2) are $(0, 0)$ and $\left(\frac{cb-ack}{bd}, \frac{a}{b}\right)$ (Derivation D3). If we assign parameter values $a = 1.5, b = 1.1, c = 2.5, d = 1.4$, and $k = 0.5$, we can create a phase portrait and generate a sample solution like we did for the Lotka-Volterra system.

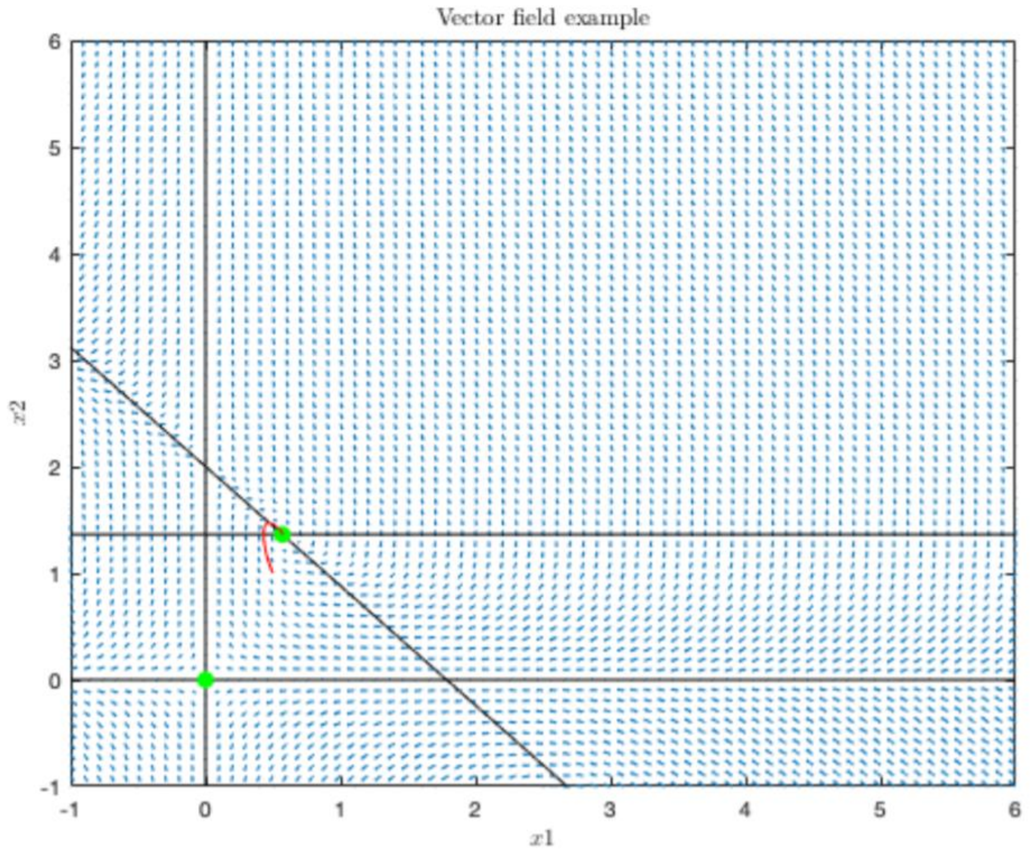


Figure 7: Phase Portrait for the Logistic Predator-Prey system. Equilibrium solutions are represented as green dots. The red curve is a solution generated from the initial condition $x_1(0) = 0.5, x_2(0) = 1$.

Looking at Figure 7, we can see that the equilibrium solution at $(0, 0)$ is unstable, and the other equilibrium solution is stable. From this, we can predict that any solution will approach this stable equilibrium. Using the initial condition $x_1(0) = 0.5, x_2(0) = 1$, a sample solution was generated. We can further study the sample solution by plotting component curves for the solution.

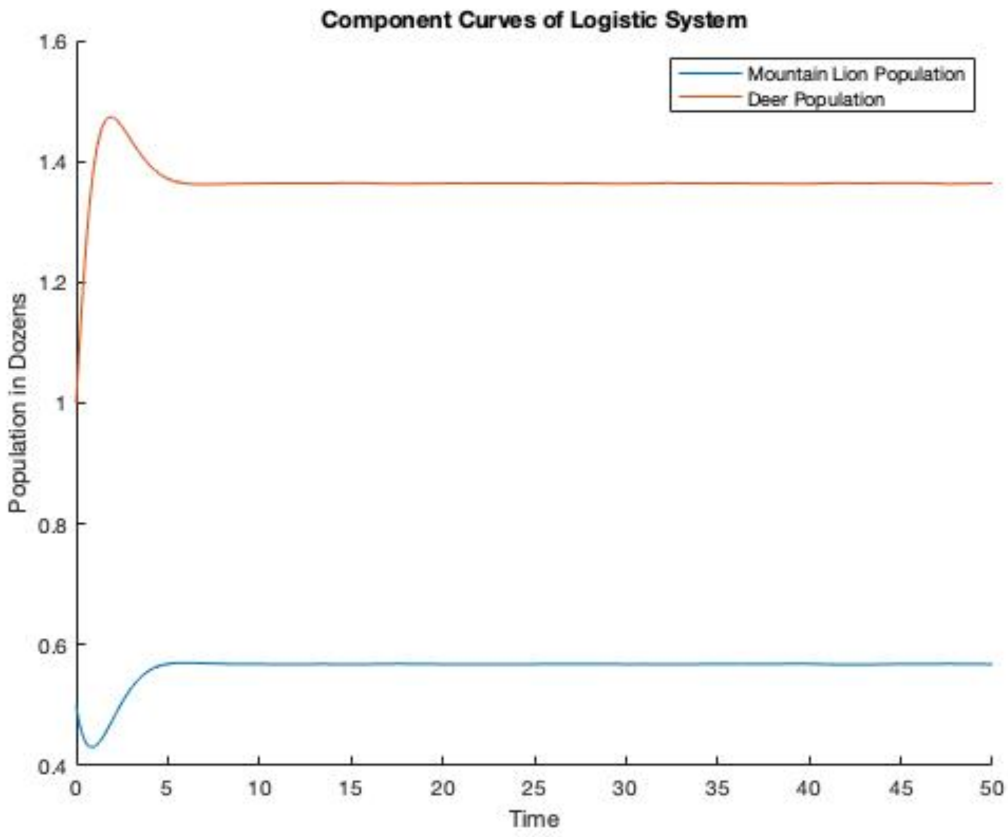


Figure 8: Component Curves for a sample solution of the Logistic Predator-Prey system

From the component curves, we can see the sample solution is non-periodic and the populations asymptotically reach a constant level.

Comparing the Lotka-Volterra and Logistic Predator-Prey Systems

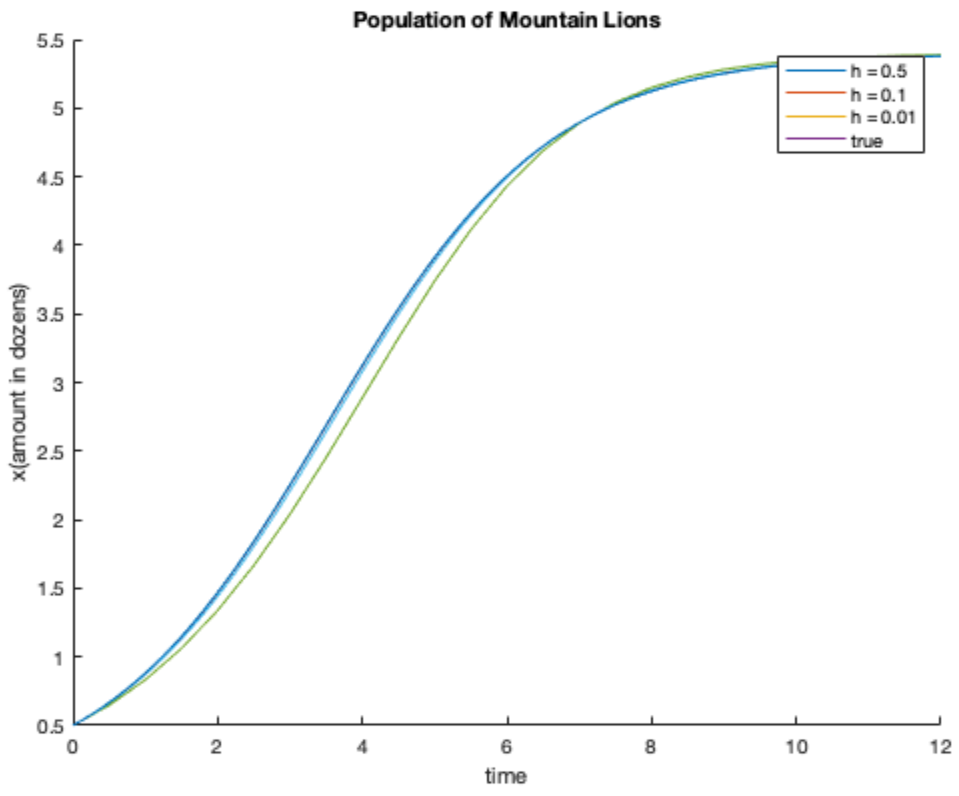
The strength of the Lotka-Volterra model is that it considers how the population of one animal depends on the population of the other. It shows how the mountain lion population follows the deer population. The disadvantage of this model, however, is that it assumes that in the absence of predators, the prey population can grow exponentially without stopping. The strength of the Logistic Predator-Prey model is that it shows that even without any predators, the prey population can not exponentially grow due to limitations on its' environment. This is more realistic for the world we live in. It also shows the inverse relationship between the growth of the predator and prey populations. The disadvantage of this model is that it does not show how the two populations interact with each other. In addition, both of the models assume that deer are the only food source for mountain lions and mountain lions are the only predator of deer. A change to make the models more accurate is to add a factor to the population growth of mountain lions that considers other food sources. This way, the mountain lion population is not as dependent on the deer population.

Conclusion

The populations of mule deer and mountain lions in the Sierra Nevada can be modeled mathematically in various ways. Modelling each population individually using the logistic equation is simple but fails to reflect interaction between the two species. A simple way to model the populations with inter-species interaction was through the Lotka-Volterra system. However, the Lotka-Volterra system assumed no limits on the prey population except for interaction with predators. This led to the development of the Logistic Predator-Prey system, under which a prey population would exhibit logistic behavior in the absence of any predators. The Logistic Predator-Prey model seems to be the most accurate model for what happens in real life but there are some changes that can be made which could increase its accuracy.

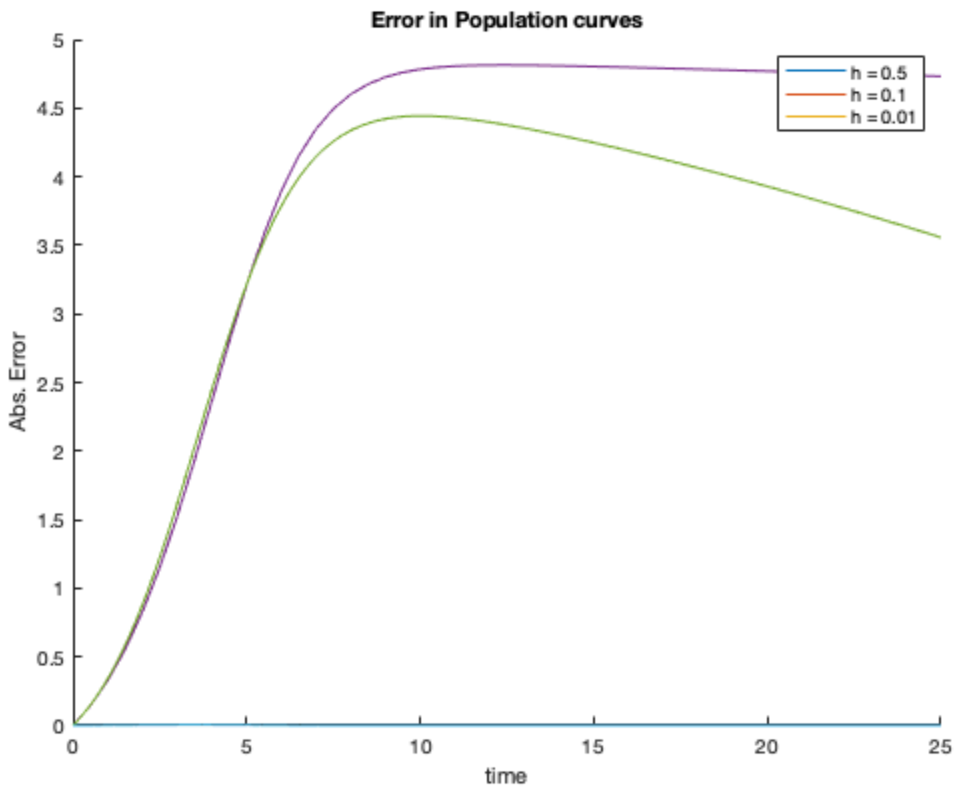
Euler approximation code for Section 2, Question 3 a

```
x_prime = @(t,x) .65*x.*(1-x/5.4);
h1 = .5;
h2 = .1;
h3 = .01;
t1 = 0:h1:25;
t2 = 0:h2:25;
t3 = 0:h3:25;
x1 = zeros(1,length(t1));
x1(1) = .5;
x2 = zeros(1,length(t2));
x2(1) = .5;
x3 = zeros(1,length(t3));
x3(1) = .5;
for n = 1:(length(t1)-1)
    x1(n+1) = x1(n) + h1*x_prime(t1(n),x1(n));
end
for n = 1:(length(t2)-1)
    x2(n+1) = x2(n) + h2*x_prime(t3(n),x2(n));
end
for n = 1:(length(t3)-1)
    x3(n+1) = x3(n) + h3*x_prime(t3(n),x3(n));
end
xo = .5;
r = .65;
lm = 5.4;
t = 0:.01:25;
x = (xo*exp(r*t)*lm)./(lm-xo+(xo*exp(r*t)));

figure(1)
hold on
plot(t1,x1,'LineWidth', .3)
plot(t2,x2,'LineWidth', .4)
plot(t3,x3,'LineWidth', .7)
plot(t,x,'LineWidth', 1)
xlabel('time'), ylabel('x(amount in dozens)'), title("Population of
 Mountain Lions")
legend("h = 0.5","h = 0.1","h = 0.01","true")
axis([0 12 .5 5.5])

hold off
```



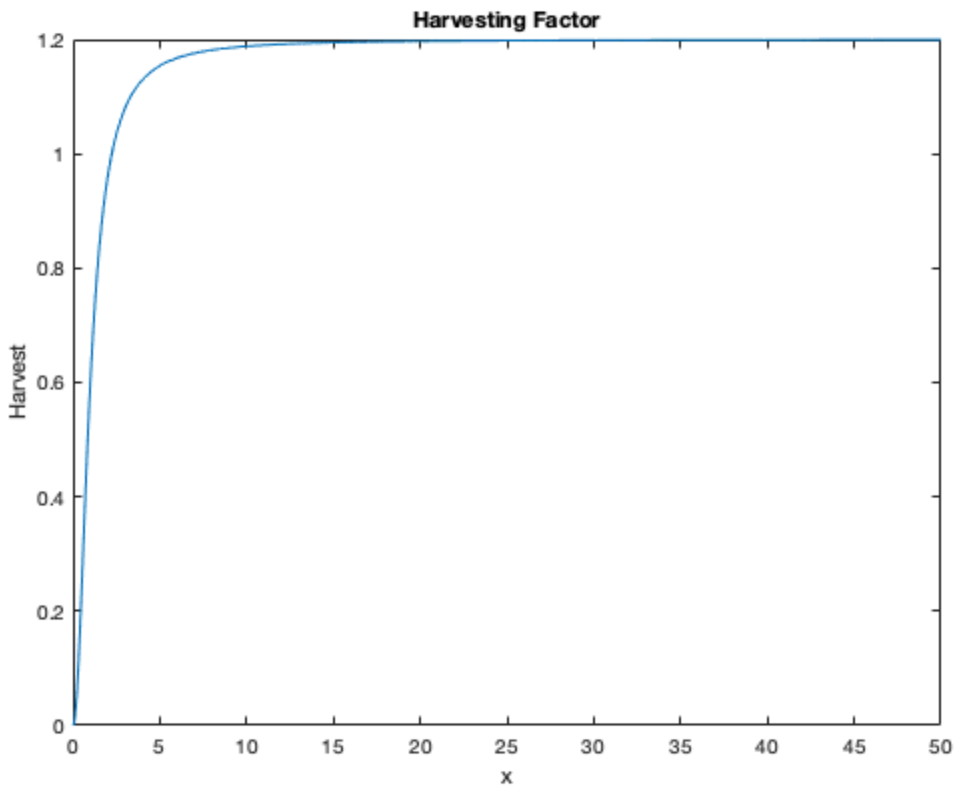
Error in Eulers method for Section 2, Question 3 b

```
e1 = abs(x(1:51)-x1);
e2 = abs((x(1:251)-x2));
e3 = abs((x-x3));
figure(2)
hold on
semilogy(t1,e1,t2,e2,t3,e3)
xlabel('time'), ylabel('Abs. Error'), title("Error in Population
 curves")
legend("h = 0.5","h = 0.1","h = 0.01")
hold off
```



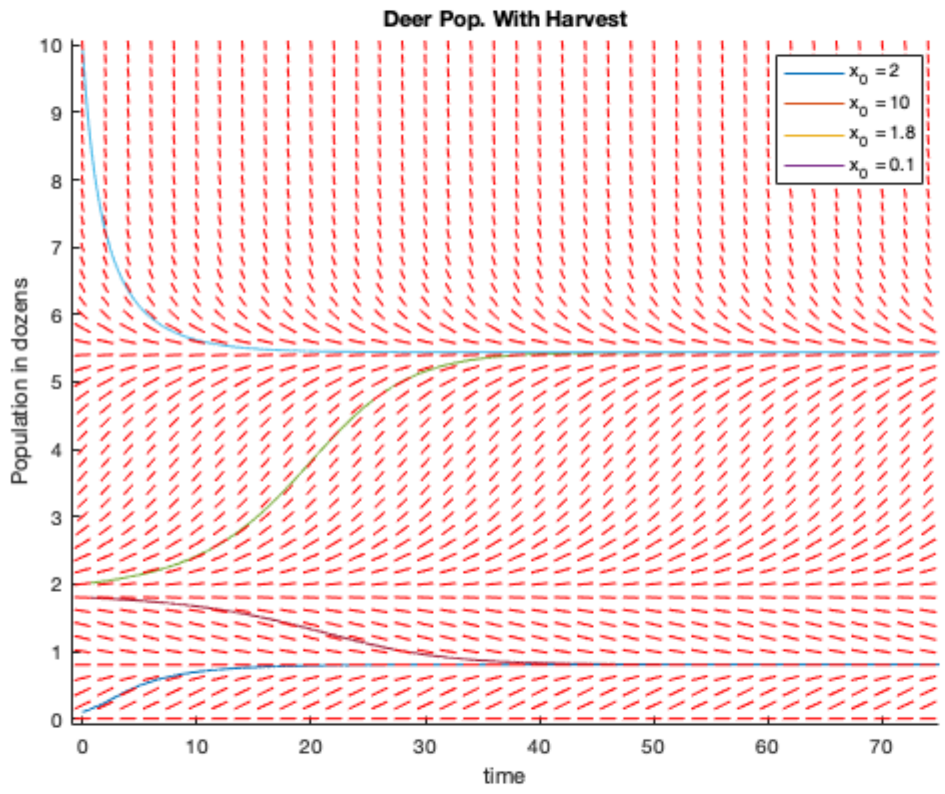
Plotting the Harvest Factor

```
p = 1.2;
q = 1;
xxx = 0:.01:50;
figure(3)
hx = p*xxx.^2./(q+xxx.^2);
plot(xxx,hx);
xlabel('x'), ylabel('Harvest'), title("Harvesting Factor")
```



Published with MATLAB® R2018b

```
% Eulers method with Harvest Factor, Section 2, Question 6
r = .65;
L = 8.1;
p = 1.2;
q = 1;
h = @(x) p*x.^2./(q+x.^2);
x_prime = @(t,x) r*x.*(1-x/L)-h(x);
h1 = .01;
t1 = 0:h1:75;
x1 = zeros(1,length(t1));
x1(1) = 2;
x2 = zeros(1,length(t1));
x2(1) = 10;
x3 = zeros(1,length(t1));
x3(1) = 1.8;
x4 = zeros(1,length(t1));
x4(1) = 0.1;
for n = 1:(length(t1)-1)
    x1(n+1) = x1(n) + h1*x_prime(t1(n),x1(n));
end
for n = 1:(length(t1)-1)
    x2(n+1) = x2(n) + h1*x_prime(t1(n),x2(n));
end
for n = 1:(length(t1)-1)
    x3(n+1) = x3(n) + h1*x_prime(t1(n),x3(n));
end
for n = 1:(length(t1)-1)
    x4(n+1) = x4(n) + h1*x_prime(t1(n),x4(n));
end
% dirfield.m
%{
function dirfield(f,tval,yval,plot_title)
[tm,ym]=meshgrid(tval,yval);
dt = tval(2) - tval(1);
dy = yval(2) - yval(1);
fv = vectorize(f);
if isa(f,'function_handle')
  fv = eval(fv);
end
yp=feval(fv,tm,ym);
s = 1./max(1/dt,abs(yp)./dy)*0.35;
h = ishold;
quiver(tval,yval,s,s.*yp,0,'.r'); hold on;
quiver(tval,yval,-s,-s.*yp,0,'.r');
if h
  hold on
else
  hold off
end
axis([tval(1)-dt/2,tval(end)+dt/2,yval(1)-dy/2,yval(end)+dy/2])

title(plot_title);
xlabel('t values');
ylabel('y values');
%}
f = @(t,x) (0.65*(1-(x./8.1)).*x)-(1.2*x.^2./(1+x.^2));
figure(1)
% Plot of direction field and solutions
hold on
plot(t1,x1,'LineWidth', 1)
plot(t1,x2,'LineWidth', 1)
plot(t1,x3,'LineWidth', 1)
plot(t1,x4,'LineWidth', 1)
% Function call for dirfield.m
dirfield(f,0:2:75, 0:0.2:10, 'Deer Pop. With Harvest')
xlabel('time'), ylabel('Population in dozens')
legend('x_0 = 2', 'x_0 = 10', 'x_0 = 1.8', 'x_0 = 0.1')
hold off
```



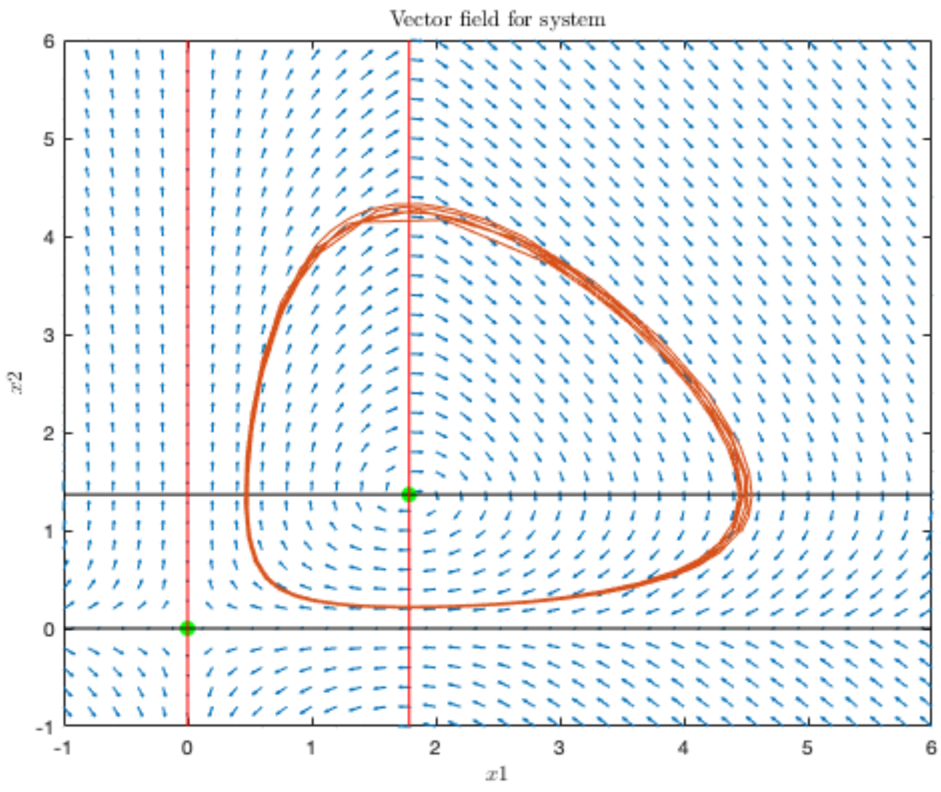
Published with MATLAB® R2018b

```
close all; clear all;
% The following code is from flow.m
```

Generating vector field for Lotka-Volterra System of equations

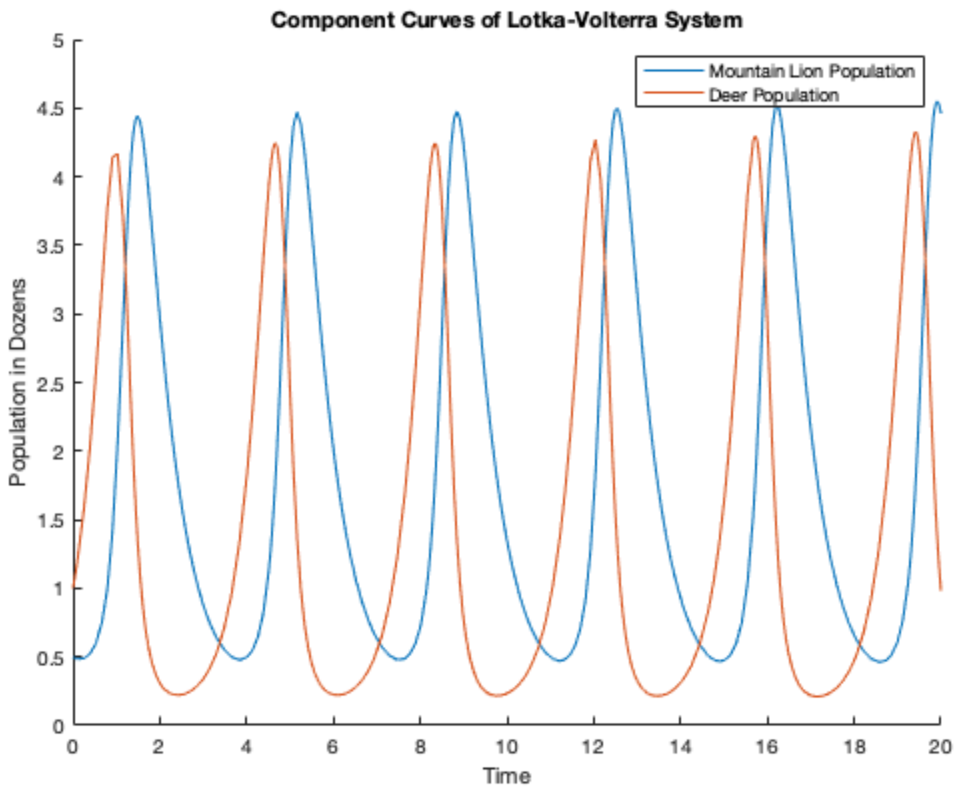
```
x1min = -1; x1max = 6; x2min = -1; x2max = 6;
x1step = .2; x2step = .2;
[x1, x2] = meshgrid(x1min:x1step:x1max, x2min:x2step:x2max);
a = 1.5; b = 1.1; c = 2.5; d = 1.4;
dx1 = -a*x1+b*x1.*x2;
dx2 =  c*x2-d*x1.*x2;
dx1 = dx1./sqrt(dx1.^2 + dx2.^2);
dx2 = dx2./sqrt(dx1.^2 + dx2.^2);
quiver(x1, x2, dx1,dx2,'AutoScaleFactor',0.5)
axis([x1min x1max x2min x2max])

% Calling project_system_3_1_5.m for ode45 to find solutions with
 initial
% conditions x1=0.5 and x2=1
%{
function vprime = project_system_3_1_5(t, v)
    x1 = v(1);
    x2 = v(2);
    a = 1.5;
    b = 1.1;
    c = 2.5;
    d = 1.4;

    vprime = zeros(2, 1);

    vprime(1) = -a*x1+(b*x1.*x2);
    vprime(2) = c*x2-(d*x1.*x2);
end
%}
[t_out, v_out] = ode45(@project_system_3_1_5, [0,20], [0.5,1]);
figure(1)
hold on
    xlabel('$x1$','Interpreter','latex')
    ylabel('$x2$','Interpreter','latex')
    title('Vector field for system','Interpreter','latex')

plot(c/d,a/b, 'g.', 'MarkerSize', 20);
plot(0,0, 'g.', 'MarkerSize', 20);

% Adding nullclines and equilibirum points
e = c/d;
n1 = refline([0 a/b]); n2 = refline([0 0]);
n1.Color = 'k'; n2.Color = 'k';
n1.LineWidth = 1; n2.LineWidth = 1;
n3 = line([0 0], ylim); n4 = line([e,e],[-1,6]);

n3.LineWidth = 1; n4.LineWidth = 1;
```

```
n3.Color = 'r'; n4.Color = 'r';
plot(v_out(:,1), v_out(:,2))
hold off
```



Component Curves of Lotka-Volterra System

```
figure(2)
hold on
plot(t_out, v_out(:,1))
plot(t_out, v_out(:,2))
legend( 'Mountain Lion Population','Deer Population')
title('Component Curves of Lotka-Volterra System')
xlabel('Time')
ylabel('Population in Dozens')
hold off
```



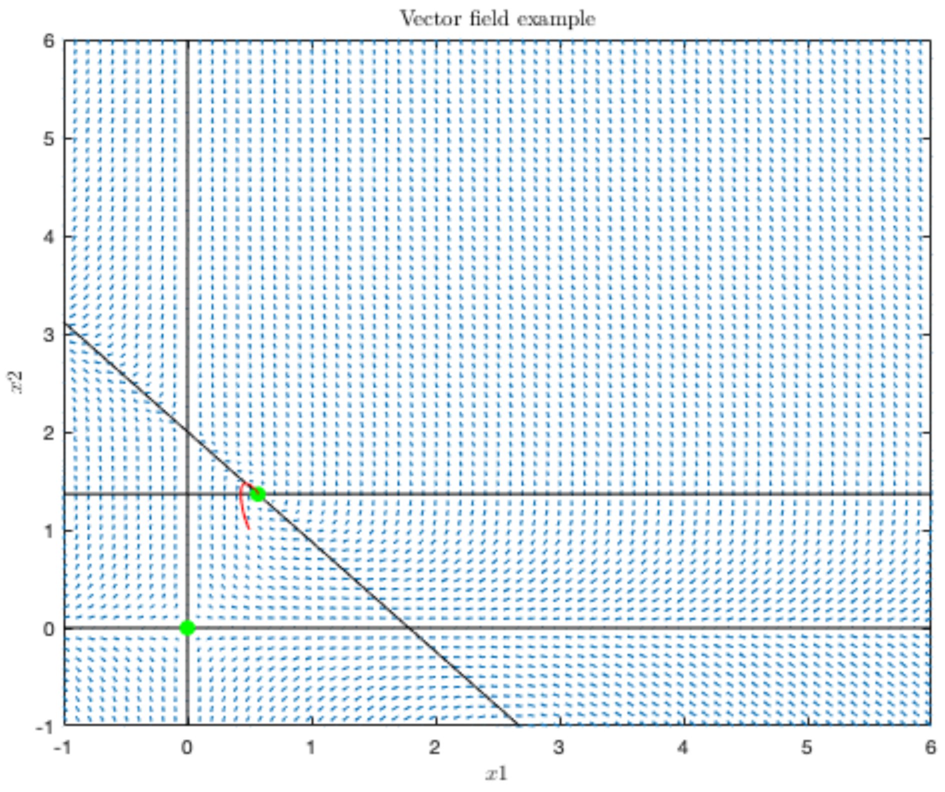
Published with MATLAB® R2018b

```
close all; clear all;
% Following code is from flow.m
```

Generating Vector field for Logistic Predator-Prey equation

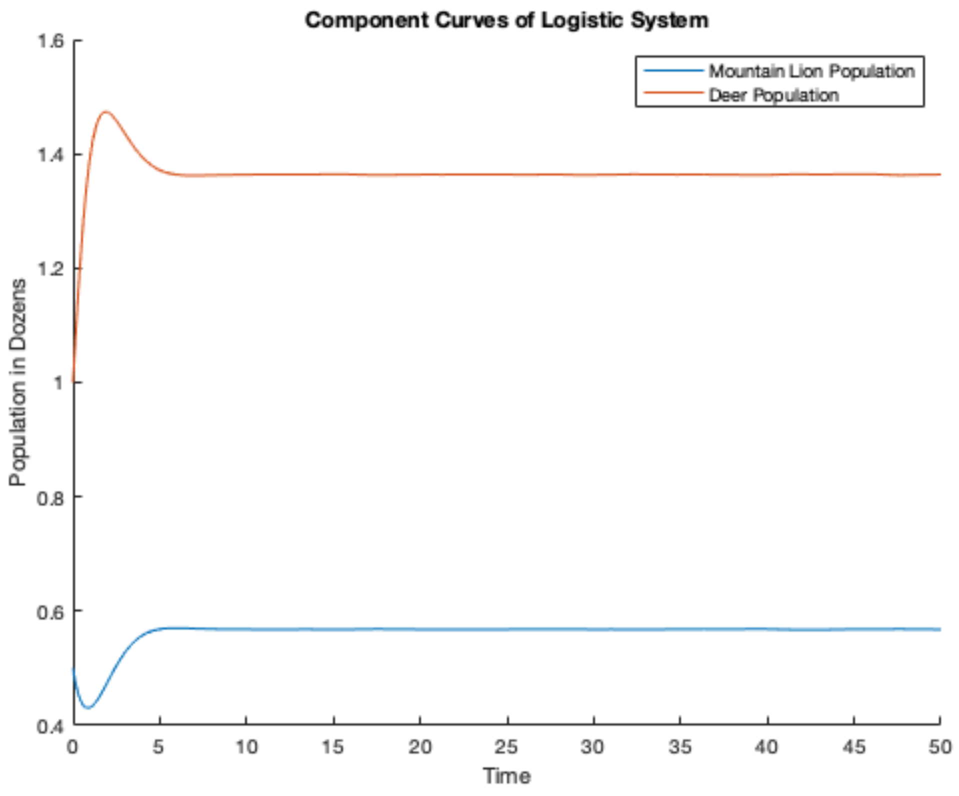
```
x1min = -1; x1max = 6; x2min = -1; x2max = 6;
x1step = 0.1; x2step = 0.1;
[x1, x2] = meshgrid(x1min:x1step:x1max, x2min:x2step:x2max);
a = 1.5;
b = 1.1;
c = 2.5;
d = 1.4;
k = 0.5;
dx1 = -a*x1+(b*x1.*x2);
dx2 = c*(1-k*x2).*x2-(d*x1.*x2);
dx1 = dx1./sqrt(dx1.^2 + dx2.^2);
dx2 = dx2./sqrt(dx1.^2 + dx2.^2);
quiver(x1, x2, dx1,dx2,'AutoScaleFactor',0.5)
axis([x1min x1max x2min x2max])
% Calling project_system_3_2_2.m for ode45 to find solutions with
 initial
% conditions x1 = 0.5, x2 = 1
%{
function vprime = project_system_3_2_2(t, v)
    x1 = v(1);
    x2 = v(2);
    a = 1.5;
    b = 1.1;
    c = 2.5;
    d = 1.4;
    k = 0.5;

    vprime = zeros(2, 1);

    vprime(1) = -a*x1+(b*x1.*x2);
    vprime(2) = c*(1-k*x2).*x2-(d*x1.*x2);
end
%}
[t_out, v_out] = ode45(@project_system_3_2_2, [0,50], [0.5,1]);

figure(1)
hold on
xlabel('$x1$','Interpreter','latex')
ylabel('$x2$','Interpreter','latex')
title('Vector field example','Interpreter','latex')
x2_=@(x) (1/k)-(d/(c*k))*x;
x=-1:0.1:6;
n0=plot(x, x2_(x));  n0.LineWidth = 1; n0.Color ='k';

% Adding nullclines and equilibrium solutions
f=a/b;
n3 = line([0 0], ylim); n1 = refline([0 a/b]);
```

```
n2 = refline([0 0]);
n1.LineWidth = 1; n2.LineWidth = 1; n3.LineWidth = 1;
n1.Color = 'k'; n2.Color = 'k'; n3.Color = 'k';
plot((c*b-a*c*k)/(b*d),a/b, 'g.', 'MarkerSize', 20);
plot(0,0, 'g.', 'MarkerSize', 20);
plot(v_out(:,1), v_out(:,2),'LineWidth',1,"Color",'r')
hold off
```



Component Curves of Logistic Predator-Prey equation

```
figure(2)
hold on
plot(t_out, v_out(:,1))
plot(t_out, v_out(:,2))
legend('Mountain Lion Population','Deer Population')
title('Component Curves of Logistic System')
xlabel('Time')
ylabel('Population in Dozens')
hold off
```



Published with MATLAB® R2018b

Solving for $x(t)$ using separation of variables D1
(derivation 1)

$$\frac{dx}{dt} = r\left(1-\frac{x}{L}\right)x$$

$$\cancel{\frac{dx}{dt} = xr - \frac{x^2 r}{L}}$$

$$\frac{dx}{dt} = r\left(\frac{xL-x^2}{L}\right) \Rightarrow \frac{L}{x(L-x)}dx = r\,dt$$

$$\int \frac{L}{x(L-x)}dx = \int r\,dt$$

$$\frac{A}{x} + \frac{B}{L-x} = \frac{L}{x(L-x)}$$

$$A(L-x) + Bx = L$$

$$AL - Ax + Bx = L$$

$$B - A = 0 \qquad AL = L$$

$$B = A \qquad A = 1$$

$$\int \frac{1}{x}dx + \int \frac{1}{L-x}dx = rt + C$$

$$\ln|x| + (-\ln|L-x|) = rt + C$$

$$e^{\ln|x| - \ln|L-x|} = e^{rt+C}$$

$$e^{\ln|x|}e^{-\ln|L-x|} = Ce^{rt}$$

$$\frac{x}{L-x} = Ce^{rt} \qquad x(0) = x_0$$

$$\longrightarrow \frac{x_0}{L-x_0} = C$$

$$\frac{x}{L-x} = \frac{x_0 e^{rt}}{L-x_0}$$

$$x = \frac{x_0 e^{rt}}{L-x_0}(L-x)$$

$$x = \frac{x_0 e^{rt} L}{L-x_0} - \frac{x_0 e^{rt} x}{L-x_0}$$

$$x + \frac{x_0 e^{rt} x}{L-x_0} = \frac{x_0 e^{rt} L}{L-x_0} \Rightarrow x\left(1 + \frac{x_0 e^{rt}}{L-x_0}\right) = \frac{x_0 e^{rt} L}{L-x_0}$$

$$x = \frac{x_0 e^{rt} L}{\cancel{L-x_0}} \cdot \frac{\cancel{L-x_0}}{L-x_0+x_0 e^{rt}}$$

$$\boxed{x(t) = \frac{x_0 e^{rt} L}{L-x_0+x_0 e^{rt}}}$$

Lotka-Volterra nullclines and equilibriums

D2

(derivation 2)

$\frac{dx_1}{dt} = -ax_1 + bx_1x_2$ $\frac{dx_2}{dt} = cx_2 - dx_1x_2$

set both equal to zero

$0 = -ax_1 + bx_1x_2$ $0 = x_2(c - dx_1)$

$x_1(-a + bx_2) = 0$ $x_2 = 0$ $c = dx_1$

$x_1 = 0$ $a = bx_2$ $x_1 = \frac{c}{d}$

$x_2 = \frac{a}{b}$

nullclines: $x_1 = 0,\ x_2 = \frac{a}{b},\ x_2 = 0,\ x_1 = \frac{c}{d}$

equilibrium points: $\left(\frac{c}{d}, \frac{a}{b}\right), (0,0)$

Logistic Predator-Prey nullclines and equilibriums

D3

(derivation 3)

$\frac{dx_1}{dt} = -ax_1 + bx_1x_2$ $\frac{dx_2}{dt} = c(1-kx_2)x_2 - dx_1x_2$

set both equal to zero

$0 = -ax_1 + bx_1x_2$ $0 = c(1-kx_2)x_2 - dx_1x_2$

$0 = x_2c - ckx_2^2 - dx_1x_2$

$x_1 = 0$ $x_2 = \frac{a}{b}$ $0 = x_2(c - ckx_2 - dx_1)$

$x_2 = 0$ $0 = c - ckx_2 - dx_1$

$x_2 = \frac{1}{k} - \frac{d}{ck}x_1$

nullclines: $x_1 = 0,\ x_2 = \frac{a}{b}$

$x_2 = 0,\ x_1 = \frac{1}{k} - \frac{d}{ck}x_1$

$\frac{a}{b} = \frac{1}{k} - \frac{dx_1}{ck} \rightarrow \frac{a}{b} - \frac{1}{k} = -\frac{dx_1}{ck} \rightarrow \frac{dx_1}{ck} = \frac{1}{k} - \frac{a}{b}$

$x_1 = \frac{\left(\frac{1}{k} - \frac{a}{b}\right)ck}{d} = \frac{cb - ack}{db}$

equilibrium points: $(0,0), \left(\frac{cb-ack}{db}, \frac{a}{b}\right)$