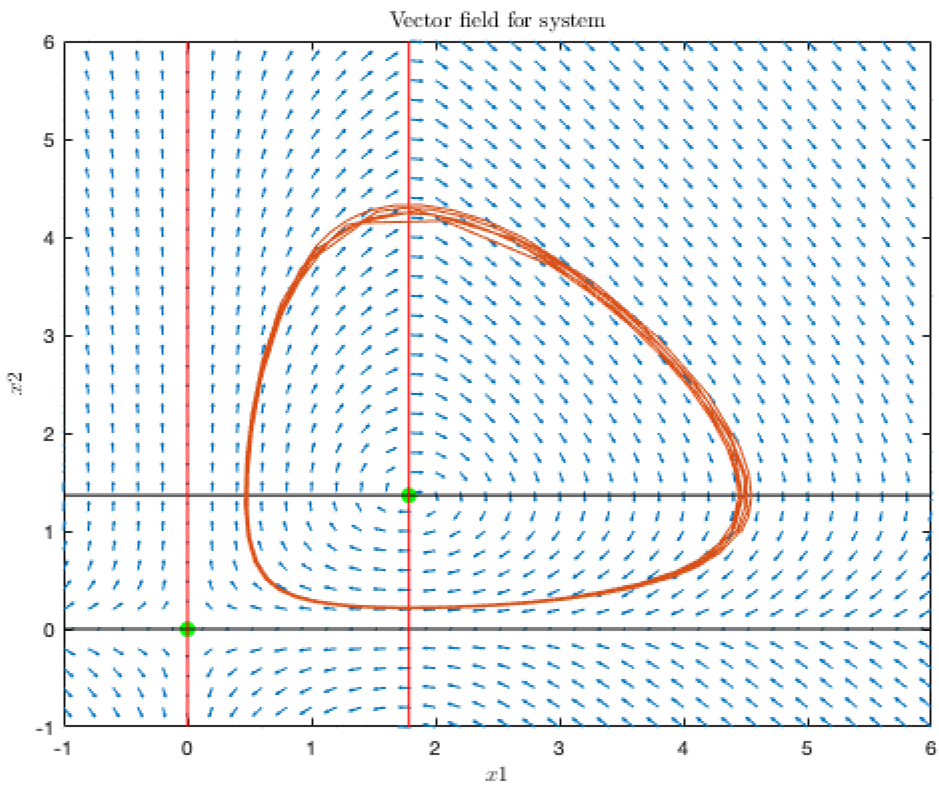
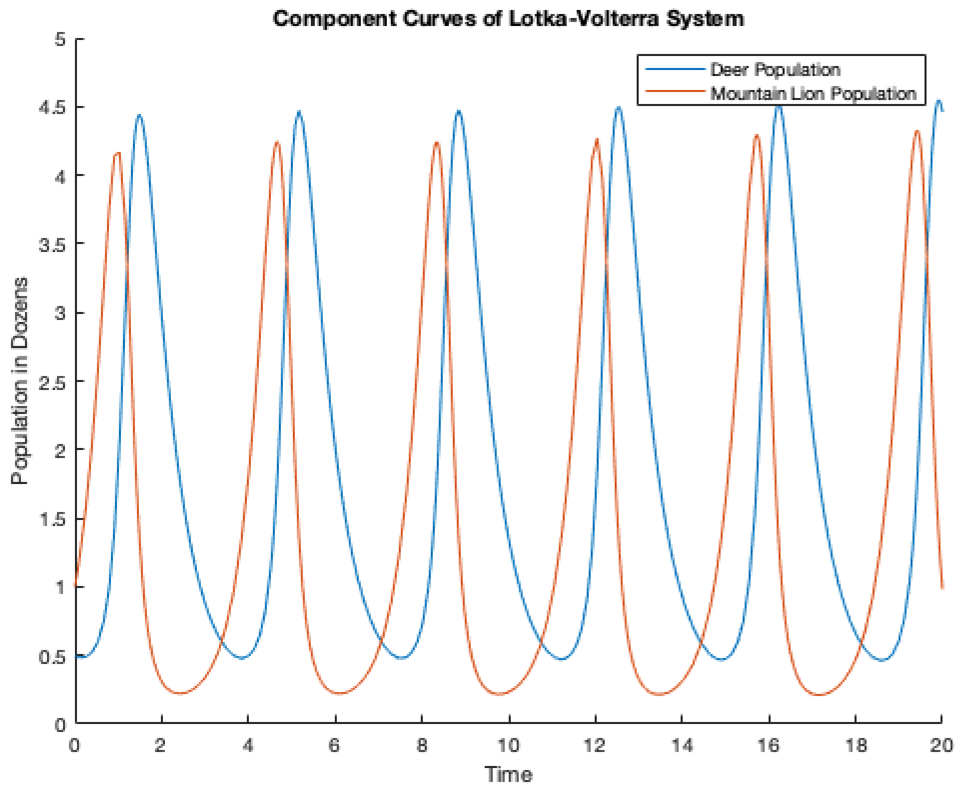
```
close all; clear all;
% This Matlab code generates a vector field for the system of ODEs
% dx1/dt = f(x1,x2), dx2/dt = g(x1,x2)

% This code currently will find the vector field for the EXAMPLE
 problem
%            dx1 = -a*x1+b*x1.*x2;
%            dx2 =  c*x2-d*x1.*x2;
%---------------------------------------------------------------------------
%         THESE ARE NOT THE PROBLEMS YOU ARE SOLVING FOR PROJECT 1!
% (To have this code generate the vector fields for the Project 1
 systems
% of equations, make any necessary adjustments in the sections of code
% labeled with "Step i" where i = 1, 2, 3, 4, or 5)
%---------------------------------------------------------------------------

% Step 1: Set the axis limits so that you plot the vector field over
 the
%         intervals x1min < x1 < x1max, x2min < x2 < x2max
    x1min = -1; x1max = 6; x2min = -1; x2max = 6;

% Step 2: pick step sizes for x1 and x2;
    x1step = .2; x2step = .2;

% generate mesh for plotting
    [x1, x2] = meshgrid(x1min:x1step:x1max, x2min:x2step:x2max);

% Step 3: define all needed parameter values
    a = 1.5; b = 1.1; c = 2.5; d = 1.4;

% Step 4: define the system of equations you are using
    dx1 = -a*x1+b*x1.*x2;
    dx2 =  c*x2-d*x1.*x2;

% normalize vectors (to help plotting)
    dx1 = dx1./sqrt(dx1.^2 + dx2.^2);
    dx2 = dx2./sqrt(dx1.^2 + dx2.^2);

% generate the vector field
    quiver(x1, x2, dx1,dx2,'AutoScaleFactor',0.5)

% specify the plotting axes
    axis([x1min x1max x2min x2max])

% Step 5: label the axes, include a title

[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);

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

% ANSWER TO QUESTION 1:
% 1st order, autonomous, linear

% ANSWER TO QUESTION 2:
% Nullclines: x1 = 0, x2 = a/b and x1 = c/d, x2 = 0
% Equilibrium Solutions: (0,0), (c/d, a/b)

% ANSWER TO QUESTION 3:
% Figure 1

% ANSWER TO QUESTION 4:
% (0,0) is semi stable, (c/d, a/b) is unstable

% ANSWER TO QUESTION 5 (along with figure 1)
% The solution does at we expected, because it will never reach an
% equilibrium solution as all equilibrium points are unstable

% ANSWER TO QUESTION 6 (along with figure 2)
figure(2)
hold on
plot(t_out, v_out(:,1))
plot(t_out, v_out(:,2))
legend('Deer Population', 'Mountain Lion Population')
title('Component Curves of Lotka-Volterra System')
xlabel('Time')
ylabel('Population in Dozens')
hold off

% The curves are out of phase, and this means that the interaction of
 the
% predator and prey depends on the population of each other. If the
 deer
% population is higher, it means the mountain lion population can rise
% because they have more food. As they kill off more deer, there is
 less
% food for them and their population goes down after the deer
 population
% decreases. Then, when the mountain lion population is low, the deer
```

```
% population can decrease because there are less predators to kill
 them,
% and the cycle repeats.
```





Published with MATLAB® R2018b