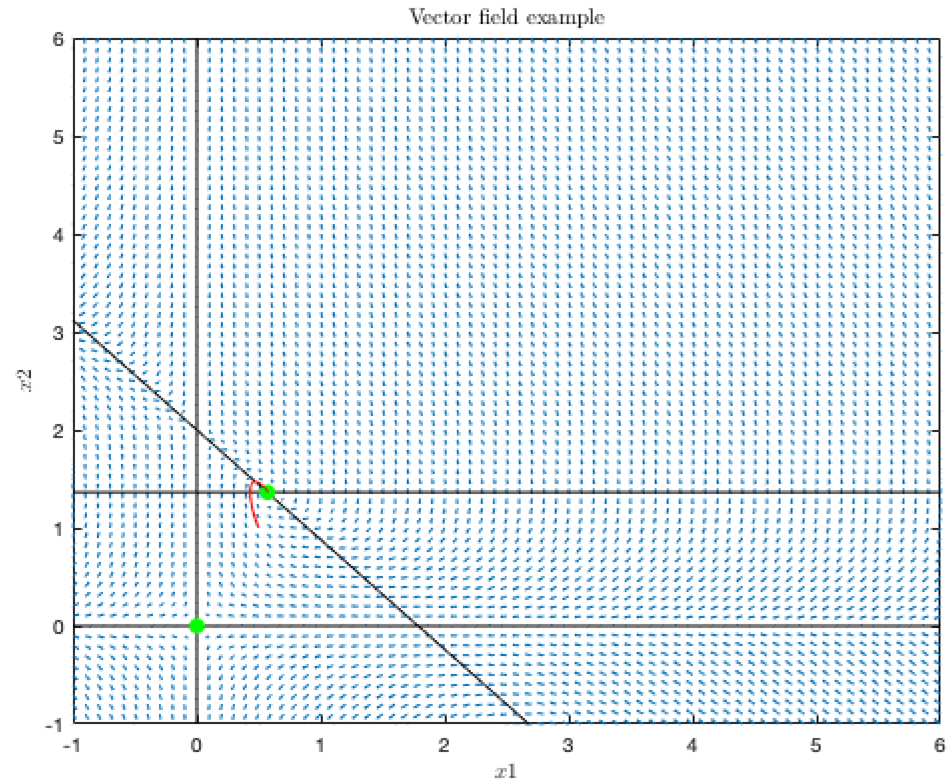
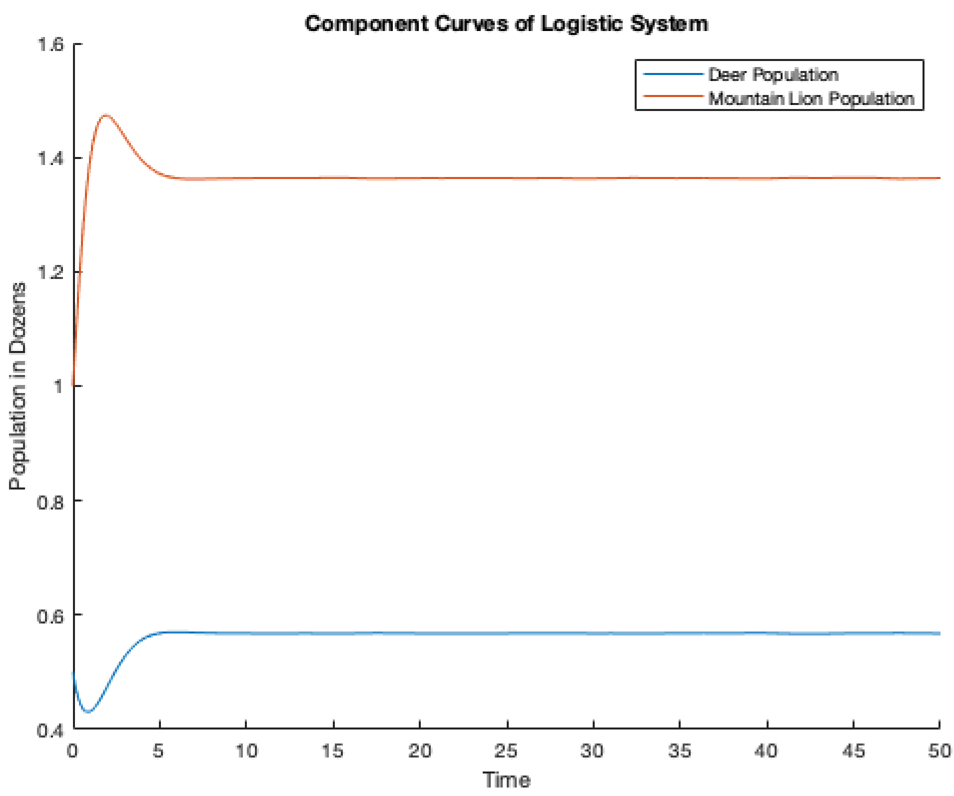
```
close all; clear all;
% This Matlab code generates a vector field for the system of ODEs
% dx1/dt = f(x1,x2), dx2/dt = g(x1,x2)

% This code currently will find the vector field for the EXAMPLE
 problem
%            dx1/dt = a*x2
%            dx2/dt = -x1
%---------------------------------------------------------------------------
%         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 = 0.1; x2step = 0.1;

% 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;
    k = 0.5;

% Step 4: define the system of equations you are using
    dx1 = -a*x1+(b*x1.*x2);
    dx2 = c*(1-k*x2).*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_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';

    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


% ANSWER QUESTION 1:
% Nullclines: x1 = 0, x2 = a/b and x2 = (1/k)-(d/c*k)*x1, x2 = 0
% Equilibruim Solutions: (0,0), ((c*b-a*c*k)/(b*d), a/b)

% ANSWER QUESTION 2:
% (0,0) is a semi-stable equilibrium, while ((c*b-a*c*k)/(b*d), a/b)
 is a
% stable equilibrium. This means that any solution will eventually end
 up
% at this equilibrium point, and that is where the population will
 remain
% stable

% ANSWER QUESTION 3 (along with figure 2) (need to discuss this)
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 Logistic System')
xlabel('Time')
ylabel('Population in Dozens')
hold off
```


Vector field example
x2
x1
6
5
4
3
2
1
0
-1
-1
0
1
2
3
4
5
6


Component Curves of Logistic System
Deer Population
Mountain Lion Population
Population in Dozens
Time
1.6
1.4
1.2
1
0.8
0.6
0.4
0
5
10
15
20
25
30
35
40
45
50

Published with MATLAB® R2018b