ECE3040

Final Project

Name: Austin Shammas

Due Date: 4/21/2017

The predator–prey problem refers to an ecological system in which we have two species, one of which feeds on the other. This type of system has been studied for decades and is known to exhibit interesting dynamics. Let $x_1(t)$ represent the number of hares (prey) and let $x_2(t)$ represent the number of lynxes (predator). The independent variable t represent time and is measured in years. The dynamics of the system are modeled as the nonlinear system of differential equations:

$$\frac{dx_1}{dt} = f_1(x_1, x_2) = rx_1\left(1 - \frac{x_1}{k}\right) - \frac{ax_1x_2}{c + x_1}$$

$$\frac{dx_2}{dt} = f_2(x_1, x_2) = b\frac{ax_1x_2}{c + x_1} - dx_2$$

In the first equation, r represents the growth rate of the hares, k represents the maximum population of the hares (in the absence of lynxes), a represents the interaction term that describes how the hares are diminished as a function of the lynx population and c controls the prey consumption rate for low hare population. In the second equation, b represents the growth coefficient of the lynxes and d represents the mortality rate of the lynxes. This model makes many simplification assumptions, such as the fact that hares decrease in number only through predation by lynxes. But, the model is often sufficient to answer basic questions about the system.

In the following, you are to assume the parameter values: $a = 3.2, b = 0.6, c = 50, d = 0.56, k = 125$ and $r = 1.6$.

1. Find the equilibrium (fixed) point(s) of the system. These are the (x_1^*, x_2^*) points at which the population remains constant (i.e., $\frac{dx_1}{dt} = 0$ and $\frac{dx_2}{dt} = 0$). Start by plotting the resulting two (implicit) nonlinear equations. The plot should give you an estimate of the locations of the equilibrium point(s), which can be used as initial search points to find numerical solutions (accurate to five digits). You may use Matlab's fsolve. (Note that only $x_1 \geq 0$ and $x_2 \geq 0$ are relevant here; positive or zero populations.)

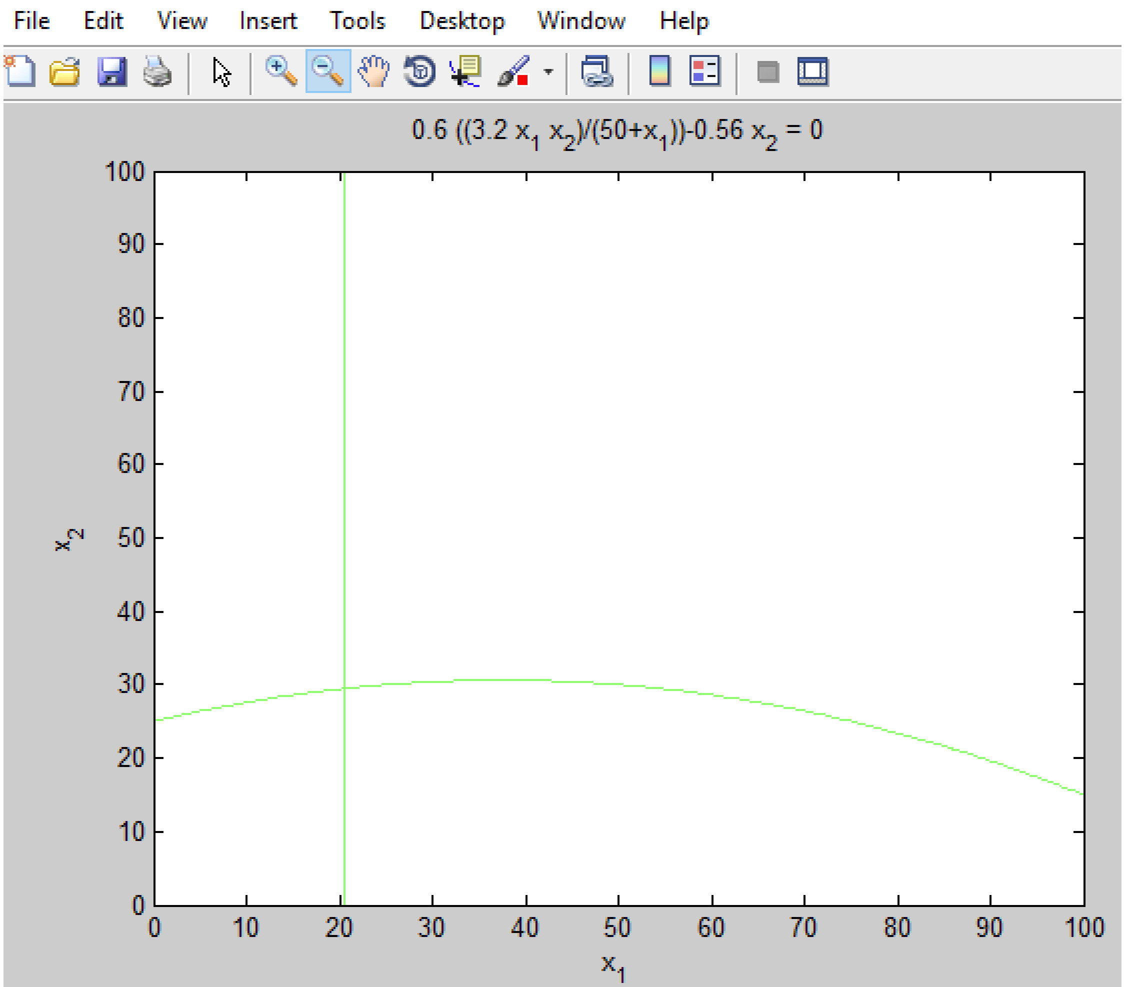
```
>> dx1dt

dx1dt =

    @(x1,x2)1.6*x1*(1-x1/125)-((3.2*x1*x2)/(50+x1))


>> dx2dt

dx2dt =

    @(x1,x2)0.6*((3.2*x1*x2)/(50+x1))-0.56*x2
```

After plotting the functions many times using ezplot, I kept getting a horizontal line and a vertical line. I then figured out that the default parameters for ezplot is -2pi to 2pi:

- If you do not specify a plot range, `ezplot` uses the interval $[-2\pi\ 2\pi]$ as a starting point. Then it can choose to display a part of the plot over a subinterval of $[-2\pi\ 2\pi]$ where the plot has significant variation. Also, when selecting the plotting range, `ezplot` omits extreme values associated with singularities.

So I set my parameters to 0 100 (population cannot be negative so I started at 0)

```
>> ezplot(dx1dt, [0 100])
Warning: Function failed to e
vectorizing the function may
speed up its evaluation and a
array elements.
> In specgraph\private\ezplot
  In ezplot>ezimplicit at 258
  In ezplot at 155
>> hold on
>> ezplot(dx2dt, [0 100])
```



We can visually see there is an equilibrium point at roughly (20.5, 29). We also know (see slide 15: lecture 22) that there is an equilibrium point at (0,0).

```
function F=shammas_fsolve(x)
F(1)= 1.6*x(1)*(1-x(1)/125)-((3.2*x(1)*x(2))/(50+x(1)));
F(2)= 0.6*((3.2*x(1)*x(2))/(50+x(1)))-0.56*x(2);

end
```

```
>> fun=@shammas_fsolve;
```

```
>> x0=[0,0]

x0 =

     0     0
```

```
>> x=fsolve(fun,x0)

Equation solved at ini

fsolve completed becau
is near zero as measur
the problem appears re

<stopping criteria det

x =

     0     0
```

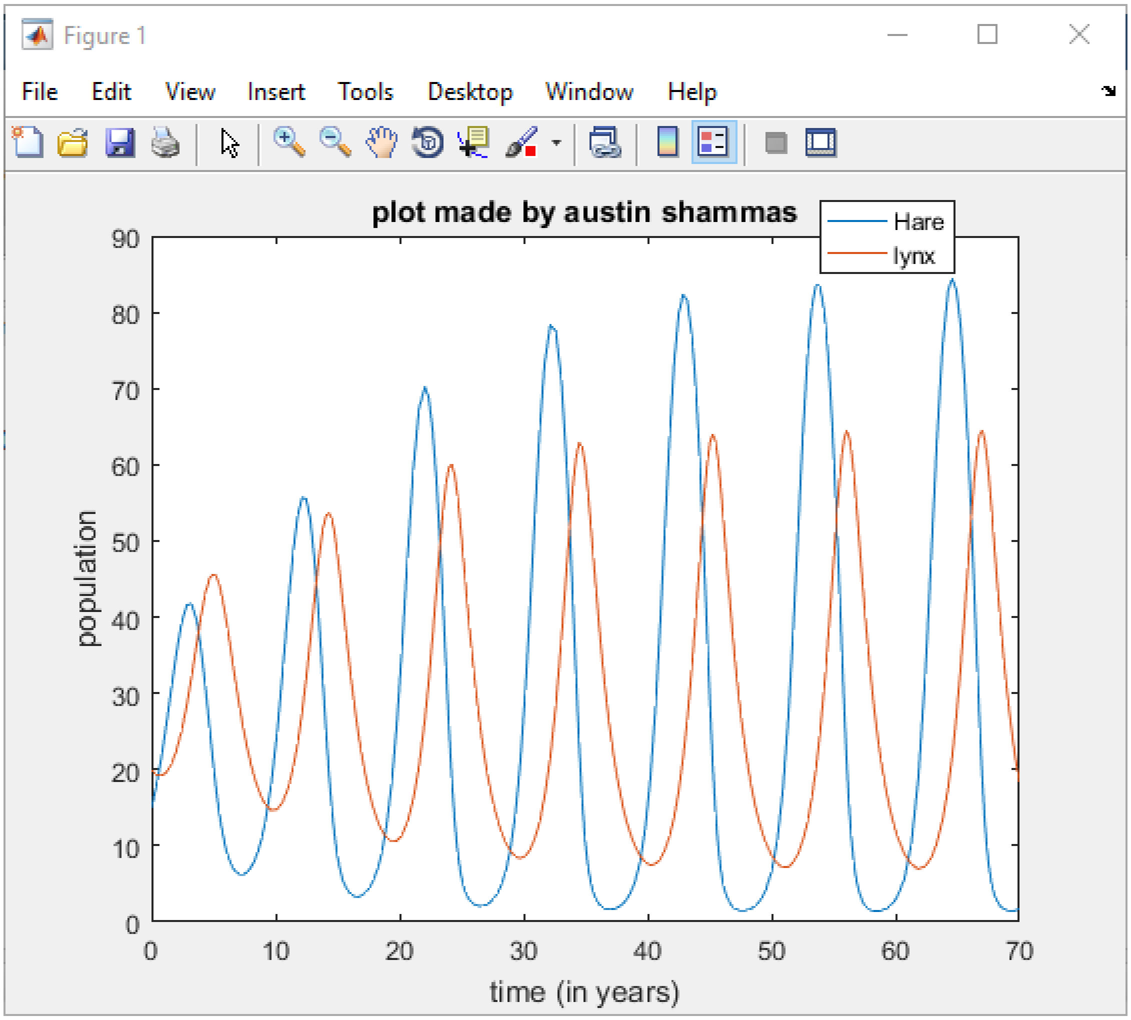
```
>> x0=[20.5,29];
>> x = fsolve(fun, x0)

Equation solved.

fsolve completed because th
as measured by the default
the problem appears regular

<stopping criteria details>

x =

   20.5882   29.4810
```

So, we can conclude that our numerical solutions are (x1*)=20.5882 and (x2*)=29.4810.

2. Employ ode45 to solve the predator-prey system. Assume the initial populations $x_1(0) = 15$ and $x_2(0) = 20$, and solve for $t \in [0\ \ 70]$. Plot the populations $x_1(t)$ and $x_2(t)$ on the same set of axis.

```
function [yp] = shammas_predpray(t,x)
yp=[1.6*x(1)*(1-x(1)/125)-((3.2*x(1)*x(2))/(50+x(1)));
0.6*((3.2*x(1)*x(2))/(50+x(1)))-0.56*x(2)];
end
```

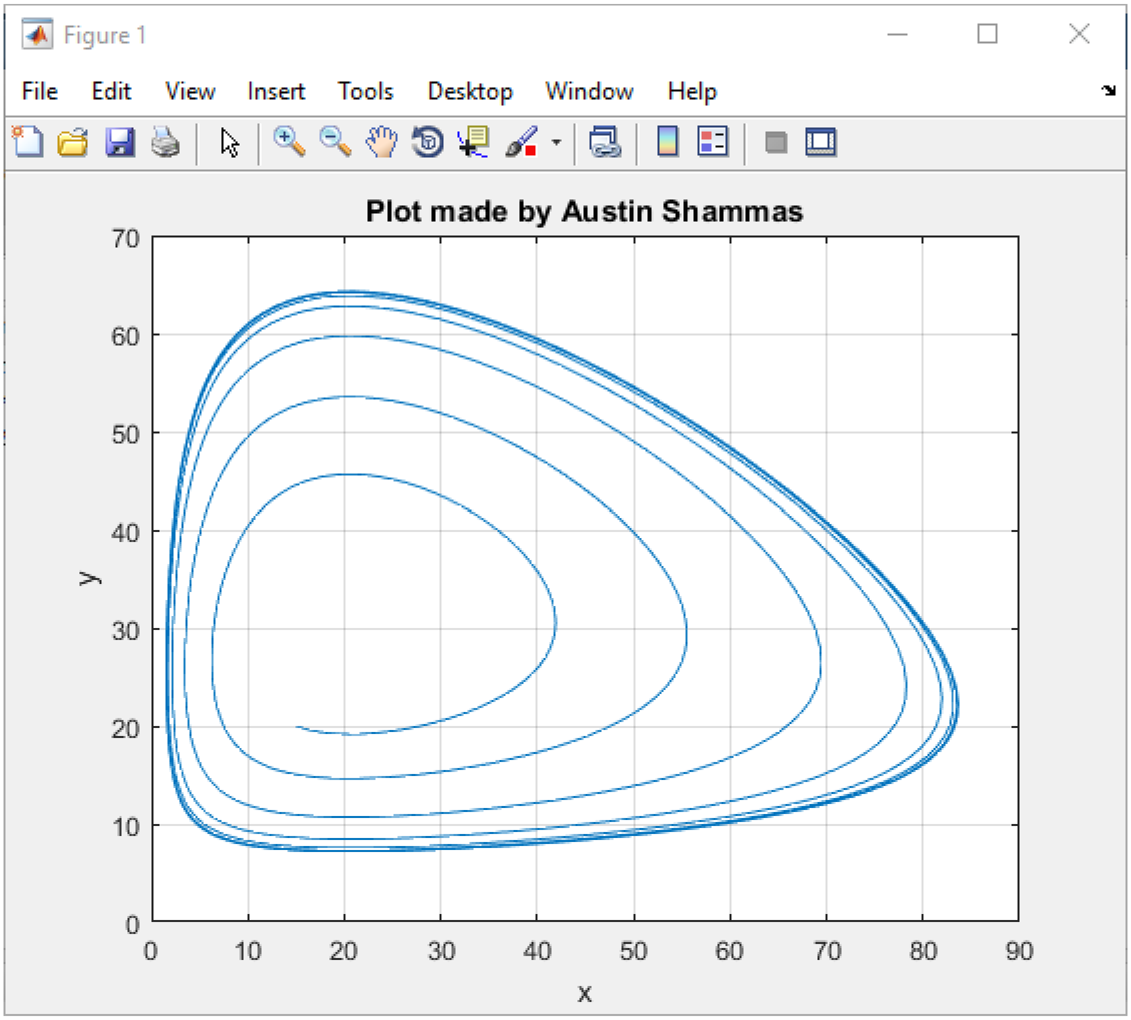
```
>> [t,x]=ode45(@shammas_predpray, [0 70], [15 20]);
>> plot(t,x)
>>
>> title 'plot made by austin shammas'
>> xlabel 'time'
>> ylabel 'population'
>> xlabel 'time (in years)'
```



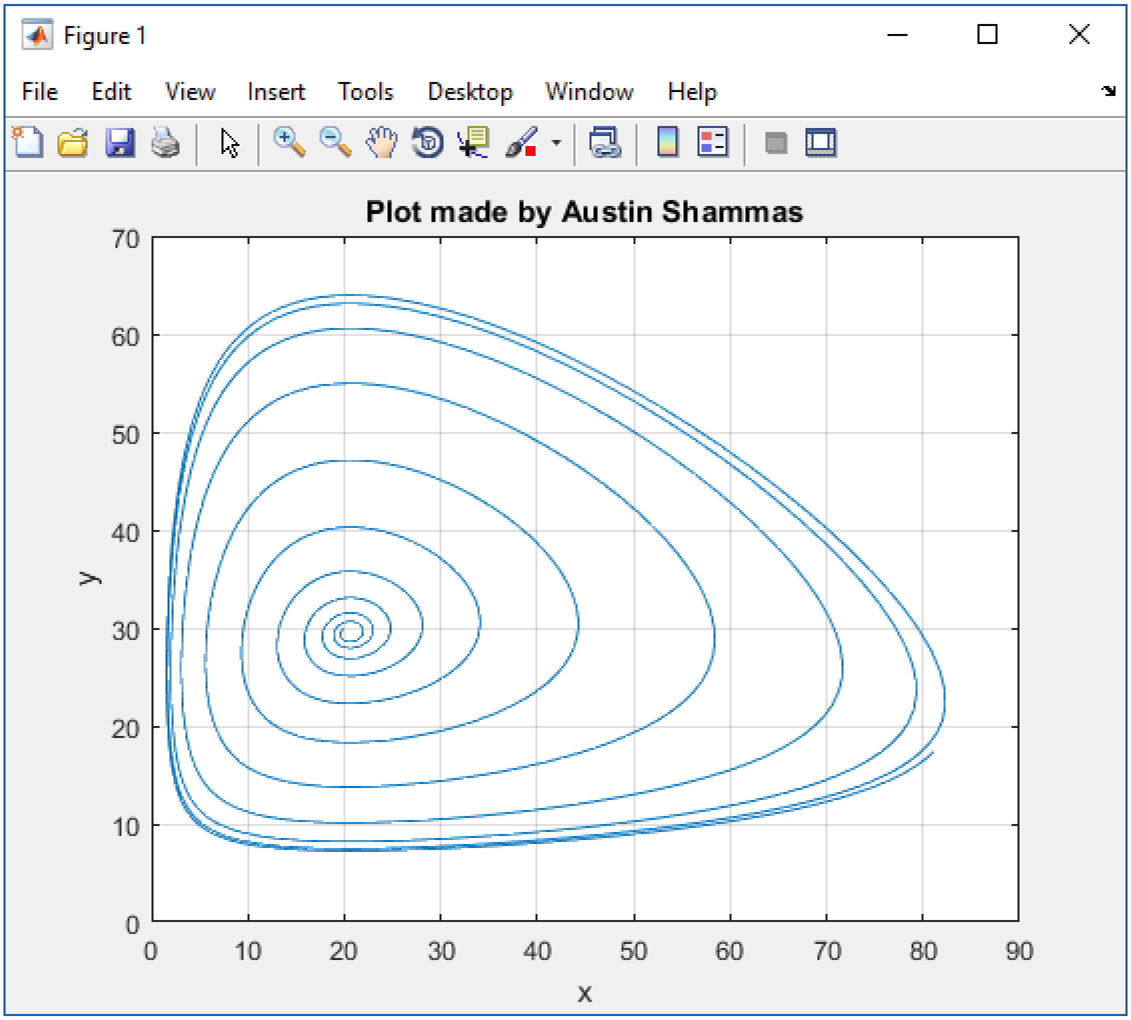
3. Generate phase-plane trajectories [plot of $x_2(t)$ vs $x_1(t)$] for each of the following initial populations: (15, 20), (20, 30) and (90, 70). Assume $t \in [0\ \ 100]$.

```
function [yp] = shammas_predpray(t,x)
yp=[1.6*x(1)*(1-x(1)/125)-((3.2*x(1)*x(2))/(50+x(1)));
0.6*((3.2*x(1)*x(2))/(50+x(1)))-0.56*x(2)];
end
```

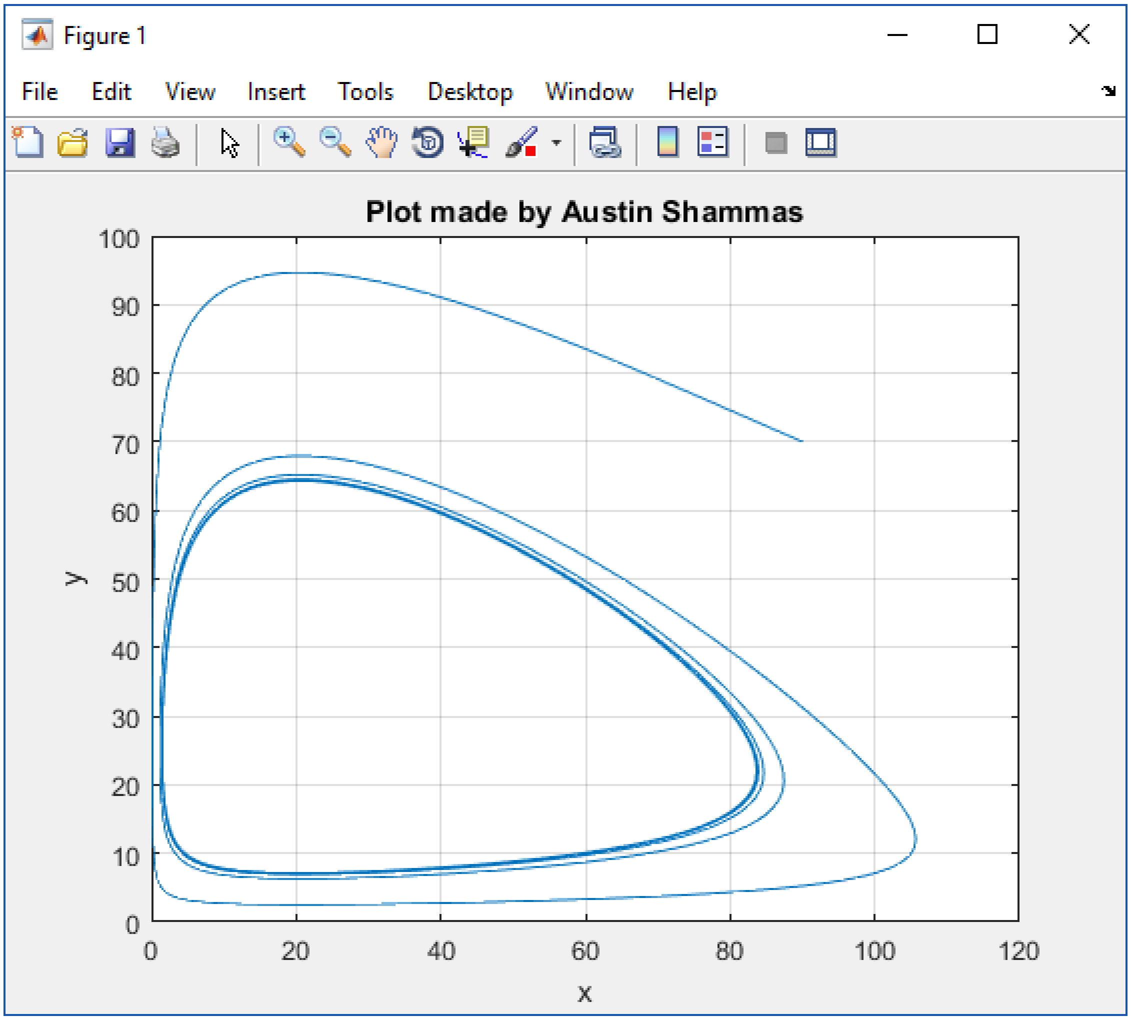
```
>> options=odeset('RelTol',1e-6);

>> [t,x]=ode45(@shammas_predpray, [0 100], [15 20], options);
>> plot(x(:,1),x(:,2))
>> grid on
>> title 'Plot made by Austin Shammas'; xlabel 'x'; ylabel 'y'
```




```
>> [t,x]=ode45(@shammas_predpray, [0 100], [20 30], options);
>> plot(x(:,1),x(:,2))
>> grid on
>> title 'Plot made by Austin Shammas'; xlabel 'x'; ylabel 'y'
```

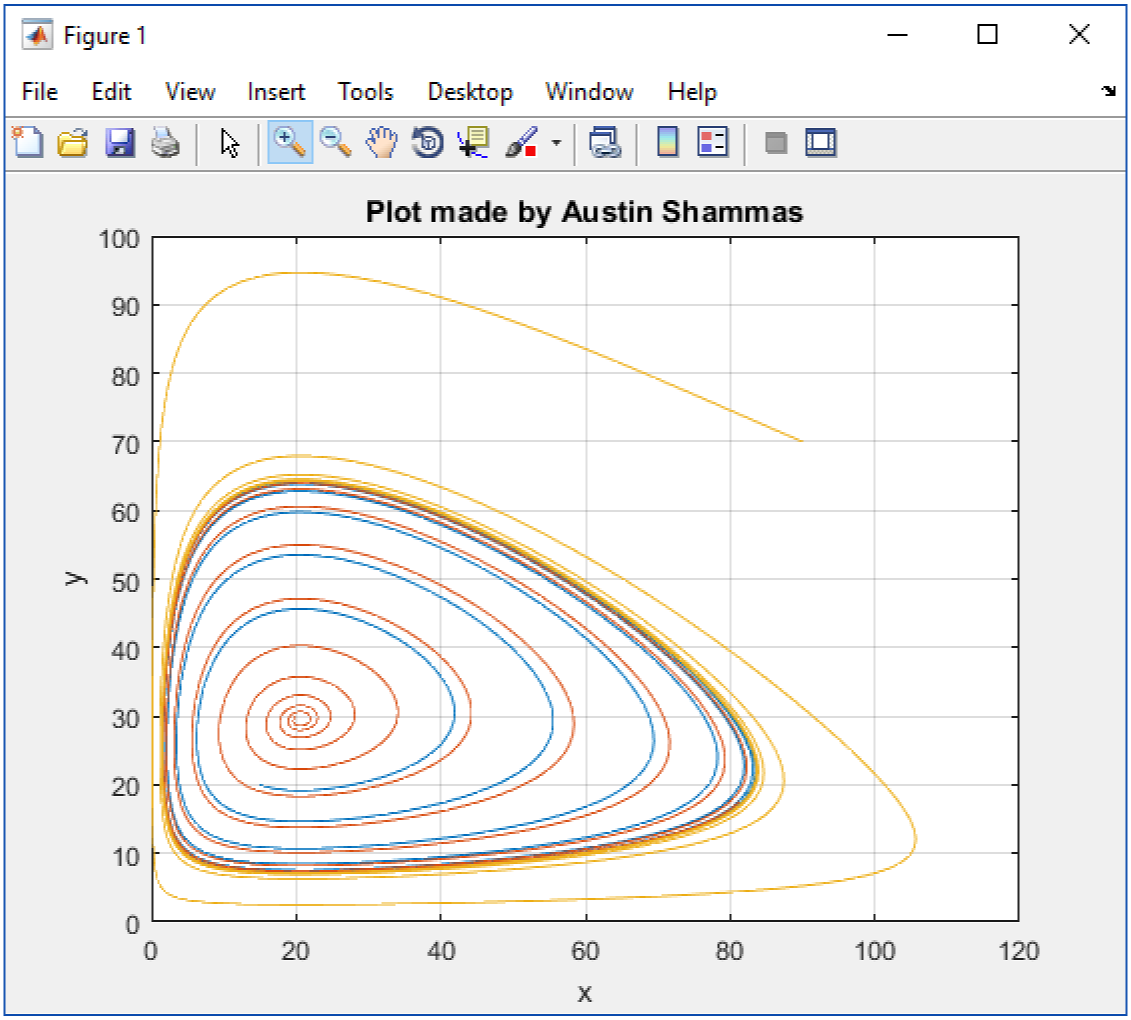



```
>> [t,x]=ode45(@shammas_predpray, [0 100], [90 70], options);
>> plot(x(:,1),x(:,2))
>> grid on
>> title 'Plot made by Austin Shammas'; xlabel 'x'; ylabel 'y'
```



Putting them all together:

```
>> [t,x]=ode45(@shammas_predpray, [0 100], [15 20], options);
>> plot(x(:,1),x(:,2))
>> grid on
>> title 'Plot made by Austin Shammas'; xlabel 'x'; ylabel 'y'
>> hold on
>> [t,x]=ode45(@shammas_predpray, [0 100], [20 30], options);
>> plot(x(:,1),x(:,2))
>> [t,x]=ode45(@shammas_predpray, [0 100], [90 70], options);
>> plot(x(:,1),x(:,2))
```



4. Discuss the stability of the equilibrium point(s) that you determined in Part 1. Give an interpretation of the solutions from Part 2 and 3.

Stability of part 1: the equilibrium points we calculated in part 1 is unstable. Reason is because the populations start to change very fast and very far from that point, which makes it an unstable point.

Interpretation for parts 2 & 3: The plots I created, assuming they are correct, show that the population is constantly changing (oscillating). In the beginning of the plot, the hare dominated the population of lynx, but over time the lynx begin to dominate. From the pattern we see, overtime the hare should again dominate the lynx.

5. Linearize the above predator-prey model at the point $(a_1, a_2) = (20.588, 29.481)$ to arrive at a system of two first-order linear differential equations. Employ Taylor series expansion to expand the functions $f_1(x_1, x_2)$ and $f_2(x_1, x_2)$ at (a_1, a_2) and truncate the series so that only linear component are retained. Hint: The Taylor series expansion of a function of two variables, $f(x_1, x_2)$, at the point (a_1, a_2) is given by:

$$f(x_1, x_2) = f(a_1, a_2) + \left[\frac{\partial f}{\partial x_1}\bigg|_{(a_1,a_2)}\right](x_1 - a_1) + \left[\frac{\partial f}{\partial x_2}\bigg|_{(a_1,a_2)}\right](x_2 - a_2) + nonlinear\ terms$$

```
>> a=3.2; b=0.6; c=50; d=0.56; k=125; r=1.6;
>> dx1=@(x1,x2) r*x1*(1-(x1/k))-((a*x1*x2)/(c+x1));

>> f1x1x2=dx1(20.588,29.481)

f1x1x2 =

  -5.8777e-05
```

This is f(a1,a2)

```
>> syms x1 x2
```

```
>> f1diffx1=diff(dx1,x1)

f1diffx1 =

(a*x1*x2)/(c + x1)^2 - (r*x1)/k - (a*x2)/(c + x1) - r*(x1/k - 1)
```

```
>> f1diffx1=@(x1,x2) (a*x1*x2)/(c + x1)^2 - (r*x1)/k - (a*x2)/(c + x1) - r*(x1/k - 1);

>> f1evalx1=f1diffx1(20.588,29.481)

f1evalx1 =

    0.1263
```

0.1263*(x1-20.588)= 0.1263x1-2.6106. This is the second term in the equation

```
>> f1diffx2=diff(dx1, x2)

f1diffx2 =

-(a*x1)/(c + x1)
```

```
>> f1diffx2=@(x1,x2) -(a*x1)/(c + x1);
```

```
>> f1evalx2=f1diffx2(20.588,29.481)

f1evalx2 =

   -0.9333
```

-0.9333*(x2-29.481)= -0.9333x2+27.5154. This is the third term in the equation

Combining all of the non variable terms:

```
>> (-5.8777e-005)-2.6106+27.5154

ans =

   24.9047
```

Combining it all, we conclude:

F(x1,x2)=24.9047+0.1263x1-0.93333x2

$$F_{1(x_1,x_2)} = 24.9047 + 0.1263x_1 - 0.9333x_2$$

```
>> dx2

dx2 =

    @(x1,x2)b*((a*x1*x2)/(c+x1))-d*x2


>> f2x1x2=dx2(20.588,29.481)

f2x1x2 =

  -1.3365e-04
>> f2diffx1=diff(dx2,x1)

f2diffx1 =

(a*b*x2)/(c + x1) - (a*b*x1*x2)/(c + x1)^2

>> f2diffx1=@(x1,x2) (a*b*x2)/(c + x1) - (a*b*x1*x2)/(c + x1)^2;
```

```
>> f2evalx1=f2diffx1(20.588,29.481)

f2evalx1 =

    0.5680

>> f2diffx2=diff(dx2,x2)

f2diffx2 =

(a*b*x1)/(c + x1) - d

>> f2diffx2=@(x1,x2) (a*b*x1)/(c + x1) - d;
>> f2evalx2=f2diffx2(20.588,29.481)

f2evalx2 =

  -4.5333e-06
```

Firs term is -1.33*10^-4

0.568*x1-11.6941 is the 2nd term

1.33565*10^-4 -4.533*10^-6x2

Cancel 1st term and number in 3rd term

I get: -11.6941+.568x1-4.533*10^-6x2

Combining it all, we get:

$$F_{2(x_1,x_2)} = -11.694 + 0.5680x_1 - 4.533 * 10^{-6}x_2$$

6. Repeat Parts 1-4 for the linearized system.

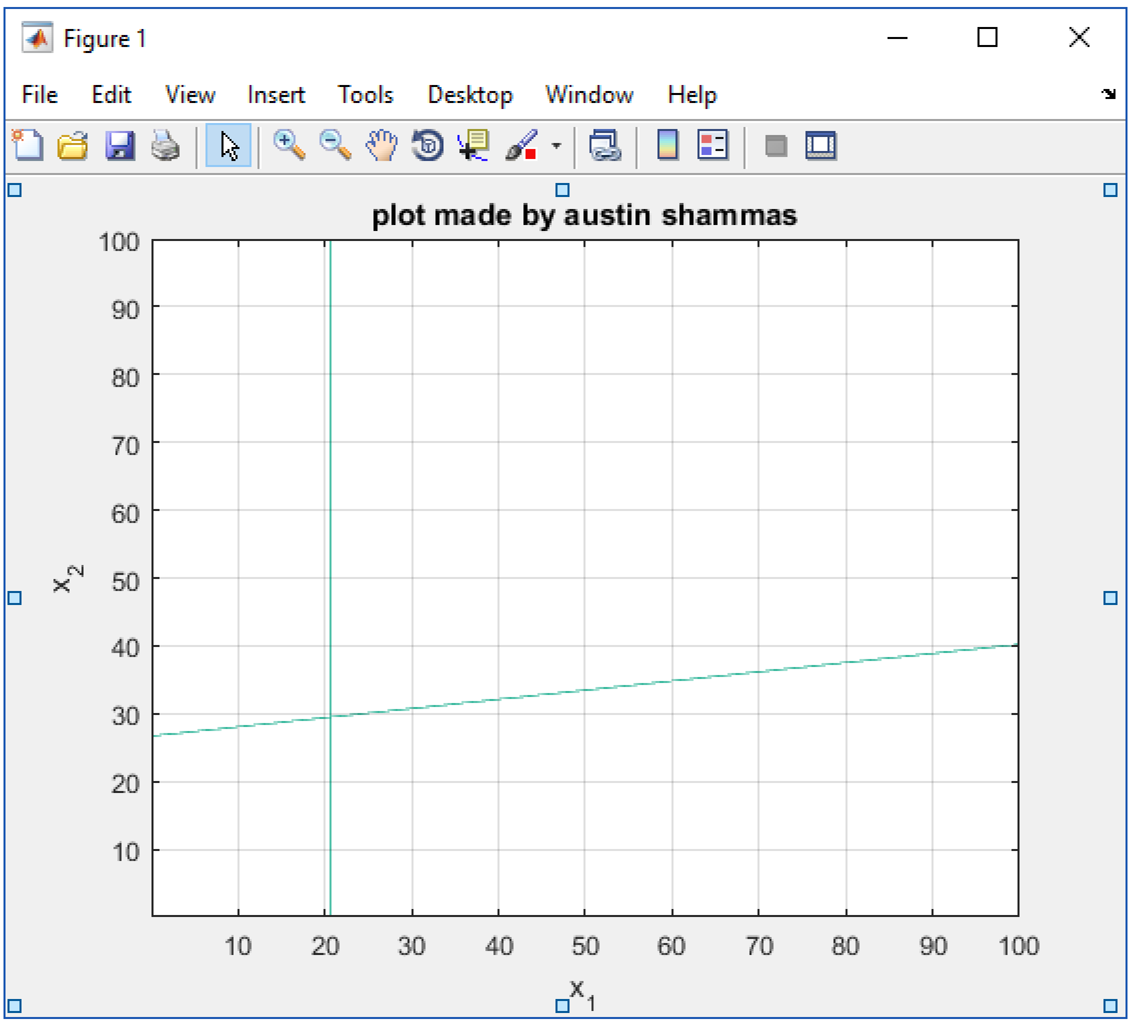
In order to keep things organized, I will label the first question: 6.1, the second question 6.2, etc.

6.1:

1. Find the equilibrium (fixed) point(s) of the system. These are the (x_1^*, x_2^*) points at which the population remains constant (i.e., $\frac{dx_1}{dt} = 0$ and $\frac{dx_2}{dt} = 0$). Start by plotting the resulting two (implicit) nonlinear equations. The plot should give you an estimate of the locations of the equilibrium point(s), which can be used as initial search points to find numerical solutions (accurate to five digits). You may use Matlab's fsolve. (Note that only $x_1 \geq 0$ and $x_2 \geq 0$ are relevant here; positive or zero populations.)

```
>> newF2=@(x1,x2) -11.694+0.568*x1-(4.533*10^-6)*x2;
>> newF1=@(x1,x2) 24.9047+0.1263*x1-0.9333*x2;
``
```

```
>> ezplot(newF1, [0 100])
>> hold on
>> ezplot(newF2, [0 100])
>> grid on
>> title 'plot made by austin shammas'
```



We can guess that the equilibrium point is roughly (20.5, 29). Also, as said before, there is an equilibrium point at (0, 0).

```
function [yp] = shammas_predpray(t,x)
yp= [24.9047+0.1263*x(1)-0.93333*x(2);
    -11.694+0.568*x(1)-(4.533*10^-6)*x(2)];
end
```

```
function F=shammas_fsolve(x)
F(1)=24.9047+0.1263*x(1)-0.93333*x(2);
F(2)= -11.694+0.568*x(1)-(4.533*10^-6)*x(2);

end
```

```
>> fun=@shammas_fsolve;
>> x0=[0,0];
```

```
>> x=fsolve(fun,x0)

Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>


x =

   20.5882   29.4810
```

```
>> x0=[20.5,29];
>> x=fsolve(fun,x0)

Equation solved.

fsolve completed because the vector
as measured by the default value of
the problem appears regular as measu

<stopping criteria details>


x =

   20.5882   29.4810

   '
```

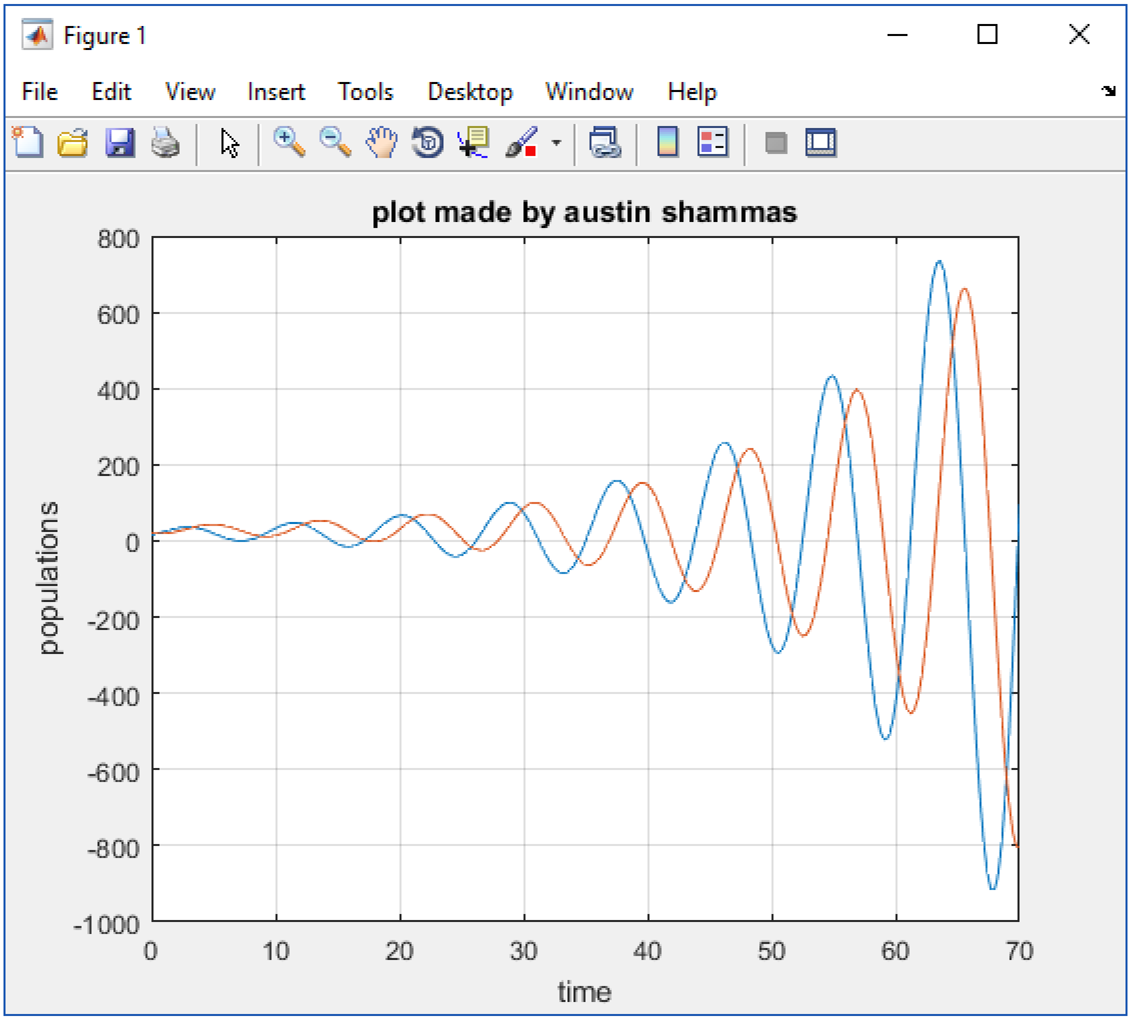
So we can say our new x1*=20.5882 and x2*=29.4810

6.2:

2. Employ ode45 to solve the predator-prey system. Assume the initial populations $x_1(0) = 15$ and $x_2(0) = 20$, and solve for $t \in [0\ \ 70]$. Plot the populations $x_1(t)$ and $x_2(t)$ on the same set of axis.

```
function [yp] = shammas_predpray(t,x)
yp= [24.9047+0.1263*x(1)-0.93333*x(2);
    -11.694+0.568*x(1)-(4.533*10^-6)*x(2)];
end
'
```

```
>> [t,x]=ode45(@shammas_predpray, [0 70], [15 20]);
>> plot(t,x)
>> title 'plot made by austin shammas'; xlabel 'time'; ylabel 'populations'
>> grid on
>>
```



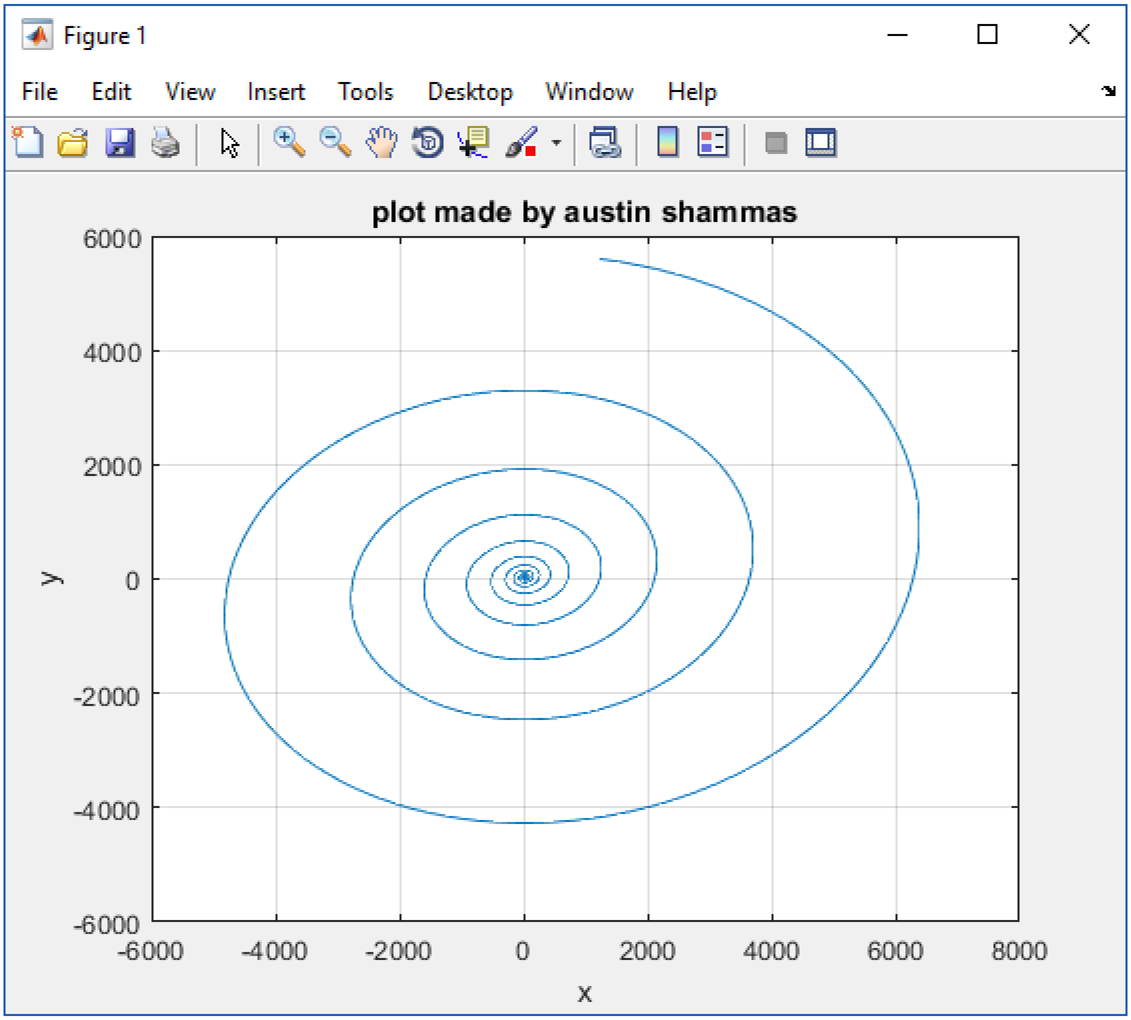
6.3:

3. Generate phase-plane trajectories [plot of $x_2(t)$ vs $x_1(t)$] for each of the following initial populations: (15, 20), (20, 30) and (90, 70). Assume $t \in [0\ \ 100]$.

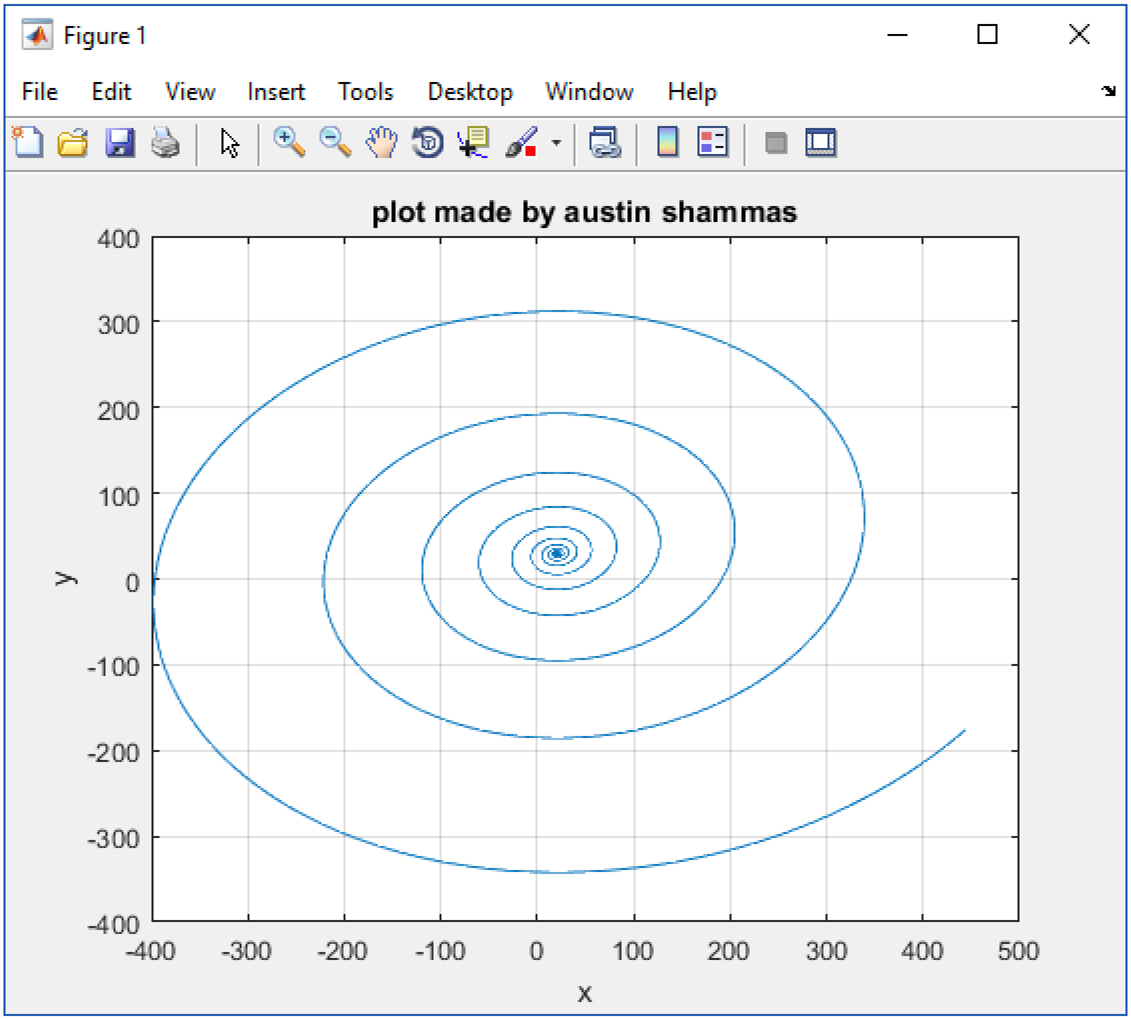
```
function [yp] = shammas_predpray(t,x)
yp= [24.9047+0.1263*x(1)-0.93333*x(2);
    -11.694+0.568*x(1)-(4.533*10^-6)*x(2)];
end
```

```
>> options=odeset('RelTol',1e-6);
```

```
>> [t,x]=ode45(@shammas_predpray, [0 100], [15 20], options);
>> plot(x(:,1),x(:,2))
>> grid on
>> title 'plot made by austin shammas'; xlabel 'x'; ylabel 'y';
>>
```




```
>> [t,x]=ode45(@shammas_predpray, [0 100], [20 30], options);
>> plot(x(:,1),x(:,2))
>> grid on
>> title 'plot made by austin shammas'; xlabel 'x'; ylabel 'y';
>>
```




```
>> [t,x]=ode45(@shammas_predpray, [0 100], [90 70], options);
>> plot(x(:,1),x(:,2))
>> grid on
>> title 'plot made by austin shammas'; xlabel 'x'; ylabel 'y';
```

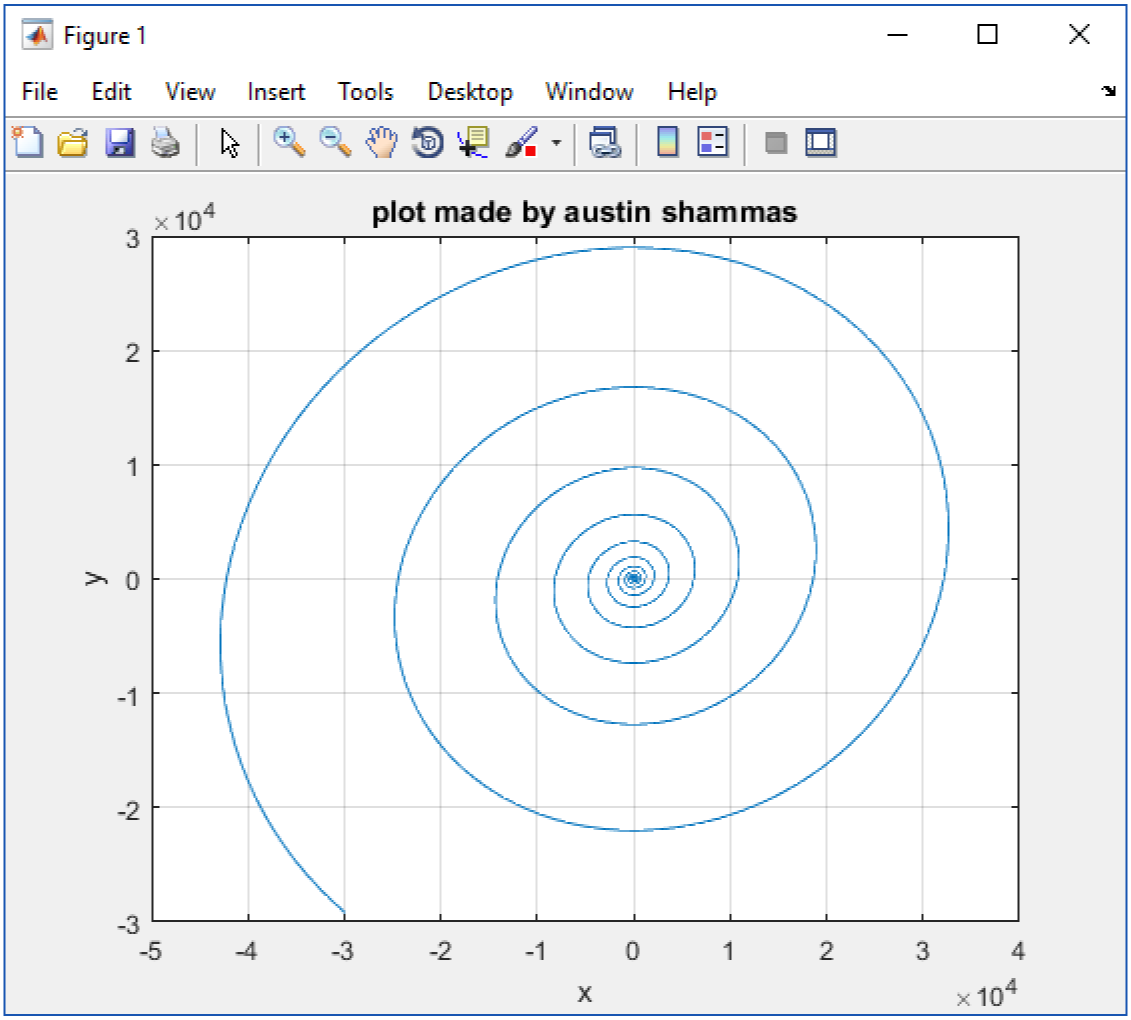
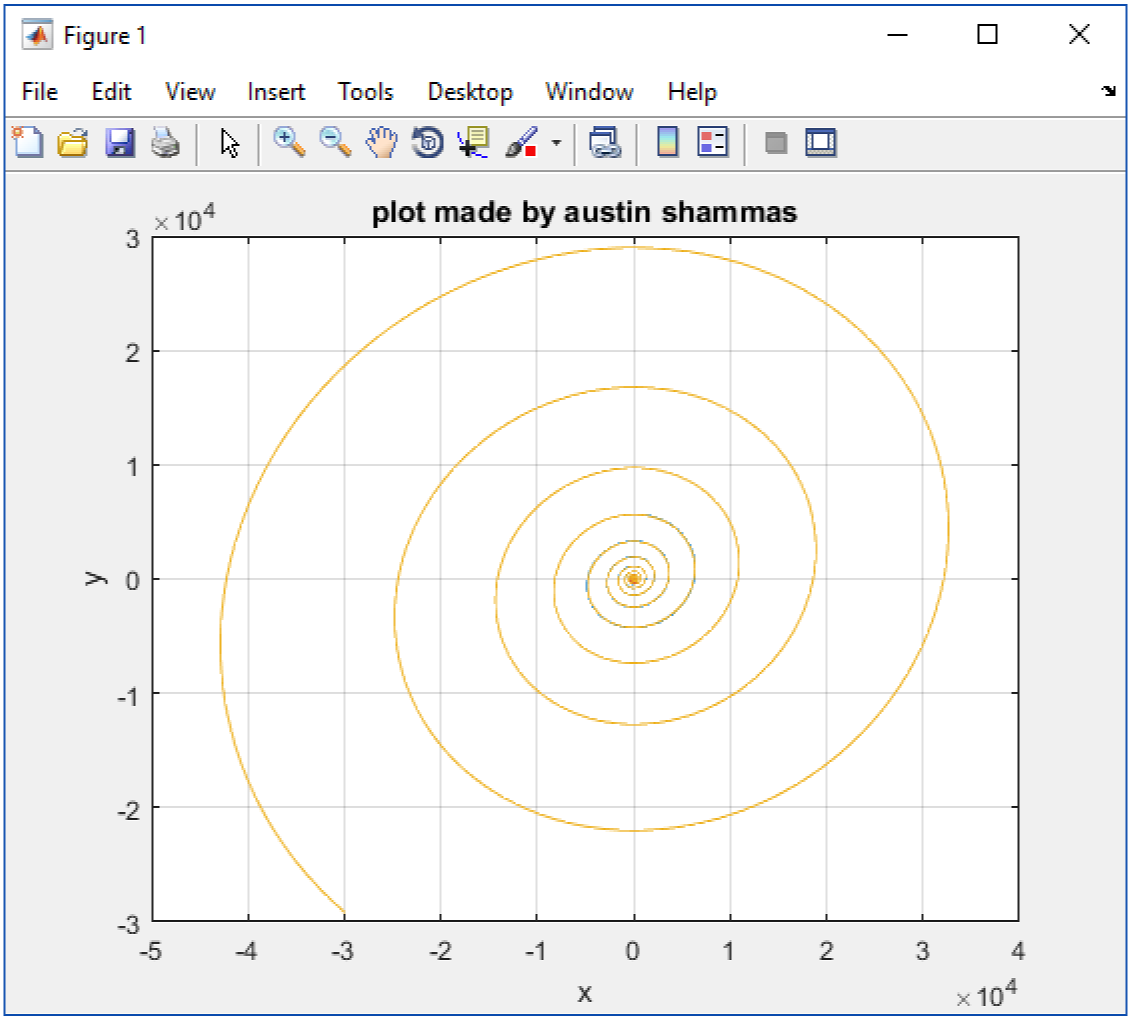


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
plot made by austin shammas
×10^4
3
2
1
0
-1
-2
-3
y
-5
-4
-3
-2
-1
0
1
2
3
4
x
×10^4

```
>> [t,x]=ode45(@shammas_predpray, [0 100], [15 20], options);
>> plot(x(:,1),x(:,2))
>> grid on
>> title 'plot made by austin shammas'; xlabel 'x'; ylabel 'y';
>> hold on
>> [t,x]=ode45(@shammas_predpray, [0 100], [20 30], options);
>> plot(x(:,1),x(:,2))
>> [t,x]=ode45(@shammas_predpray, [0 100], [90 70], options);
>> plot(x(:,1),x(:,2))
```



4. Discuss the stability of the equilibrium point(s) that you determined in Part 1. Give an interpretation of the solutions from Part 2 and 3.

Stability of the equilibrium points: the equilibrium points we found in part 1 are stable. This is because the population stays around this point for a long time and does not quickly move from it.

Interpretation of solutions from part 2&3: In the linearized model, it seems like the predator prey model is somewhat stable. The reason why I think this is because one population isn't much larger than the other, like how it was in the nonlinear model. In this model, when the population of the hare increases, the population of the lynx also increases. This goes on until time is about 45 years, at this point the hares start to increase faster than the lynx, but it is still not increasing by a substantial amount (like the non linear model).

7. Generate a single phase-plane plot that compares the trajectories of the nonlinear and the linear models. Assume initial population (20, 29), and use $t \in [0 \ \ 70]$.

```
function [yp] = shammas_predpray(t,x)
yp= [24.9047+0.1263*x(1)-0.93333*x(2);
    -11.694+0.568*x(1)-(4.533*10^-6)*x(2)];
end
```

```
>> [t,x]=ode45(@shammas_predpray, [0 70], [20 29], options);
>> plot(x(:,1),x(:,2))
```

```
function [yp] = shammas_predpray(t,x)
yp= [1.6*x(1)*(1-x(1)/125)-((3.2*x(1)*x(2))/(50+x(1)));
    0.6*((3.2*x(1)*x(2))/(50+x(1)))-0.56*x(2)];
end
```

```
>> [t,x]=ode45(@shammas_predpray, [0 70], [20 29], options);
>> plot(x(:,1),x(:,2))
>> grid on; title 'plot made by Austin Shammas'; xlabel 'x'; ylabel 'y'
```

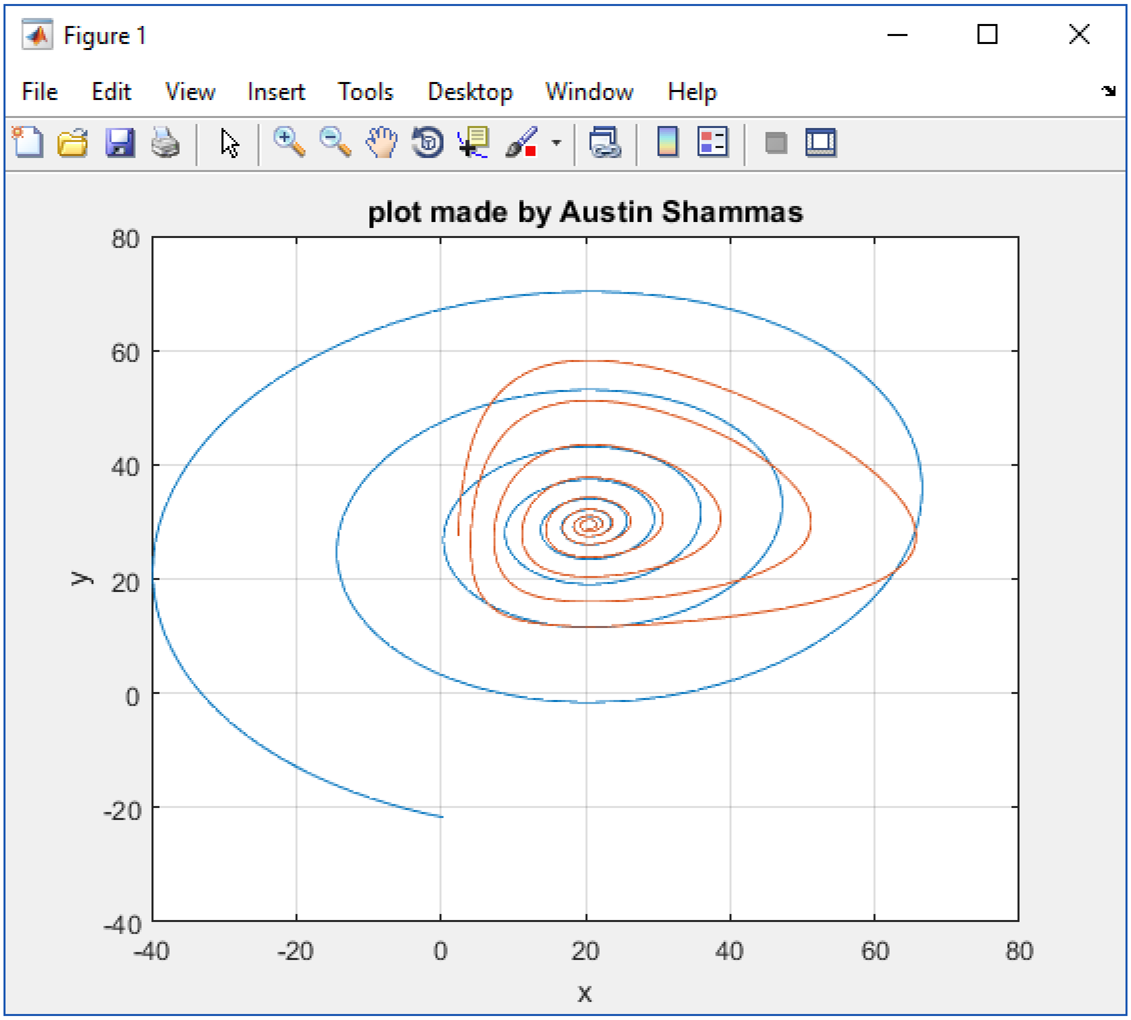


Figure 1
File Edit View Insert Tools Desktop Window Help
plot made by Austin Shammas
80
60
40
20
0
-20
-40
y
-40 -20 0 20 40 60 80
x