Lotka-Volterra Model vs. Rozenweig-MacArthur Model

1 Modeling the Lotka-Volterra Equations

For this project , we will compare the simple *Lotka-Volterra* equations with a modification of the Lotka-Volterra equations called the *Rosenzweig-MacArthur* equations. In order to do this, we will use the Hare/Lynx data from Table 5.1 in the book by Brauer and Castillo-Chavez. For the application, we will be comparing our results with the Hare/Lynx data found in Table 5.1 on page 186 of Chapter 5 in the book by Brauer and Castillo-Chavez.
We will use the following parameter values,

$$\lambda = 32.221,$$
$$b = 1.3847,$$
$$\mu = 27.432,$$
$$c = 0.5433.$$

We will find the Lotka-Volterra linearized stability equations at the equilibrium point (x^*, y^*). From there, we will plot the vector field and the orbits, comparing the orbits of the linearized equations to those of the Lotka-Volterra equations. We will then compare the solutions for the Lotka-Volterra equations to the actual data of the Hares/Lynx vs time, in an effort to figure out the correlation between Hares and Lynx, and the effects of bringing new predators into the mix.

1.: First, we will find the linearized stability equations for the Lotka-Volterra equations at the equilibrium point, (x^*, y^*), when both species coexist and determine the stability of the equilibrium point by finding the eigenvalues. We will see the type of trajectory that a point near the equilibrium point will have, and plot the orbits about the equilibrium point for the initial perturbations of

$$(u(0), v(0)) = (21 - x^*, 44 - y^*), \tag{1}$$
$$(u(0), v(0)) = (72 - x^*, 23 - y^*). \tag{2}$$

The Lotka-Volterra equations are

$$\begin{aligned} F(x,y) &= \frac{dx}{dt} = x(\lambda - by) \\ G(x,y) &= \frac{dy}{dt} = y(-\mu + cx) \end{aligned} \tag{3}$$

with partial derivatives:

$$\begin{aligned} F_x(x,y) &= \lambda - by \\ F_y(x,y) &= -bx \\ G_x(x,y) &= cy \\ G_y(x,y) &= -\mu + cx \end{aligned} \tag{4}$$

Plugging in our parameter values, we get:

$$\begin{aligned} F_x(x,y) &= 32.221 - 1.3847y \\ F_y(x,y) &= -1.3847x \\ G_x(x,y) &= 0.5433y \\ G_y(x,y) &= -27.432 + 0.5433x \end{aligned} \tag{5}$$

Now we need to find the linearization of the system at $(x^*,\ y^*)$.
To do this, we can take:

$$\frac{d}{dt}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} F_x(x^*, y^*) & F_y(x^*, y^*) \\ G_x(x^*, y^*) & G_y(x^*, y^*) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \tag{6}$$

Calculating this, we get:

$$\begin{aligned} u' &= F_x(x^*, y^*)u + F_y(x^*, y^*)v \\ v' &= G_x(x^*, y^*)u + G_y(x^*, y^*)v \end{aligned} \tag{7}$$

So,

$$\begin{aligned} u' &= (32.221 - 1.3847y^*)u + (-1.3847x^*)v \\ v' &= (0.5433y^*)u + (-27.432 + 0.5433x^*)v \end{aligned} \tag{8}$$

Since F(x,y) = $\frac{dx}{dt}$ and G(x,y) = $\frac{dy}{dt}$, then the equilibrium points will happen when F(x,y) = 0 and G(x,y) = 0.

$$\begin{aligned}
&F(x,y) = x(\lambda - by) = 0 \\
&G(x,y) = y(-\mu + cx) = 0 \\
&\implies x = 0, y = 0 \\
or& \\
&\implies \lambda - by = 0 \\
&\implies y = \frac{\lambda}{b} \\
&\implies -\mu + cx = 0 \\
&\implies x = \frac{\mu}{c}
\end{aligned} \tag{9}$$

Then, inserting the values that we know, we get two equilibrium points:

$$\begin{gathered}
(0,0) \\
(\frac{27.432}{0.5433}, \frac{32.221}{1.3847}) \approx (50.4914, 23.2693)
\end{gathered} \tag{10}$$

Since we want an equation for a situation where both species exist, the equilibrium point (0,0) would not be good to use, since this would mean that neither species exists. Instead we will focus on the second point. If we plug this point into our equations for u' and v', we get:

$$\begin{aligned}
u' &= (32.221 - 1.3847(\frac{32.221}{1.3847}))u + (-1.3847(\frac{27.432}{0.5433}))v \\
\implies u' &= 0u + (-69.915)v = -69.915v \\
v' &= (0.5433(\frac{32.221}{1.3847}))u + (-27.432 + 0.5433(\frac{27.432}{0.5433}))v \\
\implies v' &= 12.642u + 0v = 12.642u
\end{aligned} \tag{11}$$

Now we need to check the stability of this equilibrium point.
Let

$$A = \begin{bmatrix} 0 & -69.915 \\ 12.642 & 0 \end{bmatrix} \tag{12}$$

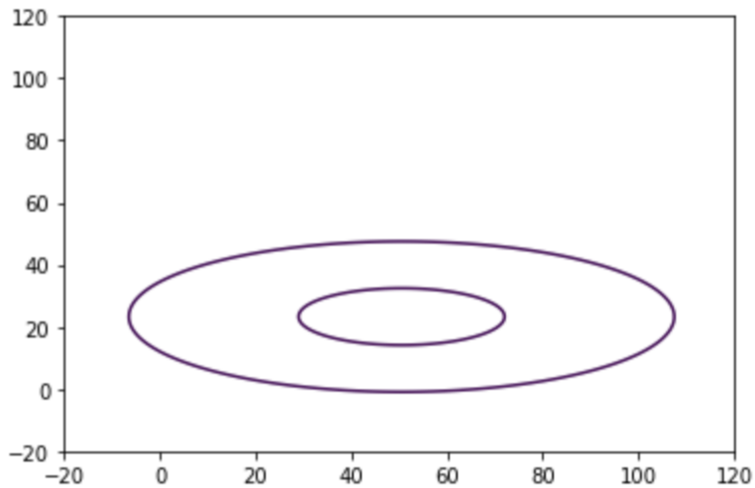
Then taking A - σ I, we get

$$A = \begin{bmatrix} 0-\sigma & -69.915 \\ 12.642 & 0-\sigma \end{bmatrix} \tag{13}$$

For the eigenvalues:

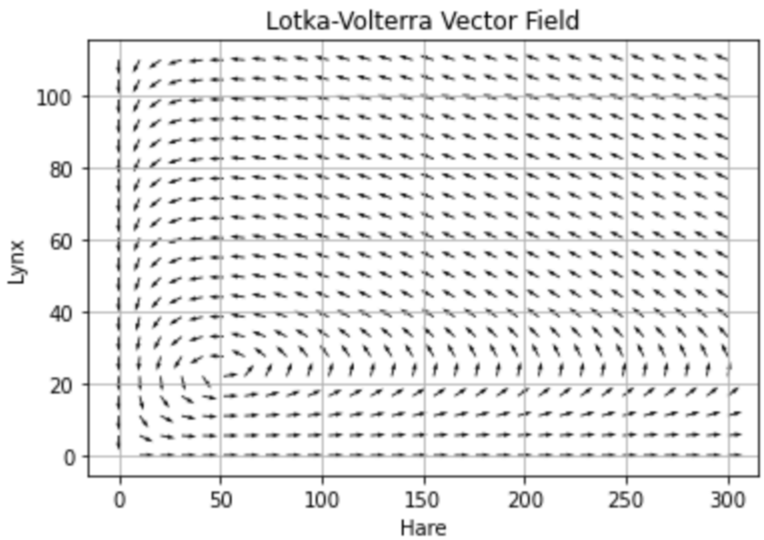
$$\begin{aligned}
&det(A - \sigma I) = (-\sigma)(-\sigma) - (-69.915)(12.642) = \sigma^2 + 883.8654 \\
&\implies \sigma = 0 \pm \sqrt{883.8654} = \pm 29.7299
\end{aligned} \tag{14}$$

Since the eigenvalues are complex and the trace of A is equal to 0, then the equilibrium point (50.4914,23.2693) is stable and the a point close to the equilibrium point will form a closed orbit around it.
The plot can be seen down below:



2.: Now, we will plot the vector field for the Lotka-Volterra equations. In addition, we will plot the orbits for the Lotka-Voltera equations on the same graph for the initial conditions, $(x_0, y_0) = (21, 44)$, $(72, 23)$, and $(2, 16)$. We will see if the vector field and orbits describe the same trajectory that you found in previously.

Shown below is the vector field for the Lotka-Volterra equations: In order to



work with the initial conditions (x_0, y_0), we need to set $\frac{dx}{dt} = \frac{dy}{dt}$. From before, we know that

$$\begin{aligned} \frac{dx}{dt} &= x(\lambda - by) = x(32.221 - 1.3847y) \\ \frac{dy}{dt} &= y(-\mu + cx) = y(-27.432 + 0.5433x) \end{aligned} \tag{15}$$

Then, setting these equal to each other, we get:

$$\begin{aligned} &\frac{dx}{x(32.221 - 1.3847y)} = \frac{dy}{y(-27.432 + 0.5433x)} \\ &\implies \frac{32.221 - 1.3847y}{y} dy = \frac{-27.432 + 0.5433x}{x} dx \\ &\implies \int \frac{32.221}{y} - 1.3847 dy = \int \frac{-27.432}{x} + 0.5433 dx \\ &\implies h = 32.221 \log y + 27.432 \log x - 1.3847y - 0.5433x \end{aligned} \tag{16}$$

Now we can use the initial conditions to calculate h.

For $(x_0, y_0) = (21, 44)$,

$$\begin{aligned} h &= 32.221 \log y + 27.432 \log x - 1.3847y - 0.5433x \\ h &= 32.221 \log(44) + 27.432 \log(21) - 1.3847(44) - 0.5433(21) \\ h &= 16.8887 \end{aligned} \tag{17}$$

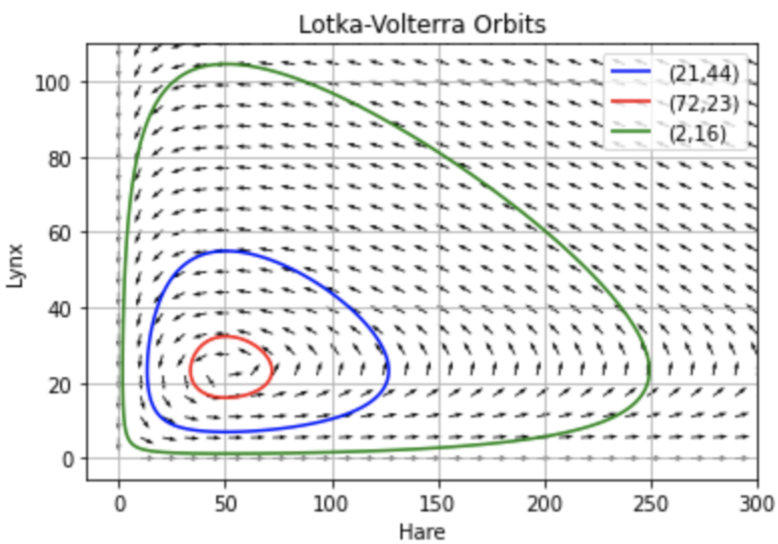
For $(x_0, y_0) = (72, 23)$,

$$\begin{aligned} h &= 32.221 \log y + 27.432 \log x - 1.3847y - 0.5433x \\ h &= 32.221 \log(23) + 27.432 \log(72) - 1.3847(23) - 0.5433(72) \\ h &= 23.8609 \end{aligned} \tag{18}$$

For $(x_0, y_0) = (2, 16)$,

$$\begin{aligned} h &= 32.221 \log y + 27.432 \log x - 1.3847y - 0.5433x \\ h &= 32.221 \log(16) + 27.432 \log(2) - 1.3847(16) - 0.5433(2) \\ h &= 23.8140 \end{aligned} \tag{19}$$

Using these values, we now get the graph below:



3.: Now we compare the two orbits we found.

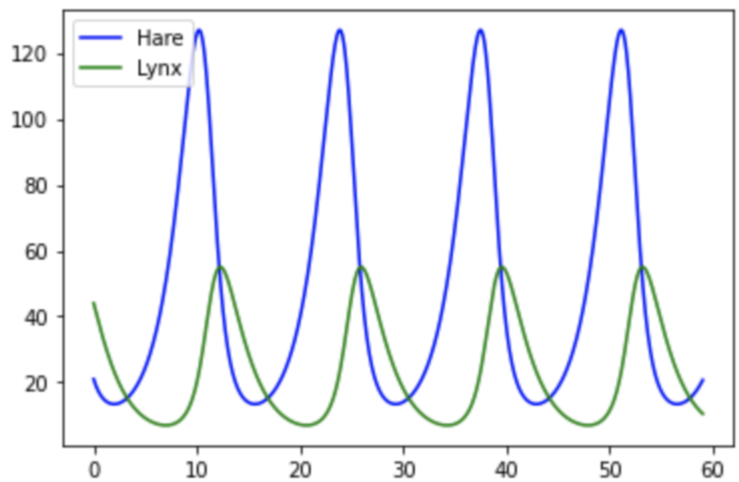
In problem (i), the orbits found are from the linearized equations. The issue arises in the fact that the system itself is not linear, so the orbits are just an approximation of the relationship between hares and lynxes.

In problem (ii), the orbits found are from the Lotka Volterra equations and finding the h values. While the orbits from problem (i) are simple ellipses, these orbits are strange shapes, though they do follow the vector field path.

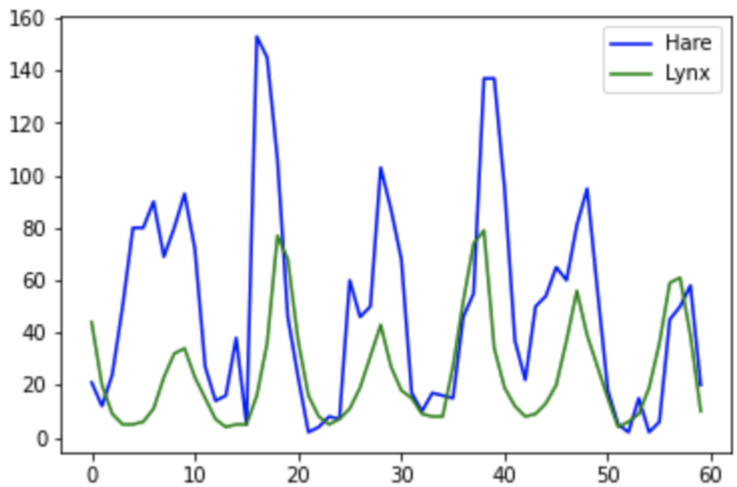
The orbits from problem (i) and the orbits from problem (ii) are similar though in their trajectory, so they both can be considered decent models. The main issue with the orbits from problem (i) is that they are tight around the equilibrium point and closed. This is an issue because they may not be quite as accurate for more extreme cases in the population amounts. Due to this, the orbits found through the linearized equations are not as accurate as the orbits found from the Lotka Volterra equations.

4.: We now plot the solutions for the Lotka-Volterra system. Then, we will plot the Hare data versus time and the Lynx data versus time on the same graph, with the time from $t = 0$ to $t = 59$. We can then see how the solutions compare to the data.

The plot for the solutions of the Lotka-Volterra system is shown below:



The plot for "Hare vs Time" vs "Lynx vs Time" is shown below: What

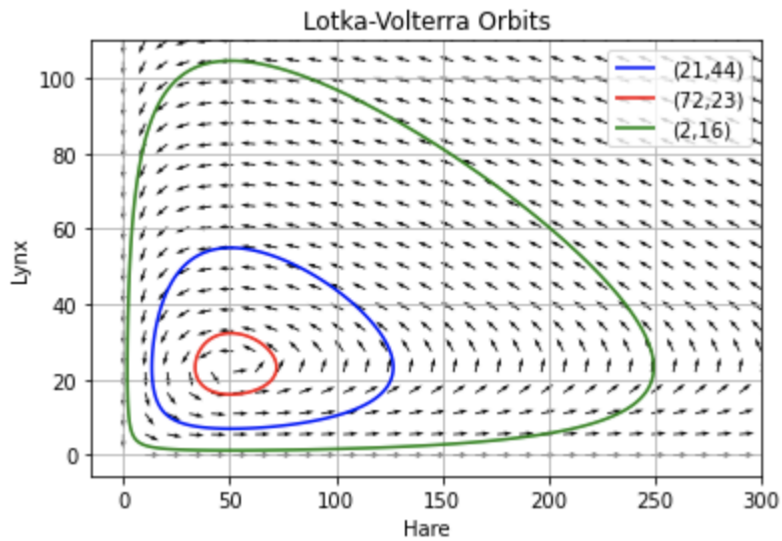


can be seen here is that the hares and the lynx have a direct correlation to each other. When the hares population rises, the lynx population rises, and vice versa. We can see that there is time between the changes to the hare population and changes to the lynx population. This indicates that the lynx are affected by the hares. If there are more hares, the lynx have more to eat, so there are more lynx. Once the hares reach their population ceiling and begin to decline, the lynx eventually decline as well due to lack of food.

5.: Now, we can estimate how the lynx population will behave based on if the initial population of hares is zero, and also how the hare population will behave if the initial population of lynxes is zero.

First, when the initial population of hares is zero, how will the lynx population behave? When the initial population of hares is zero, we can see the population of lynxes decreases really fast until t = 10. On the other case, when the initial population of lynxes is zero, the population of hare grows exponentially when t is approached to 60.

6.: Suppose that hunters begin hunting the lynxes which increases the death rate μ of the lynxes. How does this affect the solution to the Lotka-Volterra equations? What happens to the populations if we instead model the hunting of lynxes in terms of constant yield harvesting.



The hunting the lynxes means the population of lynxes decreases. So, we can see that there are some primary affects when the death rate of the lynx increases. First, the equilibrium point is changed. As you seen the graph up here. The first orbit with the initial point was (72,23) but it has changed to the (21,44) and (2,16). Second, the maximum possible population of both hares and lynx increases, When the lynxes are dead, there are more possibility chance to alive for the hares. So, the population of hares has increased. Also, when the population of hares has increased, then there are more foods for the less population of lynxes. It means the population of lynxes is going to be increased. If we look this up with the equation of the non-linearized orbits of the Lokta-Volterra system. Only $\mu logx$ is affected by μ value. It also means the maximum hare population will increase with an increased value of μ. And the increasing value of $\mu logx$ leads an increased lynx population.

$$\frac{dx}{dt} = x(\lambda - by)\frac{dy}{dt} = y(-\mu + cx) - H \tag{20}$$

when the H = 0.25 is applied. there are slightly increasing of maximum population of lynxes and hares. However, it also changes the equilibrium points as well.

2 Modeling the Rosenzweig-MacArthur Equations

There are several things that the Lotka-Volterra model does not take into account. For example, the growth of the prey is unbounded which we can correct by incorporating the logistic model instead of the simple model. Additionally, if we look at the growth of lynxes, it linearly depends on the number of hares, in other words, increasing the number of hares directly increases the number hares consumed by the lynxes. This is not practical, since a lynx does not spend all of its time consuming hares and can ideally only catch and eat one hare at a time. These considerations give rise to the Rosenzweig-MacArthur equations,

$$x' = \lambda x\big(1 - \frac{x}{K}\big) - bx\phi(x)y, \tag{21}$$

$$y' = y(cx\phi(x) - \mu), \tag{22}$$

where $x\phi(x)$ is called the *predator functional response* and represents the number of prey consumed per predator in unit time. Note that if $\phi(x) = 1$ and $K \to \infty$ we will recover the Lotka-Volterra equations.

There are many different choices for the predator functional response. So for this project, we will simply consider,

$$x\phi(x) = x^q, \quad q < 1. \tag{23}$$

We will use the same parameter values as before, and for the new parameters we use,

$$K = 400, \tag{24}$$

$$q = 0.88. \tag{25}$$

7.: Now, we can find the linearized stability equations for the Rosenzweig-MacArthur equations at the equilibrium point when both both species coexist and determine the stability of the equilibrium point by finding the eigenvalues. We will see what sort of trajectory will a point near the equilibrium point have. Then, we will plot the orbit about the equilibrium point for the initial perturbation of

$$(u(0), v(0)) = (-10, 10). \tag{26}$$

The Rosenzweig-MacArthur equations are

$$x' = \lambda x(1 - \frac{x}{K}) - bx\phi(x)y \tag{27}$$

$$y' = y(cx\phi(x) - \mu) \tag{28}$$

First, We need to find the partial derivative of F(x,y) and G(x,y)

$$\begin{aligned} F_x(x,y) &= \lambda(1 - \frac{2x}{K}) - bqx^(q-1)y \\ F_y(x,y) &= -bx^q \\ G_x(x,y) &= cyqx^(q-1) \\ G_y(x,y) &= -\mu + cx^q \end{aligned} \tag{29}$$

Plugging in our parameter values, we get:

$$\begin{aligned} &F_x(x,y) = 32.221(1 - \frac{2x}{400}) - (1.3847)(0.88)yx^{(0.88)-1} \\ &\implies F_x(x,y) = 32.221(1 - \frac{x}{200}) - 1.218536yx^{-0.12} \\ &F_y(x,y) = -1.3847x^{0.88} \\ &G_x(x,y) = (0.5433)y(0.88)x^{0.88-1} \\ &\implies G_x(x,y) = 0.478104yx^{-0.12} \\ &G_y(x,y) = -27.432 + 0.5433x^{0.88} \end{aligned} \tag{30}$$

Now we need to find the linearization of the system at (x^*, y^*).
To do this, we can take:

$$\frac{d}{dt}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} F_x(x^*,y^*) & F_y(x^*,y^*) \\ G_x(x^*,y^*) & G_y(x^*,y^*) \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} \tag{31}$$

Calculating this, we get:

$$\begin{aligned} u' &= F_x(x^*,y^*)u + F_y(x^*,y^*)v \\ v' &= G_x(x^*,y^*)u + G_y(x^*,y^*)v \end{aligned} \tag{32}$$

So,

$$
\begin{aligned}
u' &= (32.221(1-\frac{x^*}{200}) - 1.218536y^*x^{*-0.12})u + (-1.3847x^{*0.88})v \\
v' &= (0.478104y^*x^{*-0.12})u + (-27.432 + 0.5433x^{*0.88})v
\end{aligned} \tag{33}
$$

Since F(x,y) = $\frac{dx}{dt}$ and G(x,y) = $\frac{dy}{dt}$, then the equilibrium points will happen when F(x,y) = 0 and G(x,y) = 0.

$$
\begin{aligned}
&F(x,y) = \lambda x(1-\frac{x}{K}) - bx\phi(x)y = 0 \\
&G(x,y) = y(cx\phi(x) - \mu) = 0 \\
&\implies x = 0, y = 0 \\
or& \\
&\implies cx\phi(x) - \mu = 0 \\
&\implies x = \frac{\mu^{\frac{1}{q}}}{c} \\
&\implies y = \frac{\lambda}{b}\frac{\mu^{\frac{1}{q}-1}}{c}(1-\frac{1}{K}(\frac{\mu}{c})^{\frac{1}{q}})
\end{aligned} \tag{34}
$$

Then, inserting the values that we know, we get two equilibrium points:

$$
\begin{aligned}
&(0,0) \\
&(\frac{27.432^{\frac{1}{0.88}}}{0.5433}, \frac{32.221}{1.3847}\frac{27.432^{\frac{1}{0.88}-1}}{0.5433}(1-\frac{1}{400}(\frac{27.432}{0.5433})^{\frac{1}{0.88}})) \\
&\approx (79.3129, 52.7803)
\end{aligned} \tag{35}
$$

Since we want an equation for a situation where both species exist, the equilibrium point (0,0) would not be good to use, since this would mean that neither species exists. Instead we will focus on the second point. If we plug this point into our equations for u' and v', we get:

$$
\begin{aligned}
&u' = (32.221(1-\frac{79.3129}{200}) - 1.218536(52.7803)79.3129^{-0.12})u + (-1.3847(79.3129^{0.88}))v \\
&\implies u' = -3.9098u - 69.9155v \\
&v' = (0.478104(52.7803)79.3129^{-0.12})u + (-27.432 + 0.5433(79.3129^{0.88}))v \\
&\implies v' = 8.7279u + 0v = 8.7279u
\end{aligned} \tag{36}
$$

Now we need to check the stability of this equilibrium point.
Let

$$
A = \begin{bmatrix} -3.9098 & -69.9155 \\ 8.7279 & 0 \end{bmatrix} \tag{37}
$$

Then taking A - σ I, we get

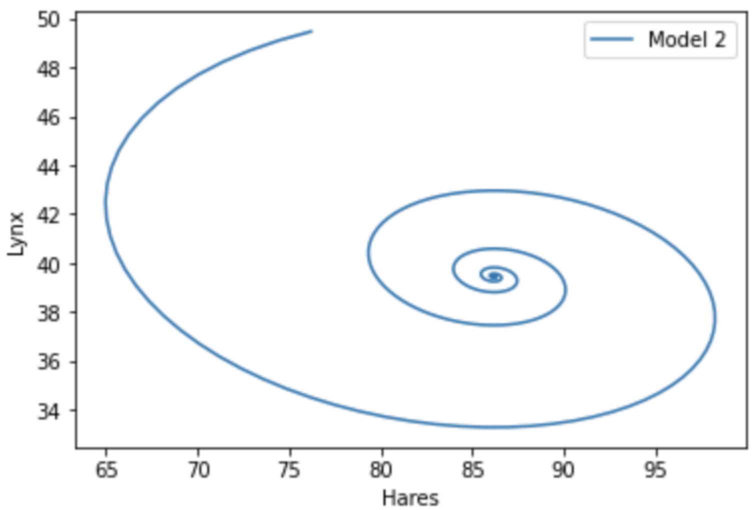
$$
A = \begin{bmatrix} -3.9098-\sigma & -69.9155 \\ 8.7279 & 0-\sigma \end{bmatrix} \tag{38}
$$

For the eigenvalues:

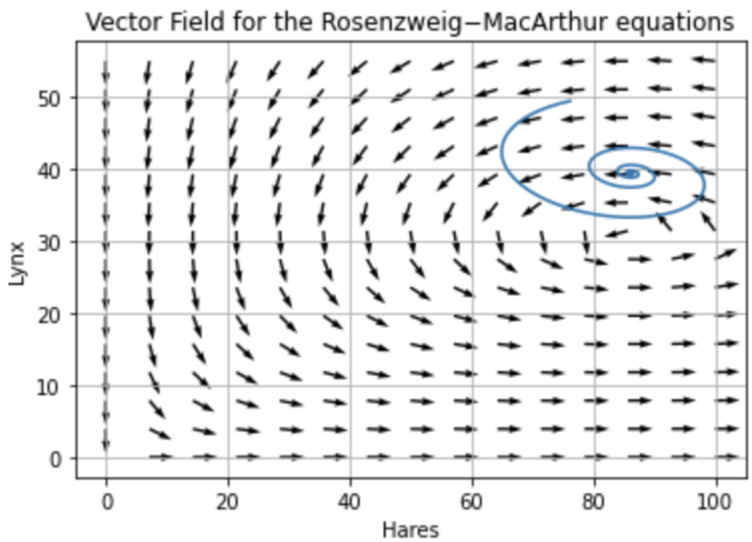
$$
\begin{aligned}
&det(A-\sigma I) = tr(A) \pm \sqrt{tr(A)^2 - 4det(A)} \\
&\implies det(A-\sigma I) = -3.9098 \pm \sqrt{-3.9098^2 - 4(610.2155)} \\
&\implies det(A-\sigma I) = -3.9098 \pm \sqrt{15.2865 - 2440.862} \\
&\implies det(A-\sigma I) = -3.9098 \pm 49.2501i
\end{aligned} \tag{39}
$$

Since the eigenvalues are complex and the trace of A is negative, then the equilibrium point (79.3129, 52.7803) is stable and the a point close to the equilibrium point will curve inward towards it.

The plot can be seen down below:



8.: We will plot the vector field for the Rosenzweig-MacArthur equations on the same plot as previously. We can see if the vector field describes the same trajectory that we found previously.



The orbits described in previously follows the same trajectory as the vector field.

9.: We plot the solutions for the Rosenzweig-MacArthur system and the Hare/Lynx data. How do the solutions compare to the data? How does the solution differ from the Lotka-Volterra solution?

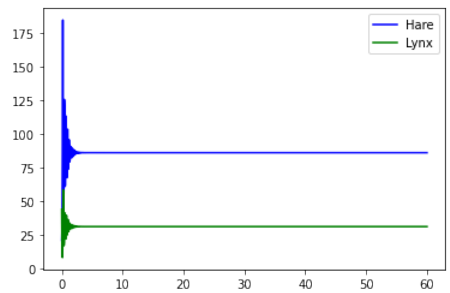


Figure 1: Rosenzweig-MacArthur Solutions

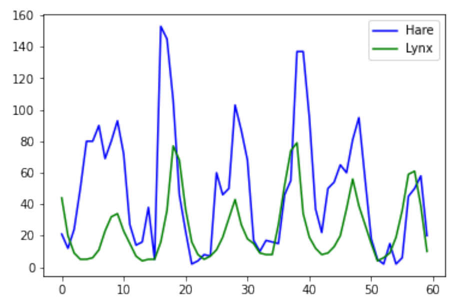
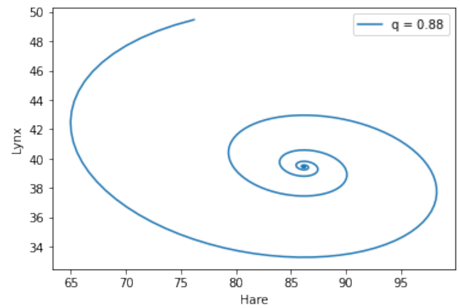
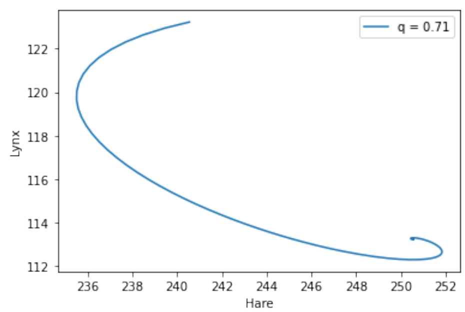


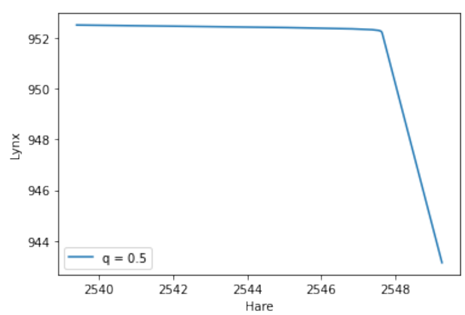
Figure 2: Hares and Lynxes vs Time

In the previous problem, we saw that the both the hares and lynxes were directly correlated. The Rosenzweig-MacArthur solutions differ greatly from the Hare/Lynx data, particularly at time t = 2. It seems that the Rosenzweig-MacArthur system is more accurate for smaller time intervals. The Lotka-Volterra plot is much more accurate in the long run, but the Rosenzweig-MacArthur system shows promise for a small time interval.

10.: If we decrease q, say $q = 0.71$, how will this affect the equilibrium point, or what if $q = 0.5$? We can interpret this physically in regards to the predator-prey interaction.







For $q = 0.88$, our equilibrium point is (86.1935,39.4572).
For $q = 0.71$, our equilibrium point is (250.5522,113.2239).
For $q = 0.5$, our equilibrium point is (2549.3856,942.4997).
As we can see, equilibrium points and our power of q function are inversely related to each other. We see that when we decrease q, our equilibrium point

values increase.
Our predator functional response tells us the number of prey consumed per predator in unit time. As we mentioned, when q decreases, we get an increase in our equilibrium values. This tells us that as consumption of hares per lynx in unit time decreases, we see an increase in hare population, eventually resulting in an increase in our lynx population as well.

11.: Now we can qualitatively describe the differences between the Lotka-Volterra and the Rosenzweig-MacArthur equations.

The Lotka-Volterra equations correlate directly to the the given hare and lynx data. In this model, we see that the lynx population increases linearly as the hare population increases. However, this is not always the case when considering predator-prey interactions. The Lotka-Volterra model implies that the lynxes spend all their time consuming hares, when in reality, it can only catch and eat one hare at a time.
The Rosenzweig-MacArthur model can help us with this issue. In this model, we get a predator functional response. This response tells us the number of prey consumed per predator in unit time. As we can see in these graphs, a decrease in q results in a decrease in our lynx population.
The orbits are different, too. The Lotka-Volterra model form elliptical orbits, while the Rosenzweig-MacArthur model forms spirals. This means, as far as equilibrium points go, the Lotka-Volterra model will never reach them, but the Rosenzweig-MacArthur model will. As K and q get larger, though, the Rosenzweig-MacArthur model actually does start to resemble an ellipses that does not reach the equilibrium point, which makes it much more close to the Lotka-Volterra model.

Figure 3: The figures 4-7 are K = 400

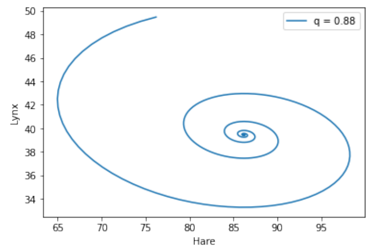


Figure 4: q = 0.5

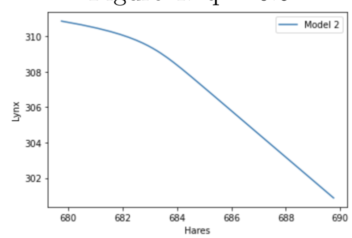


Figure 5: q = 0.6

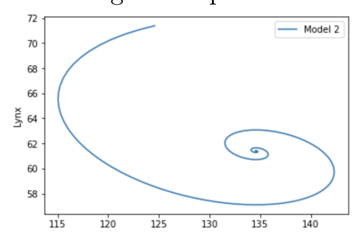


Figure 6: q = 0.8

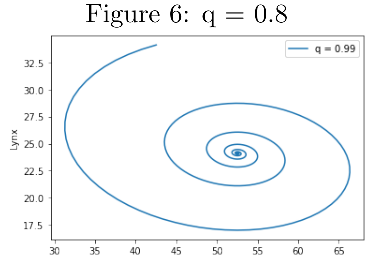


Figure 7: q = 0.99

Figure 8: The figures 9-10 are K = 1000

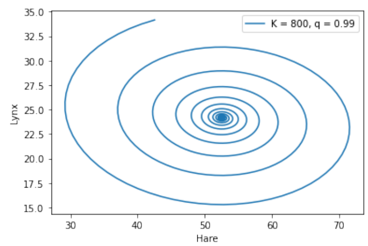


Figure 9: q = 0.88

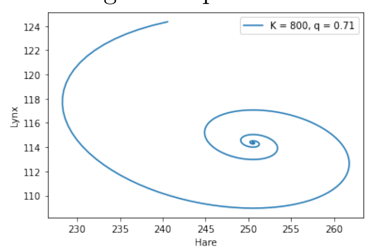


Figure 10: q = 0.99

Figure 11: The figures 12-13 are K = 4000

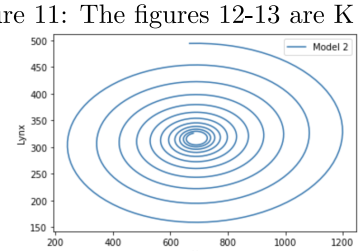


Figure 12: q = 0.6

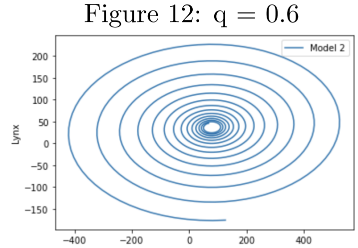


Figure 13: q = 0.9

In conclusion, if there is a situation where two populations' relationship to each other needs to be researched, the Lotka-Volterra equations are good to use, but as far as predator-prey relationships go, the Rosenzweig-MacArthur equations are actually more accurate. The issue with the Lotka-Volterra equations is that due to harvesting, the orbits turn into spirals. This limits our ability to understand what is truly happening with the relationship between the two species. That being said, the Lotka-Volterra model is still a good tool to use when modelling real data, and it is important to know.

Appendix

```
#1
import numpy as np
import matplotlib.pyplot as plt
def p(l,m,c,b,x,y):
    xs = m/c
    ys = l/b
    u = x - xs
    v = y - ys
    k = (u**2)/(b/(c*l))+(v**2)/(c/(b*m))
    print(xs)
    print(ys)
    return k,xs,ys
def main():
    l = 32.221
    b = 1.3847
    m = 27.432
    c = 0.5433
    x1 = 21
    y1 = 44
    x2 = 72
    y2 = 23
    k1,xs,ys = p(l,m,c,b,x1,y1)
    k2,xs,ys = p(l,m,c,b,x2,y2)
    u = np.linspace(-20,120,1000)
    v = np.linspace(-20,120,1000)
    u,v = np.meshgrid(u,v)
    q = ((u-xs)**2)/(b/(c*l))+((v-ys)**2)/(c/(b*m))
    plt.contour(u,v,q,[k1])
    plt.contour(u,v,q,[k2])
    plt.show()
if __name__ == "__main__":
    main()

#2
import os
import math
import numpy as np
import matplotlib.pyplot as pyplot
l = 32.221
b = 1.3847
m = 27.432
c = 0.5433
x = np.linspace(0,300,30)
```

```
y = np.linspace(0,110,21)
x,y = np.meshgrid (x,y)
dx = x*(l-b*y)
dy = y*(-m + c*x)
z = (dx**2 + dy**2)**(0.5)
dx = dx/z
dy = dy/z
pyplot.quiver(x,y,dx,dy)
pyplot.grid()
pyplot.title('Lotka-Volterra Vector Field')
x1 = 21
y1 = 44
x2 = 72
y2 = 23
x3 = 2
y3 = 16
a1 = l*math.log(y1) + m*math.log(x1)-b*y1 - c*x1
a2 = l*math.log(y2) + m*math.log(x2)-b*y2 - c*x2
a3 = l*math.log(y3) + m*math.log(x3)-b*y3 - c*x3
x4 = np.linspace(1,300,1000)
y4 = np.linspace(1,110,1000)
x4,y4 = np.meshgrid(x4,y4)
p = l*np.log(y4)+m*np.log(x4)-b*y4-c*x4
o1 = pyplot.contour(x4,y4,p,[a1],colors ='blue')
o2 = pyplot.contour(x4,y4,p,[a2],colors ='red')
o3 = pyplot.contour(x4,y4,p,[a3],colors ='green')
l1,_ = o1.legend_elements()
l2,_ = o2.legend_elements()
l3,_ = o3.legend_elements()
pyplot.legend([l1[0],l2[0],l3[0]],['(21,44)','(72,23)','(2,16)'], loc='upper right')
pyplot.title('Lotka-Volterra Orbits')
pyplot.xlabel('Hare')
pyplot.ylabel('Lynx')
pyplot.show()

#4
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
def p():
    years=[]
    for i in range(1848,1908):
        years.append(i)
    return years
def q(z,t):
    l = 32.221
```

```
    b = 1.3847
    m = 27.432
    c = 0.5433
    x = z[0]
    y = z[1]
    dx = (x*(l-b*y))/59
    dy = (y*(-m+c*x))/59
    return [dx,dy]
hares = [21,12,24,50,80,80,90,69,80,93,72,27,14,16,38,5,153,145,106,46,23,2,4,8,7,60,46,50,
lynx = [44,20,9,5,5,6,11,23,32,34,23,15,7,4,5,5,16,36,77,68,37,16,8,5,7,11,19,31,43,27,18,15
years=p()
grid = np.linspace(0,59,60)
plt.plot(grid,hares,'b',label='Hare')
plt.plot(grid,lynx,'g',label='Lynx')
plt.legend()
plt.show()
grid = np.linspace(0,59,1000)
xy = odeint(q,[hares[0],lynx[0]],grid)
plt.plot(grid,xy[:,0],'b',label ='Hare')
plt.plot(grid,xy[:,1],'g',label ='Lynx')
plt.legend()
plt.show()

#5
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
def p(z,t):
    l = 32.221
    b = 1.3847
    m = 27.432
    c = 0.5433
    x=z[0]
    y=z[1]
    dx = (x*(l - b*y))/59
    dy = (y*(-m + c*x))/59
    return [dx,dy]
hares = [21, 12, 24, 50, 80, 80, 90, 69, 80, 93, 72, 27, 14, 16, 38, 5, 153, 145, 106, 46, 
68, 17, 10, 17, 16, 15, 46, 55, 137, 137, 95, 37, 22, 50, 54, 65, 60, 81, 95, 56, 18, 5, 2,
lynx=[44, 20, 9, 5, 5, 6, 11, 23, 32, 34, 23, 15, 7, 4, 5, 5, 16, 36, 77, 68, 37, 16, 8, 5,
18, 15, 9, 8, 8, 27, 52, 74, 79, 34, 19, 12, 8, 9, 13, 20, 37, 56, 39, 27, 15, 4, 6, 9, 19,
grid = np.linspace (0,59,1000)
xy1 = odeint(p , [0,lynx[0]] , grid)
xy2 = odeint(p , [hares[0],0], grid)
plt.show()
plt.plot(grid, xy2[:,0], 'r',label='hares')
```

```
plt.plot(grid, xy2[:,1], 'g',label='lynx')
plt.legend()
plt.show()

#7
from scipy.integrate import odeint
import numpy as np
import matplotlib.pyplot as plt
def p(z,t):
    u=z[0]
    v=z[1]
    b = 1.3847
    m = 27.432
    c = 0.5433
    l = 32.221
    k = 400
    q = 0.88
    x = (m/c)**(1/q)
    y = (l*((m/c)**(((1-q)/q))) - ((1/k)*((m/c)**(((2-q)/q)))))/b
    uu = l*(1-(2*x/k)) - b*q*(x**(q-1))*y
    uv = -b*(x**q)
    vu = y*q*c*(x**(q-1))
    vv = c*(x**q) - m
    du = uu*u + uv*v
    dv = vu*u + vv*v
    return [du,dv]
u0 = -10
v0 = 10
b = 1.3847
m = 27.432
c = 0.5433
l = 32.221
k = 400
q = 0.88
xs = (m/c)**(1/q)
ys = (l*((m/c)**(((1-q)/q))) - ((1/k)*((m/c)**(((2-q)/q)))))/b
print("x^* = ")
print(xs)
print("y^* = ")
print(ys)
grid = np.linspace(0,3,1000)
xy = odeint(p,[u0,v0],grid)
plt.plot(xy[:,0] + xs, xy[:,1] + ys, label='Model 2')
plt.xlabel ("Hares")
plt.ylabel ("Lynx")
plt.legend ()
```

```
plt.show()

#8
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
def sys_diff_eq ( z , t ) :
 l = 32.221
 b = 1.3847
 m = 27.432
 c = 0.5433
 x = z [ 0 ]
 y = z [ 1 ]
 dxdt = ( x *(l - b*y ))/59
 dydt = ( y* (-m + c *x ))/59
 return [ dxdt , dydt ]
hares = [21,12,24 , 50 , 80 , 80 , 90 , 69 , 80 , 93 , 72 ,27 , 14 , 16 , 38 , 5 , 153 , 145
         , 137 , 95 , 37 , 22 , 50 , 54 , 65 , 60 , 81 , 95 , 56 ,18 , 5 , 2 , 15 , 2 , 6 ,
lynx = [ 44 , 20 , 9 , 5 , 5 , 6 , 11 , 23 , 32 , 34 , 23 , 15 , 7 , 4 , 5 , 5 , 16 , 36 , 7
        , 34 , 19 , 12 ,8 , 9 , 13 , 20 , 37 , 56 , 39 , 27 , 15 , 4 , 6 , 9 , 19 , 36 , 59

t_grid = np . linspace ( 0 ,59 ,1000 )
xy_sol_1 = odeint(sys_diff_eq , [ 0 , lynx [0]] , t_grid )
xy_sol_2 = odeint(sys_diff_eq , [ hares [0] ,0] , t_grid )
plt.plot(t_grid , xy_sol_2 [: , 0 ] , ' r ' , label= 'hares' )
plt.plot(t_grid , xy_sol_2 [: , 1 ] , ' g ' , label= 'lynx' , linewidth =10)
plt.legend( )
plt.show( )

#9
from scipy.integrate import odeint
import numpy as np
import matplotlib.pyplot as plt
def p():
    years=[]
    for i in range(1848,1908):
        years.append(i)
    return years
def q(z, t):
    x = z[0]
    y = z[1]
    b = 1.3847
    c = 0.5433
    l = 32.221
    m = 27.432
    K = 400
```

```
    q = 0.88
    dxdt = (l*x*(1 - (x/K))) - (b*(x**q)*y)
    dydt = y*(c*(x**q) - m)
    return [dxdt, dydt]
hares = [21,12,24,50,80,80,90,69,80,93,72,27,14,16,38,5,153,145,106,46,23,2,4,8,7,60,46,50,1
lynx = [44,20,9,5,5,6,11,23,32,34,23,15,7,4,5,5,16,36,77,68,37,16,8,5,7,11,19,31,43,27,18,15
years=p()
x_0 = 21
y_0 = 44
xy = odeint(q, [x_0,y_0], grid)
grid = np.linspace(0,60,1000)
xy = odeint(q,[hares[0],lynx[0]],grid)
plt.plot(grid,xy[:,0],'b',label ='Hare')
plt.plot(grid,xy[:,1],'g',label ='Lynx')
plt.legend()
plt.show()
grid = np.linspace(0,59,60)
plt.plot(grid,hares,'b',label='Hare')
plt.plot(grid,lynx,'g',label='Lynx')
plt.legend()
plt.show()


#10
from scipy.integrate import odeint
import numpy as np
import matplotlib.pyplot as plt
def eqn(z,t):
    u = z[0]
    v = z[1]
    b = 1.3847
    c = 0.5433
    l = 32.221
    m = 27.432
    K = 400
    q1 = 0.71
    q2 = 0.5
    x = (m/c)**(1/q)
    y = (l*((m/c)**((1-q)/q)) - ((1/K)*((m/c)**((2-q)/q))))/b
    u_u = l*(1-((2*x)/(K)))-(b*q*y*(x**(q-1)))
    u_v = (-b)*(x**q)
    v_u = c*y*q*(x**(q-1))
    v_v = c*(x**q) - m
    dudt = u_u*u + u_v*v
    dvdt = v_u*u + v_v*v
    return [dudt, dvdt]
```

```
u_0 = -10
v_0 = 10
b = 1.3847
c = 0.5433
l = 32.221
m = 27.432
K = 400
q0 = 0.88
q1 = 0.71
q2 = 0.5
x_star0 = (m/c)**(1/q0)
y_star0 = (l*((m/c)**((1-q0)/q0)) - ((1/K)*((m/c)**((2-q0)/q0))))/b
x_star1 = (m/c)**(1/q1)
y_star1 = (l*((m/c)**((1-q1)/q1)) - ((1/K)*((m/c)**((2-q1)/q1))))/b
x_star2 = (m/c)**(1/q2)
y_star2 = (l*((m/c)**((1-q2)/q2)) - ((1/K)*((m/c)**((2-q2)/q2))))/b
print("for q=0.88, x_star = ")
print(x_star0)
print("for q=0.88, y_star = ")
print(y_star0)
print("for q=0.71, x_star = ")
print(x_star1)
print("for q=0.71, y_star = ")
print(y_star1)
print("for q=0.5, x_star = ")
print(x_star2)
print("for q=0.5, y_star = ")
print(y_star2)
#plot
grid = np.linspace(0,3,1000)
xy = odeint(eqn, [u_0, v_0], grid)
plt.plot(xy[:,0] + x_star0, xy[:,1] + y_star0, label = "q = 0.88")
plt.xlabel("Hare")
plt.ylabel("Lynx")
plt.legend()
plt.show()
grid = np.linspace(0,3,1000)
xy = odeint(eqn, [u_0, v_0], grid)
plt.plot(xy[:,0] + x_star1, xy[:,1] + y_star1, label = "q = 0.71")
plt.xlabel("Hare")
plt.ylabel("Lynx")
plt.legend()
plt.show()
grid = np.linspace(0,3,1000)
xy = odeint(eqn, [u_0, v_0], grid)
plt.plot(xy[:,0] + x_star2, xy[:,1] + y_star2, label = "q = 0.5")
```

```
plt.xlabel("Hare")
plt.ylabel("Lynx")
plt.legend()
plt.show()
```