CS346 HW2.2 Writeup

Matt Bass, Will Johnson, Jackson Prettyman

1a) Analyzation

Through our initial runthrough, we found the general outline of our code to all be fairly similar. In terms of implementation, we all followed a structure of parameter declaration, looping, and formatting, which resulted in the (generally) correct outputs in a digestible and intuitive way. Below we address the unique strengths and weaknesses that either add or subtract from the correctness of readability of homework two exercise one's code.

Jackson's Code

A strength found in Jackson's code lies in his variable naming. The naming follows a very descriptive, albeit wordy, pattern. Each variable is given a name that entirely describes its functions with an example being shark_interaction_birth_rate. This variable, shark_interaction_birth_rate, represents the fraction of shark to tuna interactions that results in the birth of a shark, or the birth rate of sharks during interactions (line 17). This style of naming left no question for CS346 students, and would most likely be readily understandable by non-programmers with proper documentation, which is found with corresponding commenting. It's also important to note the use of Shark or Tuna used in variable naming, rather than abstract letters such as Y or P to represent predator or prey. Although this limits the model to a particular predator-prey relationship in readability, it greatly reduces confusion that could exist between even CS346 members.

There is a problem that is inherent to this style of naming. Jackson's code suffered from exceeding per-line character limits. This formatting error damages readability, and it is especially prevalent with Jackson's wordy variable naming and same line comments. Fortunately, the solution to this problem is relatively simple and existed within Matt's code. The use of ellipses as well as different commenting structures can alleviate this readability problem.

Finally, Jackson's code suffered from a correctness problem. His value for delta time used throughout, 0.03, was too large that the discrete changes in the population model resulted in an error that gradually increased both the tuna and shark population over time (line 21). This was an error address in class, and can simply be fixed using a smaller delta time value. In this case, 0.003 is effective at producing the right results without absolutely destroying the runnability of the function.

Will's Code

Jackson and Will's implementations were the two most similar, with that came a similar effectiveness as well as similar ineffectiveness. For example, both Jackson and Will suffered from exceeding the character limit. However, there was an important difference between the two in terms of a correctness error.

While Jackson's model suffered from insufficiently small value for delta t, Will did not have this issue explicitly. When running his simulation, the shark population and tuna population would maintain proper maximum bounds. However, Will's delta t was a value of 1. This was effective because he was using a time value of 5000 months, while Jackson used a time value of 12 months. This results in 5000 iterations in Will's simulation, while Jackson only had 400. However, this conflicts with an assumption made in the creation of this model, being that the human population is constant. We are able to hold this human population constant over 12 months, as it is reasonable to assume there aren't significant changes in the human population over 12 months. However, we can not do the same over 5000 months. The solution here is also relatively simply, as an alteration of the time and delta time values will alleviate it.

Matt's Code

Matt's code stood out the most. While he implemented the general solution very similarly in a parameter declaration, looping, and formatting structure, he went above and beyond to help visualize the equilibrium points in the model. He did that through creating a vector field based on the derivative functions used in the model. While the three of us shared derivative equations, his use of the vector field allowed him to identify the true equilibrium points for his specific model specifications. This is distinct from say Jackson's write up which derived the equations needed to find the equilibrium point. This inclusion of data visualization greatly helps in the understanding of the Lotka-Volterra model, not only for non-CS346 members, but also for Jackson and Will.

1b) Revisions

Comprehensive Variable Naming

Example:

```
+human_birth_fraction_tuna = 0.04;%(b_2 should probably be a float less than
+                             % 1 ,this is because represnts number of
+                             % humans produced from one shark)
```

Line Character Limits

Example:

```
+human_birth_fraction_tuna = 0.04;%(b_2 should probably be a float less than
+                             % 1 ,this is because represnts number of
+                             % humans produced from one shark)
```

Proper Simulation Length

Examples:

Model 1:

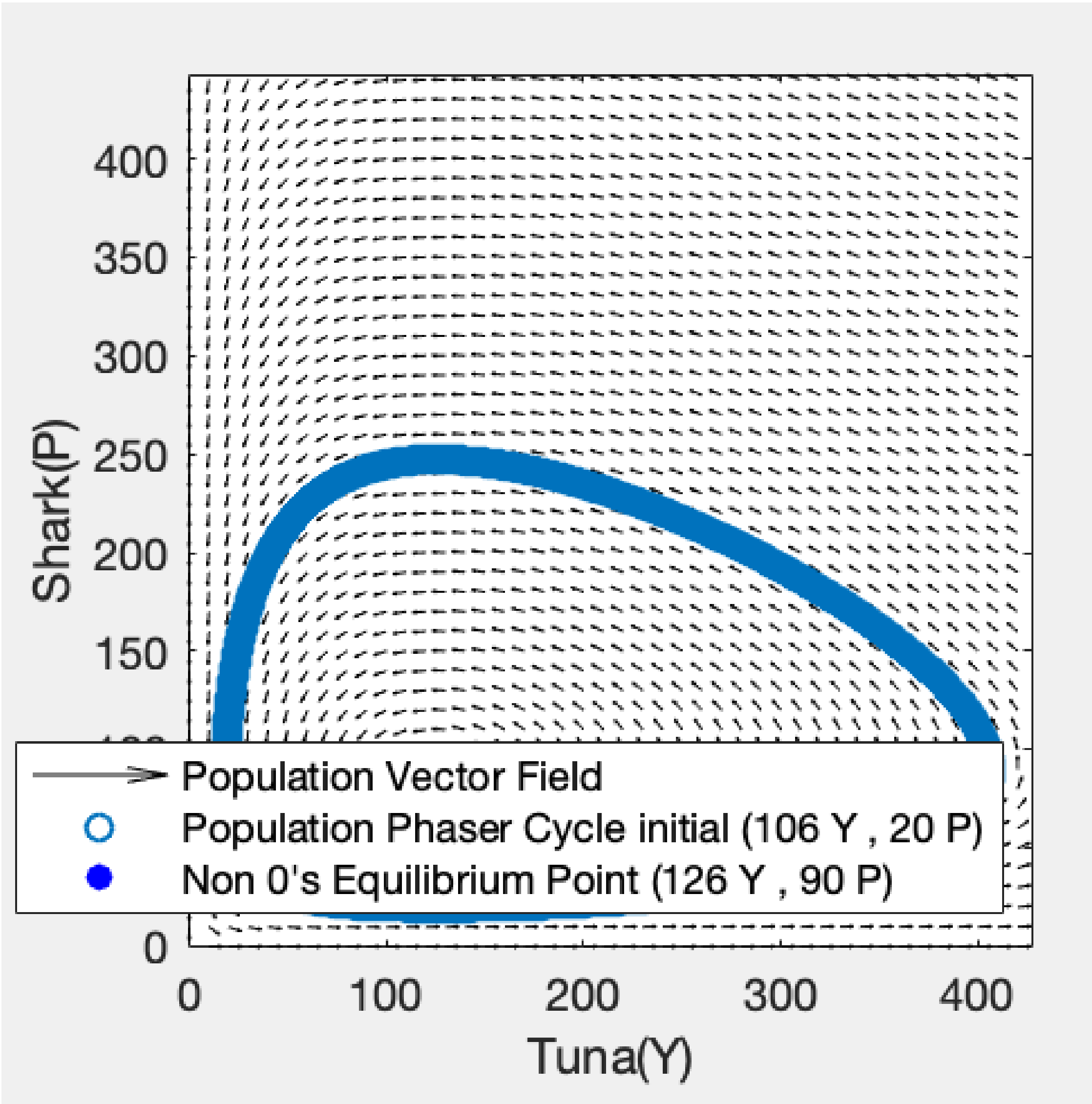
```
+numMonths = 12; %(int) This is the number of months to tun the sim
+dt = 0.0001; %(float) Time step of the sim in units of months
```

Model 2:

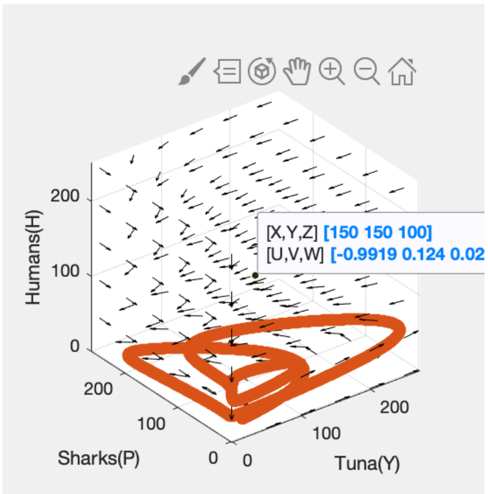
```
+numMonths = 120; %(int) This is the number of months to tun the sim
+dt = 0.0001; %(float) Time step of the sim in units of months
```

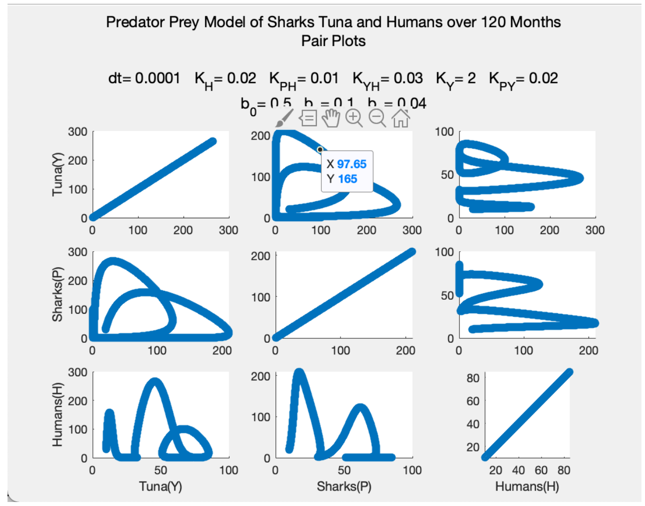
Equilibrium Point Visualization

Model 1:



Model 2:

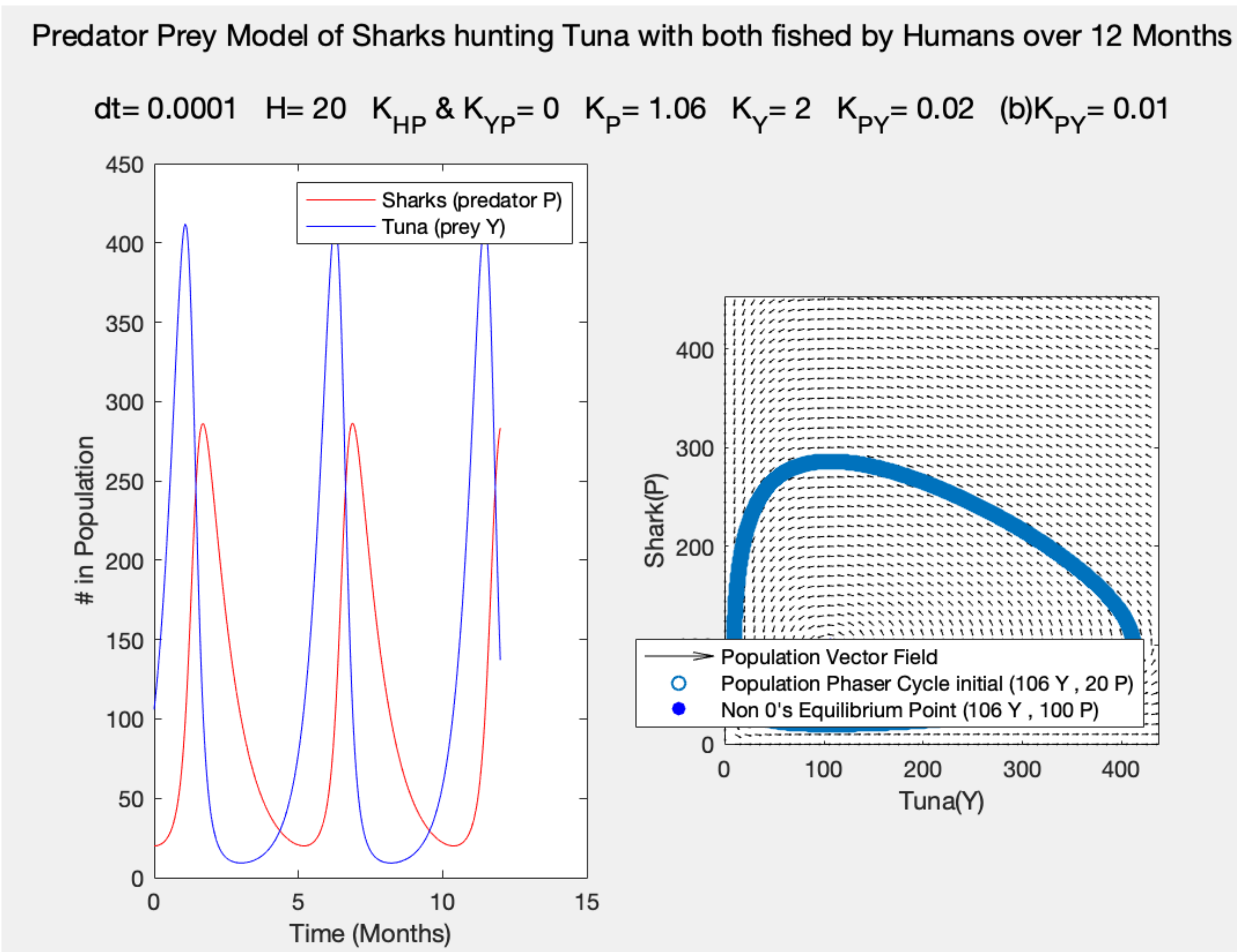

Humans(H)
200
100
0
[X,Y,Z] [150 150 100]
[U,V,W] [-0.9919 0.124 0.02
200
100
Sharks(P)
0
0
100
200
Tuna(Y)

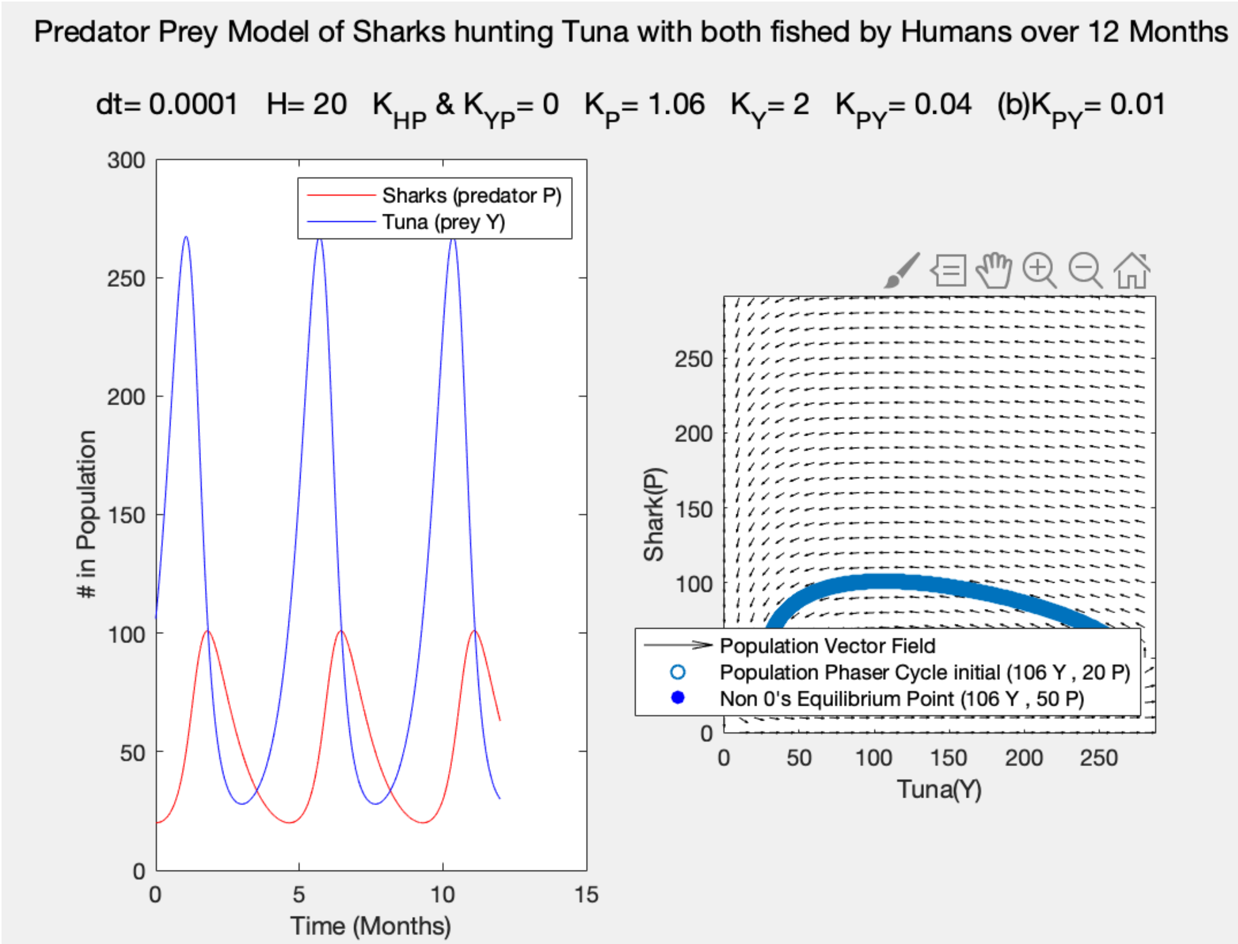

Predator Prey Model of Sharks Tuna and Humans over 120 Months
Pair Plots
dt= 0.0001 K_H= 0.02 K_PH= 0.01 K_YH= 0.03 K_Y= 2 K_PY= 0.02
X 97.65
Y 165
Tuna(Y)
Sharks(P)
Humans(H)
Tuna(Y)
Sharks(P)
Humans(H)

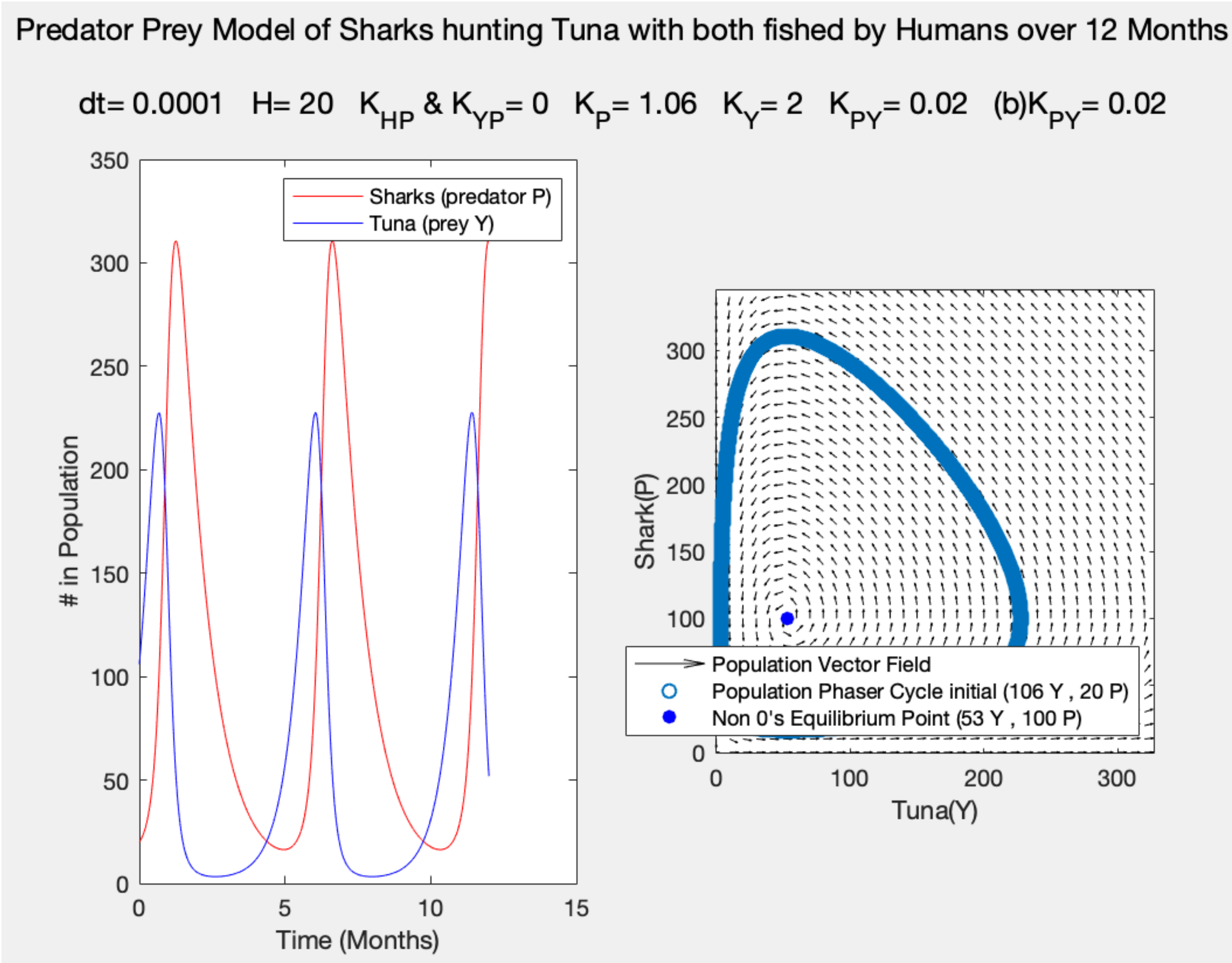
1c) Documentation

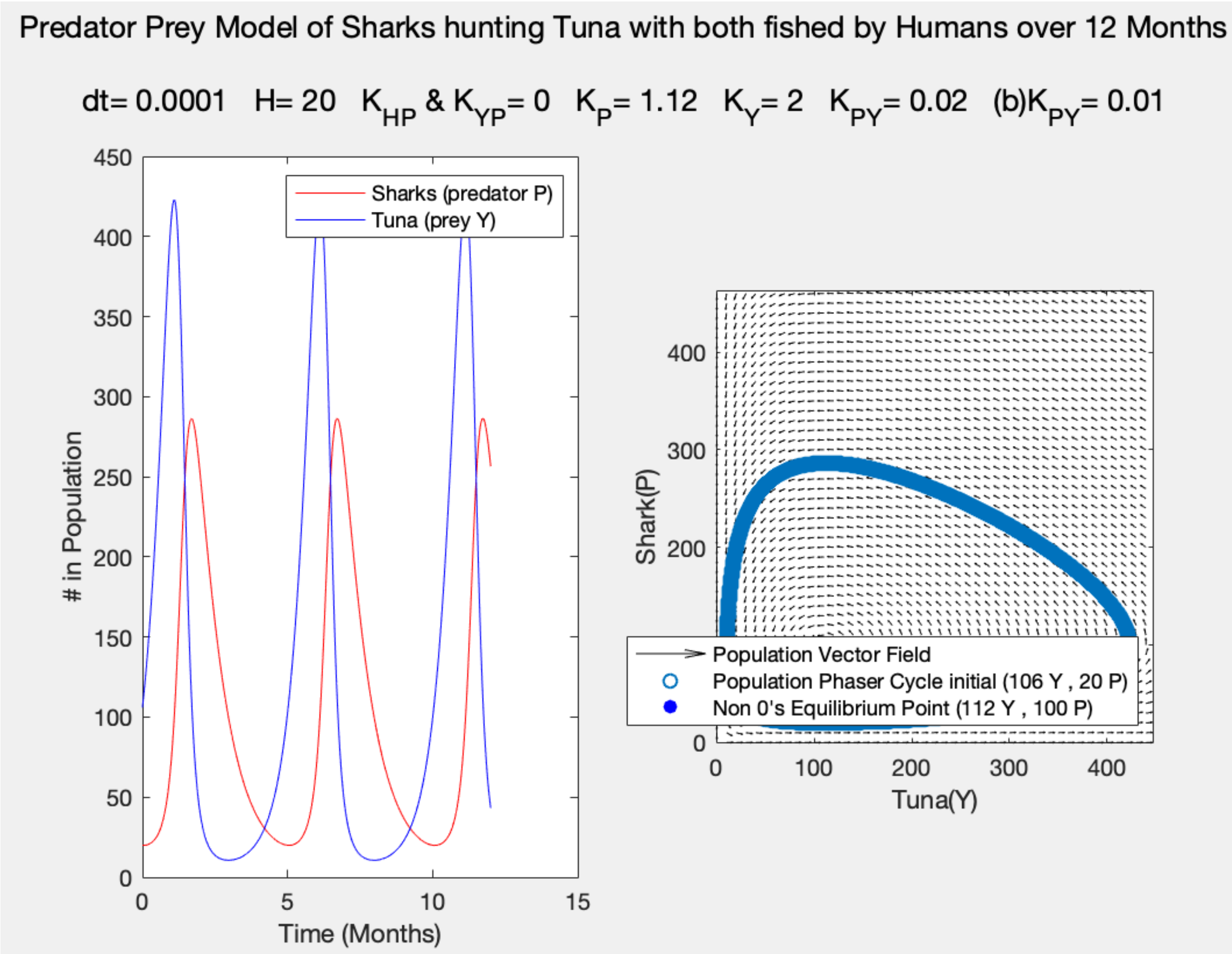
To demonstrate changes in variables, we'll have an original simulation and various alternative simulations. The values changed for each alternative simulation can be found at the top of the screenshot.

Original Simulation:

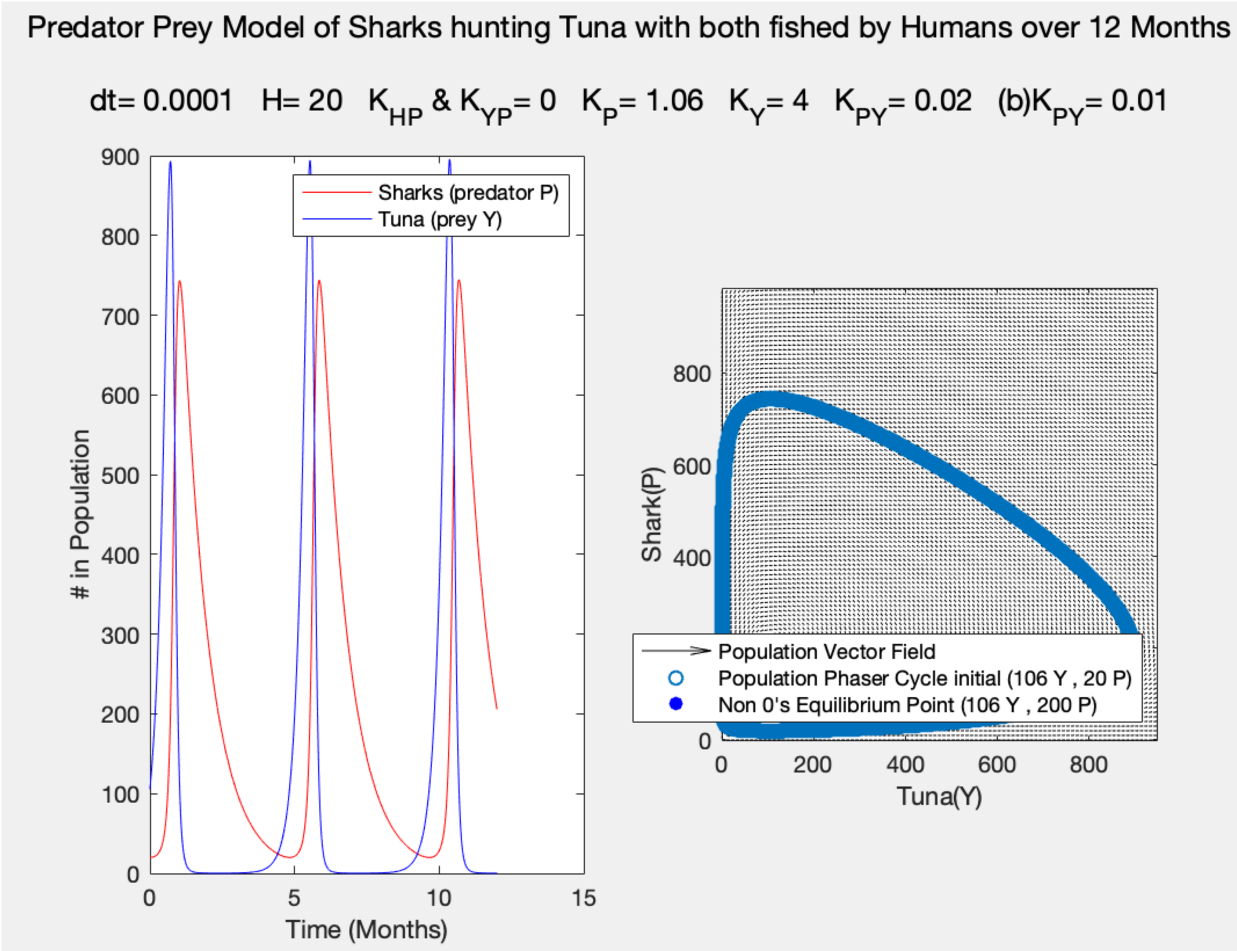


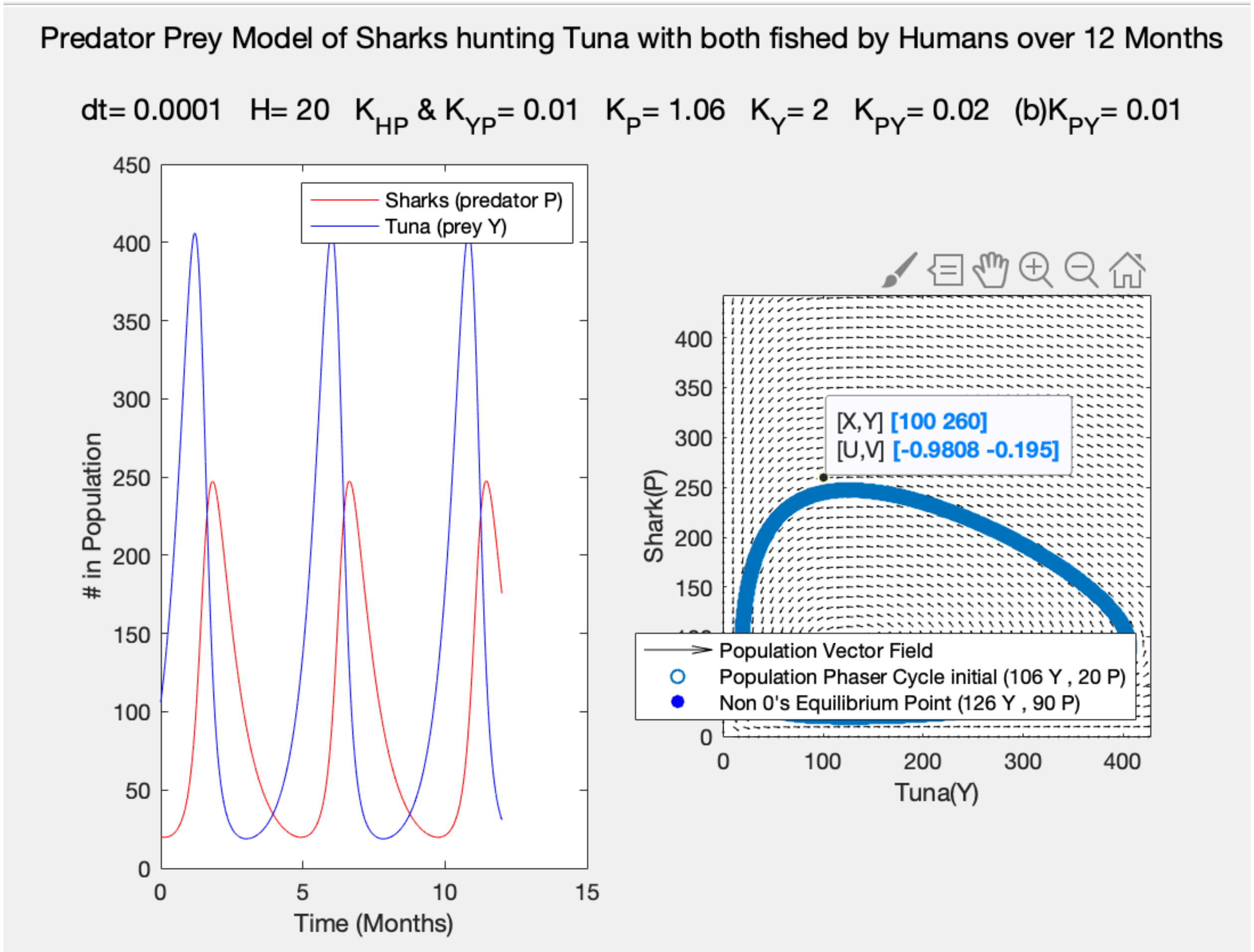

Predator Prey Model of Sharks hunting Tuna with both fished by Humans over 12 Months
dt= 0.0001 H= 20 K_{HP} & K_{YP}= 0 K_P= 1.06 K_Y= 2 K_{PY}= 0.04 (b)K_{PY}= 0.01
Sharks (predator P)
Tuna (prey Y)
in Population
Time (Months)
Shark(P)
Tuna(Y)
Population Vector Field
Population Phaser Cycle initial (106 Y , 20 P)
Non 0's Equilibrium Point (106 Y , 50 P)


Predator Prey Model of Sharks hunting Tuna with both fished by Humans over 12 Months
dt= 0.0001 H= 20 K_{HP} & K_{YP}= 0 K_P= 1.06 K_Y= 2 K_{PY}= 0.02 (b)K_{PY}= 0.02
Sharks (predator P)
Tuna (prey Y)
in Population
Time (Months)
Population Vector Field
Population Phaser Cycle initial (106 Y , 20 P)
Non 0's Equilibrium Point (53 Y , 100 P)
Shark(P)
Tuna(Y)

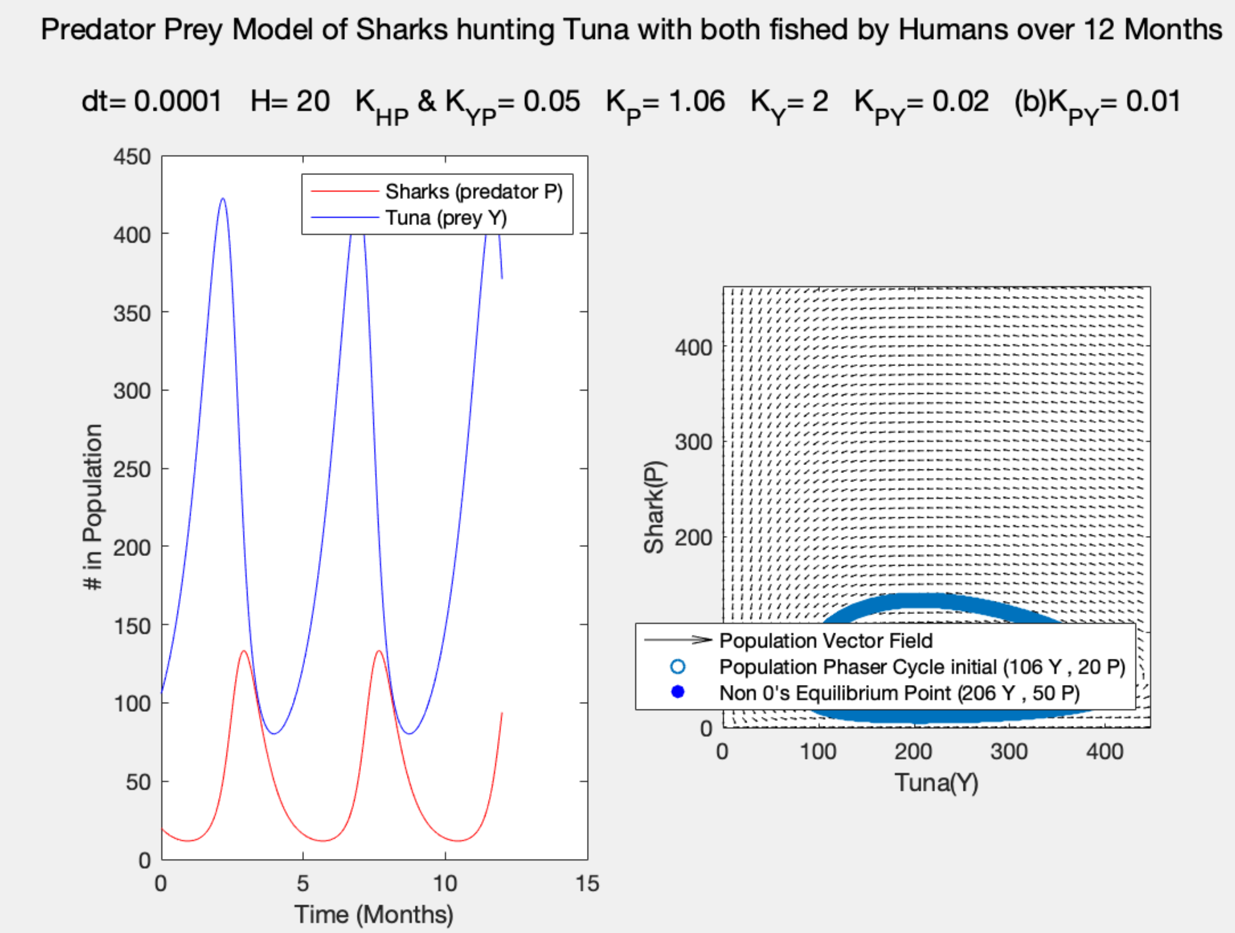


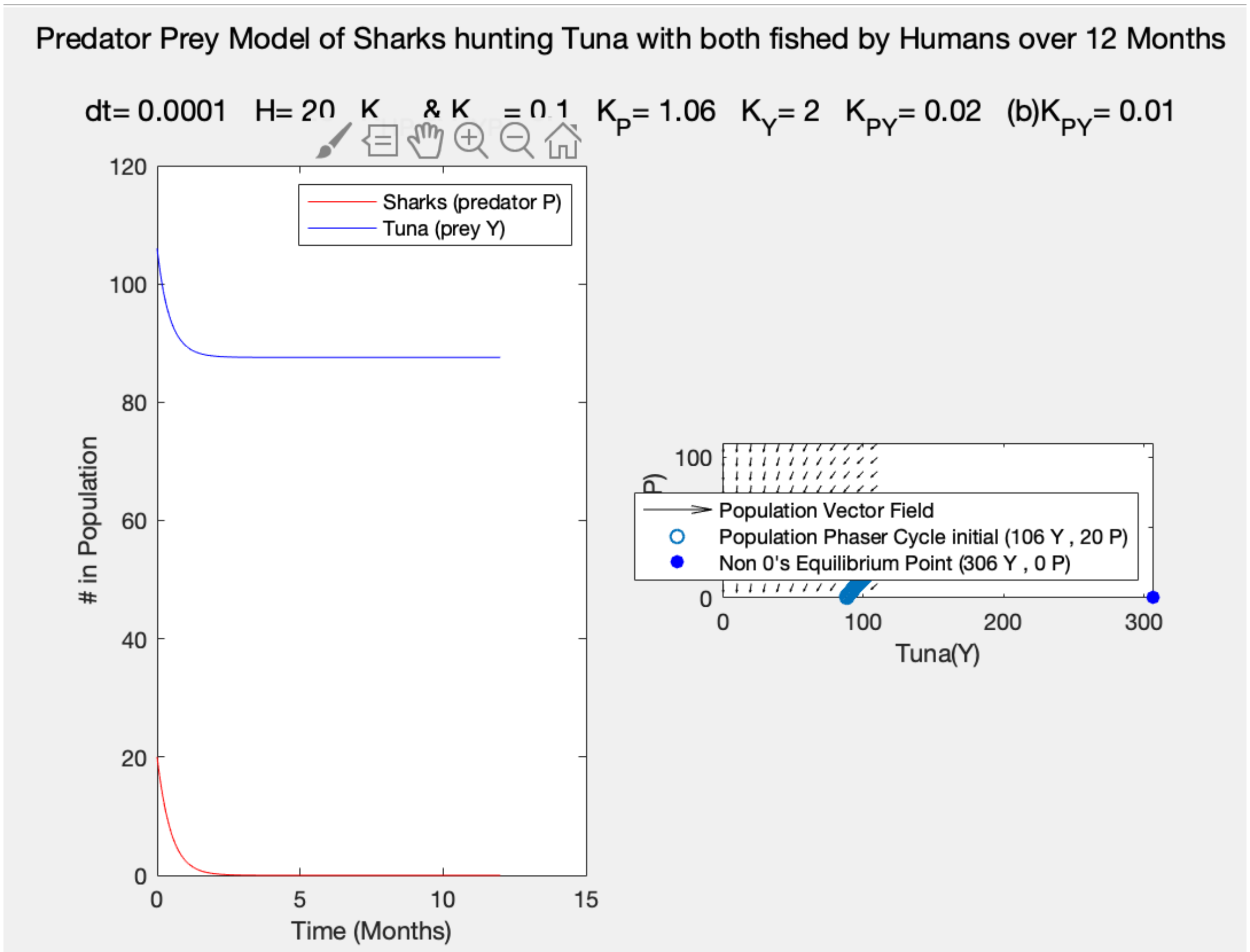
Predator Prey Model of Sharks hunting Tuna with both fished by Humans over 12 Months
dt= 0.0001 H= 20 K_HP & K_YP= 0 K_P= 1.12 K_Y= 2 K_PY= 0.02 (b)K_PY= 0.01
Sharks (predator P)
Tuna (prey Y)
in Population
Time (Months)
Population Vector Field
Population Phaser Cycle initial (106 Y , 20 P)
Non 0's Equilibrium Point (112 Y , 100 P)
Shark(P)
Tuna(Y)


Predator Prey Model of Sharks hunting Tuna with both fished by Humans over 12 Months
dt= 0.0001 H= 20 K_HP & K_YP= 0 K_P= 1.06 K_Y= 4 K_PY= 0.02 (b)K_PY= 0.01
Sharks (predator P)
Tuna (prey Y)
in Population
Time (Months)
900
800
700
600
500
400
300
200
100
0
5
10
15
Population Vector Field
Population Phaser Cycle initial (106 Y , 20 P)
Non 0's Equilibrium Point (106 Y , 200 P)
Shark(P)
Tuna(Y)
800
600
400
200
0

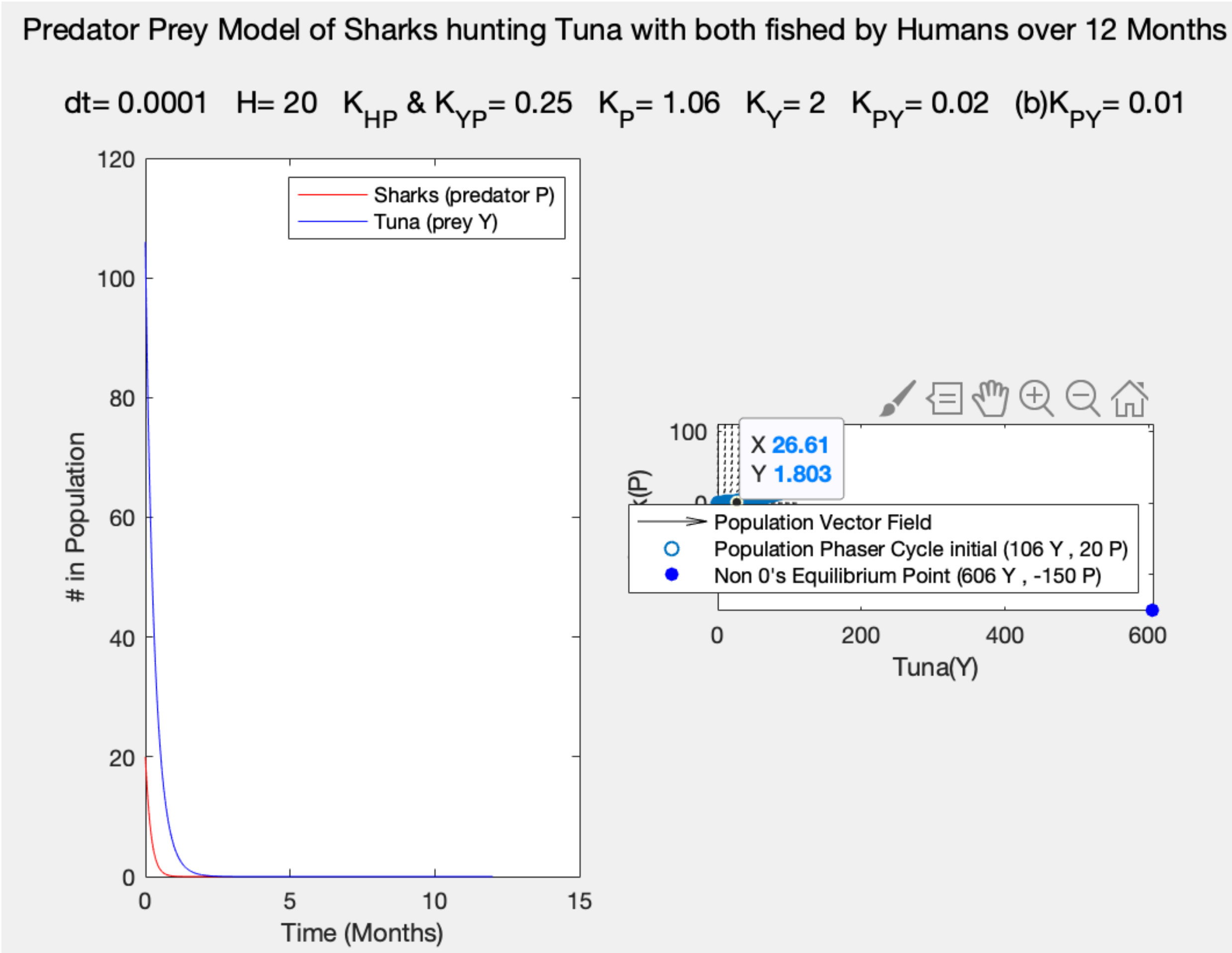


Predator Prey Model of Sharks hunting Tuna with both fished by Humans over 12 Months
dt= 0.0001 H= 20 K_HP & K_YP= 0.01 K_P= 1.06 K_Y= 2 K_PY= 0.02 (b)K_PY= 0.01
Sharks (predator P)
Tuna (prey Y)
in Population
Time (Months)
[X,Y] [100 260]
[U,V] [-0.9808 -0.195]
Shark(P)
Tuna(Y)
Population Vector Field
Population Phaser Cycle initial (106 Y , 20 P)
Non 0's Equilibrium Point (126 Y , 90 P)


Predator Prey Model of Sharks hunting Tuna with both fished by Humans over 12 Months
dt= 0.0001 H= 20 K_{HP} & K_{YP}= 0.05 K_P= 1.06 K_Y= 2 K_{PY}= 0.02 (b)K_{PY}= 0.01
Sharks (predator P)
Tuna (prey Y)
in Population
Time (Months)
Population Vector Field
Population Phaser Cycle initial (106 Y , 20 P)
Non 0's Equilibrium Point (206 Y , 50 P)
Shark(P)
Tuna(Y)



Predator Prey Model of Sharks hunting Tuna with both fished by Humans over 12 Months
dt= 0.0001 H= 20 K & K = 0.1 K_P= 1.06 K_Y= 2 K_PY= 0.02 (b)K_PY= 0.01
Sharks (predator P)
Tuna (prey Y)
in Population
Time (Months)
Population Vector Field
Population Phaser Cycle initial (106 Y , 20 P)
Non 0's Equilibrium Point (306 Y , 0 P)
Tuna(Y)


Predator Prey Model of Sharks hunting Tuna with both fished by Humans over 12 Months
dt= 0.0001 H= 20 K_{HP} & K_{YP}= 0.25 K_P= 1.06 K_Y= 2 K_{PY}= 0.02 (b)K_{PY}= 0.01
Sharks (predator P)
Tuna (prey Y)
in Population
Time (Months)
X 26.61
Y 1.803
Population Vector Field
Population Phaser Cycle initial (106 Y , 20 P)
Non 0's Equilibrium Point (606 Y , -150 P)
Tuna(Y)

1d) Reflection

Matt’s Reflection:

While looking over the code I wrote for the first model of just the tuna and shark populations changing versus the revised code that Jackson, Will and I came up with the most important change I needed to make to my original code was the readability of it using less comments and making sure that each line of code did not run off the ‘page’ with the use of ‘...’. I also could have made it more clear on how to toggle the additional graph I added to my model (the phase plane population cycle graph). I am happy that I added this second graph of the phase plane population cycle as it made it much more easy to understand how the two populations affect each other with the rate constants chosen. In the end it was a very helpful practice reviewing all our code again.

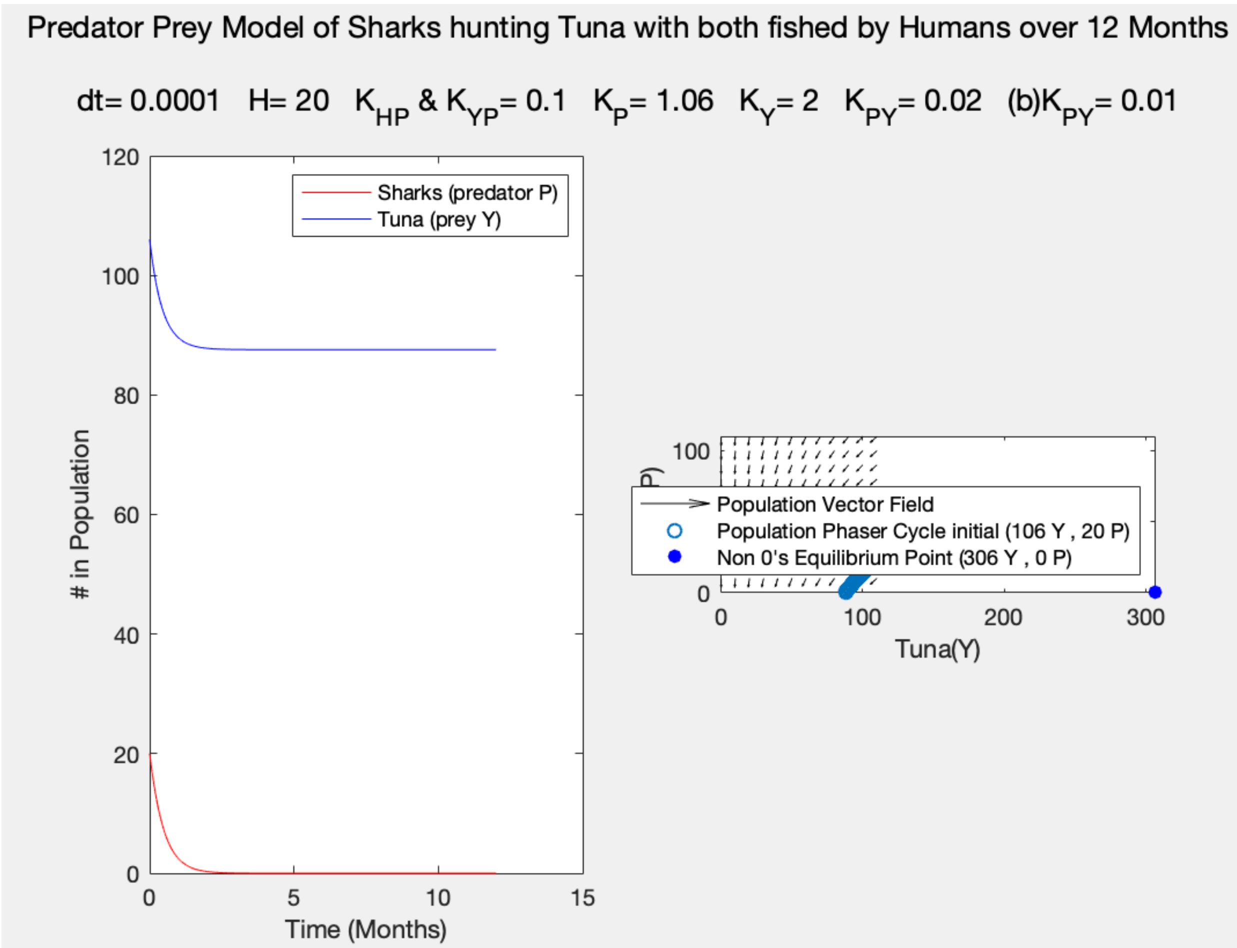
Jackson’s Reflection:

The most significant change for me between my original model and our revised model was the introduction of the vector mapping. The ability to visualize the equilibrium points helped me better understand the Lotka-Volterra model as a whole, as I was no longer algebraically manipulating equilibrium equations but rather was able to play with particular inputs. It was also helpful for me to learn of the formatting changes to prevent exceeding character limits and to increase readability. Finally, having the others notice my issue with my delta time value as well as Will’s issue with time length value helped me better understand how the discrete changes in time affects the model.

Will’s Reflection:

While comparing our code from the first part of the project, the biggest revision needed from my own code was based around the use of the ellipses ‘...’ in comments. In my original file, many comments run off the page of matlab. This affects the overall readability of the code, making it appear much more sloppy. Another point of style improvement I found when comparing was based around my documentation. The beginning of my file does not hold quite as much information when I should take the time to give a detailed description of my code in the documentation before beginning the simulation.

1e) Simulation



At a fishing rate of 0.10, sharks all die.
This is 5% of the tuna birth rate of 2.

2a) Documentation

We can start with our original equations from part one. The differences between the original equations and these from 2a are that humans are no longer hunting tuna and that the human population is no longer constant.

The only change in the birth equations is the introduction of a change in human births. Previously, it had been zero. Now it is a function of human's birth rate as a fraction of the interactions between the human population and the shark population.

Original Birth Equations:
tuna_births = tuna_population * tuna_birth_rate
shark_births = shark_population * tuna_population * shark_birth_proportionality_constant
human_births = 0

New Birth Equations
tuna_births = tuna_population * tuna_birth_rate
shark_births = shark_population * tuna_population * shark_birth_proportionality_constant
human_births = human_population * shark_population *
    human_birth_proportionality_constant_shark

Next is the death equations.
Here we must again introduce a change in human deaths as a function of human population and human death rate.
Furthemore, sharks are now being killed by humans as a function of human population, shark population, and shark fishing rate.
Also, humans are no longer fishing tuna in this model.

Original Death Equations:
tuna_deaths = (tuna_population * shark_population * tuna_death_proportionality_constant) +
    (tuna_population * human_population * fishing_rate)
shark_deaths = (shark_population * shark_death_rate) + (shark_population *
    human_population * fishing_rate)
human_deaths = 0

New Death Equations:
tuna_deaths = tuna_population * shark_population * tuna_death_proportionality_constant
shark_deaths = (shark_population * shark_death_rate) + (shark_population * human_population
    * fishing_rate_shark)
human_deaths = human_population * human_death_rate;

Finally we have the differential equations.
The only changes here are the underlying birth and death equations (already addressed) as well as the introduction of a non-constant human population.

Original Differential Equations

delta_tuna_population = (tuna_births - tuna_deaths) * dt;
delta_shark_population = (shark_births - shark_deaths) * dt;
delta_human_population = 0

New Differential Equations:
delta_tuna_population = (tuna_births - tuna_deaths) * dt;
delta_shark_population = (shark_births - shark_deaths) * dt;
delta_human_population = (human_births - human_deaths) * dt;

2b) Documentation

We can approach this similarly to 2a. The difference between 2a and 2b is that 2a introduces humans hunting tuna.

The only change in the birth equations is that humans are now hunting tuna, which factor into human's birth equation as a function of the human birth proportionality constant for tuna, the tuna population, and the human population.

2a) Birth Equations
tuna_births = tuna_population * tuna_birth_rate
shark_births = shark_population * tuna_population * shark_birth_proportionality_constant
human_births = human_population * shark_population *
    human_birth_proportionality_constant_shark

2b) Birth Equations
tuna_births = tuna_population * tuna_birth_rate
shark_births = shark_population * tuna_population * shark_birth_proportionality_constant
human_births = (human_population * shark_population *
    human_birth_proportionality_constant_sharks) + (human_population * tuna_population *
    human_birth_proportionality_constant_tuna)

The only change in the death equations is that humans are now hunting tuna, which factor into tuna's death equation as a function of the fishing rate for tuna, the tuna population, and the human population.

2a) Death Equations:
tuna_deaths = tuna_population * shark_population * tuna_death_proportionality_constant
shark_deaths = (shark_population * shark_death_rate) + (shark_population * human_population
    * fishing_rate_shark)
human_deaths = human_population * human_death_rate;

2b) Death Equations:
tuna_deaths = (tuna_population * shark_population * tuna_death_proportionality_constant) +
    (tuna_population * human_population * fishing_rate_tuna)
shark_deaths = (shark_population * shark_death_rate) + (shark_population * human_population
    * fishing_rate_shark)
human_deaths = human_population * human_death_rate;

There is no change in the differential equations between 2a) and 2b).

2a) Differential Equations:
delta_tuna_population = (tuna_births - tuna_deaths) * dt;
delta_shark_population = (shark_births - shark_deaths) * dt;

delta_human_population = (human_births - human_deaths) * dt;

2b) Differential Equations:
delta_tuna_population = (tuna_births - tuna_deaths) * dt;
delta_shark_population = (shark_births - shark_deaths) * dt;
delta_human_population = (human_births - human_deaths) * dt;

2c) Hypothesis Testing

For our hypotheses we chose the following:
Primary Hypothesis: Shark population continues to die out given equations defined in 2b. At which threshold will shark population survive as we increase the shark interaction term.
Competing Hypothesis: Regardless of an increase in the tuna_death_proportionality_constant, the population of sharks will continue to die out.

To run this simulation to test our hypotheses, we created a loop to run through and plot our model 6 times. At the end of each iteration we increment the value of the tuna_death_proportionality_constant by 0.005 while resetting all initial population values. The results can be seen in the 6 figures below:

Figure 1:

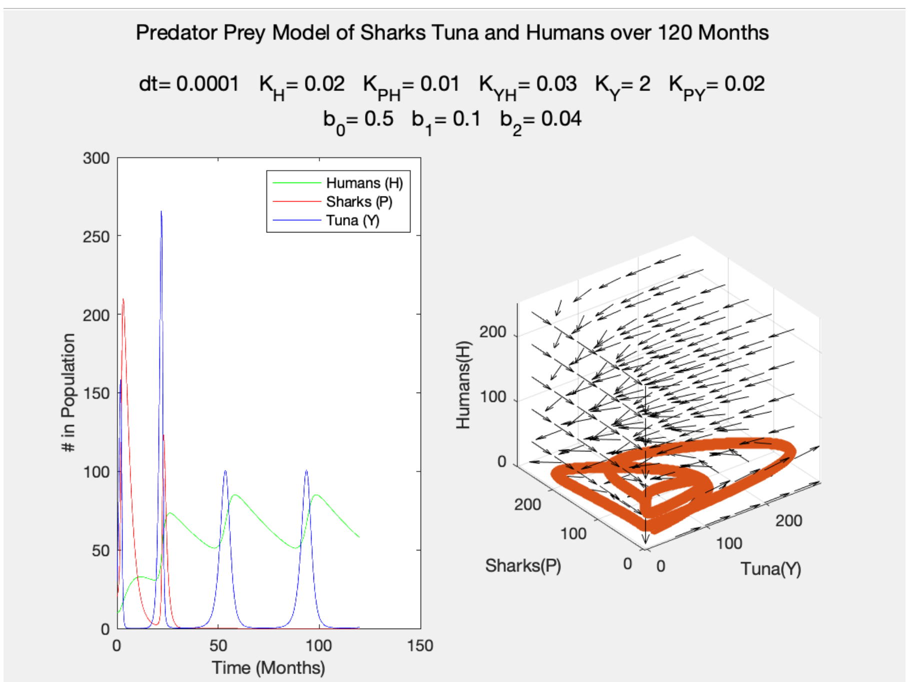


Figure 2:

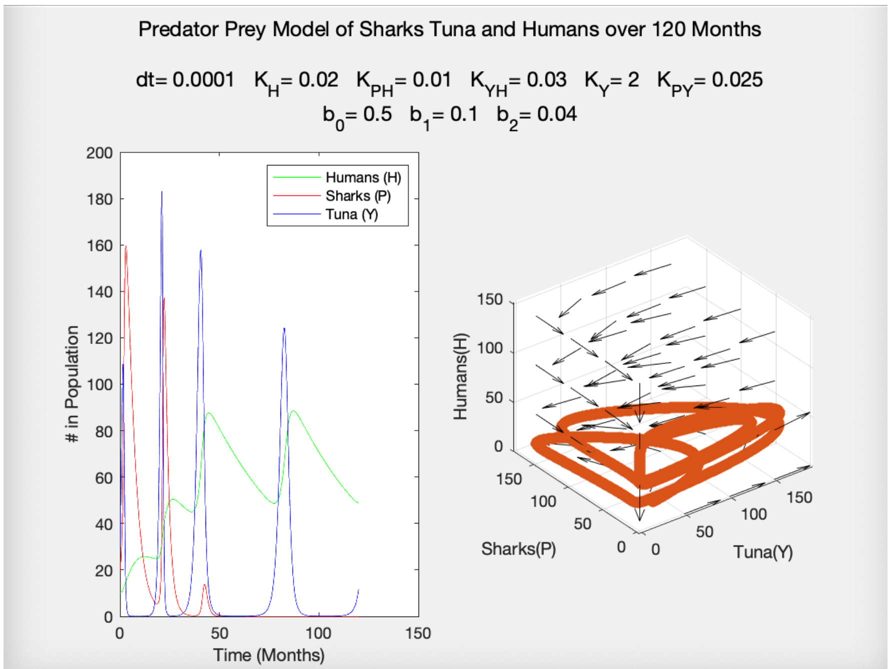


Figure 3:

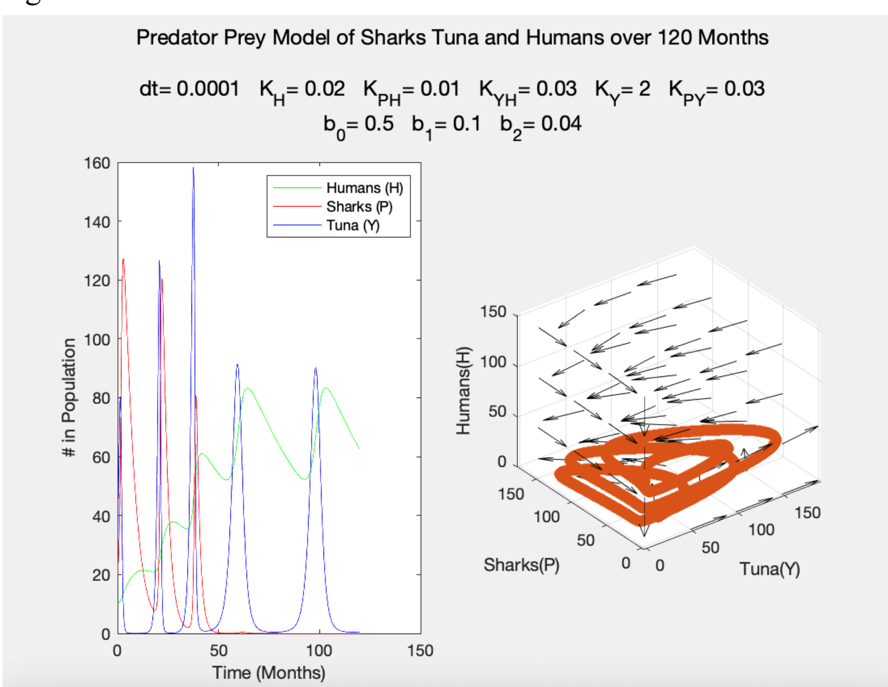


Figure 4:

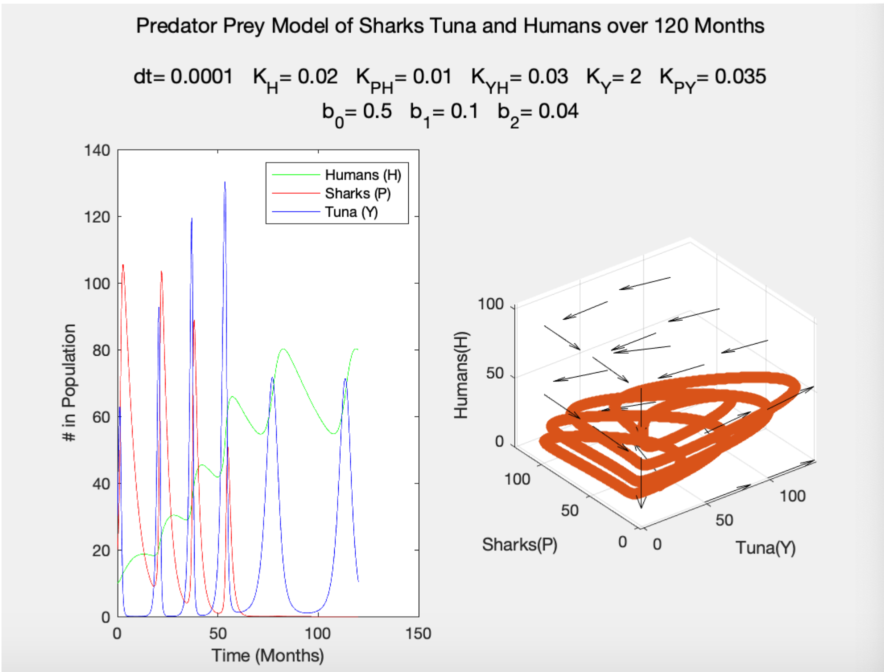


Figure 5:

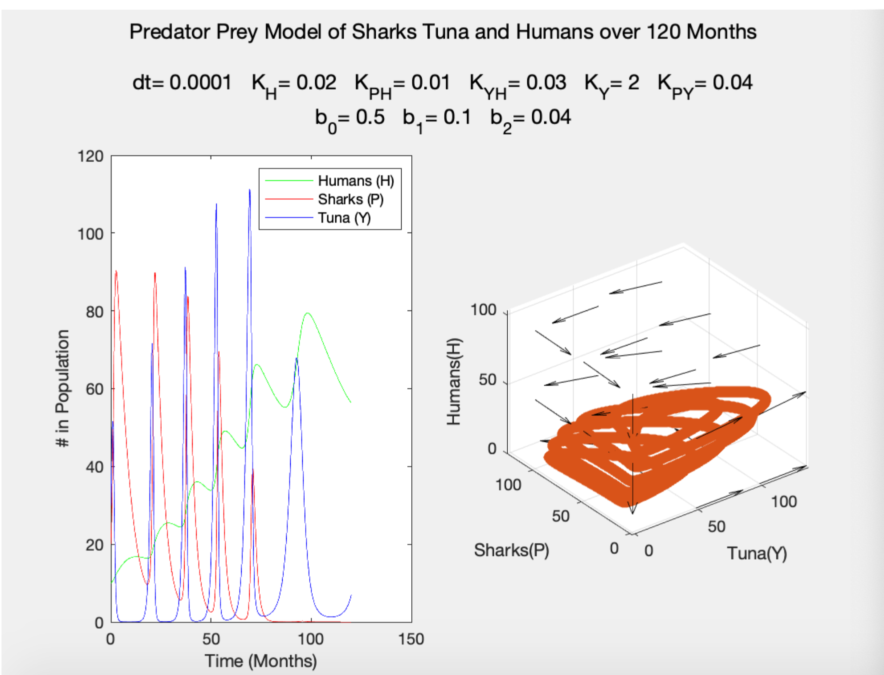
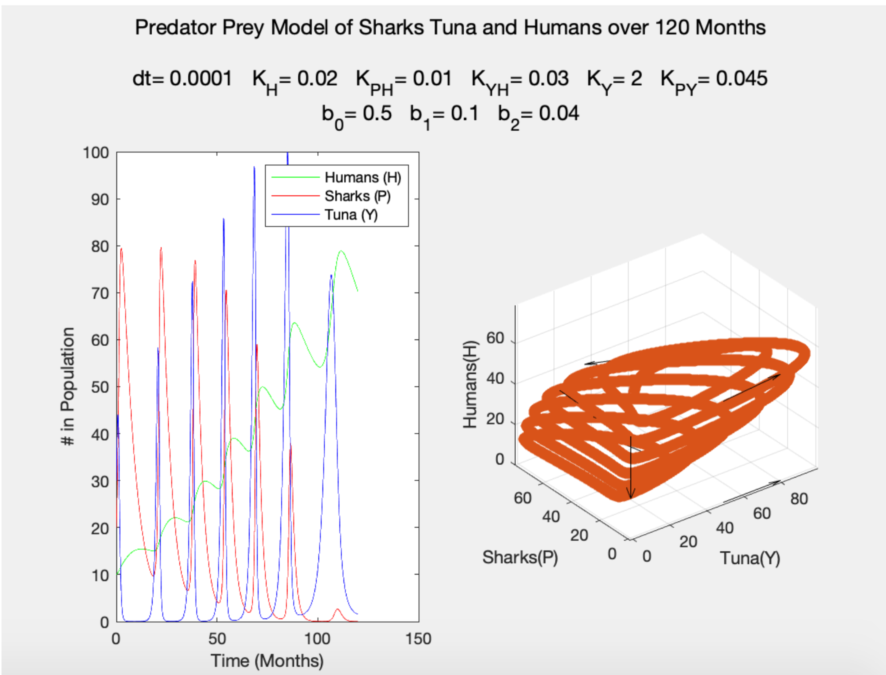


Figure 6:



To run this hypothesis simulation run the mbass21_jbpret22_wcjohn22_hw2_2e.m file. Based on the data, we conclude we would reject the primary hypothesis and support the competing hypothesis. As we can see, when we increase the tuna_death_proportionality_constant, we see that, while there are more fluctuations in the shark population, eventually the population still dies out regardless of the increase in the tuna_death_proportionality_constant