

Predator-Prey Model

Connor Hann, Xiaomeng Jia, Peifan Liu and Xinyu Wu¹

¹*Physics Department, Duke University*

(Dated: April 8, 2016)

Abstract

In this article, a classical predator-prey model is numerically explored upon several given rules. By moving a random number of sharks and fish on a 2D torus lattice and imitating the real-world rules that fish swim and breed offspring, sharks hunt fish, breed and starve to death for a given time, we dynamically show the revolution of populations of fish and sharks. It is shown that given proper initial conditions, they experience stable periodic solutions which look similar to the solutions of the Lotka-Volterra Model. For this simulation, however, a modified Lotka-Volterra Model is used due to the limited capacity of the lattice to describe the dynamics of these two species. (Github link: <https://github.com/xiaoao-duke/Group-Project-2.git>)

BACKGROUND

Modeling the interactions between predator and prey is a question of great ecological importance, as accurate models can allow one to make reliable predictions and thereby inform decisions in conservation, habitat preservation, hunting regulations, etc.

CLASSICAL LOTKA-VOLTERRA MODEL

There exist a variety of methods for modeling the interactions between predator and prey, and these models, though simple, are often in quite good agreement with what is observed. Take the classic Lotka-Volterra Model as an example. While this model is nothing more than a set of first-order nonlinear differential equations that one may solve numerically for a particular set of initial conditions, the model is capable of providing a qualitatively accurate description of the data, as is shown in Fig. [1](#).

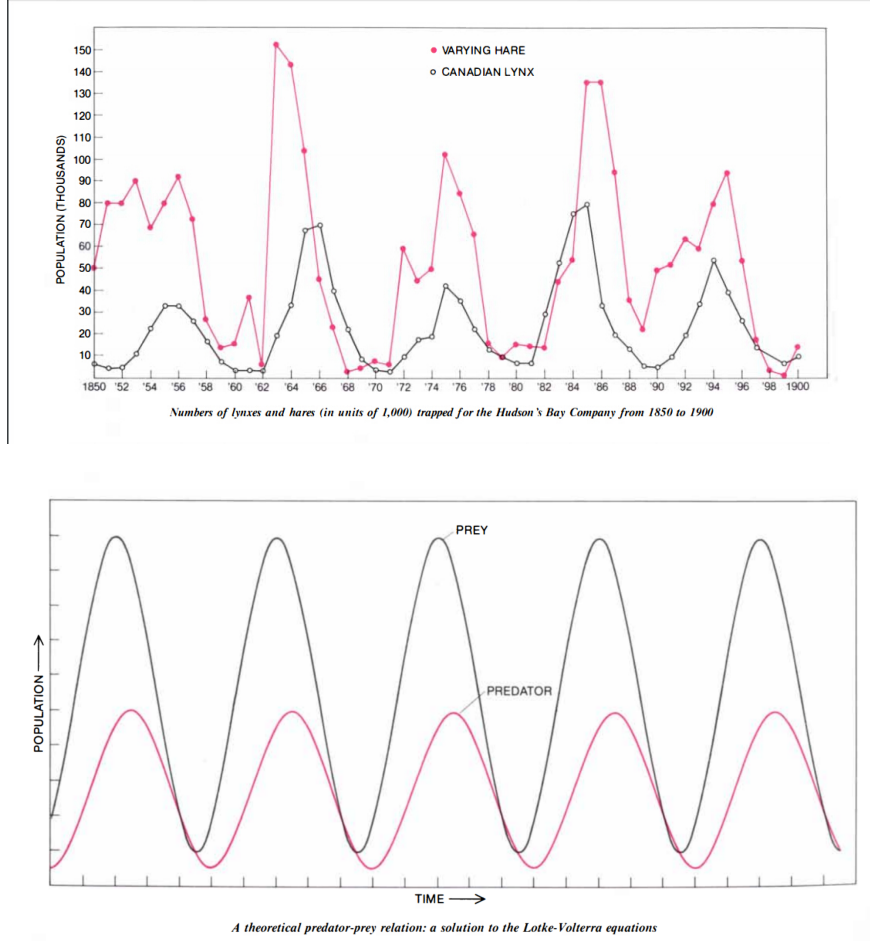


FIG. 1: (Top) experimental data of Hare and Lynx populations, (Bottom) predictions from the Lotka-Volterra equations.

In this work we study the behavior of a slightly different model, one that is similar to the Lotka-Volterra Model in that it consists of only a simple set of rules, but the nature of the actual simulation is quite different. In our case, we consider two populations, sharks and fish living on a two-dimensional lattice. We simulate the movement and behavior of each shark and fish individually and can observe fluctuations of the two populations as a function of time. Unlike the Lotka-Volterra Model, however, we are able to see exactly where all of the sharks and fish are and this allows us to attain a deeper understanding of what causes the fluctuations in populations and the conditions that may lead to extinction. The classical Lotka-Volterra Model is given by:

$$x' = x(\alpha - \beta y) \quad (1)$$

$$y' = -y(\gamma - \delta x) \quad (2)$$

where αx is the growth term which leads to exponential growth of prey population in absence of predators; $-\beta yx$ is the loss term which depends both on numbers of predators y and number of prey x ; $-\gamma y$ is the exponential decay term for predators; δxy is the gain term for predators which depends on populations of both species. The Lotka-Volterra Model has periodic solutions where peaks of predator population shifts $\pi/2$ with respect to the prey population.

IMPLEMENTATION

The model we consider is remarkably simple in that there are only five user input parameters: the initial number of sharks *n0_sharks*, initial number of fish *n0_fish*, the time it takes fish to breed *breed_age_Fish*, the time it takes sharks to breed *breed_age_Shark*, and the time it takes a shark to starve to death *starve_time*. To begin the simulation, one creates *n0_sharks* sharks and *n0_fish* fish at random locations on a two-dimensional lattice, and each fish or shark is given a random age between *breed_age_Fish* and *breed_age_Shark*, respectively. At each subsequent time step of the simulation both sharks and fish will swim around the grid, and the sharks will attempt to eat the fish. Below we discuss the specific rules that dictate the movements of the fish and sharks.

Fish Movement

During a time step, each fish looks at the four adjacent lattice sites and randomly chooses an empty site to which to move. If no lattice sites are available, the fish remains in its current position. We implement periodic boundary conditions so that if a fish moves through one boundary it reappears on the boundary on the opposite side, i.e. our two-dimensional grid is actually a topological torus. The age of the fish is increased by one in each time step, regardless it moves or not. If the age of the fish reaches *breed_age_Fish*, an additional fish is created at the site the fish has just moved from, and the ages of both fish are set to 0.

Shark Movement

The goal of the sharks is to eat fish, so during a time step each shark will look for fish on the adjacent lattice sites. If fish are found, the shark chooses one randomly and moves to the fish's position on the lattice, deleting and "eating" the fish. If no fish are available the sharks move just like the fish, choosing an available lattice site to move to randomly. Like the fish, the age of the sharks is increased by one each time step, and sharks breed in the same way as the fish once they have reached *breed_age_Shark*. Unlike the fish, however, the sharks will starve if they do not find food. An array keeps track of how long it has been since each shark has last eaten a fish, and if a shark has not found a fish within *starve_time* time steps, it "starves" to death and is deleted from the simulation.

Program Structure

We use a set of five arrays to keep track of the information necessary to run the simulation as described above. Two $N \times N$ arrays, *Sharks* and *Fish*, are used to keep track of the positions of the sharks and fish respectively. If the (i, j) site is empty the arrays hold the value -1 , otherwise the site holds the age of the shark or fish. Two additional arrays *Sharkmove* and *Fishmove* are used to store the new positions of the sharks and fish as an update is being performed, allowing us to ensure that each shark or fish is moved only once and that two animals are not moved onto the same lattice site. Finally, a fifth array *Sharkstarve* stores the amount of time since each shark has last eaten. The arrays are updated, first fish then sharks, via the rules above. One can plot the positions of the sharks and fish at any time step, and an example plot is shown in Fig. 2.

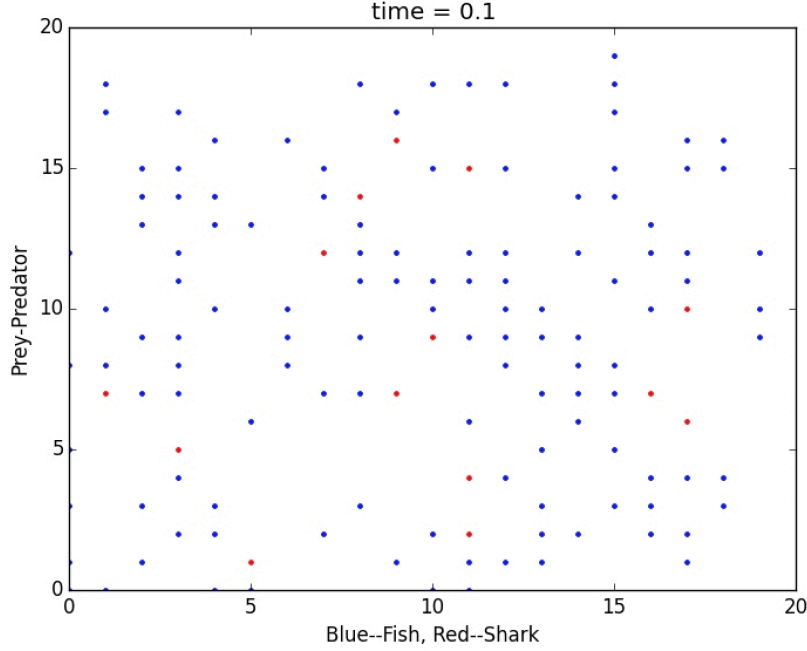


FIG. 2: A typical configuration of sharks and fish on a 20×20 lattice

NUMERICAL RESULTS

By choosing a 100×100 2D torus grid where 2000 fish and 200 sharks of random ages are randomly put as initial condition, it can be seen that as time goes on, the populations of fish and shark experience periodic variations. The number of fish increases most significantly when number of sharks reaches a valley, and decreases most sharply when shark reaches its peaks, as shown in Fig.3. The phase plane is shown in Fig.4. If we trace the dynamics of the system in each time step, it can be seen that sometimes fish fill out the whole grid and sometimes become almost extinct (Fig.6-9). This reminds us that the capacity of the grid is the upper constraint for the fish number, so the classical Lotka-Volterra equations must be modified to adapt to this simulation. Instead of exponential growth, the fish number should be described as logistic, and so does the sharks:

$$x' = x(A - Bx - Cy) \quad (3)$$

$$y' = -y(D - Ey - Fx) \quad (4)$$

which gives the analytic phase diagram (Fig.5) that properly coincide with simulation result (Fig.4).

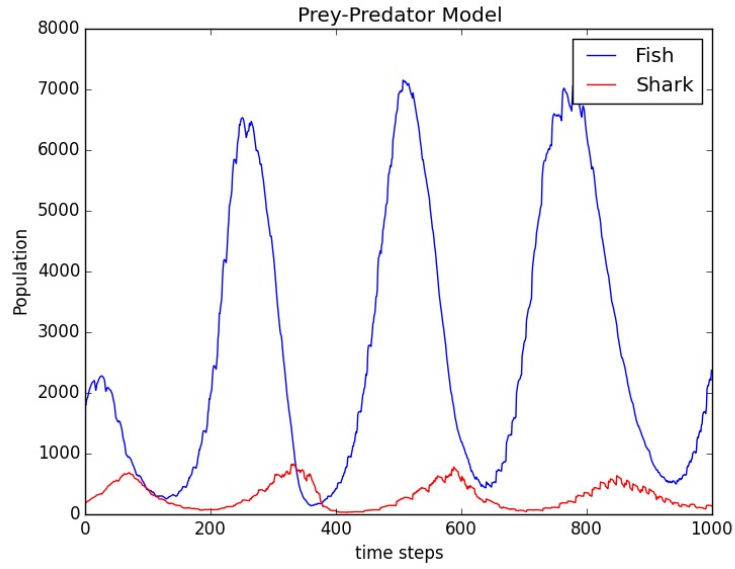


FIG. 3: Population evolution of fish and sharks.

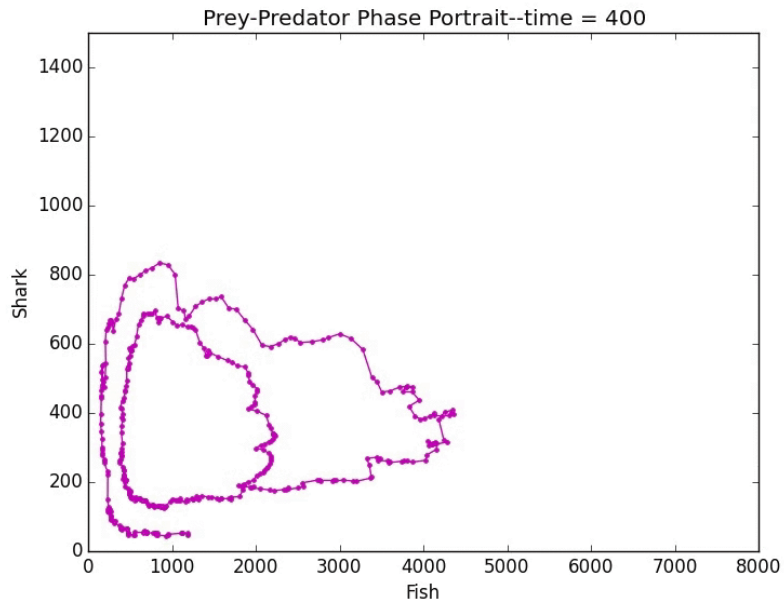


FIG. 4: Population evolution of fish and sharks in phase space.

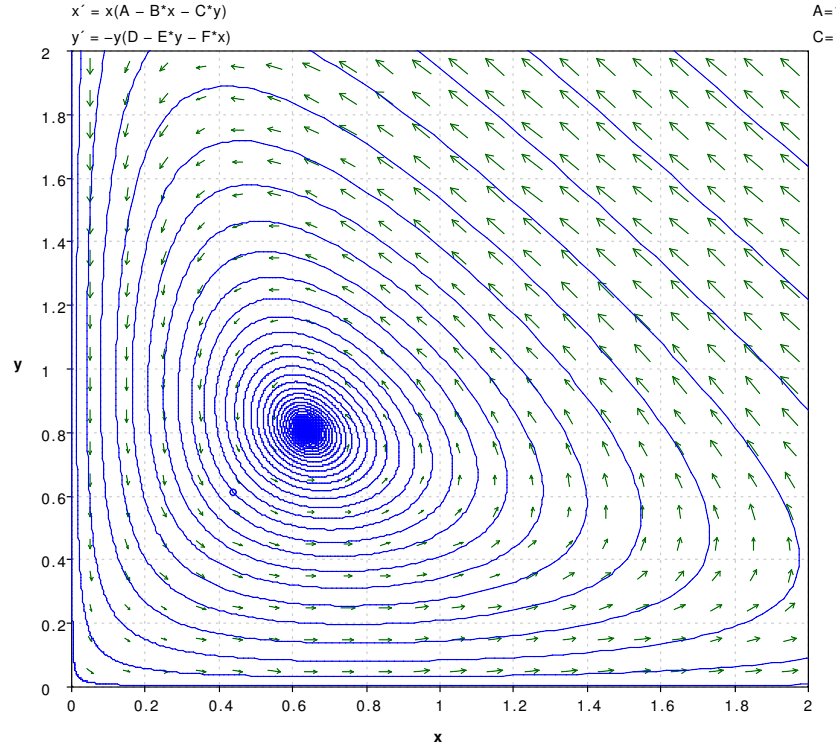


FIG. 5: Phase diagram of modified Lotka-Volterra equations.

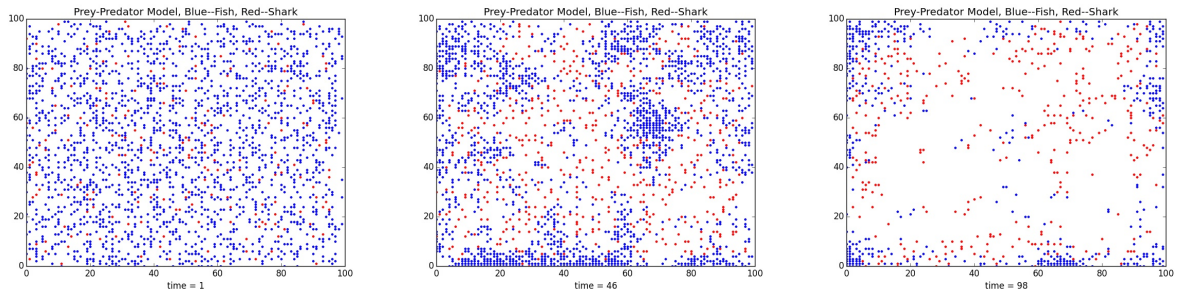


FIG. 6: Population evolution of fish and sharks.

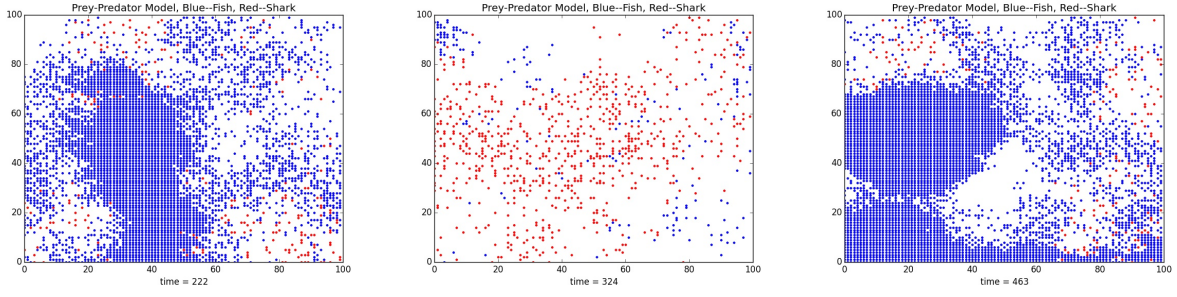


FIG. 7: Population evolution of fish and sharks.

This initial condition gives a stable periodic revolution. However, if initial conditions are far off the stable point, extinction of sharks or fish can be observed (Fig.8-9).

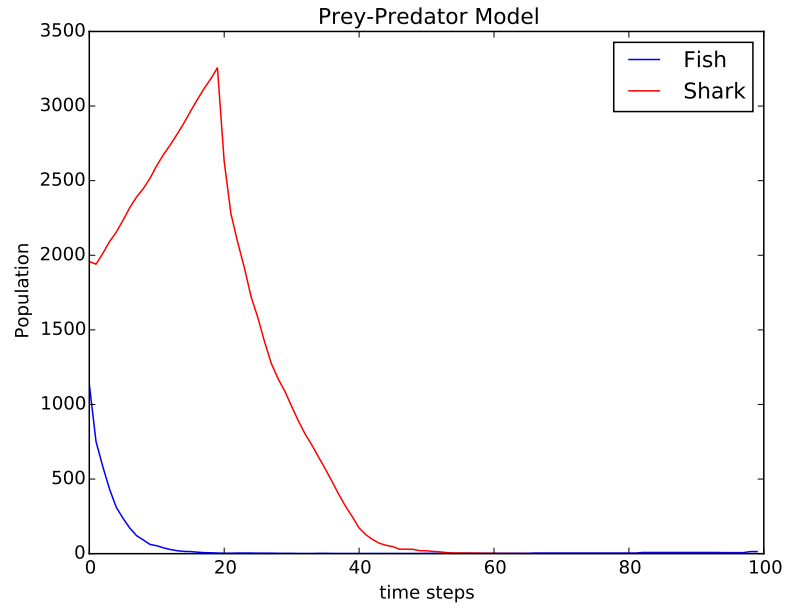


FIG. 8: Extinction of both fish and sharks.

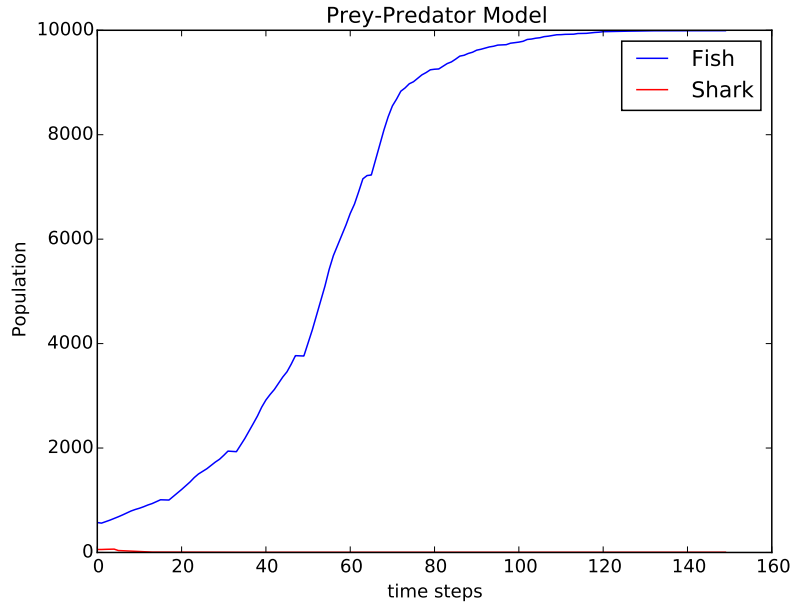


FIG. 9: Extinction of sharks.

CONCLUSION

The shark and fish predator-prey model we have described above possesses many qualities of the Lotkka-Volterra model, such as the ability of each species to go extinct and the ability to oscillate around an equilibrium. Indeed, we have shown that the results of the simulation, when considered in phase space, are in good agreement with a modified version of the Lotkka-Volterra model. What distinguishes our model, however, is that we are able to observe the individual members of the species in real time, allowing us to gain a better intuition for why fluctuations in populations occur in the first place, and why certain initial conditions are liable to produce mass extinctions.