Investigating Lotka-Volterra model using Runge-Kutta 4^{th} order method

Thinh Nguyen[1]

[1]Department of Computer Science, University of Dallas, Irving, TX USA

Abstract

In this research project we take a look at the Lotka-Volterra model, a population prediction model. This model predicts the dynamics of biological systems in which two species interact, one is the predator and one is the prey. We used the Runge-Kutta 4^{th} order algorithm to simulate this model. The Runge-Kutta 4^{th} order algorithm is the most efficient method to solve ordinary differential equations. We used this algorithm to simulate two populations (tigers and deers) based on the Lotka-Volterra model, and traced their behaviour over time. We also optimised the parameters that go with the model to get results that are as close to real data as possible.

Introduction

The Lotka-Volterra equations, also known as predator-prey equations, are a pair of first order nonlinear differential equations, used to describe the dynamics of biological systems, where 2 species interact, 1 as the predator and 1 as the prey. The population changes according to the pair of equations:

$$\frac{dx}{dt} = \alpha x - \beta xy,$$
$$\frac{dy}{dt} = \delta xy - \gamma y,$$

There are 6 parameters that come with the model: α, β, x, y, δ and γ. The populations of each species are denoted using parameters x and y, with x is the population of the prey, and y is the population of the predator. The rates in which each species reproduces and dies are denoted with parameter α, β, δ and γ, with α is the natural birth rate of the prey and β is the death rate of the predator in the presence of the predator. Δ is the birth rate of the predator in the presence of the prey and γ is the natural death rate of the predator. (1)

Procedures

We utilised the Runge-Kutta 4^{th} order method to simulate this model. The Runge-Kutta 4^{th} order method is as shown below:

Fourth order RK method

- The fourth order Runge-Kutta yields:

$$\hat{y}_{k+1} = \hat{y}_k + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

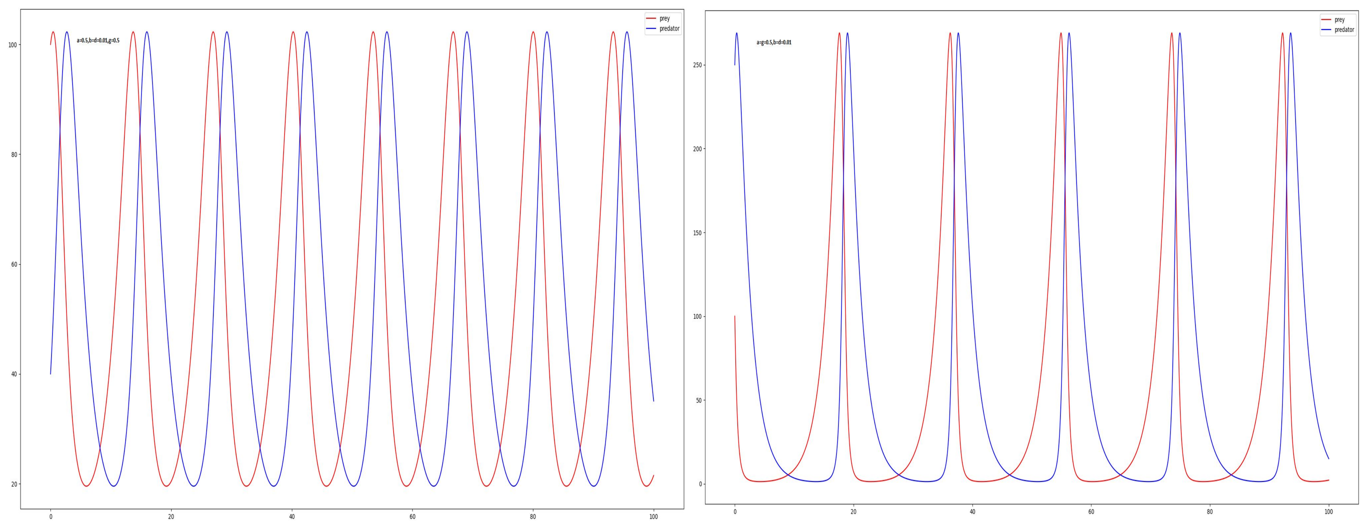
- where

$$k_1 = hf(t_k, \hat{y}_k)$$
$$k_2 = hf(t_k + \frac{h}{2}, \hat{y}_k + \frac{k_1}{2})$$
$$k_3 = hf(t_k + \frac{h}{2}, \hat{y}_k + \frac{k_2}{2})$$
$$k_4 = hf(t_k + h, \hat{y}_k + k_3)$$

The Runge-Kutta 4^{th} order method is the most widely used method to estimates differential equations because it offers a good balance between order o … on. (2)

```
#lotka_volterra.py
#using RK4 to solve the lotka_volterra equations and plot
#2 populations:

import numpy as np
import matplotlib.pyplot as plt
#init parameters

alpha = 0.5; #deer birth rate (prey)
beta = 0.01; #deer death rate (prey)
x = 100; #deer population (prey)
y = 40; #tiger population (predator)
delta = 0.01; #tiger birth rate (predator)
gamma = 0.5; #tiger death rate (predator)
t = 0; #time
h = 0.01; #time step
pc = []
pc.append(x)
Pc = []
Pc.append(y)
tc = []
tc.append(t)
#prey population function
def prey(x, y):
    return alpha*x - beta*x*y

#predator population function
def predator(x, y):
    return delta*x*y - gamma*y


while t <= 100:


    k1p = prey(x, y)*h
    k1P = predator(x, y)*h

    k2p = prey(x + k1p/2, y + k1P/2)*h
    k2P = predator(x + k1p/2, y + k1P/2)*h

    k3p = prey(x + k2p/2, y + k2P/2)*h
    k3P = predator(x + k2p/2, y + k2P/2)*h

    k4p = prey(x+ k3p, y+ k3P)*h
    k4P = predator(x+ k3p, y+ k3P)*h

    x += (k1p + 2*k2p + 2*k3p + k4p)/6
    y += (k1P + 2*k2P + 2*k3P + k4P)/6
    pc = np.append(pc, x)
    Pc = np.append(Pc, y)
    t+=h
    tc= np.append(tc, t)
```

Analysis



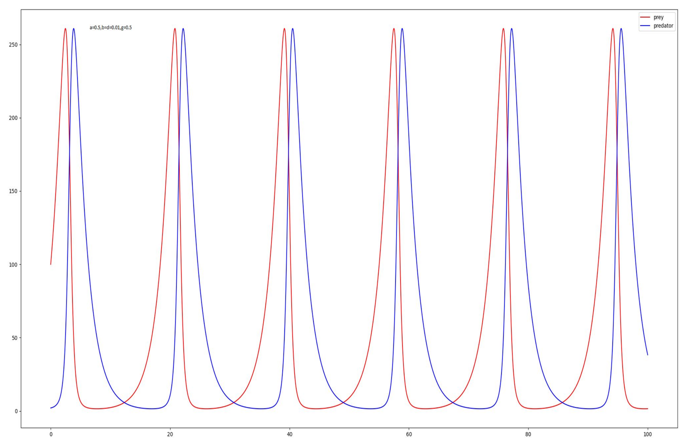


Figure 3. *Lotka-Volterra model predicting the population cycle between tigers and deers in the same ecosystem, with initial populations x=100, y=40; x=100, y = 250; and finally x=100, y=2. All parameters stay constant with α=0.5, β=0.01, δ=0.01 and γ=0.5.*

For x=100 and y=40, the graphs have a sine pattern in which one species' population goes down, the others rises shortly after, and the maximum capacity of each species stay equal over time, and the period in which the populations rise and fall also are the same over time. However, when y=2, we obtain a different graph, however it still has a cycle of each species' population rising and falling, with the period now changed. It takes longer for the populations to rise and fall when the population of the predator massively outnumbers the population of the prey, and the same phenomenon occurs when the population of the predator now massively underweight the population of the prey. The total population for each species also increases by a significant amount. This happens because when the population of the predator is reasonably lower than the prey, there are more prey being consumed by the predator, thus making the population of the prey has less opportunities to rise. With the predator population significantly less than the prey, there are more opportunities for the prey to reproduce, and thus, more food for the predator to consume, leads to the more explosive rise of both populations. The last case with the predator population significantly outnumbers the prey population, the predator will have a very small increase initially and a drastic decrease shortly after, due to the lack of prey to consume. Because there are now much less predator, the prey population has the chance to rise, and in the absence of predator, the prey population massively increases, and since there are now much more prey, the graph now follows the previous graph pattern. (3)

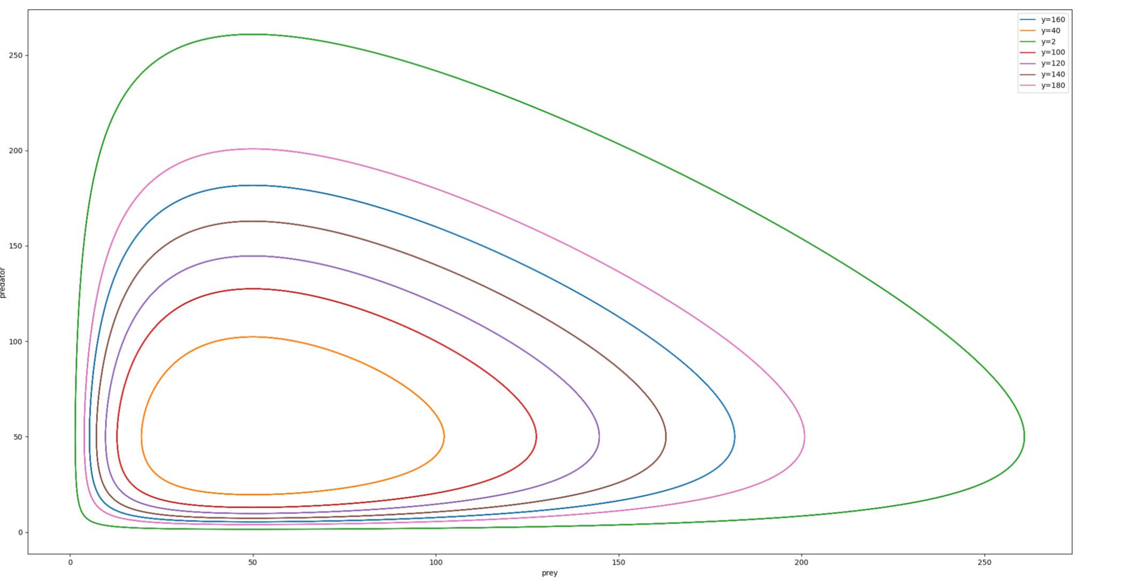


Figure 4. *Phase space diagram for each predator initial population variable change.*

With each different initial population of the predator, we get a different phase space graph. However, as shown in Figure 4, they all share the same shape. This is because of the nature of the model in which the population of one species limits the growth of the other, and therefore, its phase space diagram has a "loop" shape.

Discussion

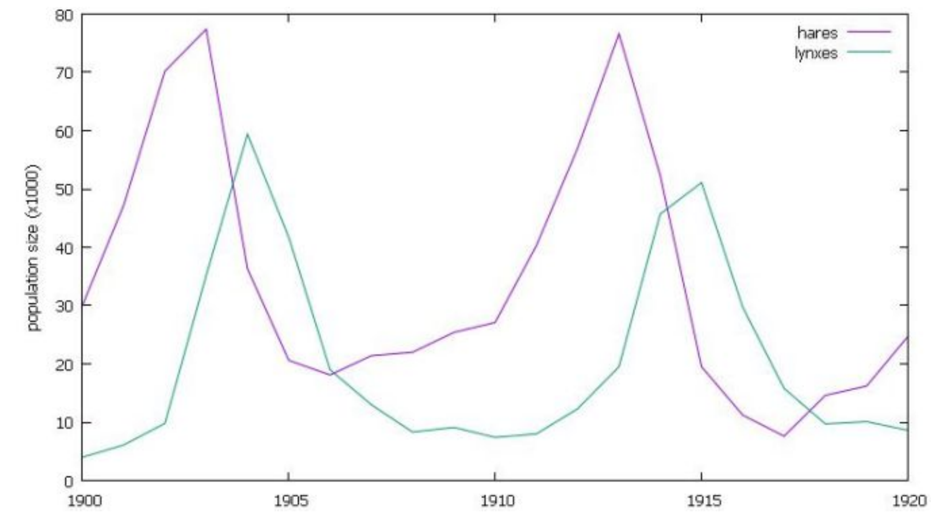


Figure 6. *Actual population graph of lynx and snowshoe hares from the Hudson Bay Company, Canada over the period 1900-1920 (Gilpin, 1973).*

The Lotka-Volterra model is the simplest model of predator-prey interactions, therefore, it does not take into account realistic events bound to happen when 2 species interact such as the predator stalking, chasing and the time it takes for each species to actually make contact, and thus, it does not produce a very accurate depiction of the realistic dynamics. However, it is still a pretty useful model to predict the dynamics of the population for each species because it does a perfect job in predicting the dynamics of the population between 2 species with perfect conditions: every event is instantaneous and no natural calamities such as diseases. The graph shown in Figure 6 even though is definitely not the same as the graphs predicting the population trends using the Lotka-Volterra model, it still follows the same dynamics pattern as the graphs using the Lotka-Volterra model.

Conclusions

In this project we have demonstrated that complex processes and phenomena in nature can be modelled using simple mathematical models. Along with these simple mathematical models, using appropriate parameters for such models also yields success in studying these complex systems and phenomena. In this project, we looked at two populations in which fit the predator and prey model (tigers and deers), and by using the Lotka-Volterra predator-prey model with the Runge-Kutta 4^{th} order algorithm, we have successfully simulated the model, which was confirmed by comparing the simulated model to the actual obtained data from the Hudson Bay Company. Even though the two models are not entirely similar, they still follow the same graph pattern consistently and therefore, it would still beneficial for us to use the Lotka-Volterra model to predict a predator and prey system's population size, under consideration that the environment they live in is perfect for their reproduction and their contacts are consistent, instantaneous and happen frequently.

References

(1) Bing Liu, Yujuan Zhang, Lansun Chen, "The dynamical behaviors of a Lotka–Volterra predator–prey model concerning integrated pest management",*Nonlinear Analysis: Real World Applications*,*Volume 6, Issue 2*, 2005

(2) Kasim Hussain, Fudziah Ismail, Norazak Senu, "Runge-Kutta Type Methods for Directly Solving Special Fourth-Order Ordinary Differential Equations", *Mathematical Problems in Engineering*, vol. 2015, Article ID 893763, 11 pages, 2015. https://doi.org/10.1155/2015/893763

(3) Kunis, F., & Dimitrov, M. (2020). Investigating Lotka-Volterra model using computer simulation. Open Schools Journal for Open Science, 3(10). https://doi.org/10.12681/osj.24890