Open Schools Journal for Open Science

Investigating Lotka-Volterra model using computer simulation

F. Kunis, M. Dimitrov

To cite this article:

Investigating Lotka-Volterra model using computer simulation

F. Kunis[12], M. Dimitrov[2]

[1]Sofia University "St. Kliment Ohridski", Sofia, Bulgaria

[2]125th School "Boyan Penev", Sofia, Bulgaria

Abstract

In this project we study the Lotka-Volterra model, also known as the model describing the population dynamics in the Predator-prey system. This model describes the interaction of the two species and also the development of their populations over time. We simulate this model using the fourth-order Runge-Kutta algorithm. This is the most widely used method for numerical solution of ordinary differential equations. Based on the obtained program, we simulated two populations and traced their behavior over time. We optimized the parameters and managed to obtain results that are very close to real data for such populations.

Keywords

computer simulations; modeling; numerical methods; Lotka-Volterra; Predator-prey.

Lotka-Volterra model

Let's get acquainted with the Lotka-Volterra model (Lotka, 1910) (Volterra, 1926). We will look at two types that interact with each other. We will call the one type a predator (for example, a cat) and the other a prey (for example, a mouse). Let's introduce the following notations:

P (t) = population of predators, p (t) = population of preys   *(1)*

Our goal is to describe this interaction. Therefore, we will introduce the coefficient a as the rate of change of the number of victims in the absence of predators. Logically, in the absence of predators, we will have an exponential increase in prey:

$$\frac{dp}{dt} = ap, \quad \Rightarrow \quad p(t) = p(0)e^{at}.$$

(2)

In the presence of predators, the exponential increase of preys cannot happen. Therefore, in order to take into account their interaction, both types must be present. We will assume the simplest case of direct proportionality of the product of their number. The interaction is proportional to:

$$b\,p\,P$$

(3)

In this formula coefficient b is the rate at which predators destroy preys. Combining equations (2) and (3) leads to a victim modification model that takes into account both the birth rate of preys and their interaction with predators:

$$\frac{dp}{dt} = a\,p - b\,p\,P$$

(4)

If we continue with the same logic, predators should also reproduce and therefore increase their population. But predators need preys. If there are no victims, they will start attacking each other, which in turn will lead to an increase in mortality, which we will denote by m:

$$\left.\frac{dP}{dt}\right|_{\text{competition}} = -mP, \quad \Rightarrow \quad P(t) = P(0)e^{-mt}$$

(5)

Therefore, m is the death rate of predators. If there are victims to serve as food, they will interact with them with the factor bpP, which will lead to an increase in the population of predators:

$$\frac{dP}{dt} = \epsilon\,b\,p\,P - m\,P$$

(6)

Here, ϵ is a constant that measures the effectiveness with which victims favor the predator population. This brings us to the following two equations in our model:

$$\frac{dp}{dt} = a\,p - b\,p\,P$$
$$\frac{dP}{dt} = \epsilon\,b\,p\,P - m\,P$$

(7)

We will solve these equations applying Runge-Kutta method, and before that we will write them in a standard form:

$$\frac{dy}{dt} = f(y, t)$$

$$
\begin{aligned}
y_0 &= p & f_0 &= ay_0 - by_0y_1 \\
y_1 &= P & f_1 &= by_0y_1 - my_1
\end{aligned}
\qquad (8)
$$

Runge-Kutta method

In general, all types of Runge–Kutta methods (Runge, 1895) (Kutta, 1901) deal with the following situation. We have the differential equation:

$$\frac{dy}{dt} = f(t, y) \qquad (9)$$

where y is some function of t and f is some function of both t and y, and the goal is to calculate the approximate values which the function y takes.
Let's denote:

$$t_{n+1} = t_n + h \qquad (10)$$

where h is some small step (the smaller the step, the better the approximation). Let's also denote:

$$y(t_n) = y_n \qquad (11)$$

All Runge–Kutta methods have the same concept for finding the values of y – they use the formula:

$$y_{n+1} = y_n + \phi h \qquad (12)$$

where ϕ is some slope (Bishob, 2013). In other words, to get to the value y_{n+1} we start from the value y_n and move along the slope ϕ with a certain small step h. That's the logic behind all Runge-Kutta methods. The only difference is how we come up with the estimate of the slope ϕ.

For example, according to the first-order Runge–Kutta method (also called Euler's method) (Bishob, 2013):

$$
\begin{aligned}
\phi &= f(t, y) \\
f(t, y) &= \frac{dy}{dt}
\end{aligned}
\qquad (13)
$$

Therefore to get to the next value y_{n+1}, we just take the slope $\frac{dy}{dt}$ at the point y_n and move a small step h along it . The equation will take the following form:

$$y_{n+1} = y_n + f(t,y)h \qquad (14)$$

This method, however, does not give an approximation that is good enough. Each next order Runge–Kutta method offers a better and better approximation. On the other hand, each next is more and more complex and harder for calculation. The fourth-order Runge–Kutta method which we are using is the happy medium – it has a very high accuracy, while at the same time it's simple enough and can be used to solve less complicated problems as well as more complex ones. It is also the most popular method for solving differential equations. The form of the fourth-order Runge–Kutta method is (Bishob, 2013):

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = hf(t,y)$$

$$k_2 = hf\left(t + \frac{1}{2}h, y + \frac{k_1}{2}\right) \qquad (15)$$

$$k_3 = hf\left(t + \frac{1}{2}h, y + \frac{k_2}{2}\right)$$

$$k_4 = hf(t + h, y + k_3)$$

Simulation

We are going to simulate the Lotka–Volterra model using a program which uses the fourth-order Runge–Kutta method to calculate the approximated values of the predators (cats) and preys (mice) populations over time (Landau, et al., 2008). In Figure *1* you can see the program (written in the programming language C++) which we used to obtain the results.

```cpp
1    #include <iostream>
2    #include <fstream>
3
4    using namespace std;
5
6    int time = 100; /// total length of time
7    double x = 100; /// initial number of mice
8    double y = 40; /// initial number of cats
9
10   double a = 0.5, b = 0.01, c = 0.5, d = 0.01;
11   /// parameters in the Lotka-Volterra equations
12
13   int numberOfSteps = 5000;
14   double h = time*1.0 / numberOfSteps; /// h = length of each step
15
16   double k1x, k1y, k2x, k2y, k3x, k3y, k4x, k4y;
17
18   /// equation of dx/dt: fx(x, t) = ax - bxy
19   double fx (double X, double Y){
20       return a*X - b*X*Y;
21   }
22
23   /// equation of dy/dt: fy(y, t) = cy - dxy
24   double fy (double X, double Y){
25       return d*X*Y - c*Y;
26   }
27
28   /// calculates the coefficients k1, k2, k3 and k4 in terms of x and y
29   void RK4 (){
30       k1x = h*fx(x, y);
31       k1y = h*fy(x, y);
32
33       k2x = h*fx(x + k1x/2, y + k1y/2);
34       k2y = h*fy(x + k1x/2, y + k1y/2);
35
36       k3x = h*fx(x + k2x/2, y + k2y/2);
37       k3y = h*fy(x + k2x/2, y + k2y/2);
38
39       k4x = h*fx(x + k3x, y + k3y);
40       k4y = h*fy(x + k3x, y + k3y);
41
42       return;
43   }

45   int main(){
46       remove ("graphX.dat");
47       remove ("graphY.dat");
48       ofstream graphX;
49       ofstream graphY;
50       graphX.open ("graphX.dat");
51       graphX << 0 << " " << x << "\n";
52       graphY.open ("graphY.dat");
53       graphY << 0 << " " << y << "\n";
54
55       for(double i=h; i<=time; i+=h){
56           RK4();
57
58           x += 1.0/6*(k1x + k2x/2 + k3x/2 + k4x);
59           y += 1.0/6*(k1y + k2y/2 + k3y/2 + k4y);
60
61           graphX << i << " " << x << "\n";
62           graphY << i << " " << y << "\n";
63       }
64
65       graphX.close();
66       graphY.close();
67
68       cout << "Done :)" << endl;
69
70       return 0;
71   }
```

Figure 1: Program for solving the Lotka-Volterra model

Using this program, we fill two files with data. Each one of them has two columns – one with the values of t and one with the size of the population at this point of time. That means the file "graphX" contains the values of $x(t)$ (*the population of the preys*) and the file "graphY" contains the values of $y(t)$ (*the population of the predators*). Having these files, we use the app "gnuplot" to plot the graphs of $x(t)$ and $y(t)$.

As we can now plot any graph we want, it's time for "investigating". A lot of research can be done on the Lotka–Volterra model, because it depends on 6 parameters (the sizes of the two initial populations and the four coefficients a, b, c and d) whose values can vary.

Let's look at *Figure 2*, which shows how the two populations change over time. We have not marked the time units in the graphs. This is because the time depends on the populations and the four parameters. In order to have a definite unit of time, there must be a specific study of two populations and the determination of the four parameters for these two populations. In our computer simulation, the time can vary from a few months to several years.
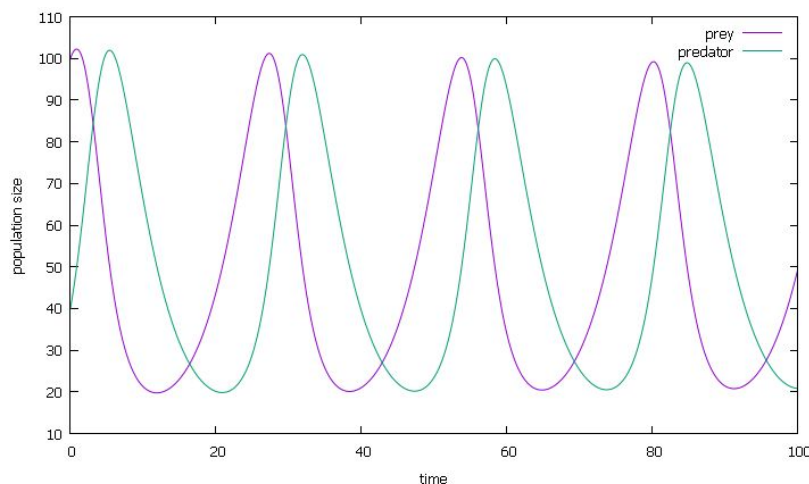
Figure 2: Number of preys and predators as a function of time from the model using
$x(0) = 100, \ y(0) = 40, \ a = 0.5, \ b = 0.01, \ c = 0.5, \ d = 0.01$

This graph looks very symmetrical and that's why it is perfect for making some important observations which will be generally true whatever initial coefficients we choose.

We will analyze the data with statements based on logic while also mathematically justifying them with the equations of the model.

In the beginning the population of the cats grows really fast (almost exponentially), because the number of mice is very big and that's why there is a lot of food for them. This corresponds to the big value of the term xyd which affects the number of cats positively.

However, as the number of cats gets bigger, the number of mice drastically decreases, because the bigger the population of cats is, the more food they consume. This corresponds to the big value of the term xyb which affects the number of mice negatively.

As the population of mice shrinks, the quantity of food for the cats becomes insufficient. What is more, as the cats are many, they will fight with each other more often. These facts explain why the population of the cats also shrinks. Mathematically, this change corresponds to the value of the term xyd getting much less than the value of yc.

As the number of cats gets really small, the population of mice starts to grow as if there are no cats i.e. exponentially. This corresponds to the negligible value of the term bxy in comparison with the term ax, both of which affect the population of mice.

Then, as the number of mice increases, we go back to the state in the beginning – a big number of mice and a small number of cats, whose population will begin to grow. That shows that the two populations oscillate and develop cyclically.

We will use that first graph to make comparisons. We will change only one or two of the parameters and compare the result with *Figure 2*.

In Figure 3.1 and 3.2 we show the changes in populations when we change the initial population of the predators. For the first one the initial population of the predators is y = 2 and for the second one y = 250.
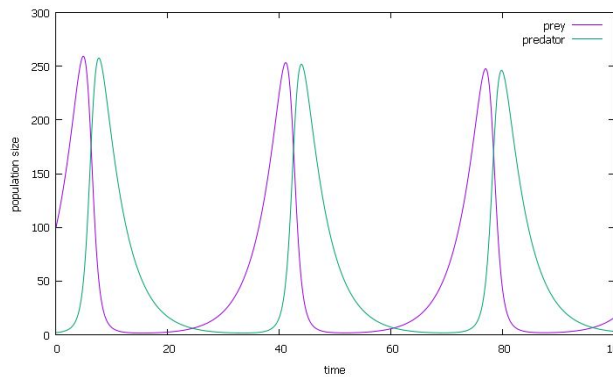

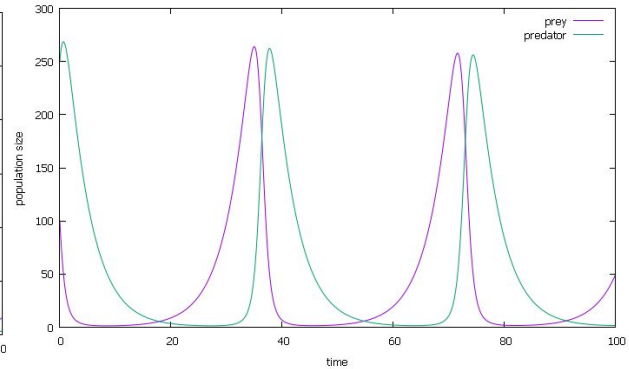Figure 3.1: Initial population $y = 2$

Figure 3.2: Initial population $y = 250$

As we can see, these graphs are almost identical. The only difference is that in the first one in the beginning the populations start to grow and in the second one they start to shrink. The interesting thing here is that after the initial difference, the two graphs follow an identical pattern of behavior.

Now, we will return to the initial values of the parameters (*x(0)=100, y(0)= 40, a = 0.5, b = 0.01, c = 0.5, d = 0.01*) and change only the parameter *a*. Look at *Figure 4.1*, *Figure 4.2* and *Figure 4.3*.
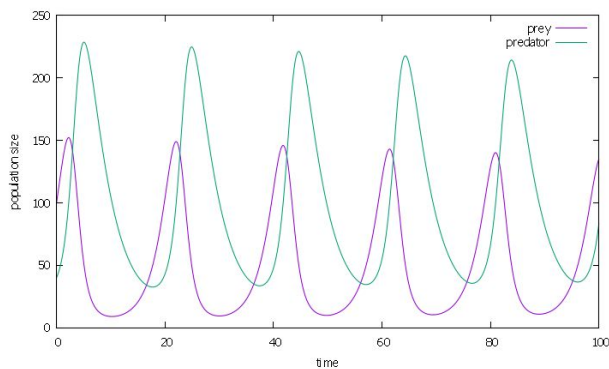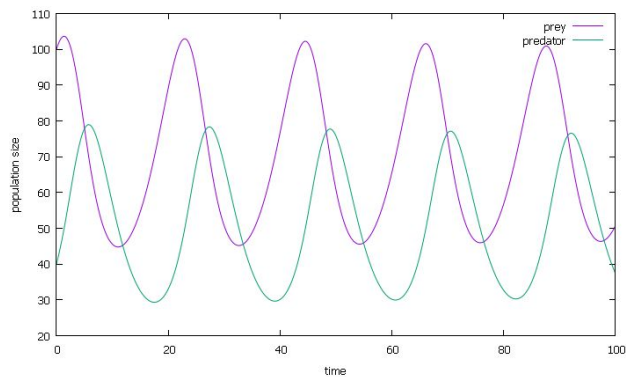

Figure 4.1: Lotka-Volterra model with $a = 1$
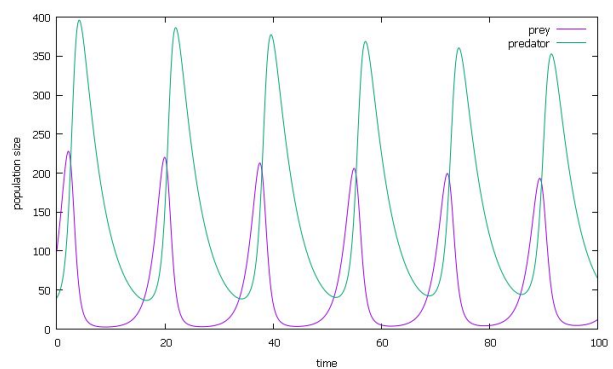
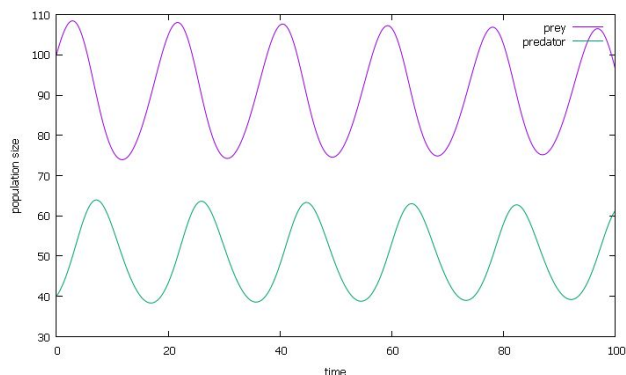Figure 4.2: Lotka-Volterra model with $a = 1.5$
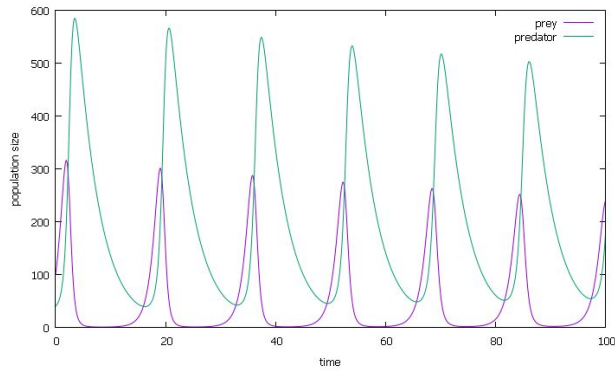
Figure 4.3: Lotka-Volterra model with $a = 2$

What happens is that by increasing the value of a we increase the rate at which the population of mice grows in the absence of cats. The number of mice increases faster and that's why it reaches a bigger maximum value than the one in the initial graph. On the other hand, that also enables the population of the cats to reach a bigger maximum.

Now, we change only the parameter c, which was initially $c = 0.5$. Look at *Figure 5.1*, *Figure 5.2* and *Figure 5.3*.

Figure 5.1: Lotka-Volterra model with c = 0.7
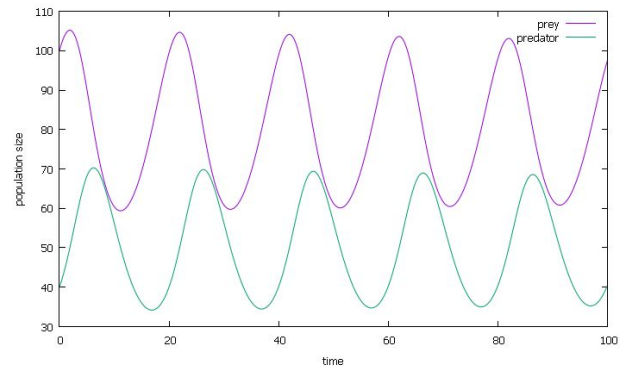
Figure 5.2: Lotka-Volterra model with c = 0.8

Figure 5.3: Lotka-Volterra model with $c = 0.9$

This apparently leads to the "separation" of the two graphs i.e. the number of mice becomes always bigger than the number of cats. That's because the term cy constantly decreases the size of the cats population.

Now, let's see what happens when we only change the parameter b, which was initially equal to $b = 0.01$. Look at *Figure 6.1* and *Figure 6.2*.
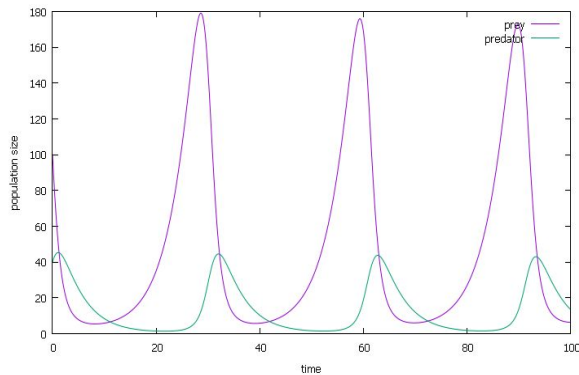

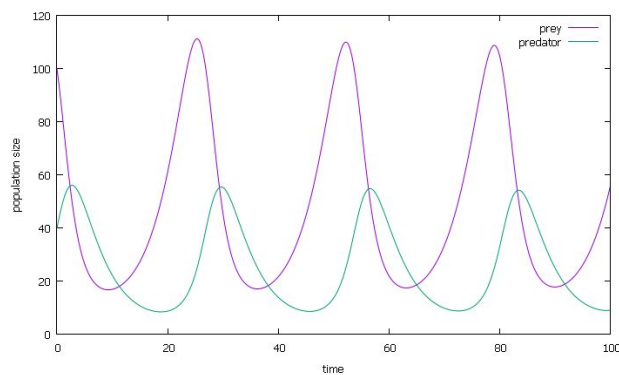*Figure 6.1: Lotka-Volterra model with* b = 0.02   Figure *6.2: Lotka-Volterra model with* b = 0.04

As we can see, when the value of b is bigger, it takes a smaller number of cats to decrease the population of mice. That is because of the term bxy which negatively affects the mice population.

Now, we change the parameter d, which was initially $d = 0.01$. Look at *Figure 7.1* and *Figure 7.2*.
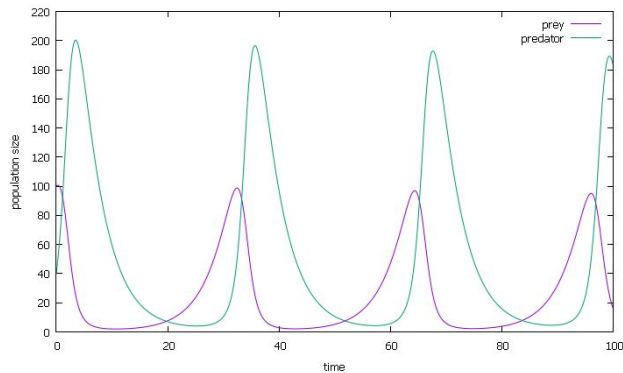

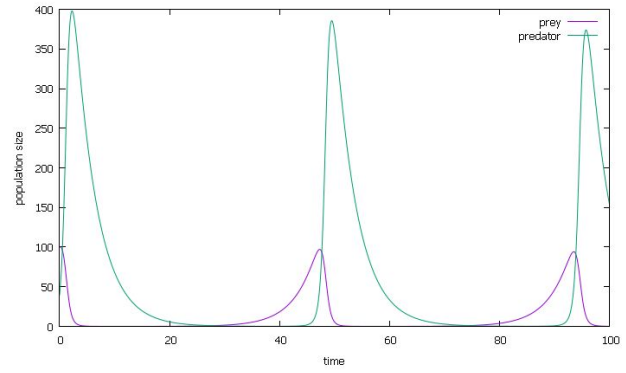*Figure 7.1: Lotka-Volterra model with* d = 0.02   Figure *7.2: Lotka-Volterra model with* d = 0.04

Analogically to the last case, here it takes a smaller number of mice to increase the population of cats. That corresponds to the term dxy which positively affects the cats population.

Now, we will change simultaneously a and c, which in the beginning were $a = c = 0.5$. Look at *Figure 8.1* and *Figure 8.2*.
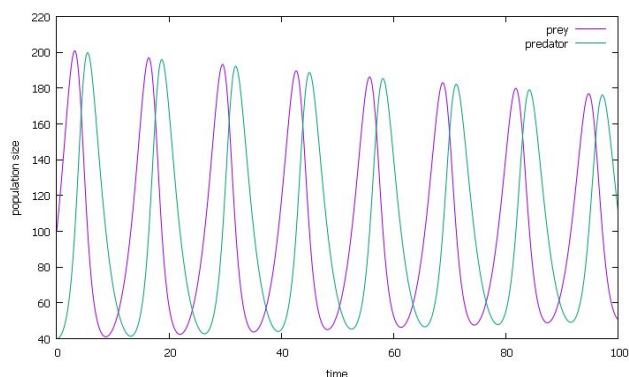
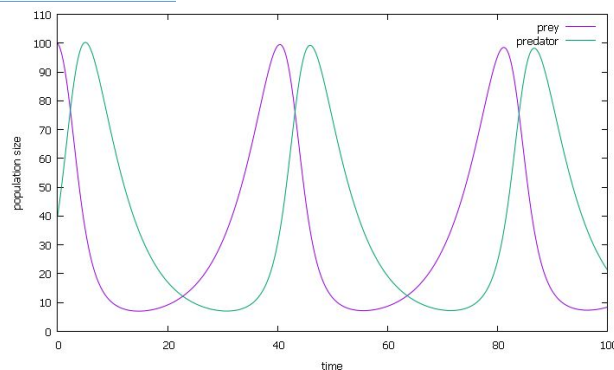*Figure 8.1: Lotka-Volterra model with $a = c = 1$*   *Figure 8.2: Lotka-Volterra model with $a = c = 0.035$*

If we combine our observations from the cases when we only changed one of these two parameters, we can say that the simultaneous increase in the values of a and c leads to an increase in the rate at which the mice reproduce in the absence of cats and the rate at which cats die in the absence of mice. That is why in the first graph the lengths of the waves are almost two times smaller than those in the second one.

If we simultaneously change the values of the parameters b and d the results are the same as the ones we got when we changed the initial population. We can intuitively prove that fact with *Figure 9*, which shows what happens when we divide the sizes of the initial populations by 10 and at the same time multiply b and d by 10.
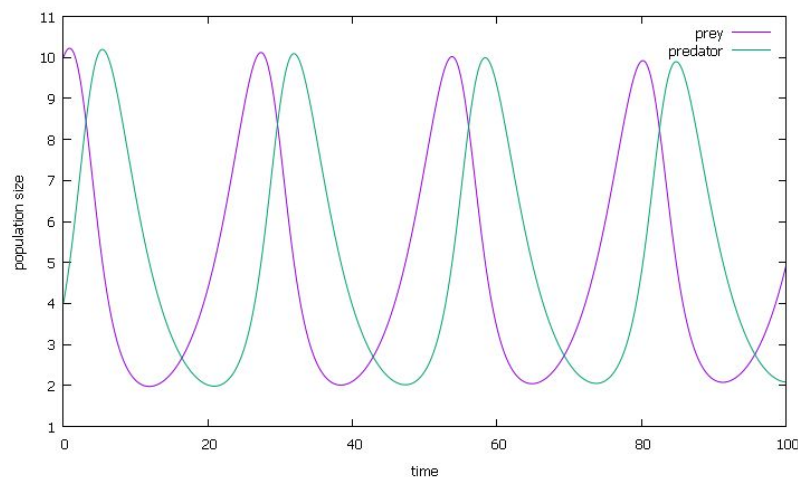

Figure 9: Lotka-Volterra model with $b = d = 0.1$, $x(0) = 10$, $y(0) = 4$

As you see, this graph is almost the same as the initial one (*Figure 2*). The only difference is that its x axis scaled by 10. We can explain that logically. By dividing the initial populations by 10 we decrease the chance of the species meeting 100 times. However, the role of the coefficients b and d is exactly the same – we use them to take into account how proportional is the rate of

change of the populations to the chance of them meeting. That's why the multiplying and dividing cancel each other out and we are left with the same graph.

When changing simultaneously the parameters a and b (which were initially a = 0.5 and $b = 0.01$) we will increase the rate of growing and shrinking of the mice population. Look at *Figure 10.1* and *Figure 10.2.*
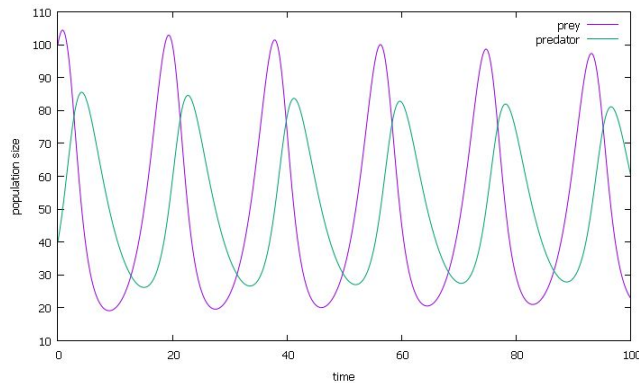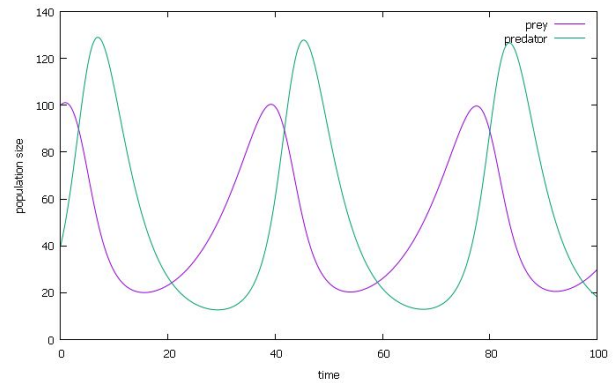

Figure 10.1: Lotka-Volterra model with a=1, b=0.02

Figure 10.2: Lotka-Volterra model with a = 0.25, b=0.005

It is logical that when we increase the values of a and b the amplitude of the wave of the mice population will also become bigger. The situation when we decrease the values of a and b is analogical.

If we change simultaneously c and d (which were initially $c = 0.5$ and $d = 0.01$) we will get similar results to the last case. Look at Figure 11.1 and Figure 11.2.
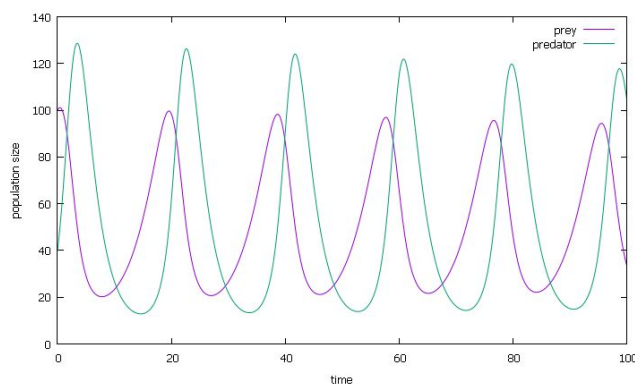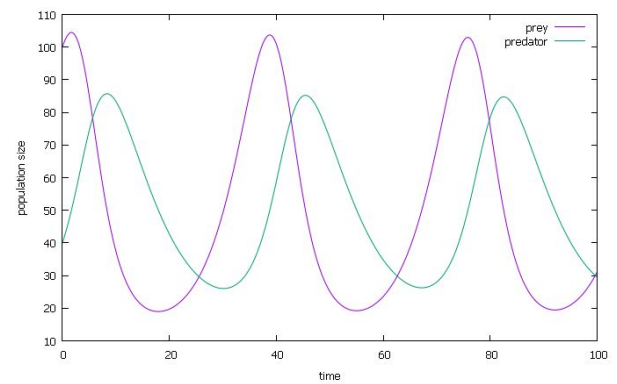

Figure 11.1: Lotka-Volterra model with c=1, d=0.02

Figure 11.2: Lotka-Volterra model with c = 0.25, d=0.005

Evaluation

The Lotka–Volterra model is simple but in quite a lot of cases it reflects reality quite well. For example, Figure 12 shows data from the *Hudson's Bay Company about the sizes of the populations of hares and lynxes over the period 1900 – 1920* (Gilpin, 1973)*. It's quite obvious that the experimental data (the cycles of growth and shrinking of the populations) matches quite well the* Lotka–Volterra model.
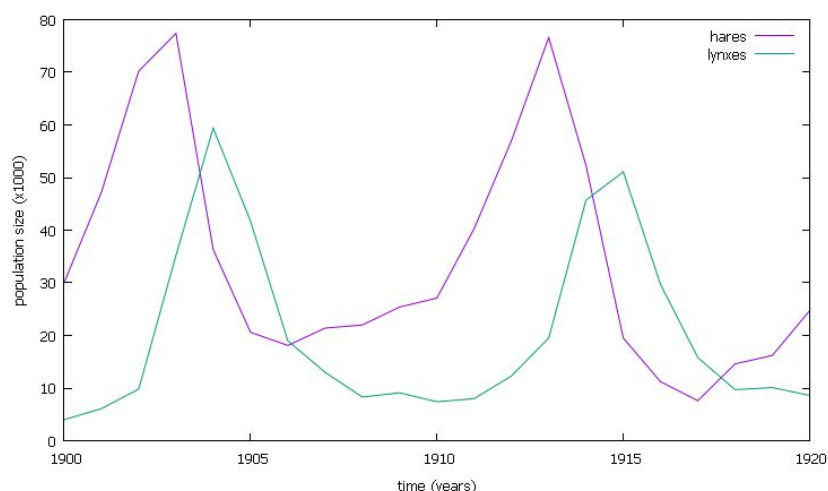

Figure 12: Number of hares and lynxes in Canada, data from the Hudson's Bay company

However, this model has some weaknesses as well. The assumption that the population of the preys can grow unlimitedly in the absence of predators is quite unrealistic. Practically, that is physically impossible. We suppose that when the predator meets the prey, the former instantly interact with the latter, which is also very unrealistic. In reality this process could be long, because it involves stalking, chasing and so on. Also, the model ignores the interaction with other species (special attention should be paid to the human influence). The age of the animals and their migration are also neglected. But despite all this, the model gives good results – the populations of oceanic phytoplankton and zooplankton develop almost exactly as in the Lotka-Volterra model.

Conclusion

In this article we have shown that complex processes and phenomena in nature can be successfully modeled. By appropriately chosen mathematical model, appropriately selected parameters of this model and by simulating this model, complex processes and phenomena in nature can be studied. In our example, we looked at two populations that belong to the predator-prey model. We chose the Lotka-Volterra model and experimented with the parameters. We were able to successfully simulate the model, which was confirmed by comparing the data obtained from the simulation and the data obtained from real observation.

Reference List

[1] Bishop, J., 2013. *7.1.8-ODEs: Classical Fourth-Order Runge-Kutta*. [video]
Available at: <https://www.youtube.com/watch?v=wr3-dWoxiY4&t=2s>
[Accessed 5 July 2020].

[2] Bishop, J., 2013. *7.1.3-ODEs: Euler's Method*. [video]
Available at: < https://www.youtube.com/watch?v=cxIbNiYM6FA>
[Accessed 5 July 2020].

[3] Bishop, J., 2013. *7.1.2-ODEs: Introduction to Runge-Kutta Methods*. [video]
Available at: <https://www.youtube.com/watch?v=b-OSyxOpxKc>
[Accessed 5 July 2020].

[4] Kutta, M., 1901. Beitrag zur näherungweisen Integration totaler Differentialgleichungen. *Zeitschrift für Mathematik und Physik.*

[5] Landau, R., Paez, M. & Bordeianu, C., 2008. *A Survey of Computational Physics. Introductory Computational Science.* s.l.:Princeton University Press.

[6] Lotka, A. J., 1910. Contribution to the Theory of Periodic Reaction. *The Journal of Physical Chemistry A.*

[7] Runge, C. D. T., 1895. Über die numerische Auflösung von Differentialgleichungen. *Springer.*

[8] Volterra, V., 1926. Variazioni e fluttuazioni del numero d'individui in specie animali conviventi. *Mem. Acad. Lincei Roma*.