Lotka Volterra Population Dynamics

Advanced Numerical Techniques Project

NIRMAL SINGH
M.Sc. Physics Semester IV

April 10, 2021

Contents

1 Introduction

We often wonder that how life came to existance on Earth and how everything turns out in the favour of life that we are still here after billions of years. Why we haven't yet found any other species out in the cosmos. If we let alone talk about our solar system it is quite surprising that life only exits approximetly $10,000\ km$ above the surface of Earth and few hundred kilometers below the surface . Every species that has ever come to existance and vanished or still fighting for their survival, every changes in the complex genetic evolution that has ever took place, it all has happened and still happening in this tiny range of vast cosmic scale. And we are here unraveling the mystery of cosmos.

In the previous semester I had Complex System and Networks where we read the necessity to study the complex systems. While exploring we discussed non-linear dynamics Rabbit and Sheep example in particular from Strogatz Book [3]. How their population affects each other and answered many questions such as when will they coexist and when will one dominate and other will go extinct.

Similarly as for my project I've selected the Lotka-Volterra model of describing predator–prey population dynamics proposed by an American physical chemist Lotka and the Italian mathematician Volterra. It can be used to describe the behaviour of biological systems and neural networks. It can even be used more widely, if modifications are made in order to make it more realistic, more powerful to give solutions to some problems occurring in Physics or in other fields of science.

Problem: *Even in this complex behivour of nature we see in general there is a balance in everything around us (ignoring the human perturbation). Every species has it's own balance numbers since nature doesn't support the exponential growth of any species. We want to achieve this natural behiviour in our simulation. We'll start with an hypothetical sitution and step by step with the help of the following book [2] we'll modify the assumptions untill we reach our interested situation. Our main interest will be the prey population where we'll use a small numbers of predators to control the population of prey so that the number of prey remains approximately constant.*

2 Used Tools in the Simulation

System Information

Linux based Operating System: Parrot 4.9, OS Type: 64-bit, Processors: 4 × AMD A6-7310 APU with AMD Radeon R4 Graphics, Memory: 3.3 GiB of RAM.

Programming Tools

1. **GCC compiler 9.3.0** for C programming.
2. **GNUPLOT Version 5.2** for plotting the datasets.
3. **Texmaker 5.0.3** for the project documentaion in LaTeX format.

Numerical Method

I've used the 4th-order Runge-Kutta method to solve the coupled differential equation of our chosen ecological system. Say we've the differential equation:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(t, x, y) \tag{2.1}$$
$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = g(t, x, y) \tag{2.2}$$

And the initial conditions at $t = 0 \Rightarrow x(0)$ and $y(0)$ are known then:

$$k_1 = h \cdot f(t_i, x_i, y_i) \qquad m_1 = h \cdot g(t_i, x_i, y_i)$$
$$k_2 = h \cdot f\left(t_i + \frac{h}{2}, x_i + \frac{k_1}{2}, y_i + \frac{m_1}{2}\right) \qquad m_2 = h \cdot g\left(t_i + \frac{h}{2}, x_i + \frac{k_1}{2}, y_i + \frac{m_1}{2}\right)$$
$$k_3 = h \cdot f\left(t_i + \frac{h}{2}, x_i + \frac{k_2}{2}, y_i + \frac{m_2}{2}\right) \qquad m_3 = h \cdot g\left(t_i + \frac{h}{2}, x_i + \frac{k_2}{2}, y_i + \frac{m_2}{2}\right)$$
$$k_4 = h \cdot f(t_i + h, x_i + k_3, y_i + m_3) \qquad m_4 = h \cdot g(t_i + h, x_i + k_3, y_i + m_3)$$

Now

$$t_{i+1} = t_i + h \tag{2.3}$$
$$x_{i+1} = x_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{2.4}$$
$$y_{i+1} = y_i + \frac{1}{6}(m_1 + 2m_2 + 2m_3 + m_4) \tag{2.5}$$

In my simulation each run will cover the time interval between 0 and 500 with the time step of $h = 0.5$.

3 Lotka Volterra Models

Before starting it is better to get familiar with the notations which have been used throughout the documention.

$$x(t) = \text{Populations of Prey}$$
$$y(t) = \text{Populations of Predators}$$
$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \text{Change in the populations of Prey}$$
$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = \text{Change in the populations of Predator}$$
$$a = \text{Prey birth rate}$$
$$b = \text{Strength of interaction}$$
$$m = \text{Predators mortality rate}$$
$$\epsilon = \text{Efficiency of predators to convert prey into food}$$

3.1 Lotka Volterra Model I

3.1.1 Constructing the problem

To begin with we assume that there is no interaction between prey and predators. In such situation we see that the prey population $x(t)$ grows at a per-capita rate of a, which would lead to the exponential growth in the population of prey:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = ax(t) \qquad \Rightarrow x(t) = x(0)e^{at}$$

Though we know, in reality it can't happen because predators y will be eating more prey as x increase. So there must be some interaction rate between prey and predators since it needs both to be present.

$$\text{Interaction rate} = bx(t)y(t)$$

where b is the strength of the interaction.

Now we can write the first, most basic equation for the prey population.

$$\boxed{\frac{\mathrm{d}x(t)}{\mathrm{d}t} = ax(t) - bx(t)y(t) \qquad \textit{(For Prey)}} \tag{3.1}$$

To get an equation for the predator population, we need to consider two things. First, predators will eat themselves if no prey present, therefore it is wise to introduce

$$\left.\frac{\mathrm{d}y(t)}{\mathrm{d}t}\right|_{\text{competition}} = -my(t) \qquad \Rightarrow y(t) = y(0)e^{-mt}$$

where m is the per-capita mortality rate. The second consideration is, once a predator caught a prey, it has an efficiency (ϵ) to convert it into food. Therefore

$$\boxed{\frac{\mathrm{d}y(t)}{\mathrm{d}t} = \epsilon bx(t)y(t) - my(t) \qquad \textit{(For Predator)}} \tag{3.2}$$

In this simple kind of model, the equilibrium values for the populations can be calculated easily which are depending on the parameters.

$$\boxed{x(t) = \frac{m}{\epsilon b} \text{ and } y(t) = \frac{a}{b}} \tag{3.3}$$

3.1.2 Results and discussion

Model	a	b	ϵ	m	$x(0)$	2.0	1.5	1.0
LVM-I	0.2	0.1	1.0	0.1	$y(0)$	1.3	1.5	1.5

Table 1: Parameters and the initial values used in the simulation.

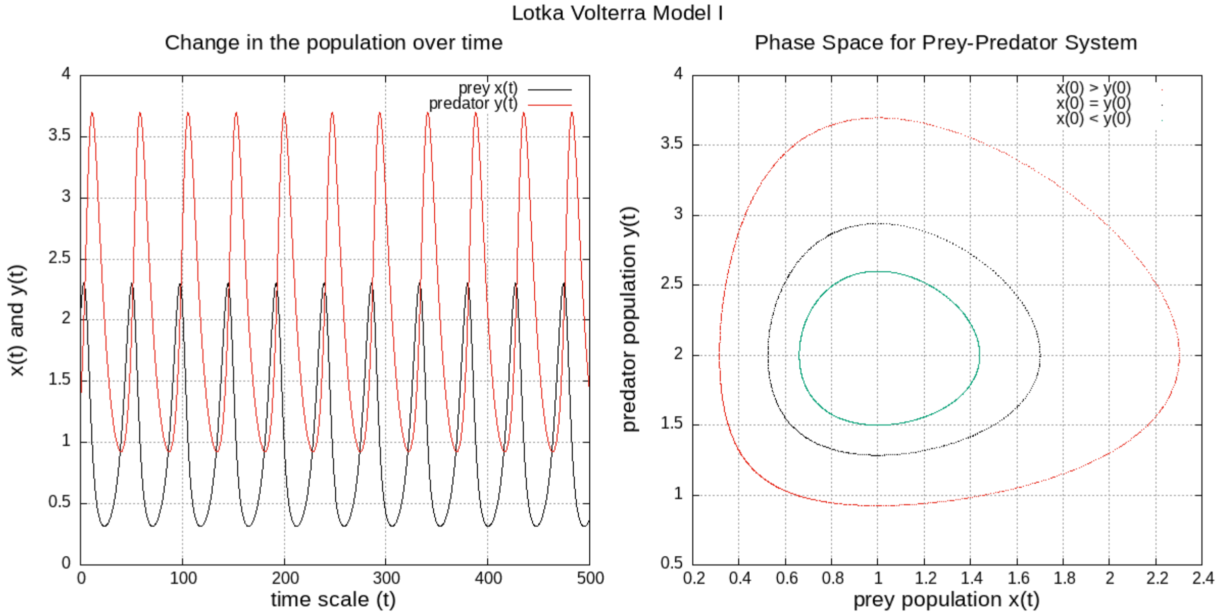

Figure 1: Left: changes in the prey population (black) and the predator population (red), when initiated with $x(0) = 2.0$ and $y(0) = 1.3$. Right: phase-space plot using three different initial conditions for the populations. (results marked with different colors for each)

It can be seen, that the amplitudes are the same, so that the prey and predator populations minimum and maximum values do not change in time and the minimum prey-maximum predator populations (and vica versa) are not occuring at the same time. From the figure it is obvious that when there are many prey, the predator population eats them and grows consequently their food supply start decreasing so their population decreases which allows the prey population to grow. This cycle goes on with time.

The same can be observed in phase-space plot, where we see a closed orbit repeating itself. Using different initial conditions wee noticed that the amplitudes are sensitive to the given initial values. But if we closer we'll see that the initial conditions are only scaling the phase space. Therefore it is not a very realistic model, we need to add some realistic modifications.

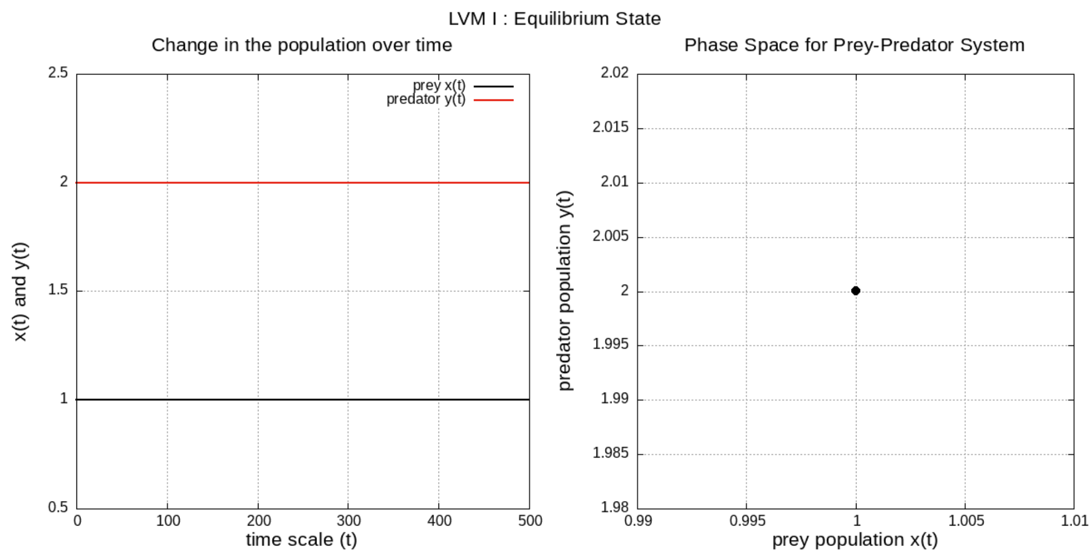

Figure 2: Equilibrium state for first kind of LVM I

3.2 Lotka Volterra Model II

3.2.1 Constructing the problem

The first step for modifying the previously presented LVM-I is to introduce a maximum limit (K) for the prey population which ensures that the food supply will decrease as the prey population grows exponentially. Now redefining the growth rate $a \to a\left[1 - \frac{x(t)}{K}\right]$ and writing the modified equation for prey.

$$\boxed{\frac{\mathrm{d}x(t)}{\mathrm{d}t} = ax(t)\left[1 - \frac{x(t)}{K}\right] - bx(t)y(t) \qquad \textit{(For Prey)}} \tag{3.4}$$

This way the growth will vanishes when the population reaches to the upper limit K also known as the carrying capacity.

Now rewriting the same equation as earlier for the predators.

$$\boxed{\frac{\mathrm{d}y(t)}{\mathrm{d}t} = \epsilon bx(t)y(t) - my(t) \qquad \textit{(For Predator)}} \tag{3.5}$$

Again calculating the equilibrium values for the populations but it wasn't as easy to calculate as before.

$$x(t) = \frac{m}{\epsilon b} \tag{3.6}$$

$$y(t) = \frac{a}{b}\left[1 - \frac{x(t)}{K}\right] = \frac{a}{b}\left[1 - \frac{m}{\epsilon b}\frac{1}{K}\right] \tag{3.7}$$

Interestingly one can see that the equilibrium of the prey population is determining the equilibrium of the predator population. The predators equilibrium is not only depending on this but also on the prey carrying capacity as a function $f\left(\frac{1}{K}\right)$.

3.2.2 Results and discussion

Model	a	b	ϵ	m	K	$x(0)$	2.0	0.5
LVM-I	0.2	0.1	1.0	0.1	20	$y(0)$	1.3	0.5

Table 2: Parameters and the initial values used in the simulation.

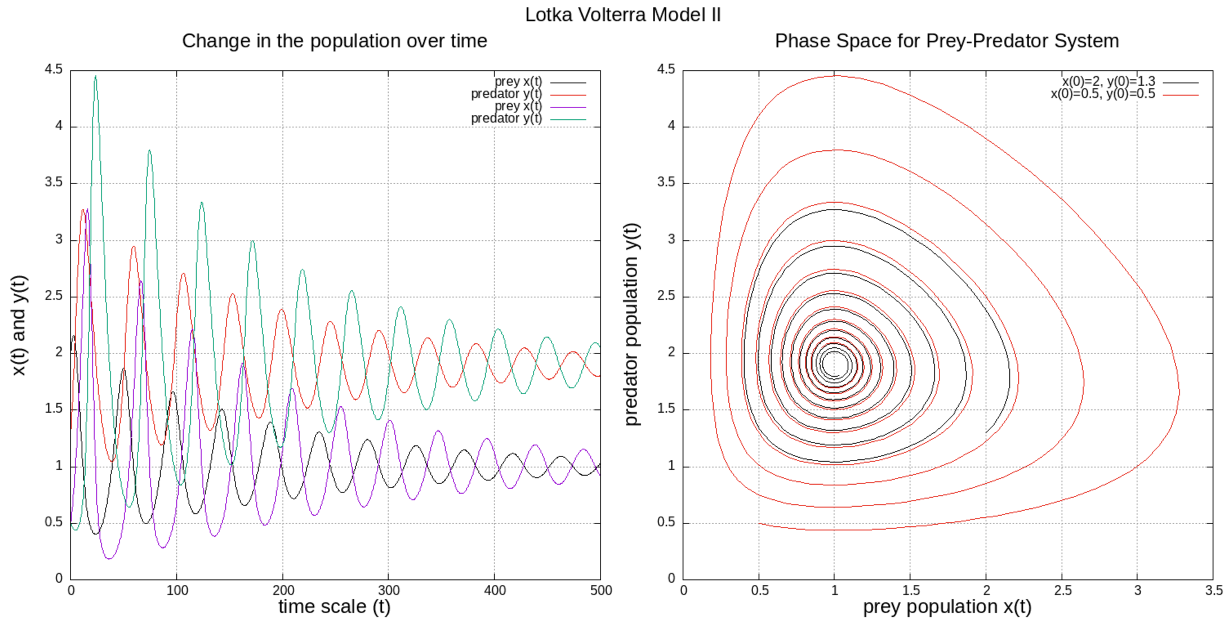

Figure 3: Left: changes in the prey population (black) and the predator population (red), when initiated with $x(0) = 2.0$ & $y(0) = 1.3$ and $x(0) = 0.5$ & $y(0) = 0.5$. Right: phase-space plot using the two mentioned initial conditions for the populations.(results marked with different colors for each)

To generate the above figure, we used the same parameters as given in the table but two sets of initial conditions for prey and predators population (also mentioned in the table). In the figure it is easy to see that no matter what the initial conditions we choose for both the prey populations we'll eventually end up to the equilibrium condition, meaning the equilibrium population are independant of initial values. But ofcourse if we change the parameters we'll get different equilibrium condition fullfilling the equilibrium equation. In the phase space plot we see spirals going inward to a single close limit cycle, on which it remains, with little variation in prey number. This is control what we mentioned while framing our problem of the project.

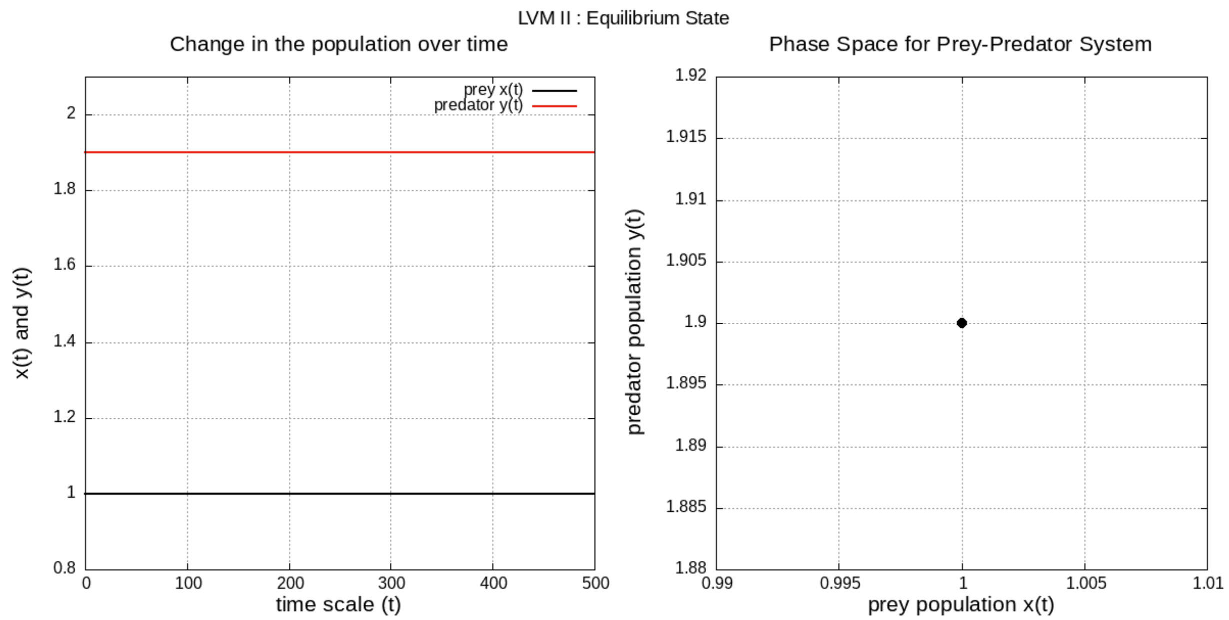

Figure 4: Equilibrium state for second kind of LVM II

3.3 Lotka Volterra Model III

3.3.1 Constructing the problem

If you recall the original LVM I and try to rethink the situation you'll notice that we took for granted an unrealistic assumption in our problem which is that predators immediately eat all the prey with which they intract. This is something that is against our intutions as we know for example the cat does spend some time looking for a mouse t_{search} and also chasing, killing, eating and digesting it t_{handling}. Further it is also wise to set a predator carrying capacity proportional to $x(t)$ as $kx(t)$. Now with these modifications rewriting the two equations describing the system.

$$\boxed{\frac{\mathrm{d}x(t)}{\mathrm{d}t} = ax(t)\left[1 - \frac{x(t)}{K}\right] - \frac{bx(t)y(t)}{1 + bx(t)t_h} \qquad \textit{(For Prey)}} \tag{3.8}$$

where t_h is the handling time for the predators.

$$\boxed{\frac{\mathrm{d}y(t)}{\mathrm{d}t} = my(t)\left[1 - \frac{y(t)}{kx(t)}\right] \qquad \textit{(For Predator)}} \tag{3.9}$$

3.3.2 Results and discussion

Model	a	b	m	K	k	t_h	$x(0)$	2.0
LVM-III	0.2	0.1	0.1	500	0.2	0.5	$y(0)$	1.3

Table 3: Parameters and the initial values used in the simulation.

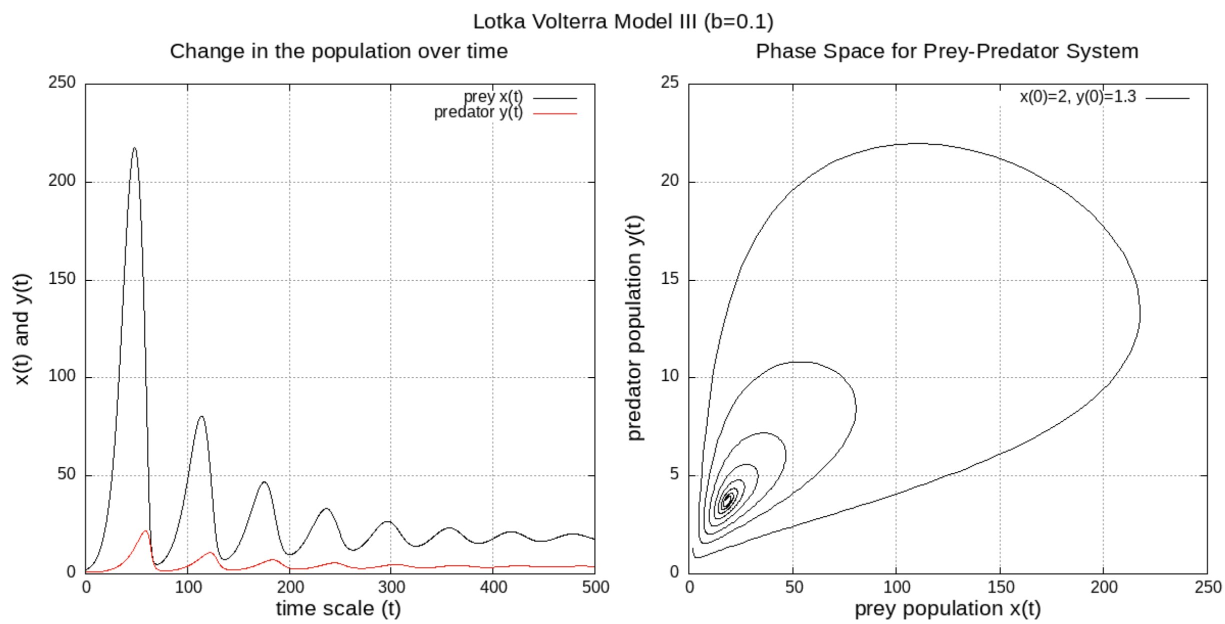

Figure 5: Left: changes in the prey population (black) and the predator population (red), when initiated with $x(0) = 2.0$ and $y(0) = 1.3$. Right: phase-space plot of populations.

We can observe the existence of three dynamic regimes as a function of the interaction rate (b) between predator and prey.

1. small b: no oscillations, no overdamping.
2. medium b: damped oscillations, which are converging to a stable equilibrium.
3. large b: limit cycle.

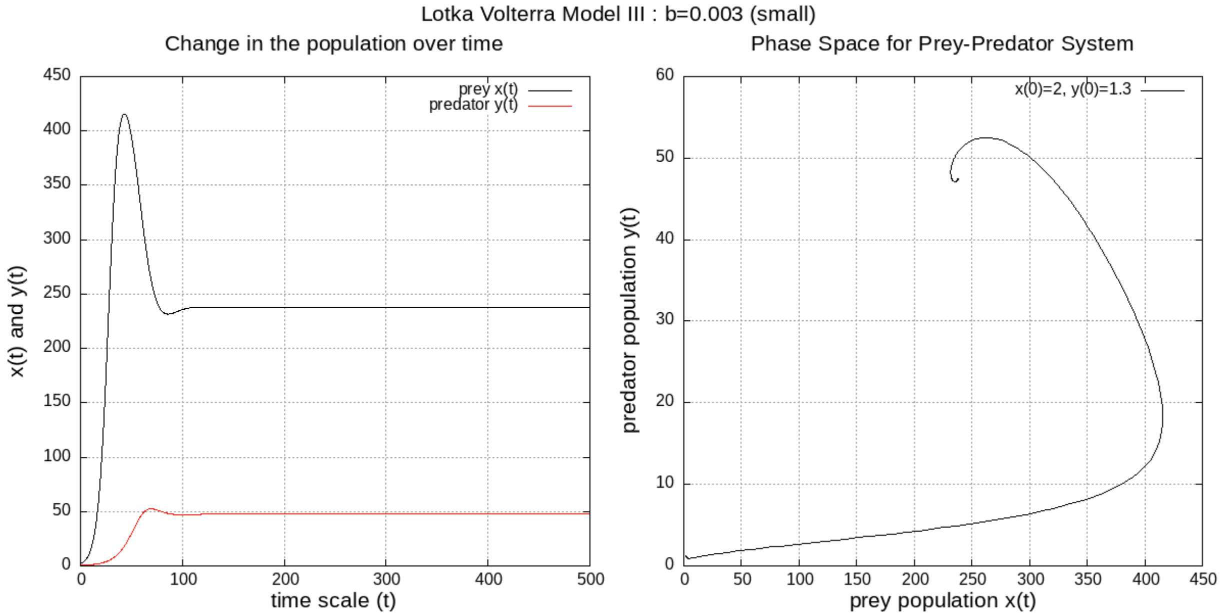

Figure 6: For small b (0.003)

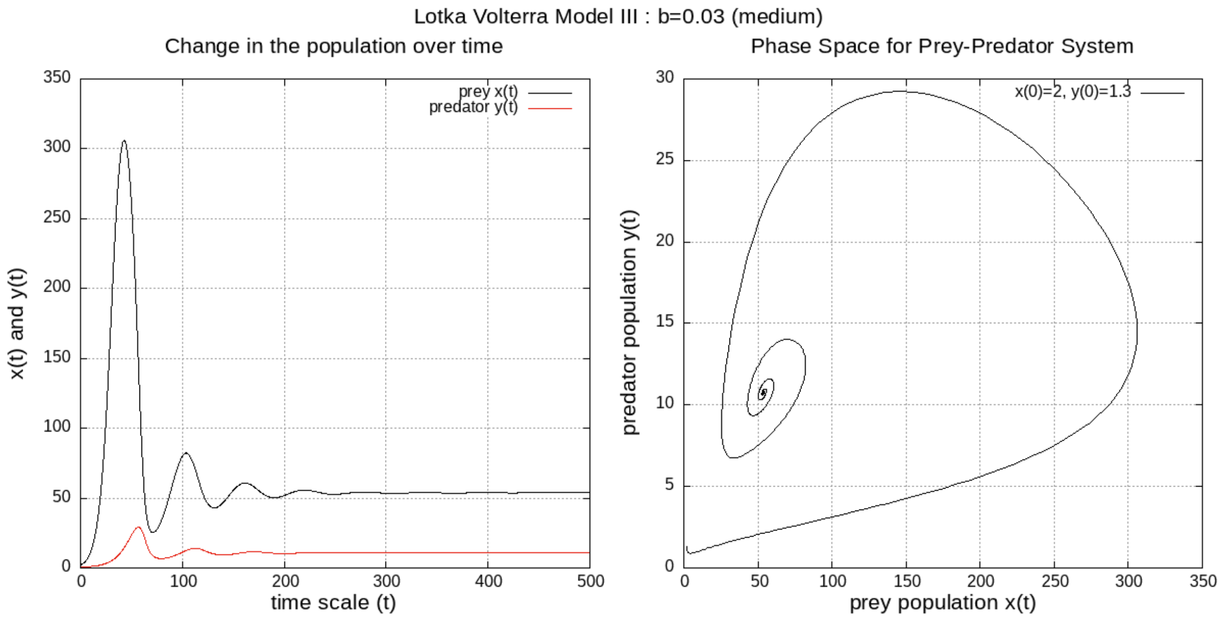

Figure 7: For medium b (0.03)

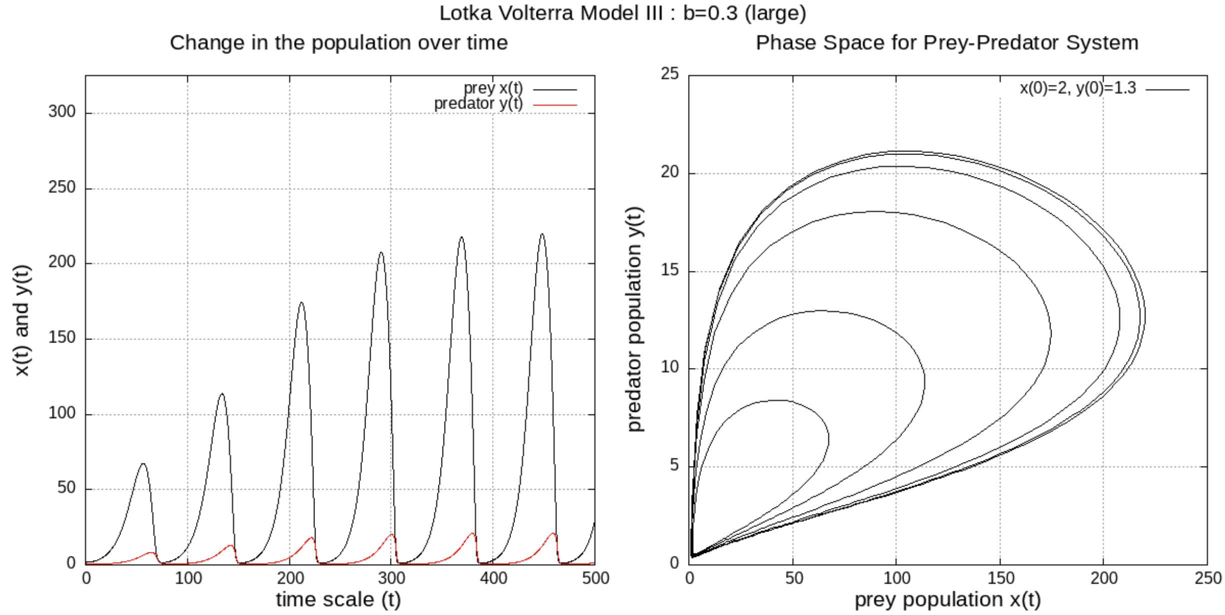

Figure 8: For large b (0.3)

The transition from equilibrium to a limit cycle is called a *phase transition*, which can be seen at a specific parameter ($b = 0.219$) with the same initial conditions and parameters as before (except b, which is changing).

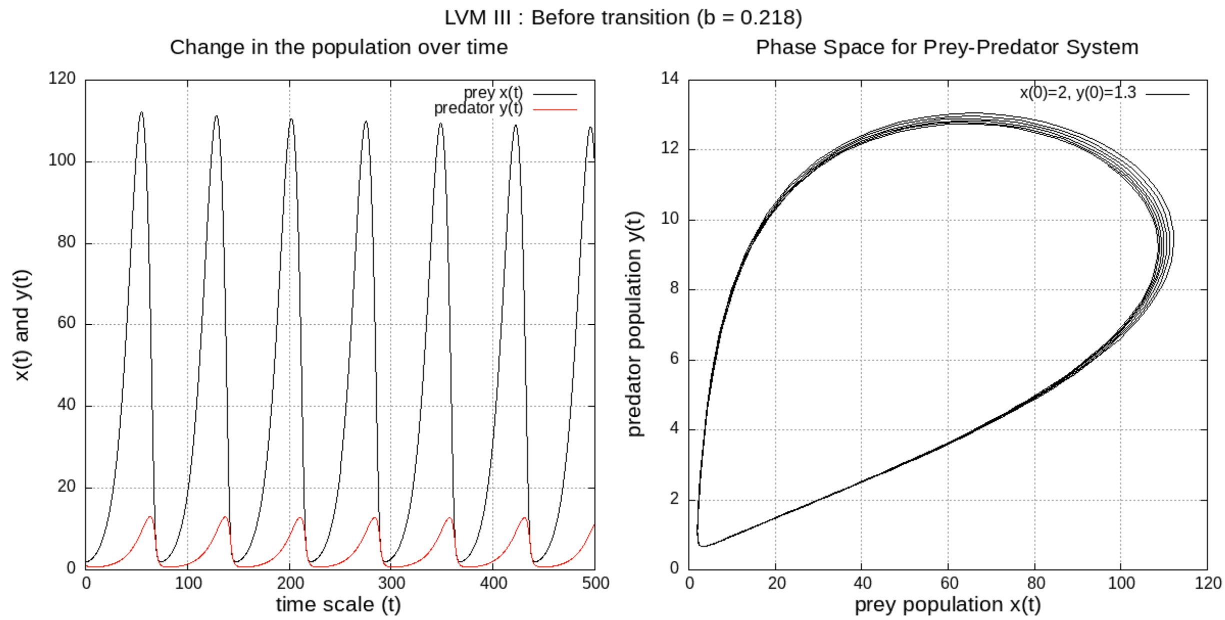

Figure 9: Before the transition (b=0.218)

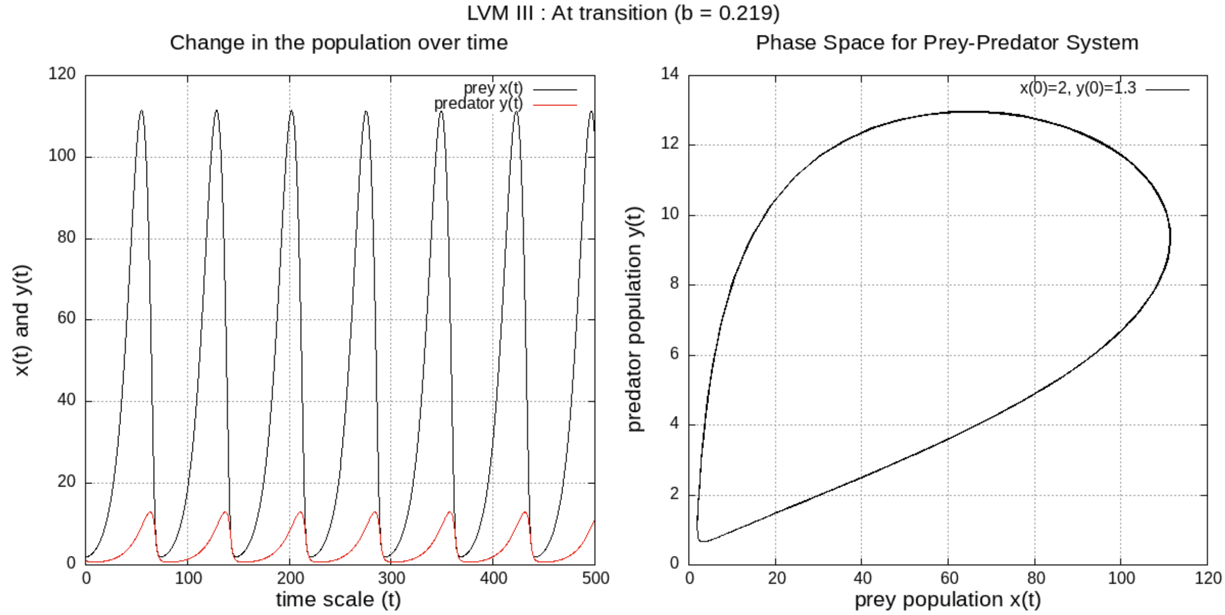

Figure 10: At the transition (b=0.219)

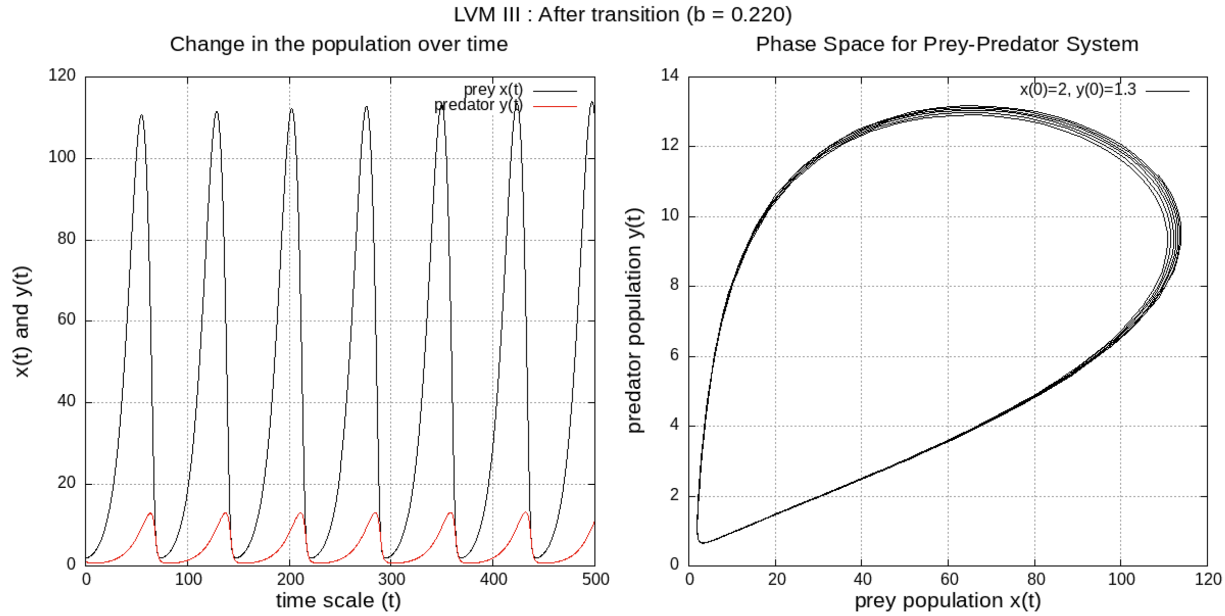

Figure 11: After the transition (b=0.220)

Therefore, we can say that this model is providing satisfactory results which we wanted in our simulation to control the prey population using small numbers of the predators. Also we observe the existence of three dynamic regimes as a function of b and found the transition value of phase transition.

3.4 Lotka Volterra Model Two Predators One Prey

3.4.1 Constructing the problem

As an additional examination it is interesting, what happens, when two predators populations share the same prey population. The equations for such a system has been adoped from LVM II.

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = ax(t)\left[1 - \frac{x(t)}{K}\right] - [b_1y_1(t) + b_2y_2(t)]\,x(t) \qquad \textit{(For Prey)} \tag{3.10}$$

$$\frac{\mathrm{d}y_1(t)}{\mathrm{d}t} = \epsilon_1 b_1 y_1(t)x(t) - m_1y_1(t) \qquad \textit{(For Predator I)} \tag{3.11}$$

$$\frac{\mathrm{d}y_2(t)}{\mathrm{d}t} = \epsilon_2 b_2 y_2(t)x(t) - m_2y_2(t) \qquad \textit{(For Predator II)} \tag{3.12}$$

where the same notations are used to all parameters and the indices are introduced to set-up different skills for each predator.

3.4.2 Results and discussion

Model	a	b	ϵ	m	K	$x(0)$	1.7
Predator-I	0.2	0.1	1.0	0.1	1.7	$y_1(0)$	1.7
Predator-II	0.2	0.2	2.0	0.1	1.7	$y_2(0)$	1.0

Table 4: Parameters and the initial conditions used in the simulation.

From the above table we can say that the second predator is twice as effective as the first one in hunting and converting the prey into food.

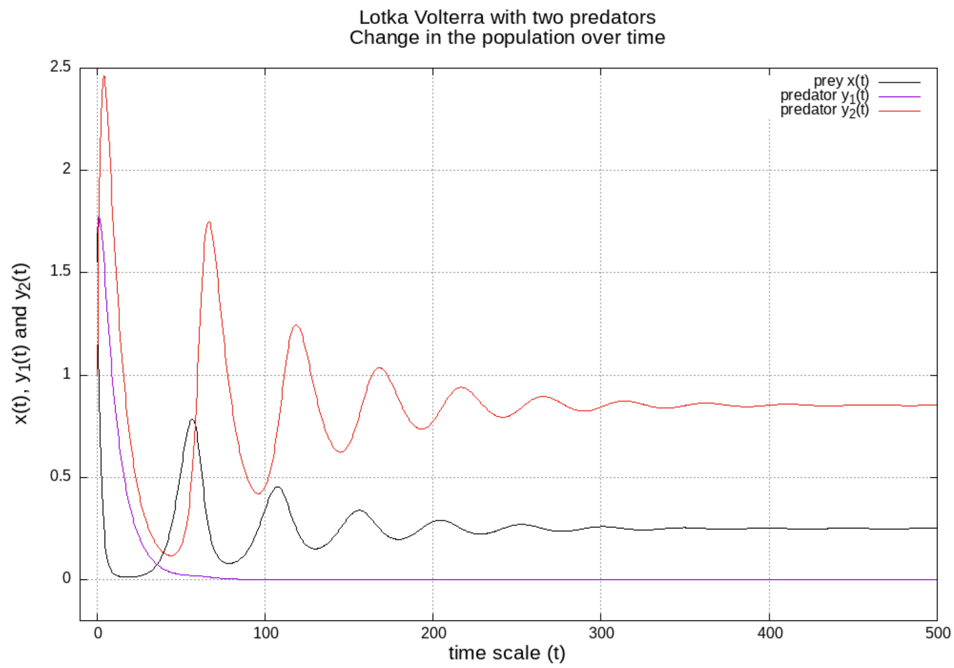

Figure 12: Changes in the prey population (black), the predator I population (violet) and the predator II population (red).

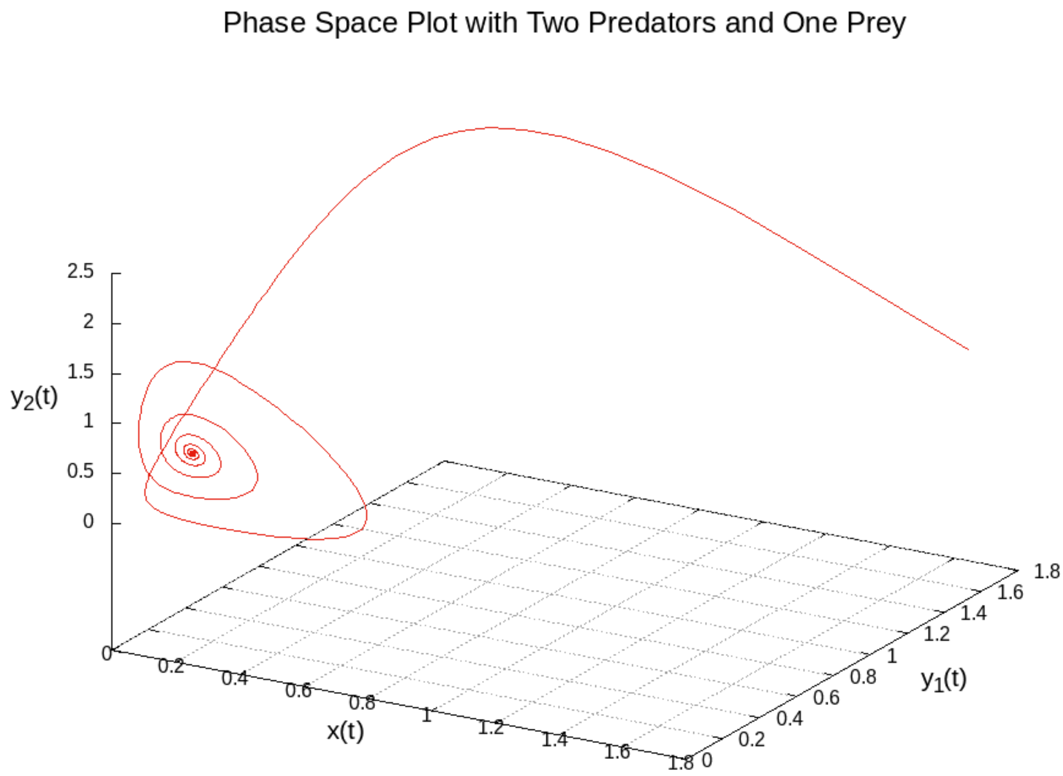

Figure 13: Phase-space plot in 3d space.

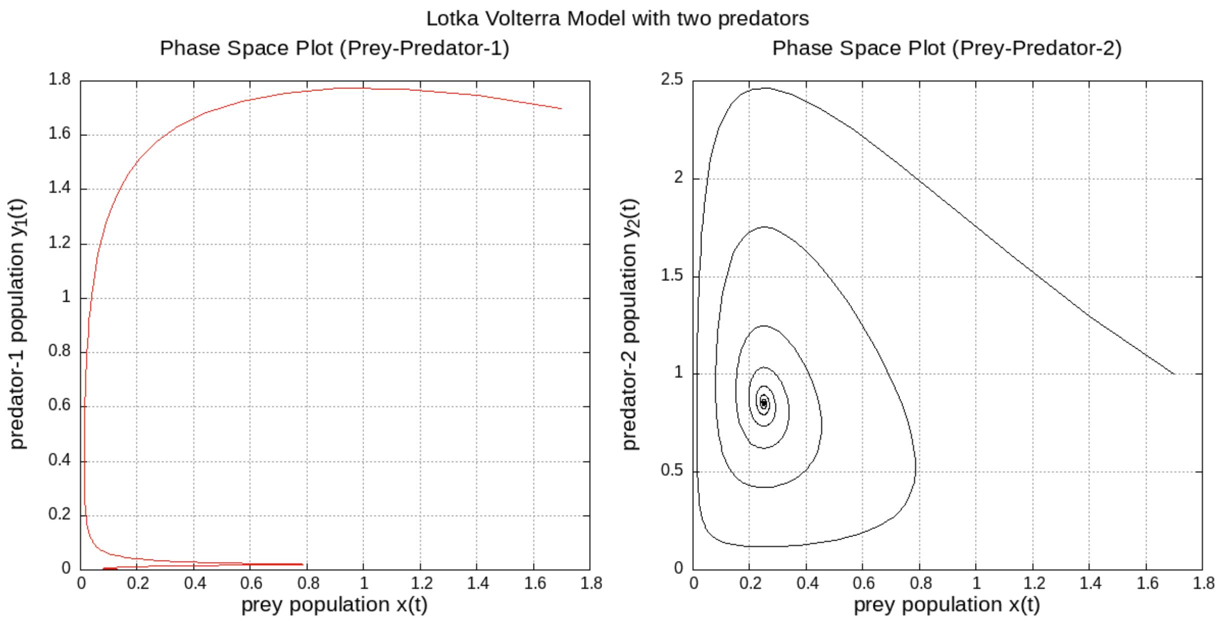

Figure 14: Phase-space plot in 2d space.

From the above figures we can see that after some time the less-skilled predator $y_1(t)$ die out while the other one's population $y_2(t)$ gets in equilibrium with the prey population $x(t)$. Alos the two predators that share the same prey can coexist only if they have the same hunting skills ($b_1 = b_2$ and $\epsilon_1 = \epsilon_2$).

4 Used Programs in the Simulation

To simulate the Lotka Volterra Models I've used the C programming for which the one kind of codes are shared here while the full source code, used files and everything related to the project could be reached in my GitHub repository. [1]

C programs

1. For Lotka Volterra Model I

```
// LVM I  system
#include<stdio.h>
#include<math.h>
// defining parameters
double a=0.2,b=0.1,e=1.0,m=0.1;
// function for prey population
double dx(double t,double x,double y) {
    return a*x-b*x*y;
}
// function for predator population
double dy(double t,double x,double y) {
    return e*b*x*y-m*y;
}
int main()
{
    int i=0,N=1000;
    double t[N],x[N],y[N],h=0.5;
    double k1,k2,k3,k4,m1,m2,m3,m4;

    FILE*fp=NULL;
    fp=fopen("lvm1.txt","w");
    // giving initial conditions
    t[0]=0,x[0]=2.0,y[0]=1.3;
    do
    {
        fprintf(fp,"%.1lf\t%lf\t%lf\n",t[i],x[i],y[i]);
        k1=h*dx(t[i],x[i],y[i]);
        m1=h*dy(t[i],x[i],y[i]);

        k2=h*dx(t[i]+h/2,x[i]+k1/2,y[i]+m1/2);
        m2=h*dy(t[i]+h/2,x[i]+k1/2,y[i]+m1/2);

        k3=h*dx(t[i]+h/2,x[i]+k2/2,y[i]+m2/2);
        m3=h*dy(t[i]+h/2,x[i]+k2/2,y[i]+m2/2);

        k4=h*dx(t[i]+h,x[i]+k3,y[i]+m3);
        m4=h*dy(t[i]+h,x[i]+k3,y[i]+m3);

        x[i+1]=x[i]+(k1+2*k2+2*k3+k4)/6;
        y[i+1]=y[i]+(m1+2*m2+2*m3+m4)/6;
        t[i+1]=t[i]+h;
        i++;

    } while(i<=N);
}
```

[1] https://github.com/singhnir/lotka-volterra-model

2. For Lotka Volterra Model II, we'll make the following changes in the above program.

```c
// LVM II   system
#include<stdio.h>
#include<math.h>
// defining parameters
double a=0.2,b=0.1,e=1.0,m=0.1,K=20;
// function for prey population
double dx(double t,double x,double y) {
    return a*x*(1-x/K)-b*x*y;
}
// function for predator population
double dy(double t,double x,double y) {
    return e*b*x*y-m*y;
}
```

3. For Lotka Volterra Model III, again we'll make the following changes in the previous program.

```c
// LVM III system
#include<stdio.h>
#include<math.h>
// defining parameters
double a=0.2,b=0.1,m=0.1,K=500,k=0.2,th=0.5;
// function for prey population
double dx(double t,double x,double y) {
    return a*x*(1-x/K)-b*x*y/(1+b*x*th);
}
// function for predator population
double dy(double t,double x,double y) {
    return m*y*(1-y/(k*x));
}
```

4. For Lotka Volterra Model of two predators and one prey populations.

```c
// LVM system with two predator and one prey populations
#include<stdio.h>
#include<math.h>
// defining parameters
double a=0.2,K=1.7;
double b1=0.1,e1=1.0,m1=0.1;
double b2=0.2,e2=2.0,m2=0.1;
// function for prey population
double dx(double t,double x,double y1,double y2) {
    return a*x*(1-x/K)-(b1*y1+b2*y2)*x;
}
// function for predator I population
double dy1(double t,double x,double y1,double y2) {
    return e1*b1*y1*x-m1*y1;
}
// function for predator II population
double dy2(double t,double x,double y1,double y2) {
    return e2*b2*y2*x-m2*y2;
}
int main()
{
    int i=0,N=1000;
    double t[N],x[N],y1[N],y2[N],h=0.5;
    double k1,k2,k3,k4,m1,m2,m3,m4,n1,n2,n3,n4;

    FILE*fp=NULL;
```

```
    fp=fopen("lvm4.txt","w");

    // giving initial conditions
    t[0]=0,x[0]=1.7,y1[0]=1.7,y2[0]=1.0;
    do {
        fprintf(fp,"%.1lf\t%lf\t%lf\t%lf\n",t[i],x[i],y1[i],y2[i]);
        k1=h*dx(t[i],x[i],y1[i],y2[i]);
        m1=h*dy1(t[i],x[i],y1[i],y2[i]);
        n1=h*dy2(t[i],x[i],y1[i],y2[i]);

        k2=h*dx(t[i]+h/2,x[i]+k1/2,y1[i]+m1/2,y2[i]+n1/2);
        m2=h*dy1(t[i]+h/2,x[i]+k1/2,y1[i]+m1/2,y2[i]+n1/2);
        n2=h*dy2(t[i]+h/2,x[i]+k1/2,y1[i]+m1/2,y2[i]+n1/2);

        k3=h*dx(t[i]+h/2,x[i]+k2/2,y1[i]+m2/2,y2[i]+n2/2);
        m3=h*dy1(t[i]+h/2,x[i]+k2/2,y1[i]+m2/2,y2[i]+n2/2);
        n3=h*dy2(t[i]+h/2,x[i]+k2/2,y1[i]+m2/2,y2[i]+n2/2);

        k4=h*dx(t[i]+h,x[i]+k3,y1[i]+m3,y2[i]+n3);
        m4=h*dy1(t[i]+h,x[i]+k3,y1[i]+m3,y2[i]+n3);
        n4=h*dy2(t[i]+h,x[i]+k3,y1[i]+m3,y2[i]+n3);

        x[i+1]=x[i]+(k1+2*k2+2*k3+k4)/6;
        y1[i+1]=y1[i]+(m1+2*m2+2*m3+m4)/6;
        y2[i+1]=y2[i]+(n1+2*n2+2*n3+n4)/6;
        t[i+1]=t[i]+h;

        i++;
    } while(i<=N);

}
```

Gnuplot scripts

Each of the previously mentioned program will generate a text file as its output file. To plot the data-sets I've written the gnuplot scripts which will produce the similiar plots as shared in the results of each model.

1. Script for the multiplot of time evolution and the phase-space trajectory.

```
set terminal png size 1200,600
set output "lvm1.png"
set key font ",14"
set multiplot layout 1,2 title "Lotka Volterra Model I"
set grid
set title "Change in the population over time"
set xlabel "time scale (t)"
set ylabel "x(t) and y(t)"
plot "lvm1.txt" u 1:2 w l lc 8 t "prey x(t)",'' u 1:3 w l lc 7 t "predator y(t)"

set title "Phase Space for Prey-Predator System"
set xlabel "prey population x(t)"
set ylabel "predator population y(t)"
set autoscale
plot "lvm1.txt" u 2:3 w l lc 8 t "x(0)=2,y(0)=1.3"
```

2. Script for the 3D plot in the LVM of two predators and one prey.

```
set terminal png size 1000,700
set output "3d.png"
set key font ",16"
set grid

set title "Phase Space Plot with Two Predators and One Prey"
set xlabel "x(t)"
set ylabel "y_1(t)"
set zlabel "y_2(t)"
splot "lvm4.txt" u 2:3:4 w l lc 7 t ""
```

To obtain all kind of plots shared here one need to make the obvious changes in the given scripts.

5 Conclusion

Rather than doing some mathematically challenged analysis it was quite easy to solve the non-linear differential equations using the numerical methods on computer. However it was not as easy to understand the solutions as I've experinced in the usual pen and paper way. As for my problem the numerical simulations have been done in C programming and plotted using Gnuplot. The results are shared here with the work as well as in my GitHub repository. One can see that I've got the expected results.

References

[1] Professor Sanjay Jain. Complex system and networks. `http://people.du.ac.in/~jain/`, September 2020. Class conversation.

[2] Rubin H Landau, Manuel J Paez, and Cristian Bordeianu. *A Survey of Computational Physics - Introductory Computational Science*. Princeton University Press, 2008.

[3] Steven H. (Steven Henry) Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Westview Press, a member of the Perseus Books Group, 2 edition, 2015.