Numerical Solution for the Lotka-Volterra Model by using Runge Kutta Method

by
E M C H Madushani
SC/2020/11486

Project report submitted to the
Department of Mathematics
in partial fulfillment of the requirement for the
Level I Industrial Mathematics course unit (IMT1b2β)
of the

Bachelor of Science (General) Degree

in the
Faculty of Science
University of Ruhuna
Matara
Sri Lanka.

December 2022

DECLARATION

I, E M C H Madushani, declare that the presented project report titled, "Numerical Solution for the Lotka-Volterra Model by using Runge Kutta Method" is uniquely prepared by me based on the group project carried out under the supervision of Dr. A. W. L. Pubudu Thilan Department of Mathematics, Faculty of Science, University of Ruhuna, as a partial fulfillment of the requirements of the level I Industrial Mathematics course unit (IMT1b2β) of the Bachelor of Science (General) Degree in Department of Mathematics, Faculty of Science, University of Ruhuna, Sri Lanka. It has not been submitted to any other institution or study program by me for any other purpose.

Signature:. . . . .

Date:. . . . .

SUPERVISOR'S RECOMMENDATION

I/We certify that this study was carried out by E M C H Madushani under my/our supervision.

Signature:. . . . .

Date:. . . . .

Dr. A.W.L. Pubudu Thilan,

Department of Mathematics,

Faculty of Science,

University of Ruhuna.

ACKNOWLEDGMENTS

First of all, I want to sincerely thank our supervisor Dr. A.W.L. Pubudu Thilan for choosing this project as well as for all the directions and instruction he provided us during this time. I want to express my deep gratitude to the laboratory demonstrators for industrial mathematics for all of their assistance. I also want to sincerely thank everyone in my group who contributed to the success of this project and helped us all work together to achieve our shared goal.The rest of my thanks go to my parents and all of my friends for their encouragement and good comments during the effort.

ABSTRACT

The Lotka–Volterra model is mainly a two-dimensional configuration. The two-dimensional form limits the application scenarios of the Lotka–Volterra model. The major goal of this project is to investigate the numerical solution of the Lotka Volterra model by using Runge Kutta method. In this project, we talk about recognizing mathematical model that describe the size of the populations of the predators and prey. The Lotka-Volterra model makes the assumption that a predator's rate of prey consumption is inversely related to its abundance. As a result, the only factor affecting predators' ability to feed is the availability of prey. While this may be true at low prey concentrations, it is undoubtedly false at high prey densities where there are few predators.

We used 'Runge Kutta' method to build the connection with the code when thinking about the mathematical models that could be applied. The Runge-Kutta method is a reliable and popular approach for resolving differential equation initial-value issues. Without the necessity for high order derivatives of functions, the Runge-Kutta method can be utilized to build high order accurate numerical methods by functions alone. The Runge-Kutta technique aims to solve the Euler's method's issue with selecting a small enough step size to provide a respectable degree of precision in the problem resolution. We have the ability to obtain the numerical solutions using these methods.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Introduction

1.1 Identifying the project

1.1.1 Lotka Volterra Model

Alfred Lotka first presented the Lotka Volterra, Predator Prey model in the theory of autocatalytic chemical processes in 1910. Lotka utilized the equation to study predator-prey interactions in 1925 after extending the model through Andrey Kolmogorov to "Organic systems" utilizing the species and a herbivorous animal species in 1920. In 1926, Vito Volterra published the same set of equations. Through his interactions with the marine biologist Umberto D'Ancona, Volterra was motivated to conduct his investigation. Volterra independently created his model and applied it to d'Ancona's observation. This model is used to describe dynamics of biological system. When two or more species coexist close together and have similar basic needs, they frequently engage in competition for resources, food, habitat, or territory.

There are two species interact,

1. Predator
2. Prey

The population sizes of the predator and prey oscillate, with the peak of the predator's oscillation lagging behind the peak of the prey's oscillation. The model makes several simplifications, including the following:

Figure 1.1: Predator Prey Behavior

1) That the prey population will increase exponentially in the absence of the predator;

2) That the predator population will starve in the absence of the prey population (rather than switching to a different kind of prey);

3) That predators can consume an infinite amount of prey; and

4) That the environment is simple.

These are the simplifications of the Lotka Volterra Model.

1.1.2 Lotka Volterra Equation

The boom-bust periodic oscillation behavior of the relative populations can be shown by numerically solving the coupled differential equations of the Lotka-Volterra system. The Lotka Volterra equations explain an ecological predator-prey model.

$$\frac{dx}{dt} = ax - bxy \tag{1.1}$$

$$\frac{dy}{dt} = cxy - dy \tag{1.2}$$

x - Number of preys at time t

y - Number of predators at time t

a - Growth rate of the prey

b - Rate of shrinkage relative to the product of the population size

d - The shrinkage rate of the predator population

c - The growth rate of the predator population as a factor of the product of the population size

Those are the assumption that for a given set of parameters that are always positive.

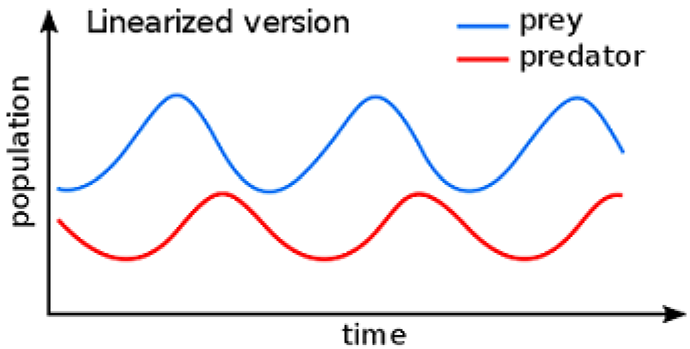

Figure 1.2: Graph of Lotka Volterra model

1.1.3 Physical meanings of the equations

Numerous presumptions are made by the Lotka-Volterra model regarding the environment and the development of the predator and prey population.

• Prey population finds ample foods.

• The size of the prey population is the defining factor of the predator population's food supply.

• The rate of change of population is proportional to the size

• The procedure doesn't affect the environment in a way that benefits a single species, and the genetic adaptation is clearly visible.

• Predator have limitless appetites.

The solution of this differential equation is deterministic and continuous. Also the generations of both the predator and prey are continually overlapping.

Prey

$$\frac{dx}{dt} = ax - bxy \tag{1.3}$$

In this model we assumed that the prey have an unlimited food supply. They reproduce exponentially unless subject to predate. This exponentially growth represented in equation by 'ax'. The rate of prediction upon the prey is proportional to rate at which the predators and preys meet. This is represented in equation by 'bxy'.

Predator

$$\frac{dy}{dt} = cxy - dy \tag{1.4}$$

In here we use a constant as the rate at which the predator population grows is not necessarily equal to the rate. The growth of predator population is represented by 'cxy'. Also the loss rate of the predators due to either natural death or emigration is represented by 'dy'. According to the equation, the rate at which a predator consumes prey less its intrinsic death rate determines how quickly its population is changing.

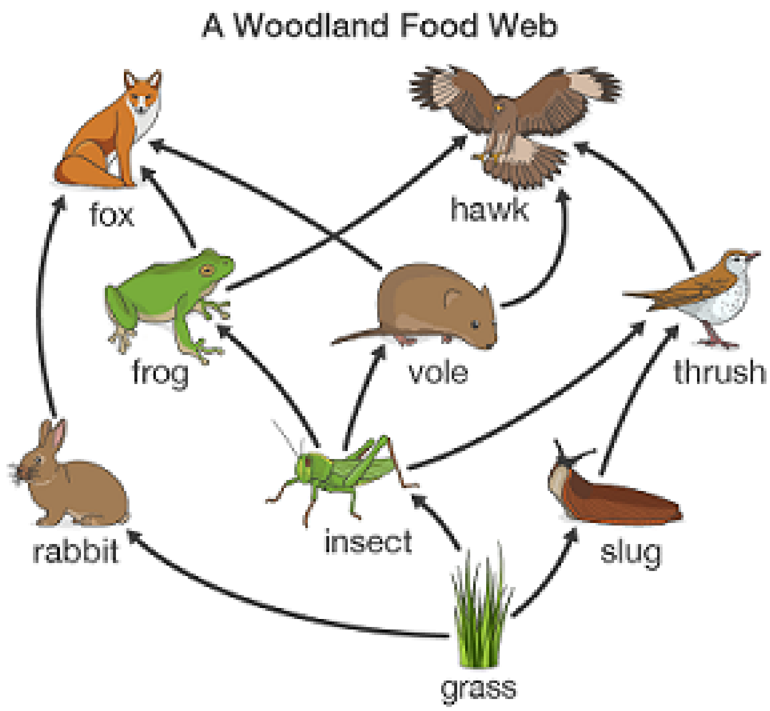

Figure 1.3: A Food Web of Animals

All the food chains in a single ecosystem make up a food web. Each organism in an ecosystem is a link in a number of food chains. Energy and nutrients can go along different food chains as they move through the ecosystem. A food web is made up of all the linked and overlapping food systems in an ecosystem.
The above figure shows a food web of animals.

[7]

[1]

CHAPTER 2
Literature Review

2.1 Background of the study

In this work, we consider creating a proper mathematical model for the find the solution of Lotka Volterra model.

2.1.1 How mathematical modeling combines with the Lotka Volterra Model

Different interventions are created by mathematical models to increase the probability of solving the Lotka Volterra model. To comprehend the dynamical process of this model, the majority of mathematical models have been created. The numerical solution to this Lotka-Volterra model can be found with the aid of mathematical models.

2.2 Mathematical Models used in Lotka Volterra Model

In this section we study about the mathematical models that can be used to find the solution of Lotka Volterra Model.

2.2.1 Logistic Equation

The formula of logistic equation is:

$$\frac{dN}{dt} = rN(1 - \frac{N}{K}) \tag{2.1}$$

The exact solution for the equation:

$$N(t) = \frac{N_0 K e^n}{[K + N_0(e^n - 1)]} \tag{2.2}$$

Where r is growth rate and K is carrying capacity. In the first equation, there is a certain value K which makes zero at bracket term. Three key features of the logistic growth are:

1. $\lim_{t\to\infty} N(t) = K$, The population will ultimately reach its carrying capacity.

2. The relative growth rate, $\frac{1}{N}\frac{dN}{dt}$, declines lie=nearly with increasing population size,

3. The population at the inflection point (where the growth rate is maximum), N_{inf} is exactly half the carrying, $N_{inf} = \frac{K}{2}$

2.2.2 Exponential Method

The formula of the Exponential method is:

$$\frac{dX(t)}{dt} = aX(t) \tag{2.3}$$

In here a - is the kinetic parameter and V(t) is Population size of the Predator or Pray. This model estimates the maximum population at t time.

2.2.3 Gompertz Model

The formula of the Gompertz model is:

$$\frac{dX(t)}{dt} = aV \ln \frac{b}{X + c} \tag{2.4}$$

This model was obtain from generalizing the logistic model. This model has ability to explain the curve of the graph and rate of change of the population size.

[4]

[5]

CHAPTER 3
Problem Statement

3.1 What is a Problem?

The main problem is solving the Lotka Volterra equation by using 'Runge Kutta' method. For this we use c programmming.

• Suppose there are two species of animals, a baboon (Prey) and cheetah (Predator).

• If the initial conditions are 10 baboons and 10 cheetahs, one can plot the progression of the two species over time;

• Given the parameters that the growth and death rates of baboon are 1.1 and 0.4 while that of cheetahs are 0.1 and 0.4 respectively.

• The choice of time interval is arbitrary.

• One may also plot solutions parametrically as orbits in phase space, without representing time, but with one axis representing the number of prey and the other axis representing the number of predators for all times.

3.2 Why it is needed to find a solution?

Simplified mathematical models based on differential equations include the Lotka-Volterra equation for exponential population increase and modified equations for logistic growth and interspecies interactions.When two species—one a predator and one its prey—interact in an ecological system, the dynamics are commonly described by the Lotka-Volterra model. Then as the final result it will provide the important

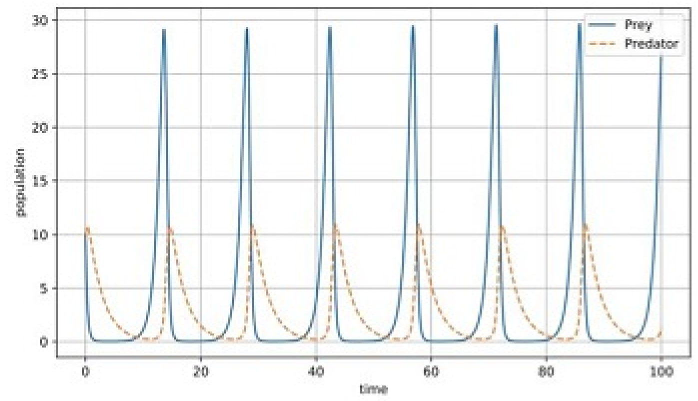

Figure 3.1: Population of Cheetahs and Baboons

data to analyze the population of animals.

3.3 Objectives of the Study

3.3.1 Main Objectives

- Correct comprehension of how predators and preys interact.
- To construct a c code, use numerical method solutions obtained using the "Runge Kutta" approach.
- How to write computer programs using mathematical models that have one, two, or three equations.

3.3.2 Other Objectives

- Develop advanced latex understanding and enhance latex writing abilities.
- Develop your team's ability to collaborate toward a common objective.

[3]

CHAPTER 4
Methodology

4.1 Ordinary Differential Equation

An ordinary differential equation (ODE) is an equation that involves some ordinary derivatives of function. If you need to solve an ODE, determine what function or functions satisfy the equation.

If you know what the derivative function is, then you need to find the anti derivative. Following example shows the simplest ODE question and solution.

Example

Solve the following ordinary differential equation.

$$\frac{dx}{dt} = 7x - 4 \tag{4.1}$$

Solution,

$$\frac{dx}{7x-4} = dt \tag{4.2}$$

$$\int \frac{dx}{7x-4} = \int dt, \tag{4.3}$$

$$\frac{1}{7}\ln 7x - 4 = t + C_1 \tag{4.4}$$

$$x(t) = Ce^{7t} + \frac{4}{7} \tag{4.5}$$

4.2 Numerical Method

Finding numerical approximations to the solutions of ordinary differential equations is done using numerical methods for ordinary differential equations (ODEs).

The process of using them is referred to as ”numerical integration,” while this term can also be used to describe computing integrals.
In this project we used 'Runge Kutta Method' as a type of numerical method to study the model as well as build the c code.

4.3 Runge Kutta Method

These methods were developed by Carl Runge and Wilhelm Kutta. The Runge-Kutta method is a reliable and repeated applications for resolving differential equation initial-value issues. Without the necessity for high order derivatives of functions, the Runge-Kutta method can be utilized to build high order accurate numerical methods by functions alone.

Runge Kutta methods are singe step methods, however with multiple stages per step.

- First order Runge Kutta method
- Second order Runge Kutta method
- Third order Runge Kutta method
- Fourth order Runge Kutta method

In here we used Runge Kutta fourth order method to solve the equations of Lotka Volterra model.

4.3.1 Fourth Order Runge Kutta Method

Let the given differential equation be

$$\frac{dy}{dx} = f(x, y) \tag{4.6}$$

with initial conditions $y = y_0$ when $x = x_0$

Then need to find the value of $y = y_0 + k$ at $x = x_0 + h$

we calculate the following terms:

$$k_1 = h * f(x_0, y_0) \tag{4.7}$$

$$k_2 = h * f(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}) \tag{4.8}$$

$$k_3 = h * f(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}) \tag{4.9}$$

$$k_4 = h * f(x_0 + h, y_0 + k_3) \tag{4.10}$$

$$k = \frac{1}{6}(k_1 + 2 * k_2 + 2 * k_3 + k_4) \tag{4.11}$$

The required value of y

$$y = y_0 + k \tag{4.12}$$

After finding the above terms, We can plot the graph for the Runge Kutta Method.

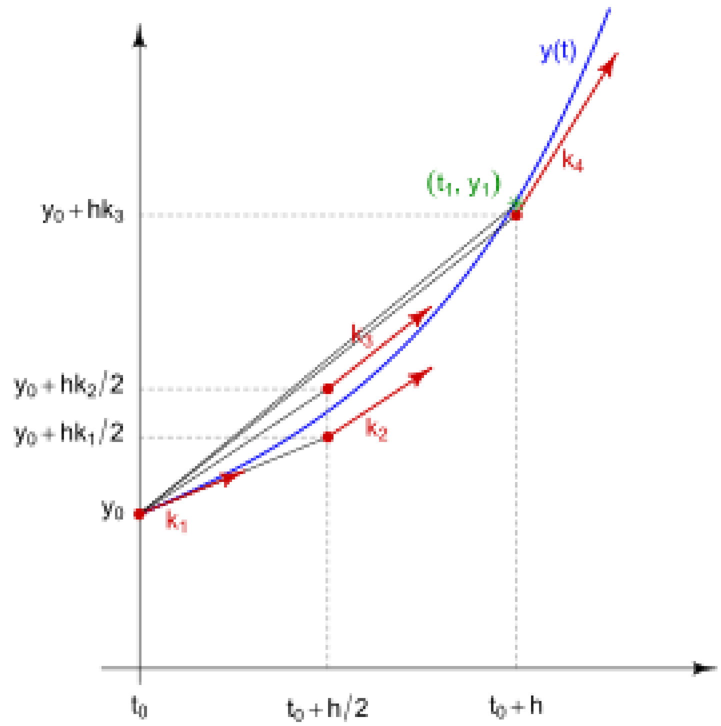

Figure 4.1: Slopes used by the classical Runge-Kutta method

Example

Apply R.K method of 4th order to find approximate value of y at x = 0.2 if

$$\frac{dy}{dx} = x + y^2 \tag{4.13}$$

Given that y = 1 when x = 0 in steps of h = 0.1.

Solution

$$\frac{dy}{dx} = x + y^2 \tag{4.14}$$

$$f(x,y) = x + y^2 \tag{4.15}$$

$$k_1 = h * f(x_0, y_0) \tag{4.16}$$

k_1 = 0.1 * 1 = 0.1

$$k_2 = h * f(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}) \tag{4.17}$$

k_2 = 01 * f(0.05, 1.05) = 0.11525

$$k_3 = h * f(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}) \tag{4.18}$$

k_3 = 0.1 * f(0.05, 1.05762) = 0.11686

$$k_4 = h * f(x_0 + h, y_0 + k_3) \tag{4.19}$$

k_4 = 0.1 * f(0.05, 1.11686) = 0.13474

$$k = \frac{1}{6}(k_1 + 2 * k_2 + 2 * k_3 + k_4) \tag{4.20}$$

k = (0.1 + 2*0.11525 + 2*0.11686+0.13474)/6 = 0.1165

The approximate value of y = y_0 + k = 1.1165

y(0.1) = 1.1165

y at x = 0.2

$$f(x,y) = x + y^2 \tag{4.21}$$

f(x_0, y_0) = 0.1 + $(1.165)^2$ = 1.3466

$$k_1 = h * f(x_0, y_0) \tag{4.22}$$

k_1 = 0.1 * f(1.3466) = 1.3466

$$k_2 = h * f(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}) \tag{4.23}$$

k_2 = 01 * f(0.15, 1.18383) = 0.15514

$$k_3 = h * f(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}) \tag{4.24}$$

k_3 = 0.15758

$$k_4 = h * f(x_0 + h, y_0 + k_3) \tag{4.25}$$

k_4 = 0.18233

$$k = \frac{1}{6}(k_1 + 2 * k_2 + 2 * k_3 + k_4) \tag{4.26}$$

k = 0.1571

Approximate value of y = y_0 + k = 1.2736

4.4 Equation model used in this project

Under this topic we discuss about the mathematical models which is used in this project.

State Variable	Meaning
x	Number of prey
y	Number of predator
a,b,c,d	Positive real parameters

Table 4.1: Notation used in figure

[2]

CHAPTER 5
Results

5.1 The exhibit two behaviors of two equation model

Let's assume that, the number of preys and predators as a function of time from the model with following initial conditions:

The number of preys $X_0 = 100$,

The number of predators $Y_0 = 40$,

The initial time $t_0 = 0$,

Total Length of time = 100,

Number of steps = 5000,

with the a = 0.5, b = 0.01, c = 0.5, d = 0.01 positive parameters.

5.2 Solutions by using the Runge Kutta Method

The answer obtained at the end of this project is the C code that was written for the mathematical model. When addressing the end result, it is important to know what that result is. To estimate the size of the populations of predators and prey, this is highly useful.

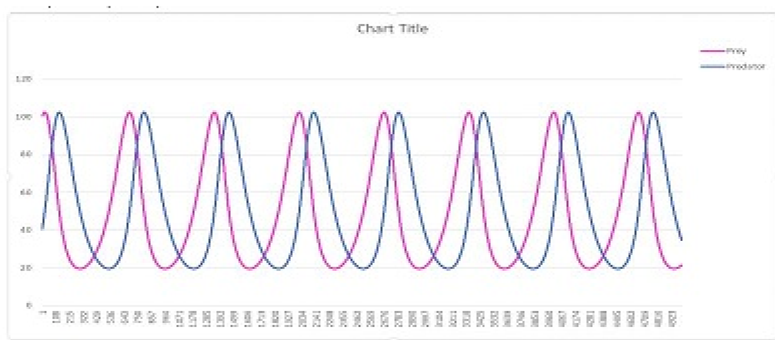

Figure 5.1: the number of preys and predators as a function of time from the model using a = 0.5, b = 0.01, c = 0.5, d = 0.01

The above graph was drawn by using the following output from the c code. (Only a part of output is included here.)

```
0.00000000    100.19617348    40.40280444
0.02000000    100.38458796    40.81123480
0.04000000    100.56508487    41.22531489
0.06000000    100.73750619    41.64506615
0.08000000    100.90169467    42.07050749
0.10000000    101.05749393    42.50165517
0.12000000    101.20474866    42.93852271
0.14000000    101.34330480    43.38112072
0.16000000    101.47300966    43.82945676
0.18000000    101.59371216    44.28353527
0.20000000    101.70526295    44.74335736
0.22000000    101.80751466    45.20892074
0.24000000    101.90032206    45.68021953
0.26000000    101.98354225    46.15724416
0.28000000    102.05703486    46.63998123
0.30000000    102.12066227    47.12841334
```

```
0.32000000     102.17428981    47.62251901
0.34000000     102.21778595    48.12227248
0.36000000     102.25102253    48.62764363
0.38000000     102.27387498    49.13859780
0.40000000     102.28622250    49.65509571
0.42000000     102.28794833    50.17709325
0.44000000     102.27893991    50.70454145
0.46000000     102.25908916    51.23738626
0.48000000     102.22829264    51.77556850
0.50000000     102.18645181    52.31902370
0.52000000     102.13347324    52.86768198
0.54000000     102.06926882    53.42146800
0.56000000     101.99375597    53.98030076
0.58000000     101.90685786    54.54409357
0.60000000     101.80850364    55.11275395
0.62000000     101.69862861    55.68618351
0.64000000     101.57717446    56.26427789
0.66000000     101.44408942    56.84692669
0.68000000     101.29932851    57.43401338
0.70000000     101.14285369    58.02541529
0.72000000     100.97463405    58.62100350
0.74000000     100.79464598    59.22064287
0.76000000     100.60287336    59.82419194
0.78000000     100.39930766    60.43150297
0.80000000     100.18394815    61.04242190
0.82000000     99.95680201     61.65678836
0.84000000     99.71788446     62.27443568
0.86000000     99.46721887     62.89519095
```

```
0.88000000     99.20483688     63.51887500
0.90000000     98.93077849     64.14530250
0.92000000     98.64509212     64.77428201
0.94000000     98.34783473     65.40561605
0.96000000     98.03907179     66.03910119
0.98000000     97.71887742     66.67452818
1.00000000     97.38733434     67.31168204
1.02000000     97.04453389     67.95034222
1.04000000     96.69057606     68.59028269
1.06000000     96.32556946     69.23127218
1.08000000     95.94963125     69.87307430
1.10000000     95.56288711     70.51544774
1.12000000     95.16547119     71.15814648
1.14000000     94.75752601     71.80091999
1.16000000     94.33920232     72.44351348
1.18000000     93.91065906     73.08566811
1.20000000     93.47206316     73.72712128
1.22000000     93.02358943     74.36760688
1.24000000     92.56542036     75.00685554
1.26000000     92.09774596     75.64459499
1.28000000     91.62076358     76.28055026
1.30000000     91.13467764     76.91444410
1.32000000     90.63969946     77.54599721
1.34000000     90.13604701     78.17492860
1.36000000     89.62394460     78.80095595
```

[6]

CHAPTER 6
Discussion and Conclusion

6.1 Discussion

One of the simplest models, the Lotka-Volterra model was created to simulate how prey and predator populations interact in the environment. To obtain the numerical solutions for this model, there are essentially two main approaches. Laplace Adomian Decomposition Method and the Runge-Kutta Method (LADM). But in this project, we are merely employing the Runge-Kutta method to solve the problem..

The biggest problem with this project was that the numerical values we obtained from solving the equations weren't exactly the same as the numbers we obtained from the coded program. As a result, we had to manually fix them and contrast the two differences. On the other hand, when we want to approximation the model behavior, having the time-consuming RK technique automated is really handy.

The objective of this study is to compare and contrast the numerical approach and the programmed method for solving the Lotka-Volterra model. We can observe the distinction between the two approaches when we display the graph using the program's output and contrast it with the numerical answer. By looking into this more, we might be able to narrow or perhaps erase the difference between those two approaches.

6.2 Conclusion

In conclusion, the Lotka-Volterra Predator-Prey Model is a fundamental approximation of the intricate ecology of our planet. It assumes the predator will only have one prey. However, as the number of variables increases, the equations get more complicated and demand additional numerical simulations and mathematical analysis. Complex processes can be examined by selecting the right mathematical model (Runge Kutta Method), selecting the right model parameters, and simulating the model. In our example, the numerical solution was obtained using the Runge-Kutta method and C code was created.

6.3 Summary

- Chapter 01 - This chapter contain the Introduction about the Lotka Volterra Method.
- Chapter 02 - The literature review was included here.
- Chapter 03 - Problem statement was simply included to the third chapter.
- Chapter 04 - Methodology- method that was used to obtain the final goal was thoroughly discussed in this chapter.
- Chapter 05 - Fifth chapter include the final out come and difficulties that need to be discuss.

References

[1] Lennart Edsberg. *Introduction to computation and modelling for the differential equations*. Wiley; 1st edition, 2008.

[2] Elsevier. Runge kutta 4th order de. https://www.youtube.com/watch?v=q87L9R9v274, 2014.

[3] Elsevier. Lotka–volterra model. https://www.sciencedirect.com/, 2022.

[4] E. Juarlin. Journal of physics: Conference series. *Solution of Simple Prey-Predator Model by Runge Kutta Method*, 2019.

[5] D V G Rao K N Murty. Approximate analytical solutions of general lotka volterra equations. *Journal of mathematical analysis and applications*, 1985.

[6] Dimitrov M Kunis F. Investigating lotka volterra model using computer simulation. *Open schools journal for open science*, 2020.

[7] unknown. Lotka–volterra equations. https://en.wikipedia.org/wiki/Lotka%E2%80%93Volterra_equations, July 2022.

CHAPTER 7
AppendixB

7.1 C implementation of The Lotka-Volterra model

```
#include<stdio.h>
#include<math.h>

double fx( double x, double y)
{
double a = 0.5;
double b = 0.01;
return a*x-b*x*y;
}
double fy(double x,double y)
{
double c = 0.5;
double d = 0.01;
double var_5;
var_5 = d*x*y-c*y;
return var_5;
}

int main(){
FILE *fp;
fp =fopen("My_code2.txt","w+");
if(fp==NULL)
{
```

```
printf("\n unable to open");
return 0;
}
double i , l;
double t_0, x_0 , y_0 ,  t_n , h , x_1 , y_1 ;
double k_1, k_2, k_3, k_4, k;
double m_1, m_2, m_3, m_4, m;

printf("t_0 = "); //Initial value for t
scanf("%lf",&t_0);
printf("x_0 = "); //Initial value for x
scanf("%lf",&x_0);
printf("y_0 = "); //Initial value for y
scanf("%lf",&y_0);
printf("Enter the calculation point tn = ");
scanf("%lf",&t_n);
printf("Enter the number of steps = ");
scanf("%lf",&l);
h = (t_n - t_0) / l;
printf("\n tn\t\t x1\t\t y1\t\t");
printf("\n");

for ( i=0 ; i<=l ; i++ )
{
k_1 = h * ( fx ( (x_0) , (y_0)  ) );
m_1 = h * ( fy ( (x_0) , (y_0)  ) );

k_2=h*(fx((x_0+k_1/2),(y_0+m_1/2)));
```

```
m_2=h*(fy((x_0+k_1/2),(y_0+m_1/2)));

k_3=h*(fx((x_0+k_2/2),(y_0+m_2/2)));
m_3=h*(fy((x_0 +k_2/2),(y_0+m_2/2)));

k_4=h*(fx((x_0+k_3),(y_0+m_3)));
m_4=h*(fy((x_0+k_3),(y_0+m_3)));

k = ( k_1 + ( 2 * k_2 ) + ( 2 * k_3 ) + k_4 ) / 6;
m = ( m_1 + ( 2 * m_2 ) + ( 2 * m_3 ) + m_4 ) / 6;

x_1 = x_0 + k;
y_1 = y_0 + m;

printf("%.8lf\t\t %0.8lf\t\t %0.8lf\n",t_0,x_1,y_1);
fprintf(fp,"%0.8lf\t\t %0.8lf\t\t %0.8lf\n",
t_0,x_1,y_1);
t_0 = t_0 + h;
t_n = t_0 + h;
x_0 = x_1;
y_0 = y_1;

}
}
```