Name: Emma Furr
Perm #: 4076063
Math 104B Atzberger

Final Project Report - Lotka Volterra (Predator-Prey Equation)

Motivation:

In this final project, we examined the Lotka Volterra Model for predator-prey and wanted to solve the systems of ordinary differential equations and analyze the population trends over a fixed amount of time for both predator and prey. The system of equations was created to describe the dynamics of an ecosystem, is the simplest model of predator-prey interactions and are well-posed first-order non-linear ODEs described by

$$\boldsymbol{dx} / \boldsymbol{dt} = \mathrm{a}\boldsymbol{x} - \mathrm{b}\boldsymbol{xy}$$
$$\boldsymbol{dy} / \boldsymbol{dt} = \mathrm{c}\boldsymbol{xy} - \mathrm{d}\boldsymbol{y}$$

where
x = number of prey
y = number of predators
dx / dt = the instantaneous rate of prey respectively
dy / dt = the instantaneous rate of predators respectively
t = time
a = the natural growing rate of rabbits, when there are no fox
b = the natural dying rate of rabbits, due to predation
c = the factor that describes how many rabbits eaten creates the reproduction of a fox
d = the natural dying rate of fox when there are no rabbits,
and a, b, c, and d are biologically determined.

The prey equation can be described as the rate at which new prey are born minus the rate at which prey are killed off. The predator equation can be described as the rate at which predators consume prey minus the natural death rate of predators.

In this project, we used three numerical analysis methods to approximate the solution to our Lotka Volterra ODEs including forward Euler's method, fourth-order Runge-Kutta method, and Adam's Bashforth-Moulton Predictor-Corrector method of order four. We will compare these approximations to the actual solutions of the ODEs (or the SciPy integration method for ODEs). We will then compare graphs and relative errors between the methods to see which method gives the most accurate approximation.

The questions we plan to answer are as follows: What are the differences in effects and oscillations that occur for the various methods implemented? Do all the methods used seem to give accurate trends for the populations in comparison? Which approximation methods are the most accurate compared to the accurate SciPy integration method? How do initial values affect the results?

Name: Emma Furr
Perm #: 4076063
Math 104B Atzberger

Problem Statement:

The problem we are focusing on deals with two different populations. We have one predator population (foxes, respectively) and one prey population (rabbits, respectively). In this project, we intend to investigate the trends of both populations and compare them against one another. We will see how the approximation methods compare to the SciPy integration method, and the difference in the population trends.

Methods:

The three methods we used, including forward Euler, fourth-order Runge-Kutta, and Adam's Predictor-Correct of order 4, were the ones we found most appropriate in approximating our well-posed system of ordinary differential equations or the Lotka Volterra model.

The first method we chose to use was forward Euler's method, which is the most basic explicit first-order numerical method for solving ODEs with a given initial value.

Euler's method constructs $w_i \approx y(t_i)$, for each $i = 1, 2, \ldots, N$, by deleting the remainder term. Thus, Euler's method is

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + hf(t_i, w_i), \quad \text{for each } i = 0, 1, \ldots, N-1. \tag{5.8}$$

Euler's method is a Taylor's method of order one and has a local truncation error of O(h). The different step size values change the severity of the method's accuracy. With h = 0.1, Euler's method will not accurately implement the Lotka Volterra Model, so a smaller step-size will mitigate the inaccuracy. We predict this method to be the least accurate in comparison with the other two methods.

The next method we used was fourth-order Runge-Kutta, which has the high-order local truncation error of the Taylor Methods but eliminates the need to compute and evaluate the derivatives of f(t,y). This method has a local truncation error of O(h^4) and is very accurate at small step-sizes. The fourth-order Runge-Kutta method is as follows:

$$\begin{aligned}
w_0 &= \alpha, \\
k_1 &= hf(t_i, w_i), \\
k_2 &= hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_1\right), \\
k_3 &= hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_2\right), \\
k_4 &= hf(t_{i+1}, w_i + k_3), \\
w_{i+1} &= w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),
\end{aligned}$$

Name: Emma Furr
Perm #: 4076063
Math 104B Atzberger

We predict that this method will give very close approximations to the actual solution and will be the best method to solve for the ODEs.

The final method we used was Adam's Predictor-Corrector of order four, which is the combination of the explicit Adams Bashforth method to predict:

$$w_{4p} = w_3 + \frac{h}{24}[55f(t_3, w_3) - 59f(t_2, w_2) + 37f(t_1, w_1) - 9f(t_0, w_0)].$$

and the implicit Adams Moulton method to improve the prediction, or correct:

$$w_4 = w_3 + \frac{h}{24}[9f(t_4, w_{4p}) + 19f(t_3, w_3) - 5f(t_2, w_2) + f(t_1, w_1)].$$

The starting values are acquired from the fourth-order Runge-Kutta method. This method is, also, very accurate at small step-sizes. We predict this method to give very adequate approximations.

In this project, we observed two populations, predator and prey. We observed these two populations over time from t = 0 to t = 100 and used a step-size of h = 0.01. The parameters a, b, c, and d are biologically determined, so we will examine a system where a = 0.9, b = 0.1, c = 0.1, and d = 0.4. Our initial values for prey and predators are x_0 = 10 and y_0 = 5, respectively.

Results:

We utilized the three methods to solve for approximations to the Lotka Volterra equations. We found the relative errors of the three methods at different time steps and implemented graphs of the oscillations, which is shown at the bottom of the report. We compared these results to see which method gave the most accurate approximation to the true values.

To find the actual solutions to the Lotka-Volterra equations, we used SciPy integration, which is a scipy.integrate.odeint() function. We then calculated the relative errors at every time step, in which times t = 10, t = 30, t = 50, t = 70, and t = 90 are shown in the tables below.

From our results, we found that forward Euler's method performs the worst out of all the methods used, as predicted. It is evident in the table of relative errors as well as the graph of oscillations that the approximations are the least accurate to the real solutions. In addition, we found that Runge-Kutta of order four performs the best and is, therefore, the most accurate. This is evident from both the graph and relative errors for both the predator and prey. Lastly, we found that Adam's Predictor-Corrector of order four performs extremely well, just like Runge-Kutta. However, this method is slightly worse than Runge-Kutta, and the small difference is evident in the relative error table.

Name: Emma Furr
Perm #: 4076063
Math 104B Atzberger

For example, at t = 70, Euler's method has a relative error for predators of 0.458808 and a relative error for prey of 0.36351, Adam's Predictor-Corrector has a relative error for predators of 0.027304 and a relative error for prey of 0.0090041, and finally, Runge-Kutta has a relative error for predators of 0.0046246 and a relative error for prey of 0.002451. It is evident by these relative errors and the relative errors in the table below that Runge-Kutta performs the best and has the most accurate approximate solutions.

These results were expected because in theory, usually error in Euler method is higher than higher order Runge-Kutta because the truncation error in higher order methods is less compared to Euler. Adam's predictor-corrector and Runge-Kutta were expected to be very accurate and similar in accuracies and in fact, our results confirm this.

In our code, we showed graphs of how the population of prey and predators interacted over time, which are present at the end of this report. If we take a look at the graphs from time t = 0 to t = 100 of all three methods and the SciPy integration, we can see that as the population of prey increases, the number of predators also increases until the predators begin to kill off the prey faster than they can reproduce. The population of prey starts to decrease which causes the predators to starve and their population begins to decline. It creates an oscillation of both populations rising and falling over time. The graph of Euler's method is the least comparable to the actual solution. The graph of Adam's Predictor-Corrector oscillates very closely to the actual graph, but slightly off in comparison with the graph of Runge-Kutta, which closely resembles the true solution.

The Lotka Volterra Model can be relevant to real life in the fact that it is an interest to biologists who are trying to understand how biological communities are structured and sustained. However, there are some problems that do arise with the Lotka Volterra equations because it does not factor in realism and environmental factors. This model focuses only on the predator and prey interactions and ignores competition, disease, and mutualism. This model is not very robust, or stable. One main problem with the model is that the population can never become extinct, but they can become arbitrarily small. When you look at the values for prey and predators at x_0 = 0 and y_0 = 0, it shows to be very unstable. The model behavior shows to be unnatural and having no asymptotic stability.

With numerical methods comes error when trying to solve real-life problems. This is due to several factors that the methods cannot take into consideration when approximating real solutions. In our project, we studied the Lotka Volterra Model and solved the well-posed system of ODEs using three different numerical methods. We analyzed the population trends over a time interval from t = 0 to t = 100 for both predator and prey, and found that fourth-order Runge-Kutta gave the best results. Although this model may not follow the true solutions exactly because the solutions are deterministic and time is continuous, our approximations to this initial value problem were very close to the real solution. This project provided a practical application of the numerical methods we have been learning in this course to a real-life model.
Table of Approximations:

Name: Emma Furr
Perm #: 4076063
Math 104B Atzberger

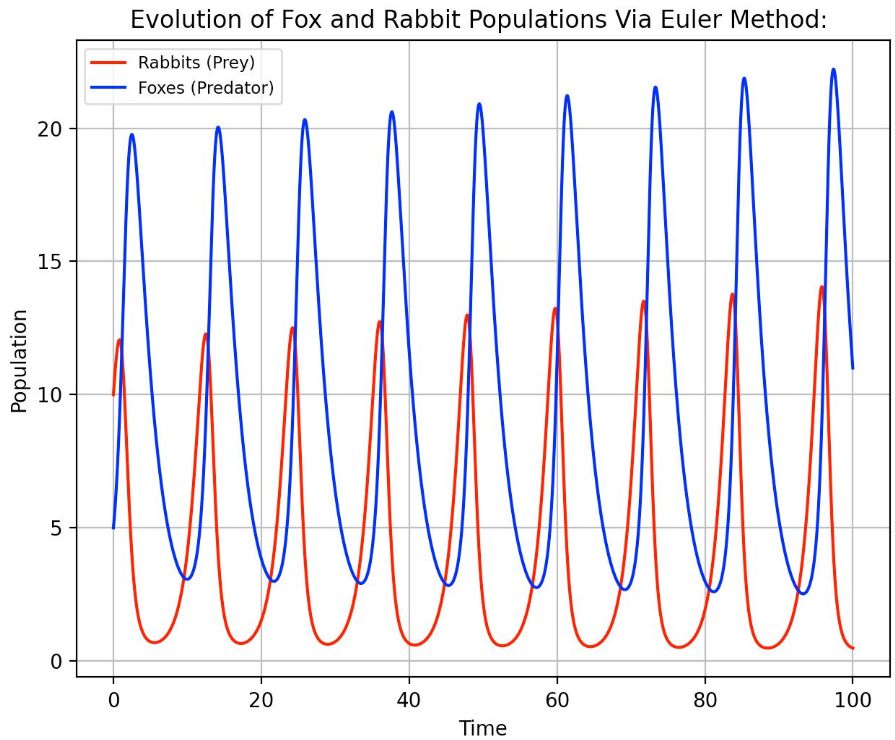
	Euler's		Runge-Kutta		Adam's PredCor		Actual ODE	
Time (t)	Predator	Prey	Predator	Prey	Predator	Prey	Predator	Prey
10	3.068604	4.097594	3.13795	4.18336	3.139547	4.18608	3.13789	4.18091
30	6.368668	0.724742	6.17029	0.84675	6.168734	0.85052	6.17614	0.84603
50	19.85005	2.179018	17.1758	1.59969	17.10569	1.59371	17.1964	1.60626
70	2.971906	6.73186	5.51679	10.6024	5.613419	10.6717	5.4914	10.5765
90	5.295038	0.658576	3.74542	1.80396	3.727839	1.86131	3.75285	1.79539

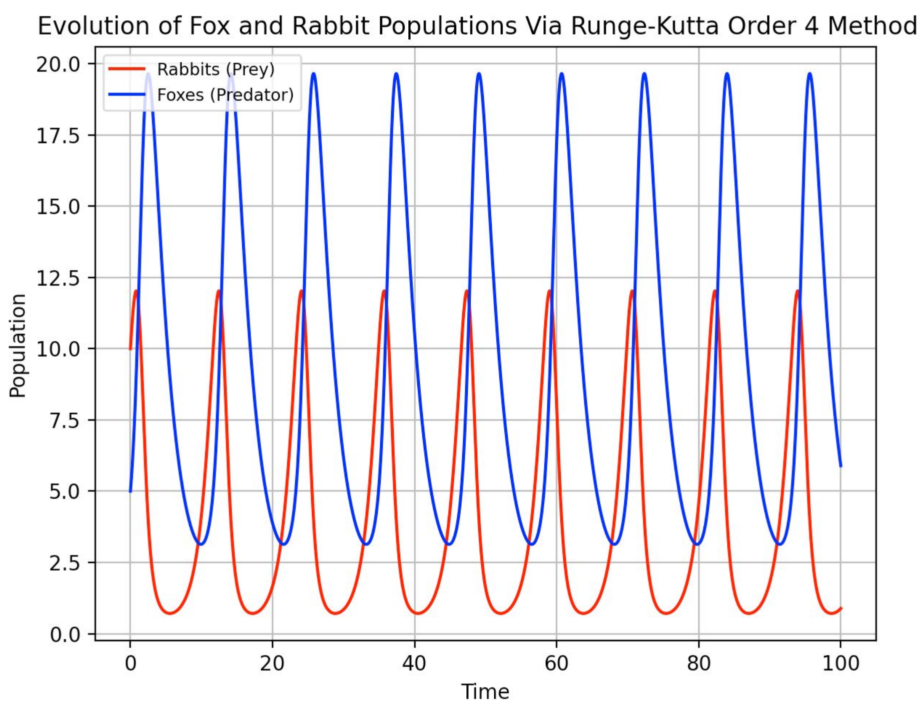
Table of Relative Errors:

	Euler's		Runge-Kutta		Adam's PC	
Time (t)	Predator	Prey	Predator	Prey	Predator	Prey
10	0.022082	0.01993	1.81x10^-5	0.00058	0.000526	0.0012356
30	0.031173	0.14336	0.0009458	0.000848	0.001198	0.0053076
50	0.154315	0.35658	0.0011981	0.004085	0.005274	0.0078127
70	0.458808	0.36351	0.0046246	0.002451	0.027304	0.0090041
90	0.410939	0.63318	0.0019788	0.004739	0.006663	0.0367188

Name: Emma Furr
Perm #: 4076063
Math 104B Atzberger

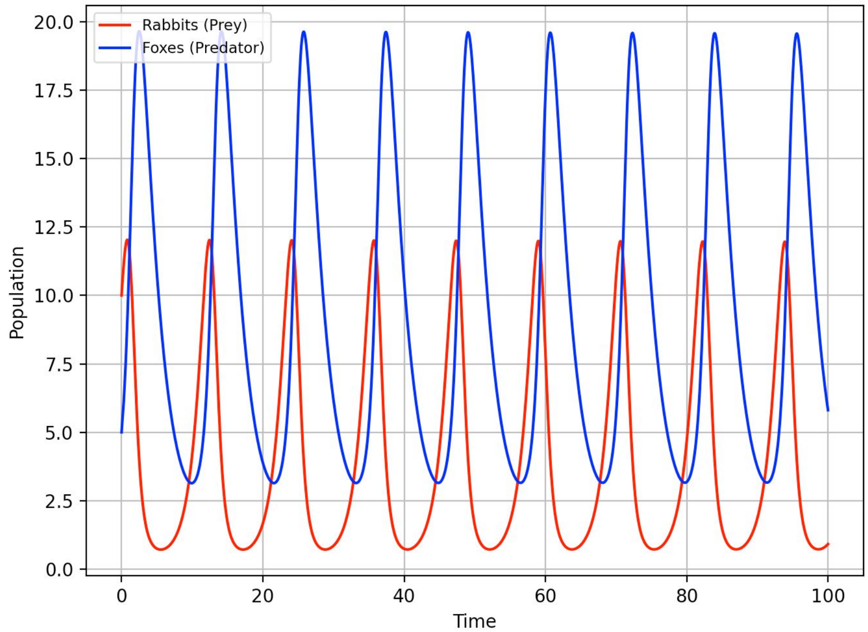
<u>Graphs:</u>



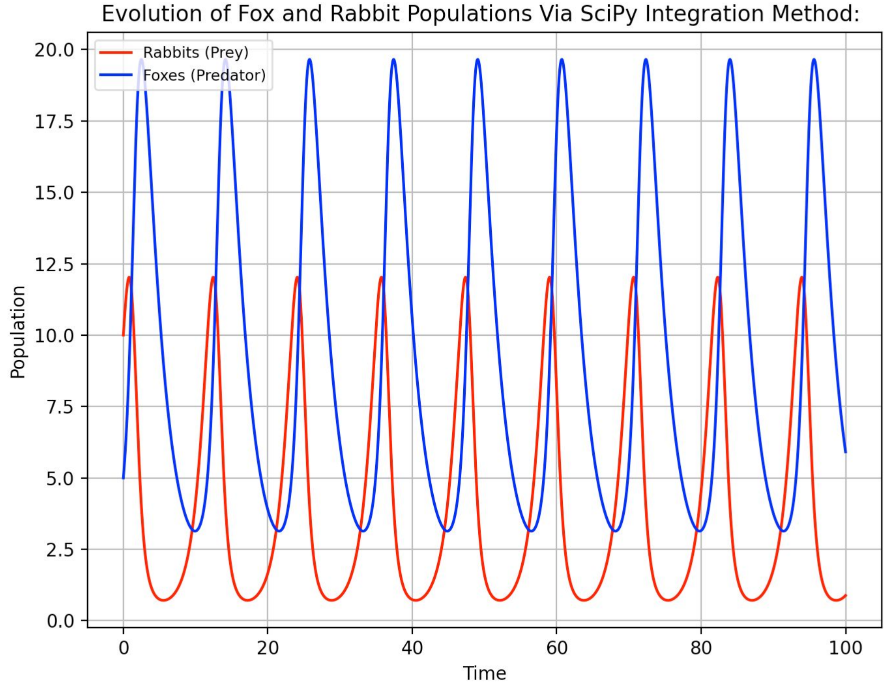


Name: Emma Furr
Perm #: 4076063
Math 104B Atzberger

Evolution of Fox and Rabbit Populations Via Adams Predictor Corrector Order 4 Method:



Evolution of Fox and Rabbit Populations Via SciPy Integration Method:



Name: Emma Furr
Perm #: 4076063
Math 104B Atzberger

Group Members:
Kelcie Opp 3754850
Amit Yativ 6657977
Shawden Soltani 4446191
Kameron Quackenbush 8868416
Niko Kalpakian 4416723