PRA3021 Assignment 3

Tom Kores Lesjak

September 2024

1 Task 1

1.1 Question 1

Question: What is the maximum and minimum number of deer and wolves in the 50 year time span?

The maximum and minimum population of deer during the 50 year time span is; 375 and 40 deer. The maximum and minimum population of wolfs during the 50 year time span is; 148 and 9 wolfs. The phase diagram of deer and wolf population is given in figure 1.

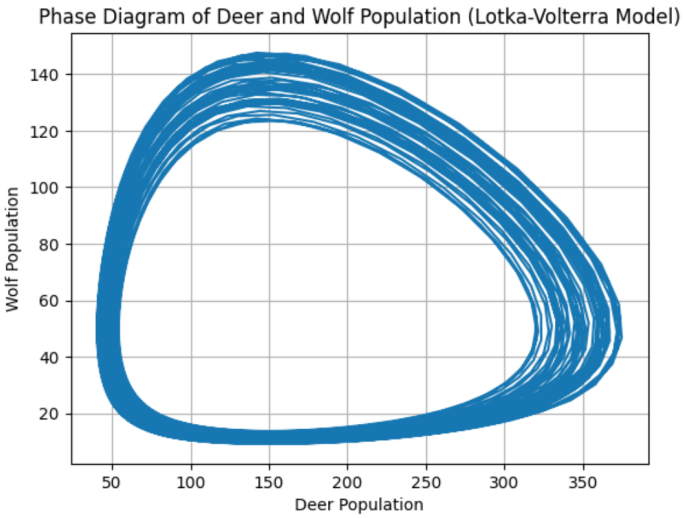


Figure 1: Phase diagram with deer population plotted against wolf population over 50 years

1.2 Question 2

Question: How does changing the parameters of the model affect the modelled population? Describe the effects for each parameter.

$$\frac{dx}{dt} = \alpha x - \beta xy \tag{1}$$

$$\frac{dy}{dt} = \delta xy - \gamma y \tag{2}$$

In the Lotka-Volterra predator-prey model, parameters α, β, δ, and γ influence the dynamics of prey (deer) and predator (wolves) populations. α is the prey growth parameter and increases deer population growth, leading to larger oscillations, while lower α slows growth and can result in collapse. β is

the parameter which determines how quickly wolves consume deer. Higher β results in sharper prey declines and stronger predator oscillations, while lower β stabilizes the system. δ controls predator population growth in response to prey. Higher δ speeds up predator growth. Whilst lower δ slows predator cycles. γ is the predator death parameter which dictates how fast wolves die off when prey is scarce. A higher γ causes predator numbers to drop sharply, allowing prey recovery, while a lower γ leads to more prolonged predator populations. The plot for the solution for these equations can be seen in figure 2.

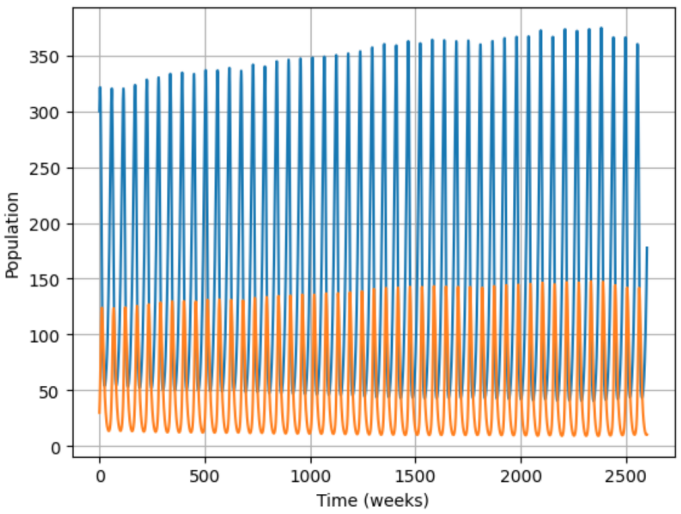


Figure 2: The populations of deer and wolves with respect to time over the 50 years in weekly intervals

1.3 Question 3

Question: The pattern obtained should be cyclical. Approximately how long does it take for the deer population population to go from its minimum to its maximum number within one cycle?

It takes 36 weeks for the deer population to go from its minimum to its maximum.

1.4 Question 4

Question: Run your Euler's function on the deer-wolf model introduced earlier, Compare the results of Euler's method against SciPy's solver by plotting the deer and wolf populations from each method over the time span. Are the results of two methods the same? Why or why not?

The results of Euler's method and SciPy's `solve_ivp` solver are not the same, and the differences grow more noticeable over time. This discrepancy

arises due to the nature of the two methods. Euler's method is a simple first-order numerical technique that approximates the solution by stepping forward in time using the derivative at the current point. Euler's method is prone to significant errors. This is because it only considers the first-order approximation of the rate of change, leading to error accumulation with each step. As a result, the method may exhibit instability or drift away from the true solution over long periods. Figure 3 shows the comparison for the deer and wolf populations from each method, yet it is not accurate since I could not get the amplitudes to line up by correctly.

In contrast, SciPy's `solve_ivp` solver uses more advanced numerical methods. These methods evaluate the function at multiple points within each time step, providing a more accurate estimate of the solution. Additionally, `solve_ivp` automatically adjusts the step size during the integration process, reducing the error accumulation that is common with fixed-step methods like Euler's. Therefore, `solve_ivp` produces a solution that is more accurate and stable over time, especially for systems that exhibit oscillatory or nonlinear behavior like the Lotka-Volterra model.

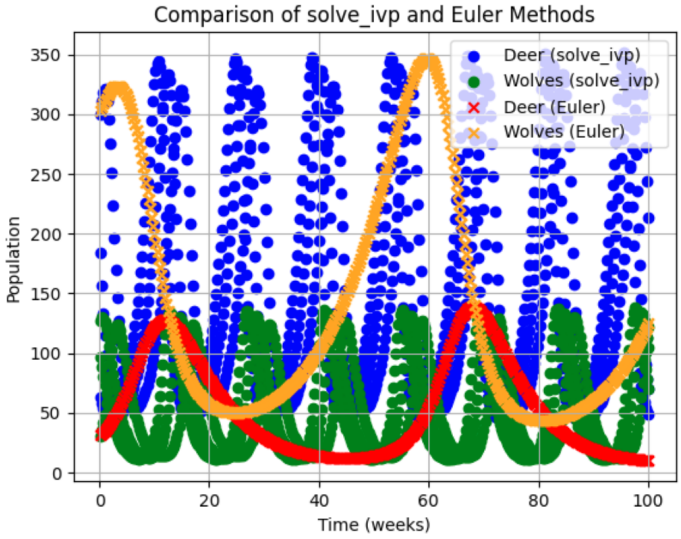


Figure 3: Results of Euler's method against SciPy's solver by plotting the deer and wolf populations from each method over the time span of 50 years

1.5 Question 5

Question: What is the effect of the time step, h, in Euler's method?

The time step h significantly impacts both the accuracy and stability of the numerical solution. A smaller time step improves accuracy because the method evaluates the derivative more frequently, reducing error and providing a better

approximation. However, smaller time steps also increase the computational cost as more iterations are required to cover the same time span. Larger time steps reduce the computational load but can introduce greater errors. Additionally, larger time steps can cause stability issues, leading to oscillations or divergence from the true solution.

1.6 Question 6

Question: Give a suggestion for how you could improve on Euler's method.

One way to improve upon Euler's method is by using higher-order numerical methods. This would significantly increase accuracy without a dramatic increase in computational complexity. While Euler's method only evaluates the derivative at the beginning of each time step, you could improve it by taking multiple derivative evaluations at intermediate points within each step. This would provide a much better estimate of the solution's behavior across the entire time step, reducing error and increasing accuracy. Another improvement could be the use of adaptive step sizes, where the step size h is dynamically adjusted based on the local error estimate. In regions where the solution changes rapidly, the method reduces the step size to maintain accuracy, and in regions where the solution changes slowly, it increases the step size to reduce computational cost.

2 Task 2

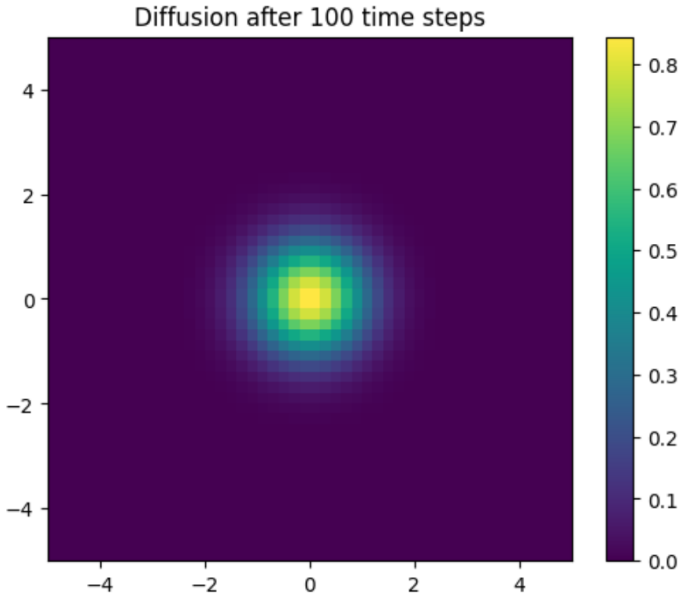


Figure 4: Implementing the finite-difference scheme for the diffusion equation.

To implement the finite-difference scheme for the diffusion equation, we break the continuous space and time into discrete points. The diffusion equation describes how quantities like heat spread over time. In the finite-difference approach, we approximate the spatial and time derivatives using the values of the quantity at neighboring points. For the spatial part, we use the central difference method, which approximates the second derivative by looking at the difference between a point and its neighbors. For the time derivative, we use the forward difference method, which calculates the rate of change of the quantity by comparing its value at the current time step and the next. These approximations allow us to update the value of the quantity at each point in space as time progresses. By repeating this process across all points and time steps, we can simulate how the quantity spreads over time.

2.1 Question 1

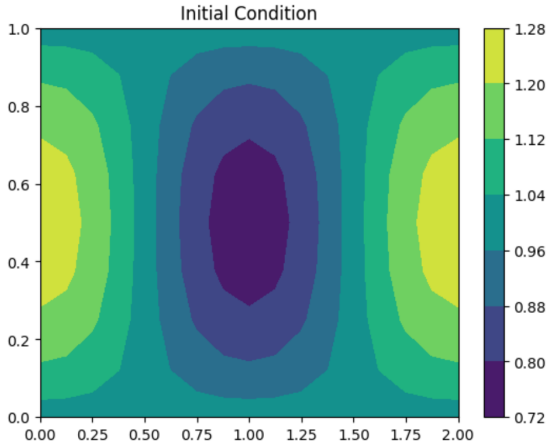
The visualizations show how the initial distribution

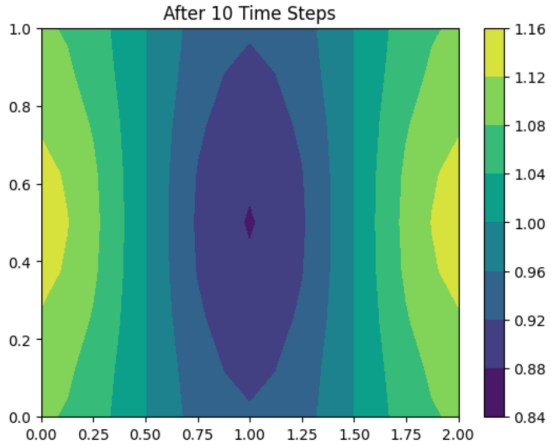
$$u(x, y) = 1 + \cos(x) \cdot y(1 - y)$$

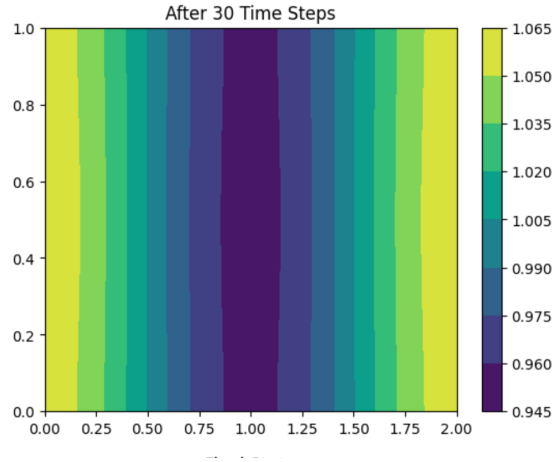
spreads over time due to diffusion. At the start ($t = 0$), the function is tightly concentrated around the initial condition. As time progresses (after 10 and 30 steps), the function gradually smooths out, indicating that the diffusion process is spreading the initial concentration evenly across the grid. By the final state, the solution has diffused significantly, showing a more uniform distribution as the system approaches equilibrium. This process can be seen in section 3.1.

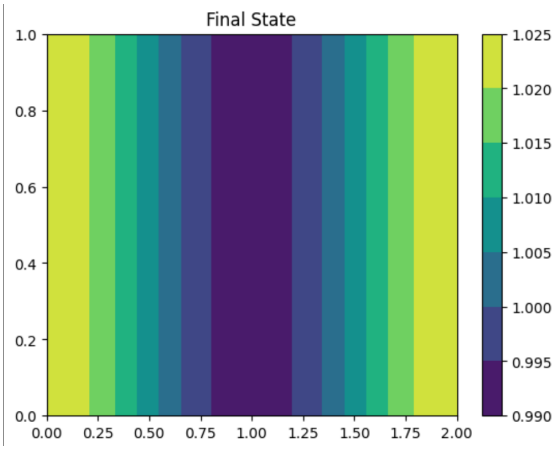
3 Appendix

3.1 Appendix A






After 30 Time Steps
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
1.065
1.050
1.035
1.020
1.005
0.990
0.975
0.960
0.945


Final State
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
1.025
1.020
1.015
1.010
1.005
1.000
0.995
0.990