Three-Species Predator-Prey Model: Rabbits, Foxes, and Wolves

Introduction

The predator-prey model explores the dynamic interactions between different species within an ecosystem, focusing on how populations of predators and prey affect one another's growth and survival. In this simulation, we observe the interplay between rabbits, foxes, wolves, and resources. Rabbits, as herbivores, rely on the available resources for survival, while foxes and wolves, as predators, hunt rabbits and each other. The model simulates how these interactions evolve over time, shedding light on the delicate balance of nature and how changes in one population can lead to cascading effects across the entire system.

Model

Variables and Parameters

1. Rabbit Population - x

$$\frac{dx}{dt} = ax - bxy - cxw$$

- Terms
    - ***ax*** : Rabbit reproduction rate.
    - ***-bxy*** : Rabbits consumed by predator foxes.
    - ***-cxw*** : Rabbits limited by environment (e.g. grass, vegetation).

2. Fox Population - y

$$\frac{dy}{dt} = dxy - ey - fyz$$

- Terms
    - ***dxy*** : Fox population growth through predation on rabbits.
    - ***-ey*** : Natural death of foxes.
    - ***-fyz*** : Foxes consumed by predator wolves.

3. Wolf Population - z

$$\frac{dz}{dt} = gyz - hz$$

- Terms
    - ***gyz*** : Wolf population growth through predation on foxes.
    - ***-hz*** : Natural death of wolves
4. Environmental Factor - w

$$\frac{dw}{dt} = i - jw - kxw$$

- Terms
    - ***i*** : Natural replenishment of environmental resource.
    - ***-jw*** : Natural decay of environment.
    - ***-kxw*** : Rabbits consuming vegetation.

Constants

- ***a*** : Rabbits reproduction rate.
- ***b*** : Predation rate by foxes.
- ***c*** : Vegetation dependency rate for rabbits.
- ***d*** : Growth rate of foxes due to hunting on rabbits.
- ***e*** : Natural death rate of foxes.
- ***f*** : Predation rate by wolves.
- ***g*** : Growth rate of wolves due to predation on foxes.
- ***h*** : Natural death rate of wolves.
- ***i*** : Resource replenishment rate.
- ***j*** : Natural decay of resources.
- ***k*** : Rate of resource depletion by rabbits.

Numerical Solution

When solving differential equations like the predator-prey model, **choosing an appropriate numerical method** is crucial for ensuring that the results are accurate and stable. The main methods for solving ordinary differential equations (ODEs) are:

1. **Euler's Method** - Used in my code implementation
2. **Runge-Kutta Methods**
3. **Implicit Methods**

Method

Using **Euler's method**, a simple and computationally efficient numerical approach for solving ordinary differential equations (ODEs). It approximates the solution by discretizing time and updating the system's state based on the derivative (rate of change) at each step.

It's easy to implement and requires only the current state and its derivative to update the next state.

It's computationally inexpensive, ideal when dealing with many time steps or when high precision isn't critical.

It's suitable for exploring general trends in population dynamics, which is your main goal.

It's often used as a starting point before considering more accurate or complex methods, especially when you're testing model behavior.

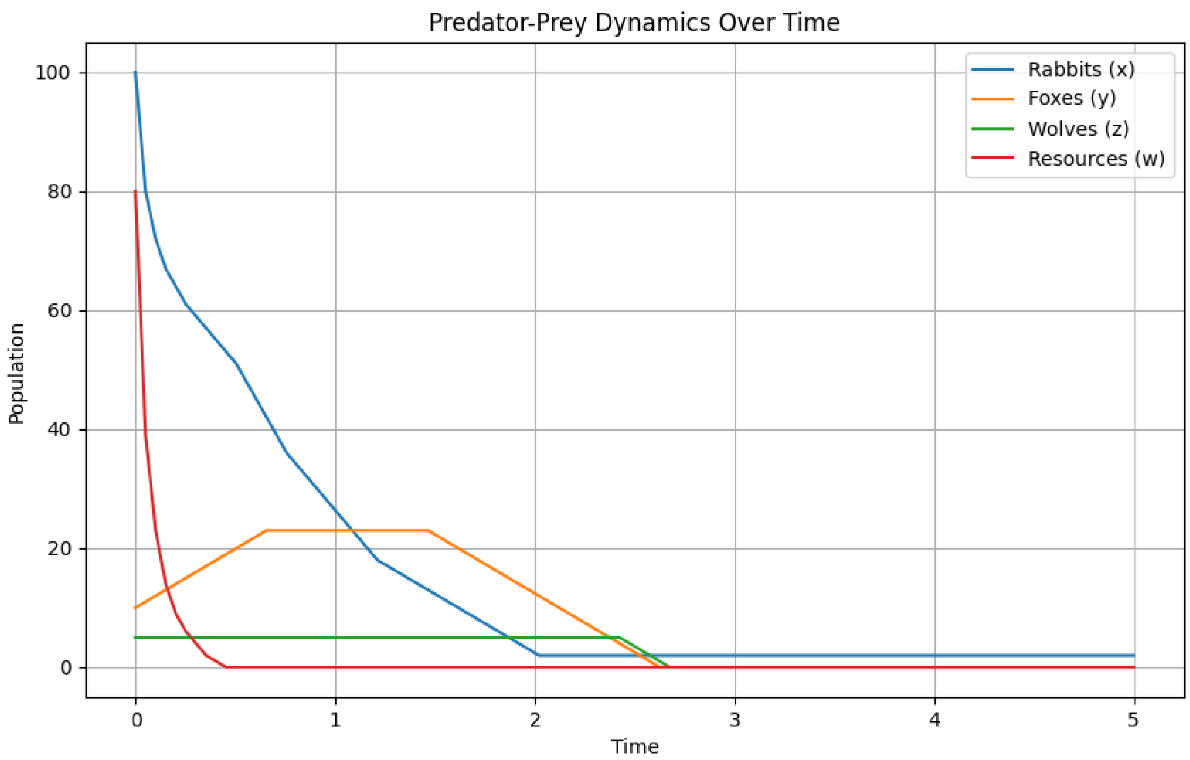
Implementation

```
import numpy as np
from src.plot import plot_results

# Derivative function for predator-prey model
def predator_prey_derivatives(state, a, b, c, d, e, f, g, h, i, j, k):
    x, y, z, w = state  # Unpack state variables

    # Derivatives
    dx_dt = a * x - b * x * y - c * x * w
    dy_dt = d * x * y - e * y - f * y * z
    dz_dt = g * y * z - h * z
    dw_dt = i - j * w - k * x * w

    return [dx_dt, dy_dt, dz_dt, dw_dt]


# Function to run the simulation
def simulate(initial_state, params, t_span, t_points):
    # Initialize time step size and time points
    dt = (t_span[1] - t_span[0]) / (t_points - 1)  # Step size
    t = np.linspace(t_span[0], t_span[1], t_points)  # Time grid
    state = np.array(initial_state)  # Convert initial state to numpy array

    # Initialize solution array to store results (time x state variables)
    num_variables = len(initial_state)  # Dynamic size of state variables
    solution = np.zeros((t_points, num_variables))  # Solution array

    # Store initial state
    solution[0] = state

    # Perform Euler integration
    for i in range(1, t_points):
        # Calculate derivatives
        derivatives = predator_prey_derivatives(state, *params)

        # Update state using Euler's method (state[i] = state[i-1] + dt * derivatives)
        for j in range(len(state)):
            state[j] = state[j] + dt * derivatives[j]  # Update all state variables

        # Store the new state in the solution array
        solution[i] = state
    return t, solution


if __name__ == '__main__':
    # Example inputs for simulation
    initial_state = [100, 10, 5, 80]  # [rabbits, foxes, wolves, resources]
    params = (
        1.2,   # a Rabbit reproduction rate
        0.1,   # b Predation rate of rabbits by foxes
        0.05,  # c Vegetation dependency rate for rabbits
        0.03,  # d Growth rate of foxes due to predation on rabbits
        0.15,  # e Natural death rate of foxes
        0.05,  # f Predation rate of foxes by wolves
        0.03,  # g Growth rate of wolves due to predation on foxes
```

```
        0.15,  # h Natural death rate of wolves
        0.5,   # e Resource replenishment rate
        0.1,   # i Natural decay rate of the resource
        0.1    # j Rate of resource depletion by rabbits
    )
    t_span = (0, 5) # Time span for simulation
    t_points = 100  # Number of time points for the simulation

    # Run the simulation
    t, solution = simulate(initial_state, params, t_span, t_points)
    print(solution)
    plot_results(t, solution)
```

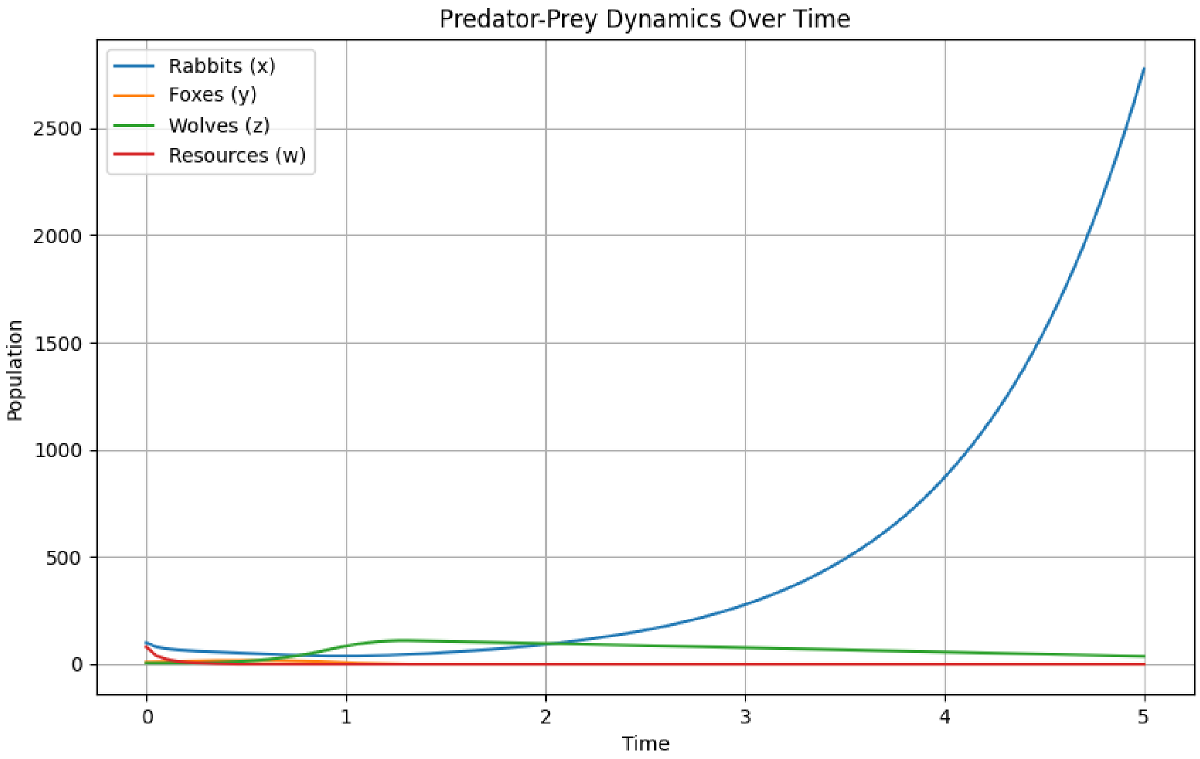
Visualization

Result depends on input parameters and is very sensitive to 11 constants.



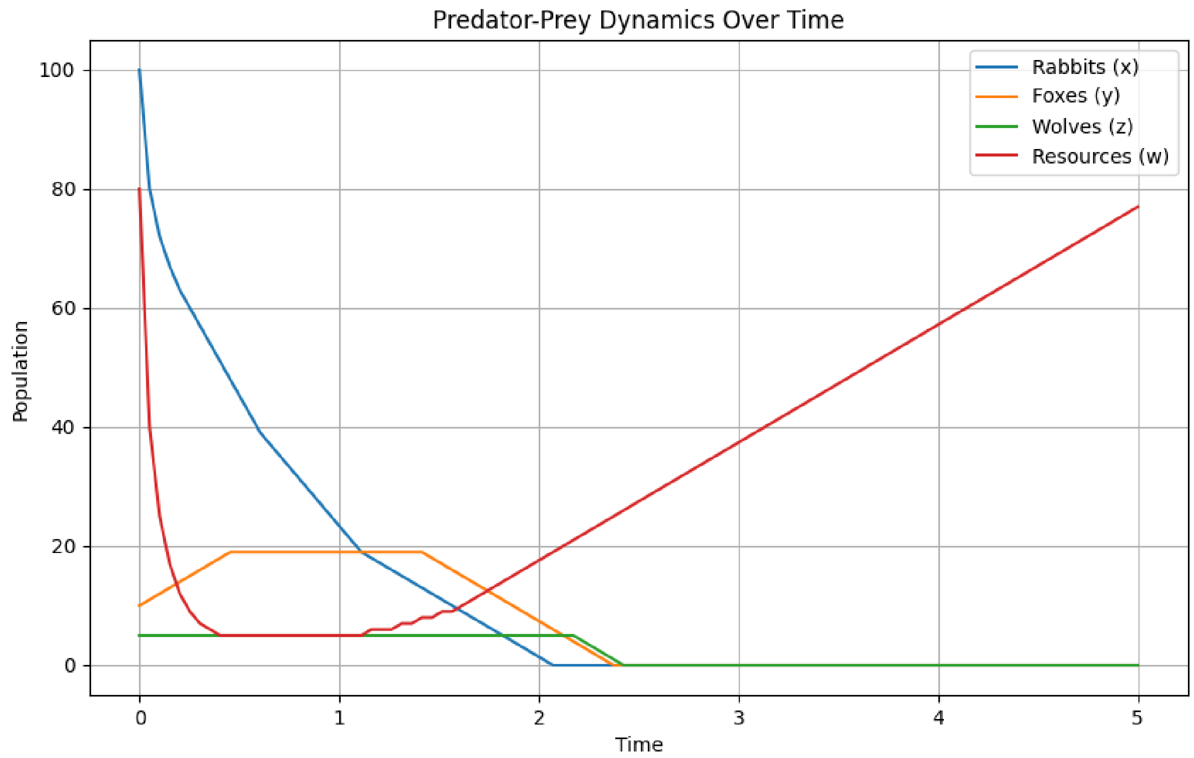
As the rabbit population decreases, foxes increase due to predation, followed by wolves preying on foxes, eventually leading to the collapse of all populations. In predator-prey models, when populations die out and equilibrium is reached, it indicates a steady state where the rates of change for all species become zero, often due to unsustainable interactions between species and resources.

Now lets see what happens if we change ***g*** parameter which is growth rate of wolves due to hunting on foxes from ***0.03 → 0.3.***



Without direct predators, rabbit populations skyrocket, as wolves cannot hunt them. However, as their numbers grow, they deplete resources, eventually leading to resource scarcity and a natural growth constraint, stabilizing the population.

Now what happens when we set replenishment of resources from ***0.5 → 30.***



As the rabbit population declines, pressure on resources decreases, allowing them to replenish steadily over time, leading to a consistent increase in resource levels.

Conclusion

In conclusion, the predator-prey model highlights the complex interdependencies within ecosystems, demonstrating how species interactions can lead to both stability and collapse. In this simulation, the unchecked growth of rabbit populations, due to the absence of direct predators like wolves, leads to resource depletion, which in turn impacts the entire system. The eventual equilibrium reached, where some populations decline and others stabilize, emphasizes the importance of balance in nature. Understanding these dynamics can provide valuable insights into real-world ecosystems and the consequences of disrupting these natural relationships.