Numerical-Analysis

Numerical Analysis methods in Python by ***Theofilos Panagiotou 9174 & Panagiotis Petridis 9286***

First Sector

We implemented four methods of ***Numerical Integration***: 1. ***Left rectangle*** 2. ***Right rectangle*** 3. ***Trapezoid*** 4. ***Simpsons***

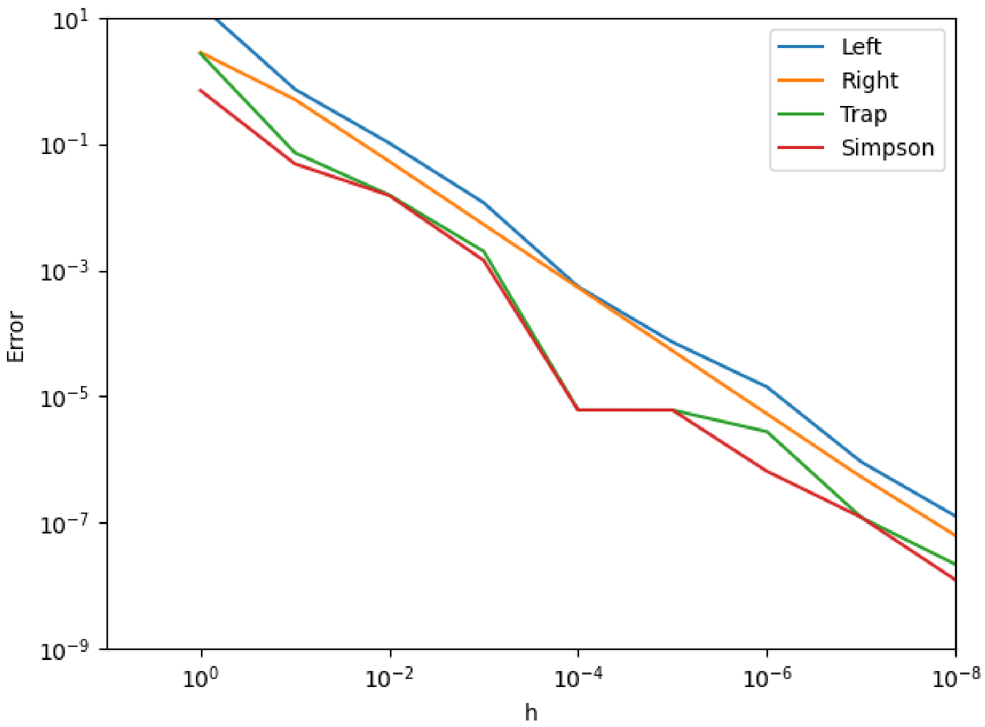
The target function to estimate its integral using the above methods is $e^{3x}sin(2x)$. Especially we calculated the following integral

$$\int_0^{\frac{\pi}{4}} e^{3x}sin(2x)dx = \frac{3}{13}e^{3\pi/4} + \frac{2}{13}$$

.

Error Diagramm

Below the diagramm presents the difference between each method and the calculated from the above integral value.



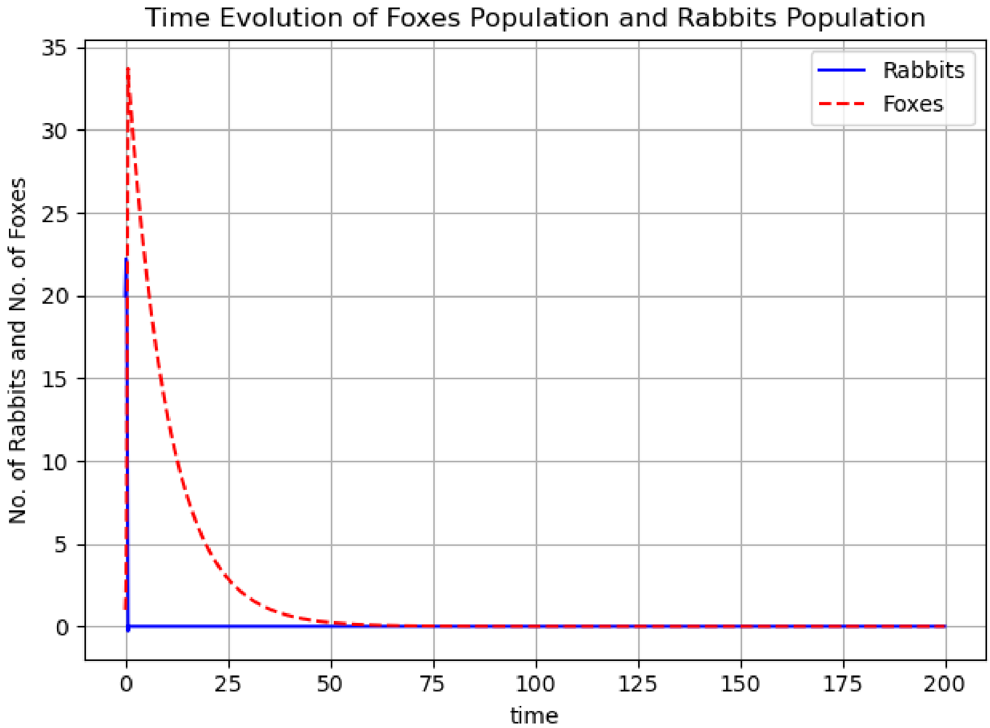
At the horizontal axis the step of discretation h scales from ***1*** to ***10***$^{-10}$. At the vertical axis we see how the error is changing during h changes. Simpsons method seems to be the most efficient between others. In general as h is getting minimized the error also decreases. As we know from theory the trapezoid absolute error upper bound is given by

$$|E_{T,N}| \leq \frac{h^2}{12}(\beta - \alpha) \max_{\alpha<z<\beta} |f''(z)|.$$

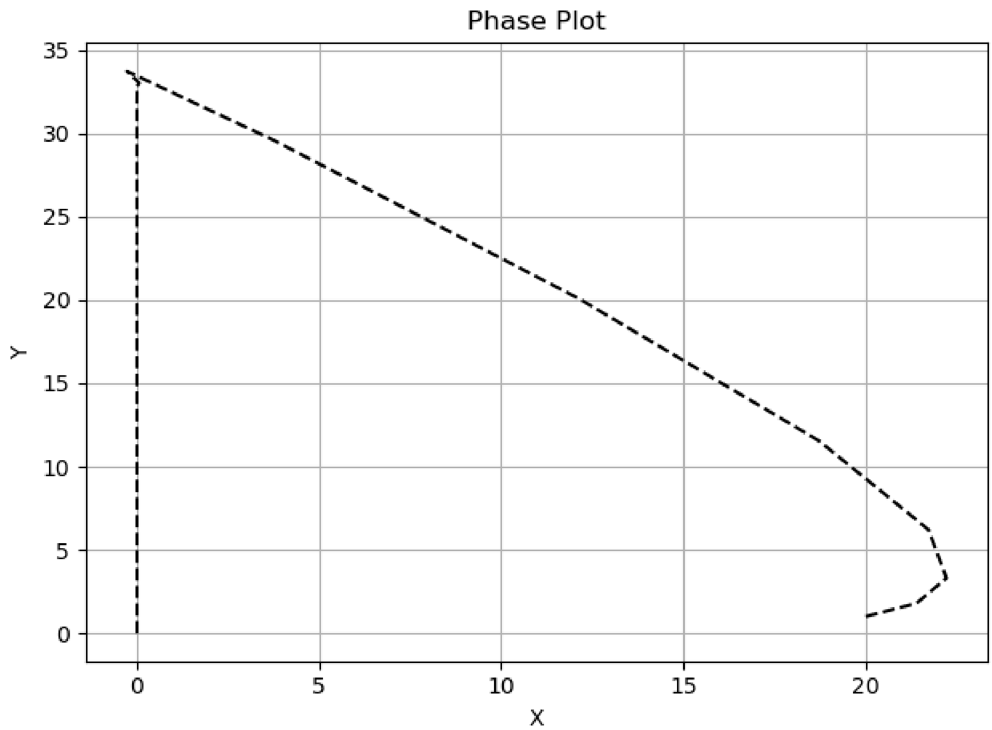
. So Error has an analogous relation with h. Similar analogous relation have also the other methods (left-right rectangle and simpsons). That is why we have the above diagramm.

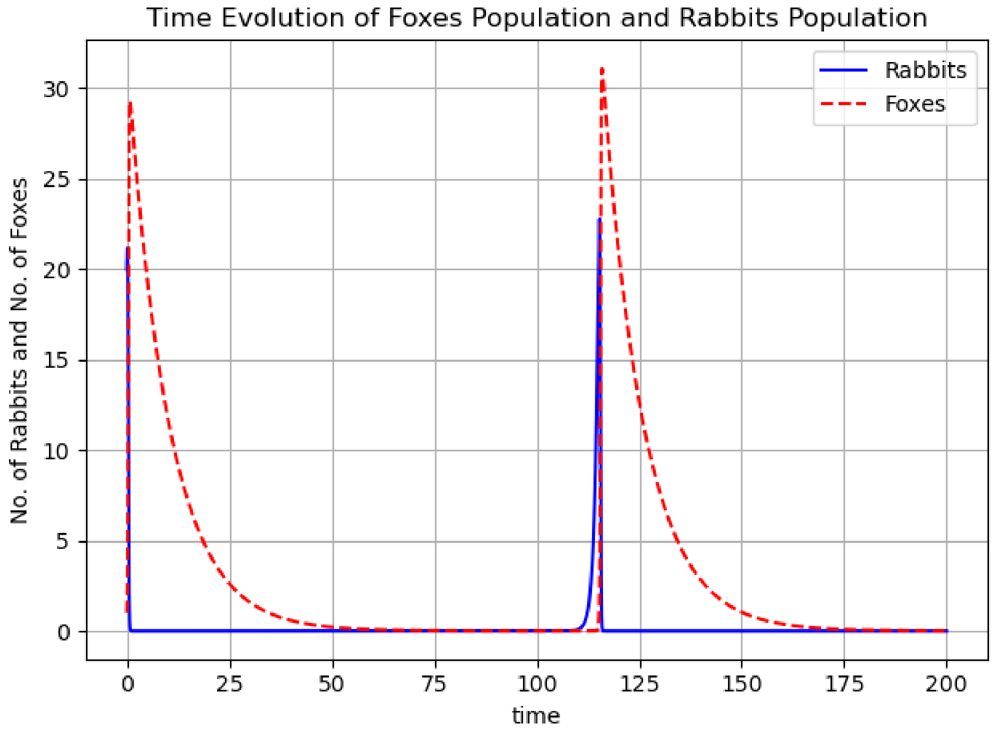
Second Sector

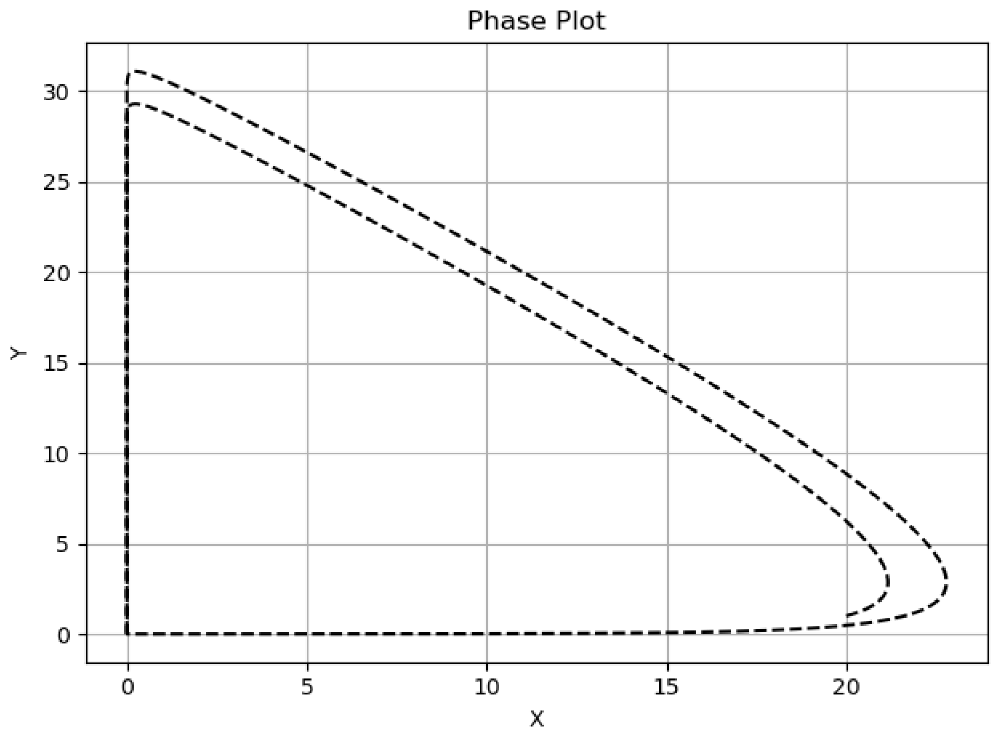
In this section we implemented the euler method for system of differential equations. The two pair of plots is the results from running the method with dx=0.1 and the



second pair with dx=0.01







What we observe is that with lower dx the number of rabbits and foxes after instantly changing, then we have a period of time where both are zero and then come back to initiall state and then we have a revision. As with the numerical integration, decreasing the h length we take a more presice solution. Since the error in euler has an quadratic analogous relation with h, we think that in the second try with h=0.01 we have less error.