Programming numerical methods for ODE's

1. We aim to solve numerically the ODE :

$$\begin{cases} y' & = f(y) \\ y(0) & = y_0 \end{cases}$$

What is the exact solution of this problem when $f(x) = -x$?

We want to use the Euler method to solve numerically this problem from the time $t = 0$ to the time $t = 1$. We decompose the interval $[0, 1]$ into n sub-intervals $[k/n, (k+1)/n]$, each of them having the size $\Delta t = 1/n$. The approximate value of y at the time k/n is denoted y_k.

Write a python function that computes an approximated solution of the equation $y' = -y$ between the times 0 and 1, once the initial condition y_0 and the number of iterations n are provided.

2. Plot the numerical solution obtained for the initial condition $y_0 = 1$ with $n = 30$. Plot on the same figure the exact solution. Repeat it for several values of n between 3 and 500. What do you observe ?

3. Plot $y_n - e^{-1}$ as a function of n and $log(y_n - e^{-1})$ as a function of $log(n)$. What is the slope of the curve obtained in the case of $log(y_n - e^{-1})$?

4. Define the function f such that $f(x) = -x + \dfrac{x^3}{6}$. Write a new function `euler_ex2`, whose inputs are the initial condition y_0, the time step dt and the final time T_f, and whose output is a vector containing an approximate solution of $y' = f(y)$ with the explicit Euler's method between the times 0 and T_f. Test it with the function f defined as above.

5. The Euler method gives good results if the time step is small. To increase the time step and thus diminish the computational time, we consider now the implicit Euler method. In the case of the function $f(y) = -y$, it is indeed possible to compute easily an approximate solution with this method.

Write a function `euler_imp` similar to the function `euler_ex` but using the implicit Euler method instead of the explicit one. Use it to represent on the same figure the exact solution and the approximate solutions obtained with the explicit and implicit methods. What do you observe ?