**Computational Tools**

# Project 3: Shooting

Assume the following second order differential equation

$$\ddot{x}(t) = \frac{3}{2}x(t)^2$$

- **TASK 1:** Write a code that numerically integrates the equation using the Runge-Kutta method of 4th order. Solve the equation for initial conditions $x(0) = 4$, $\dot{x}(0) = -40, -35, \ldots, -5$ in the time interval $t \in [0, 1]$. Plot your solutions.

We will now try to solve the boundary value problem for our differential equation with $x(0) = 4$ and $x(1) = 0$, using the shooting method. This means that, starting from $x(0) = 4$, we will try to find for which initial velocity $\dot{x}(0)$ we arrive at the position $x(1) = 0$, after time $t = 1$. To do so, we need to define the function $F(\dot{x}(0))$ which takes as input the initial velocity and returns the position $x(1)$.

- **TASK 2:** Make the plot of $F(\dot{x}(0))$, with $\dot{x}(0) \in [-40, -5]$. Using a numerical method of your choice compute the roots of $F$ in the given interval. Then compute the solution of the differential equation for the initial velocities you have found as roots of $F$. Plot your solutions. How accurate is the numerical solution of your differential equation and how close to zero is $x(1)$?

If you think your solutions in TASK 2 are accurate enough, you can skip TASK 3. Otherwise:

- **TASK 3:** Try improve the accuracy of your solution using Python's build-in functions, either a combination of scipy.integrate.solve_ivp and scipy.optimize.newton, or scipy.integrate.solve_bvp. Make sure you use appropriate tolerances.