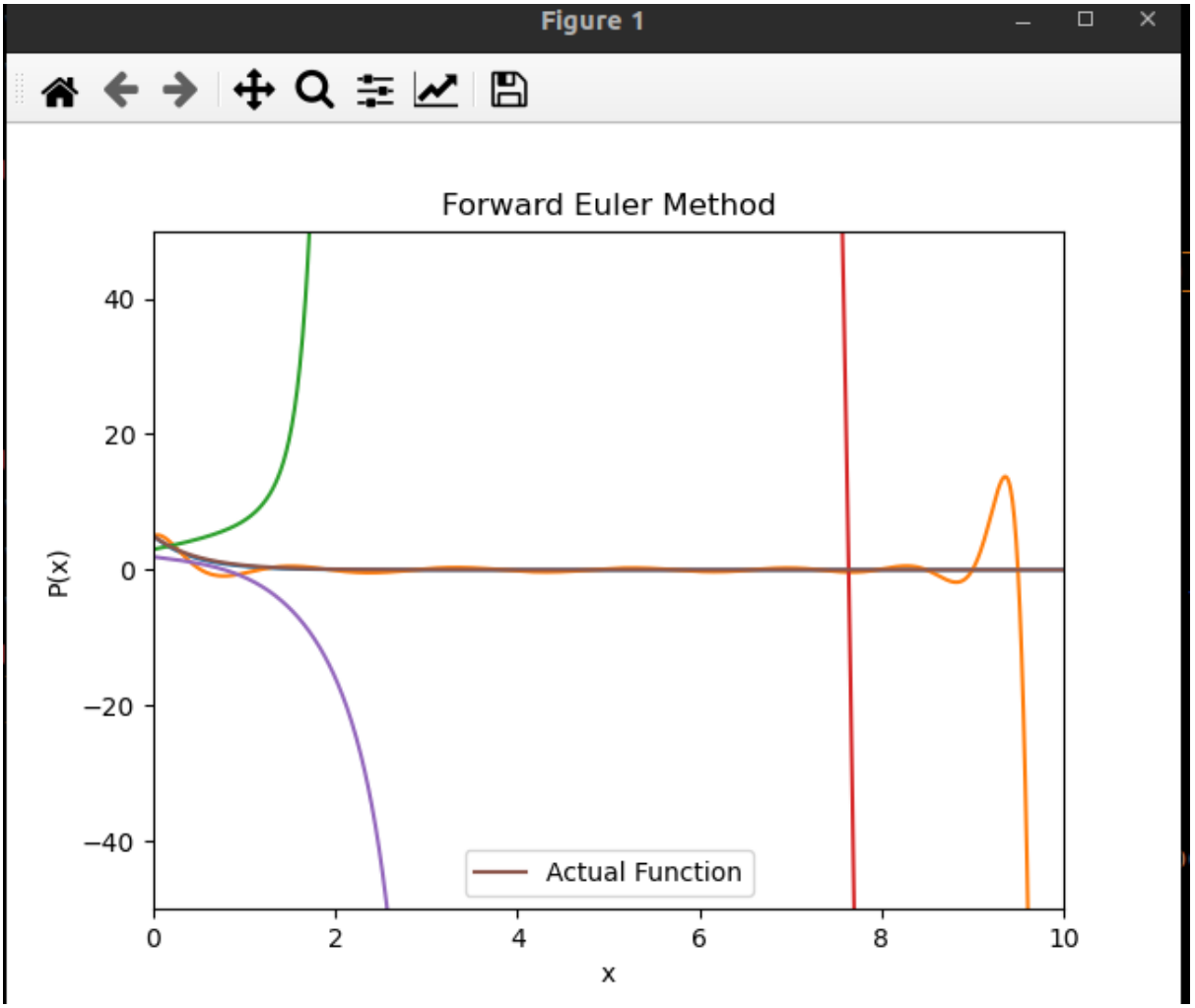
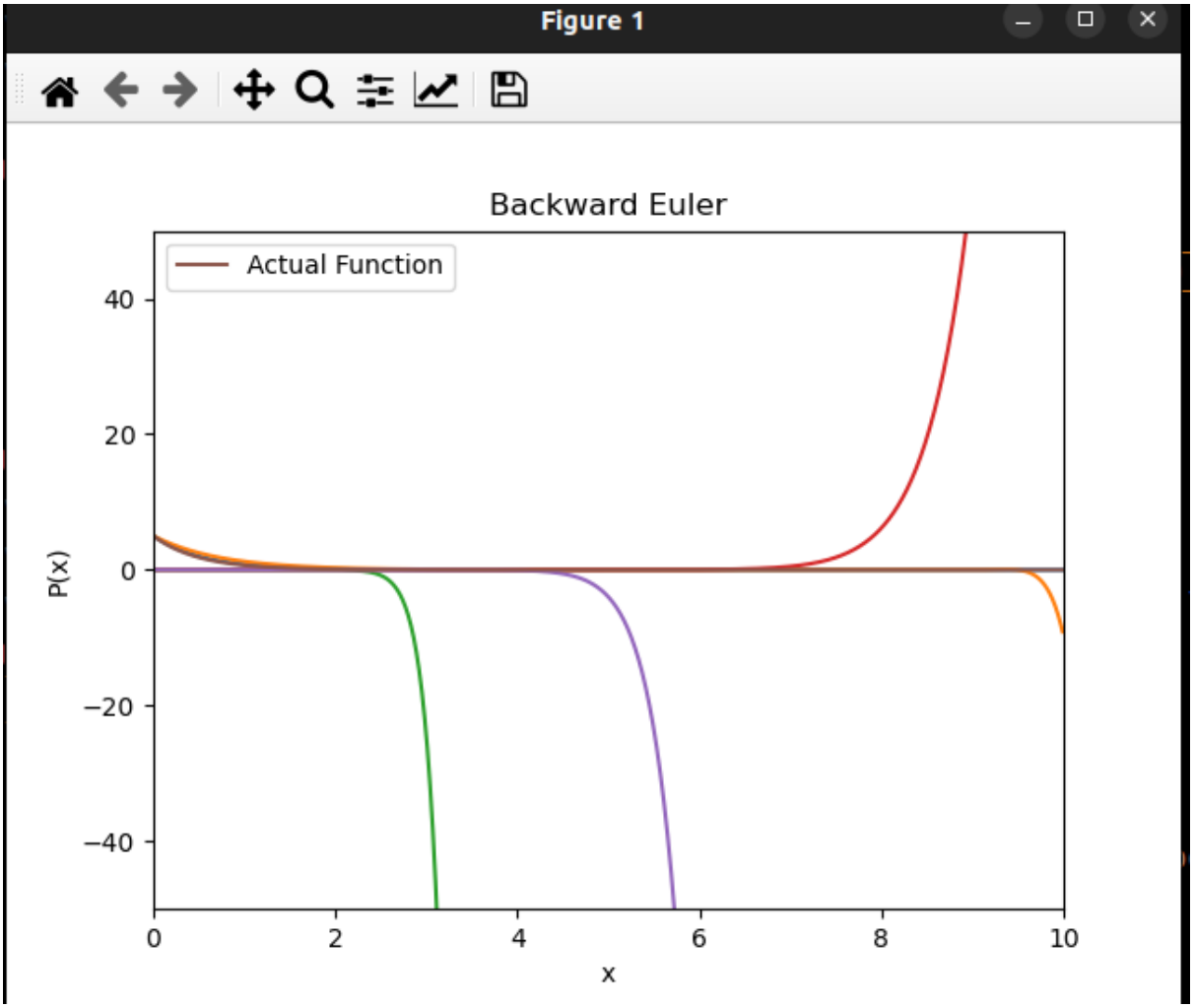
Lab report 6
112001041

1)
We will generate points in the range of [0, 10] using the Forward Euler formula for different discretization steps. Next, we will use the best_fit() function developed in a previous lab assignment to create a polynomial of degree 20 that fits the generated points. Finally, we will plot the actual function and the polynomial for visualization.



2) this is similar to previous just using the formula of backward euler



3)
Forward euler ->

●theta(n) = theta(n-1) + h * theta'(n-1)
●theta’(n) = theta’(n-1) + h * theta”(n-1)

We use theta(t) to get coordinates

4)
We uses the solve_ivp function from the scipy.integrate module to solve the system of ODEs for different initial conditions and plots an animation of the resulting trajectories.

5)
We again use solve_ivp with inputs x and y derivatives