Bisection vs. the secant method

In this tutorial, you will reproduce the plot of residual versus the number of completed iterations on slide 7 of lecture 4. It will help you understand the enormous difference between linear and quadratic convergence.
Python functions for bisection and Newton iteration are available from the course code repository (https://github.com/2072U/Course_codes_2025.git). You will have to change them to record and return the (estimated or upper bound of the) error and the residual for each iteration for plotting.

(**a**) Use bisection to solve the "unknown interest rate problem" from Lecture 2 (slide 7 & 8). Set the parameters as on the slides: $n = 12$, $T/P = 10$. Compute the result up to at least 10 significant digits. If small values of i, the interest rate, cause problems, consider changing the equation by multiplying by any number of factors i and $(1 + i)$.

(**b**) Use Newton iteration to solve he unknown interest rate problem with the same parameters and accuracy as in part (**a**).

(**c**) Plot the (estimated or upper bound of the) error versus the number of completed iterations for the two methods in one graph. How do they compare? Do the same for the residuals. Do they behave the same way?

Here are some questions to consider:

- How do you choose the initial domain for bisection and how do you choose the initial guess for Newton Iteration?
- What scale should you plot the error on? Linear? Log-log? Log-linear? Why?
- Newton iteration requires more computations per iteration (how many more?) but converges faster. On the whole, which method is more efficient (i.e. gets you the answer with a given accuracy with the fewest computations)?