Bisection vs. the secant method

In this tutorial, you will reproduce the plot of residual versus the number of completed iterations on slide 3 of lecture 5. It will help you understand the enormous difference between linear and quadratic convergence.
A Python function for bisection is available from the course code repository. You will have to program a function for the secant method, but you can wait until after Wednesday's lecture.

(**a**) Use bisection to solve the cannon ball problem from lecture 3. Assume the enemy is 100(m) away, the initial speed of the cannon ball is 40(m/s) and the friction coefficient is $c = 0.001\,(1/\mathrm{m})$. You can pull the function from page 7 of the lecture from the tutorial repository. Start by plotting the function for θ, the angle of inclination, between 0 and $\pi/2$. Find a good initial domain $[a, b]$ from the graph. Record the error and the residual for each iteration.

(**b**) Use the secant method solve the cannon ball problem with the same parameters. Find two initial guesses from the graph you made for (**a**). Record the error and the residual for each iteration.

(**c**) Plot 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.

Here are some questions to consider:

- What scale should you plot the error on? Linear? Log-log? Log-linear? Why?
- The secant method 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)?