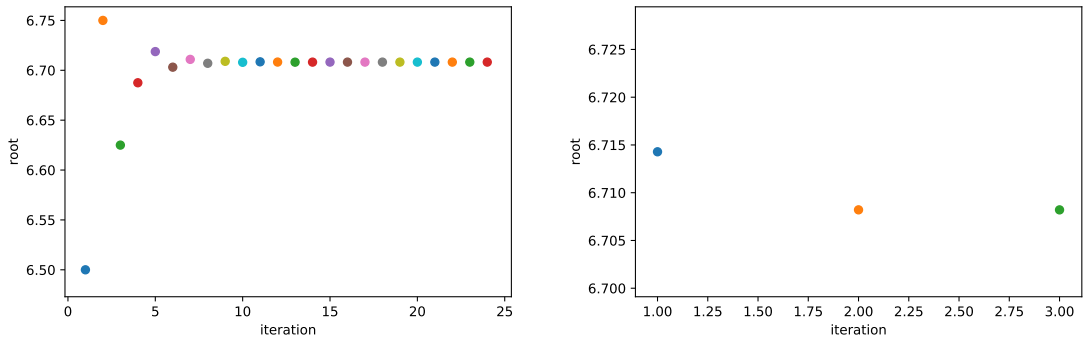
Computational Methods in Astrophysics- HW1

Anjali Yelikar

1 September 2020

I wrote all my codes in a Jupyter notebook, but when I upload it to GitHub the interactive element goes away(sorry I did not anticipate this sooner). The codes for each problem are in a separate cell in the notebook. The link to my GitHub repository is: **https://github.com/AnjaliYelikar/Computational-Methods-RIT**.

1. **Problem** 1
   As the problem states, root finding algorithms using Bisection, Newton and Secant method were written.



(a) Bisection method (b) Newton method

Figure 1: Problem 1

Above is a figure describing the number of iterations it takes for the Bisection and Newton method to find the root of the function x^2 for the same set of starting points and convergence limit. X-axis is the iteration number and Y-axis is the value of the root. The number of iterations in Newton method does seem less, we don't get to see the long stable behaviour as in the Bisection method, but it does do its job quicker. I kept changing the convergence limit for Newton so as to get more points, but that would mean a lot(lot!) more points for the Bisection method plot, hence did not do it.

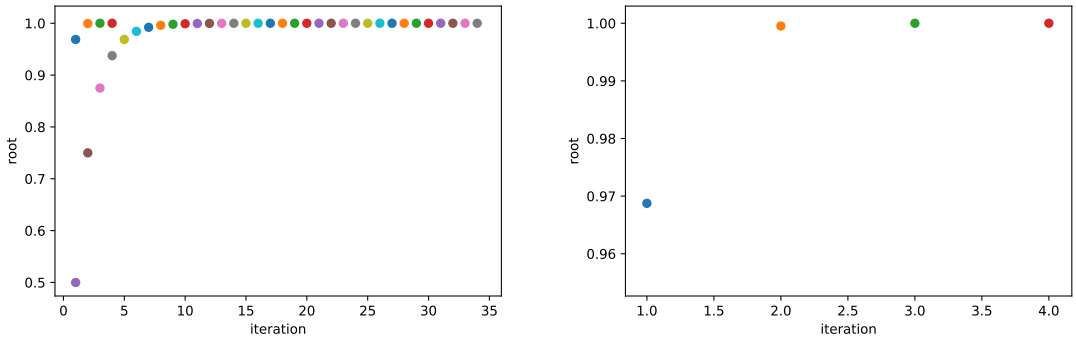
2. **Problem 2**

$$N_e(x) = N_0[1 + \frac{x}{r_c}^2]^{-1/2} \tag{1}$$

   Setting $N_e(x) = \frac{N_0}{2}$ and solving for x using the root finding algorithms.

   The root comes out to be close to 1(0.9999999994..) by both the methods. Next page is the iteration number v/s root value plot for both the methods.

3. **Problem 3**



(a) Bisection method (b) Newton method

Figure 2: Problem 2

The value of x' should vary from 0 Au to 2 AU. My code right no does not do that, it just gives the root if you specify the value of x'. I was trying to defining multiple functions one for each value of x', but the I was not able to figure out the exact way to loop this.

4. **Problem 4**

Using Eq.15, Eq.6 and Eq. 7, we get

$$x' = x + \frac{\lambda^2 * r_e * D * x * N_0}{2 * \pi * r_c^2 * [1 + \frac{x}{r_c}^2]^{3/2}} \tag{2}$$

5. **Problem 5**

Piecewise linear interpolation, no plot. Interpolated data points taken from a sine function.

6. **Problem 6**
Interpolated function to the lens density data output: $0.001 * x - 0.001$
$0.007 * x - 0.013$
$0.022 * x - 0.058$
$0.035 * x - 0.11$
$0.038 * x - 0.125$
$0.027 * x - 0.0590000000000001$
$0.01 * x + 0.0599999999999999$
$0.204 - 0.00800000000000001 * x$
$0.303 - 0.019 * x$
$0.353 - 0.024 * x$
$0.342 - 0.023 * x$
$0.306 - 0.02 * x$
$0.254 - 0.016 * x$
$0.184 - 0.011 * x$
$0.124 - 0.007 * x$
$0.092 - 0.005 * x$
$0.058 - 0.003 * x$
$0.04 - 0.002 * x$
$0.021 - 0.001 * x$