ASTP720 - Computational Methods

Homework 1

Liza Matrecito

6 September 2020

Problem 1

Write a library (i.e., in a separate file that you can call) for the three root-finding algorithms we discussed in class: Bisection, Newton, Secant. These functions should each take functions rather than data points. Make sure that each takes an optional argument the threshold and that it also takes a variable that allows the user to print out or return the number of iterations it took to hit that threshold.

For coded solution, click this Github link.

Problem 2

For the pseudo-isothermal sphere, using your root-finding algorithms, numerically calculate the full width at half maximum, i.e., what is the width (in terms of r_c when $N_e(x) = N_0/2$, half the amplitude. Drawing pictures for yourself might be useful! Do so with each of your root-finding algorithms and show how many iterations each takes as a function of your threshold. Please plot the results.

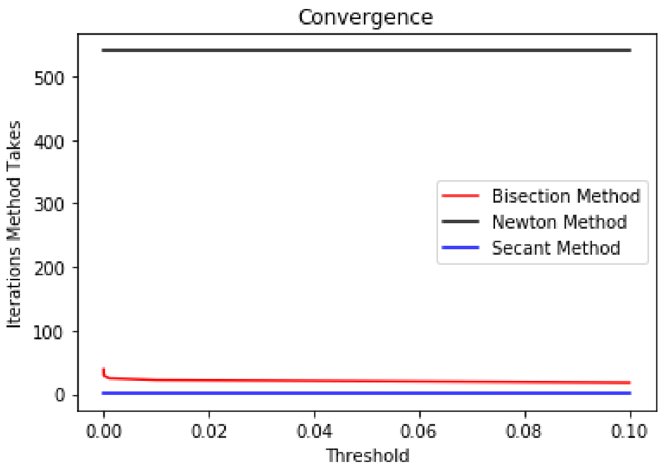

Figure 1: This plot shows the convergence of each root-finding algorithm methods. I believe the convergence does depend on the initial function f(x) that is being used (from problem 1) because I tried multiple ones and received different results. This version uses f(x) = x^2.

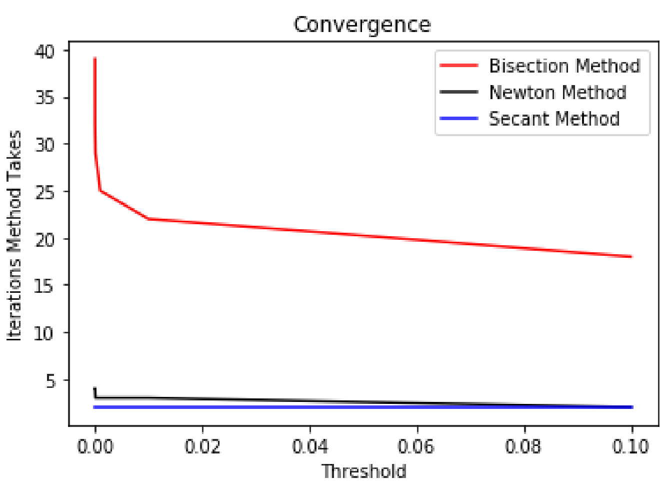

Figure 2: I just added this section to show how it really does depend on the function used. This version uses f(x) = $1/(\sqrt{1+x^2}$-(1/2)).

For coded solution, click this Github link.

Problem 3

You can probably see from the Gaussian Lens equations that if you have a light ray hitting x, and you know the other parameters of the lens (a, N_0, D, etc.), then you know what x' is. But that's boring and not what you actually observe. Let's instead say you are an observer in a "circular orbit" along the x' axis with radius 1 AU and a period of 1 year but centered at x' = 1 AU. Then, you know where your position x' is but not where the light rays from the source are intersecting the lens plane at x - as expected, analytically solving for x is not really an option. Using one of your root-finding algorithms, solve the lens equation for each value

of x' and make a raytracing plot as on the first page. Assume D = 1 kpc, a = 1 AU, λ = 21 cm, and N_0 = 0.01 pc cm^{-}3 (these are observer units, probably best to convert to something like cm^{-}2.

For coded solution, click this Github link.

Problem 4

Repeat but for the pseudo-isothermal sphere with the same parameters but $r_c = 1$ AU.

For coded solution, click this Github link.

Problem 5

Write a library for piecewise linear interpolation, given a set of x and y data points. This should return a function f that one can use to calculate a new point $f(x_{new}) \rightarrow y_{new}$.

For coded solution, click this Github link.

Problem 6

In the file lens_density.txt are a series of values of x and $N_e(x)$ for some shape. Use your interpolator to plot the values of $N_e(x)$ halfway in between all of the given x values, i.e., when x = 0.5,1.5, $\cdots$.

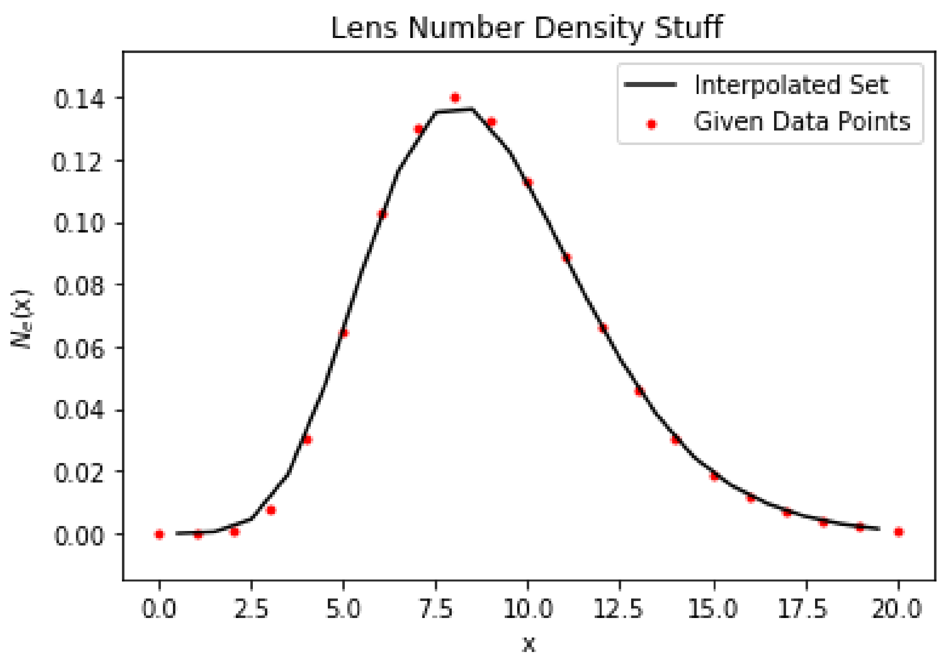

Figure 3: This plot shows the results of the density of a lens with a data set (red data points), given by the lens_density.txt file, interpolated to x-values that are halfway of the given x values from the text file. You can see how the interpolated set is a much smoother Gaussian curve (better fit).

For coded solution, click this Github link.