Homework 1 Write-up

Sarah Vaughn

The first python file is root finding and contains 3 different functions: Bisection, Newton, and Secant. These algorithms where used to calculate the the pseudo-isothermal sphere full width at half maximum.

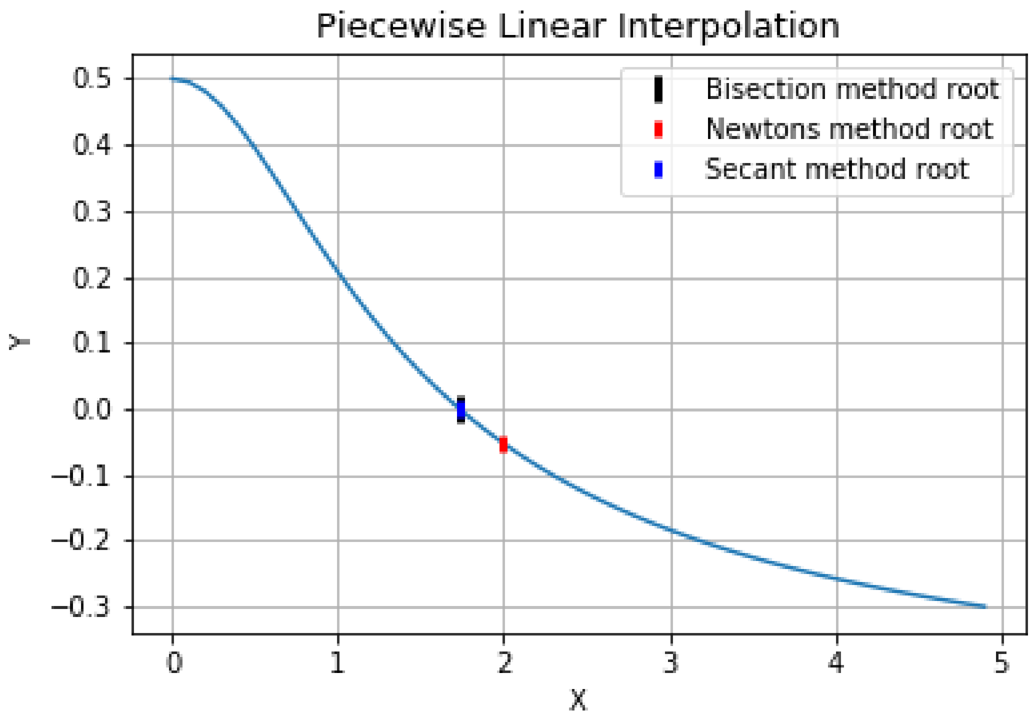
$$(1 + 1/\sqrt{u^2})) - (1/2) \tag{1}$$

$$u/((1 + u^2)^{3/2}) \tag{2}$$

where:

$$u = x/r_c \tag{3}$$

Using the Bisection method, the root value is: 1.734375 in 7 iterations.
Using the Newtons method, the root value is: 2.1890255716125564 in 4 iterations.
Using the Secant method, the root value is: 1.7325967388187353 in 6 iterations.
The results for the Newton method are not as good as the the other two methods. This is probably a result of the Newtons method function not working properly. I believe it might be the threshold not being as small as the the other two methods. However, changing the threshold to be any small causes an error within the code.

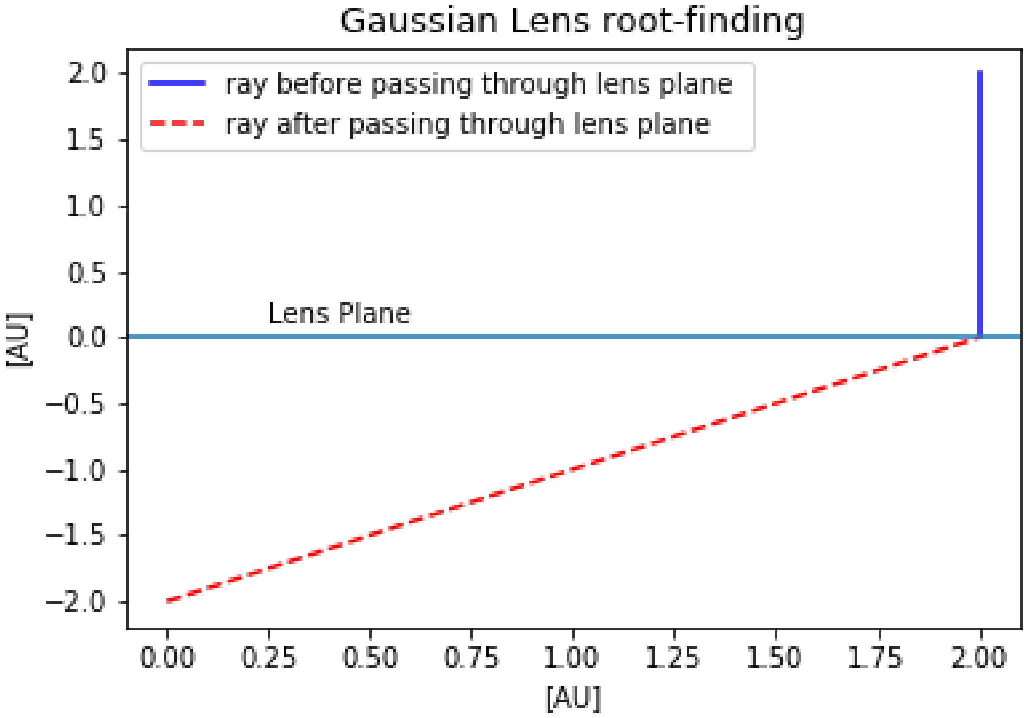


As can be seen in this plot, the Bisection and the Secant method give good and reasonable values for the root while the Newton method doesn't quite make it.

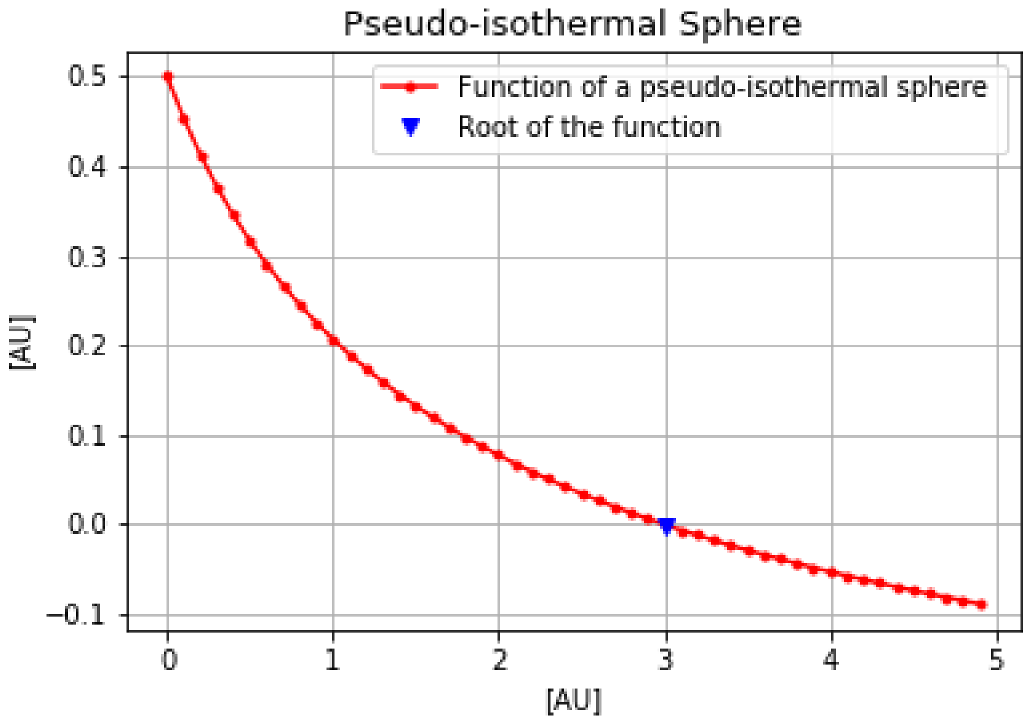
The Gaussian Spherical lens equation is:

$$x' = x * (1 + (((\lambda^2) * r_e * N_0 * D)/(\pi * a^2)) * e^{-(x/a)^2}) \tag{4}$$

x' is the axis with radius 1 AU centered at x' = 1 AU. With this information as well as Assuming $\lambda = 1.403 * 10^{-12}[AU], N_0 = 2.238 * 10^{24}[AU], D = 2.603 * 10^8[AU], a = 1[AU], r = 1.884 * 10^{-26}[AU]$. Again, the root finding algorithms where implemented. Specifically the Secant method was applied and the result was the following ray tracing plot:



By using the root-finding library to solve for x in equation 4 and plotting it like above, the ray after it passes through the lens plane should be visible. Unfortunately, the value for x from the root-finding function returned 2.0 [AU] in only 1 iteration. This is suspicious and leads me to believe that there is an issue here. Another thing that stands out to me that makes me believe this isn't working correctly is that the ray after passing through the lens plane has a very large angle which wouldn't make sense in the situation. The angle should be much smaller and the distance between the lens plane and the observer plane should be $D = 2.603 * 10^8[AU]$ but is only about $2.0[AU]$.
This same method was again applied to the pseudo-isothermal sphere equation(1) for $r_c = 1)AU$ for equation 3. The resulting root value using the Secant method, the root value is: 3.0080200845195 in 4 iterations. Plotting these results:



The final library for a piecewise linear interpolation is a function that will interpolate a new value from a given set of points, x and y. Using the equations:

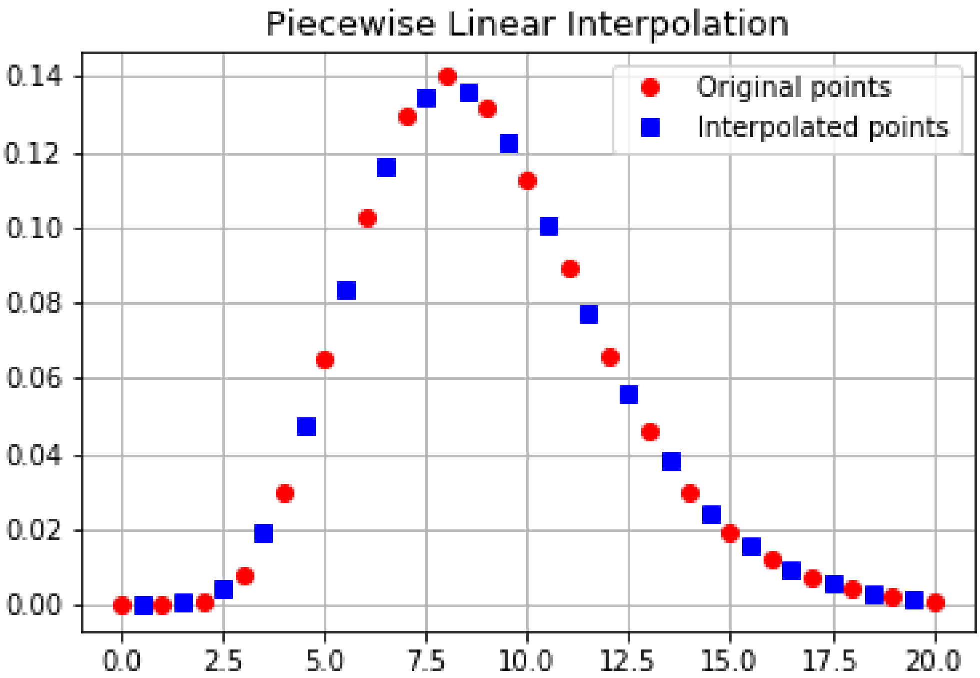
$$y_{new} = y_0(1 - x_d) + y_1 * x_d \tag{5}$$

where,

$$x_d \equiv \frac{(x_{new} - x_0)}{(x_1 - x_0)} \tag{6}$$

by passing a list of x and y points and a new value of x (x_{new}) to equation 5 and 6 a new value of y (y_{new}). Doing this will make a curve smoother giving a better result.

The library that I wrote doesn't return any values only zeros because there is an issue that I cant find but there is a library in numpy that takes a one-dimensional piecewise linear interpolant and this returns the new values for y. Using this function, and applying it to the lens density data set, the following data points will be the result:


Piecewise Linear Interpolation
Original points
Interpolated points
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0