Link to HW1 code

1 Task 1

The library containing the root finding algorithms can be found in **root_finding_algos.py**. The bisection and secant methods only require one function, as well as a lower and upper guess on the initial range bounding the root, while Newton's method requires both a function and its derivative (in functional form) but only one value for the initial guess. The threshold under which the absolute value of $f(x_r)$ must be in order for root-finding to cease is set by the **eps** variable. The **verbose** flag dictates whether or not the the number of iterations is returned along with the root, as well as if both of these findings are printed to stdout. **iters** is used as a variable due to the recursive nature of the function and should not be altered by the user.

2 Task 2

To find the full-width half-maximum of the pseudo-isothermal sphere, we want $N_e = N_0/2$ such that, starting from Equation (14) in the homework,

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

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

$$0 = \left[1 + \left(\frac{x}{r_c}\right)^2\right]^{-1/2} - \frac{1}{2} \tag{3}$$

$$0 = \left[1 + x^2\right]^{-1/2} - \frac{1}{2} \tag{4}$$

where the last line comes from the simplification of reporting x in terms of r_c. For Newton's method, we use Equation (15) with the left-hand side set to 0 since the derivative of a constant is zero. The performance of the three different root-finding algorithms as a function of the threshold is shown in Figure 1.

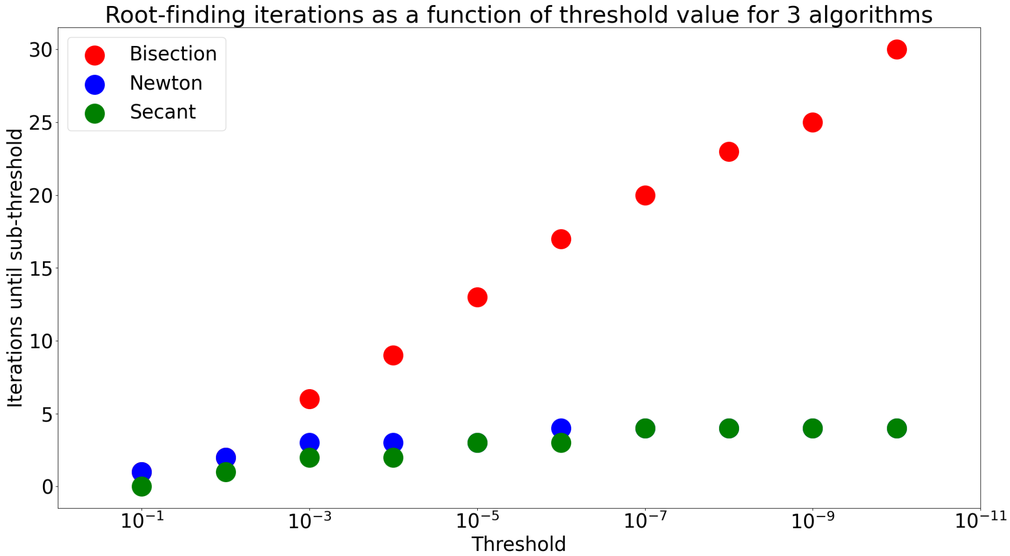

Figure 1: Number of iterations as a function of threshold for the bisection (red), Newton's (blue) and secant (green) methods. It is clear that Newton's method and the secant method both significantly outpeform the bisection method as the threshold becomes more constraining.

3 Task 3

The observer is moving in the observer plane with some position x'. Since we only care about the x-axis portion of the observer's position, we can take $x' = r\cos\theta$ where it is important to differentiate the θ of the observer's orbit from the θ_r which denotes the refraction of the light rays after passing through the lens. We know that $r = 1$ AU and the orbit is centered at 1 AU, so we have $x' = 1 + cos\theta$. We can re-write θ as some fraction of the orbit and since we have a circular orbit, we can divide this evenly into integer values representing the months of the year, divided by the number of months in a year, times the circumference of the circle (2π). Thus, we can write

$$x' = 1 + \cos\theta \tag{5}$$

$$x' = 1 + \cos\left(\frac{2\pi m}{12}\right) \tag{6}$$

$$x' = 1 + \cos\left(\frac{m\pi}{6}\right) \tag{7}$$

Feeding this into Equation (11) from the homework, we find that the lens equation which we give to the root-finding algorithms is written as

$$x' = x\left[1 + \frac{\lambda^2 r_e N_0 D}{\pi a^2} e^{-(x/a)^2}\right] \tag{8}$$

$$1 + \cos\left(\frac{m\pi}{6}\right) = x\left[1 + \frac{\lambda^2 r_e N_0 D}{\pi a^2} e^{-(x/a)^2}\right] \tag{9}$$

$$0 = x\left[1 + \frac{\lambda^2 r_e N_0 D}{\pi a^2} e^{-(x/a)^2}\right] - \left(1 + \cos\left(\frac{m\pi}{6}\right)\right). \tag{10}$$

Using the secant method, we find x values where the rays from the source intersect the lens plane. Figure 2 shows light rays connecting the source (located arbitrarily at (0, 2)) and the x values on the lens plane as well as the refracted light rays between the lens plane and the observer's location x'.

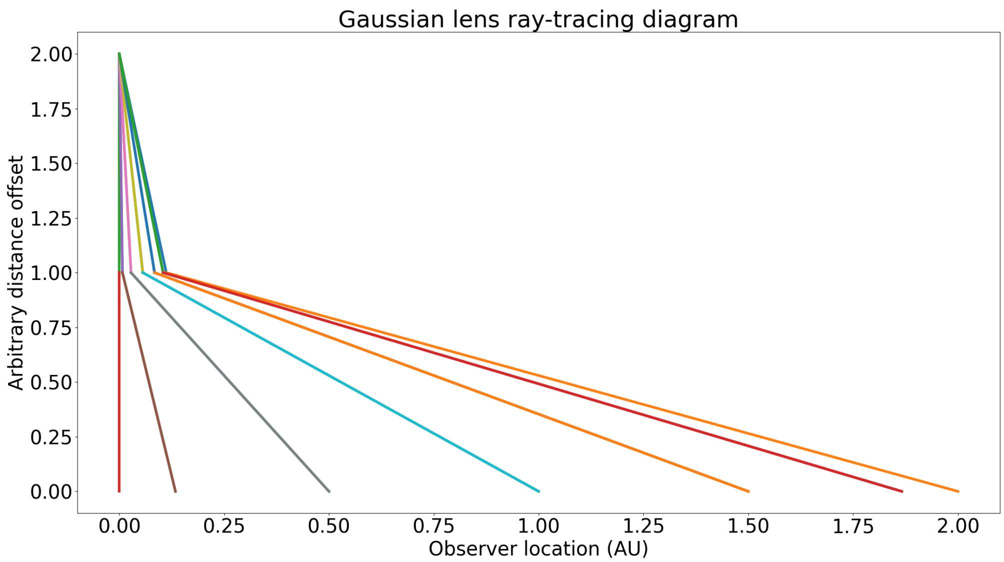

Figure 2: Ray tracing diagram for light rays emitted at the source (top-most point) which pass through a Gaussian lens at y=1 and then refract the rays to the observer, located at x' values on the x-axis represented by integer values indicating how far through one orbit the planet is within a given year.

4 Task 4

From Equations (6) and (7), we are given that $\theta_r = \frac{\lambda^2 r_e}{2\pi} \frac{d}{dx} N_e(x)$ and $x' = x - \theta_r(x) D$. Using the representation of $\frac{d}{dx} N_e(x)$ for a pseudo-isothermal spherical lens given in Equation (15), we get that

$$x' = x - \theta_r(x)D \tag{11}$$

$$x' = x - \frac{\lambda^2 r_e D}{2\pi} \frac{d}{dx} N_e(x) \tag{12}$$

$$x' = x - \frac{\lambda^2 r_e D}{2\pi} \left(- \frac{N_0 x}{r_c^2 \left[1 + \left(\frac{x}{r_c}\right)^2\right]^{3/2}} \right) \tag{13}$$

$$x' = x \left(1 + \frac{\lambda^2 r_e D N_0}{2\pi r_c^2 \left[1 + \left(\frac{x}{r_c}\right)^2\right]^{3/2}} \right) \tag{14}$$

$$1 + \cos\left(\frac{m\pi}{6}\right) = x \left(1 + \frac{\lambda^2 r_e D N_0}{2\pi r_c^2 \left[1 + \left(\frac{x}{r_c}\right)^2\right]^{3/2}} \right) \tag{15}$$

$$0 = x \left(1 + \frac{\lambda^2 r_e D N_0}{2\pi r_c^2 \left[1 + \left(\frac{x}{r_c}\right)^2\right]^{3/2}} \right) - \left(1 + \cos\left(\frac{m\pi}{6}\right)\right). \tag{16}$$

Following the same approach as in Task 3, we get a ray-tracing diagram for the pseudo-isothermal spherical lens.

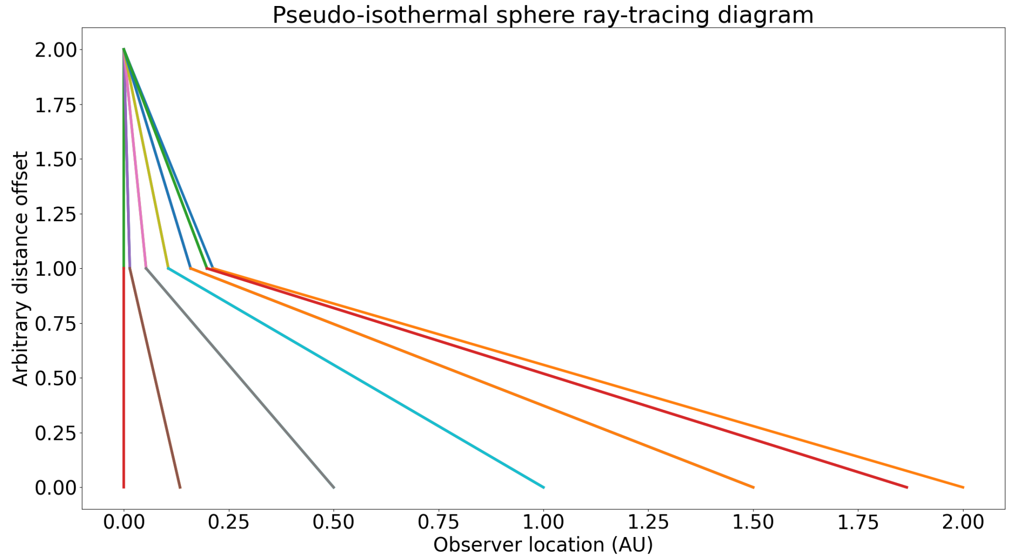

Figure 3: Ray tracing diagram for light rays emitted at the source (top-most point) which pass through a pseudo-isothermal spherical lens at y=1 and then refract the rays to the observer, located at x' values on the x-axis represented by integer values indicating how far through one orbit the planet is within a given year. The locations where the light rays are incident on the lens plane are much more spread out than for the Gaussian lens, likely due to the lack of an exponentially decaying factor as in the Gaussian lens case.

5 Task 5

The library containing the linear piece-wise interpolation can be found in **lin_interp.py**. The linear_interpolator function accepts a set of points (x_0, y_0) and (x_1, y_1) which set the bounds and values for the linear interpolation and return a function. The returned function accepts a point and returns the interpolated value based on the y_0 and y_1 values provided to the linear interpolator. While the interpolator accepts all points, the only logical ones are those which are within the bounds set by the lower (x_0) and upper (x_1) limits.

6 Task 6

Given the data file *lens_density.txt* of length n, the x values and their respective N_e values are plotted in black and labeled as training data. Each consecutive pair x and N_e values are fed to the interpolator as training data, resulting in a collection of $n-1$ pairs of training points. For each pair, the average of the x values is taken to represent the midway point between the upper and lower bounding values for the interpolator. This midway point is then fed into the function and the resulting output is plotted in red as the interpolated data. Figure 4 shows a plot of the training data in black and the outputs of the linear interpolation in red.

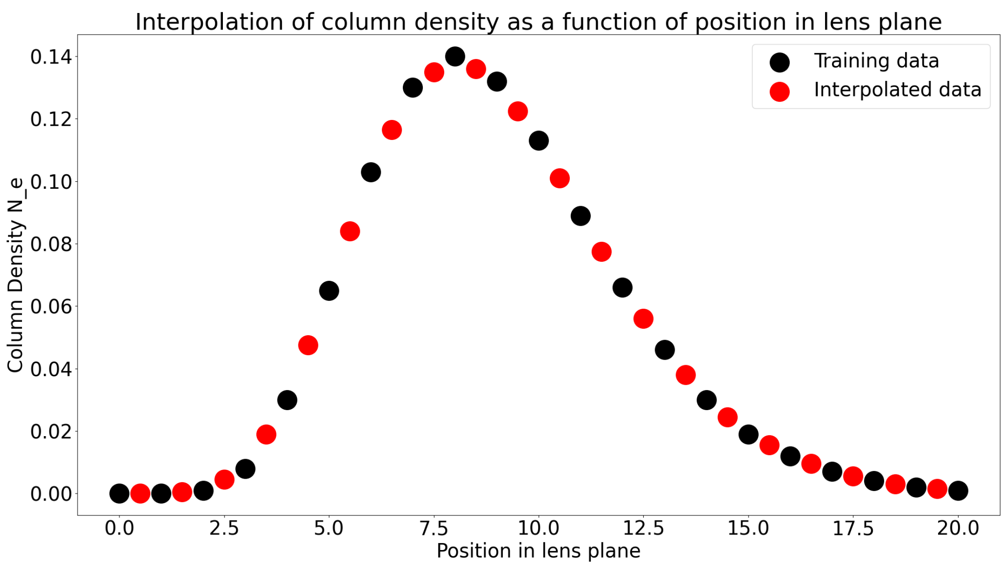

Figure 4: Plot of the training data in black and the linearly interpolated values at the midway points in red. It is clear, especially near the implied curves of the training data, that the interpolation is not fully representative of the function's true nature. This is most evident at the peak, where the interpolated values undershoot the imagined smooth peak that the data appears to follow.