Q1) Write an RK4 integrator with prototype to take one step. Use $\frac{dy}{dx} = \frac{y}{1+x^2}$, from $x = -20$ to $x = 20$, with $y(-20) = 1$ using 200 steps. Now write another stepper that takes a step of length h, compares that to two steps of $h/2$, and uses them to cancel out the leading-order error term from RK4. How many function evaluations per step does this one use? Use this modified stepper to carry out the same ODE solution using the *same number of function evaluations* as the original. Which is more accurate?

(i) Integral value from regular RK4 method (200 steps): 19.940049188005364

We can write a new modified stepper that takes a full step and then two half steps and compares them to cancel the leading order term from the standard RK4. Given the original RK4 integrator `rk4_step` uses 4 function evaluations per step, and the modified stepper `rk4_stepd` uses 12 function evaluations per step (calls the original `rk4_step` 3 times), we will need to run `rk4_stepd` for one third of the steps as `rk4_step` to obtain an ODE solution with the same number of function evaluations. Therefore, we will evaluate `rk4_step` for 600 steps and `rk4_stepd` for 200 steps and compare their accuracy.

(ii) Integral value from regular RK4 method (600 steps, 2400 function evaluations): 19.940303857712177

(iii) Integral value from modified RK4 (200 steps, 2400 function evaluations): 19.940307381100723

Integral value from analytic solution of $y(x) = c_0 e^{\arctan(x)}$: 19.940306913620812

Accuracy of (i):

$$|19.940049188005364 - 19.940306913620812| = 0.000257725615448 \tag{1}$$

Accuracy of (ii):

$$|19.940303857712177 - 19.940306913620812| = 3.055908635474225 * 10^{-6} \tag{2}$$

Accuracy of (iii):

$$|19.940307381100723 - 19.940306913620812| = 4.674799107817762 * 10^{-7} \tag{3}$$

Therefore, using `rk4_stepd` for 200 steps has an accuracy improvement of ≈ 1 order of magnitude over `rk4_step` for 600 steps (and therefore the same number of total function evaluations). See Figure 1 for a visualization.

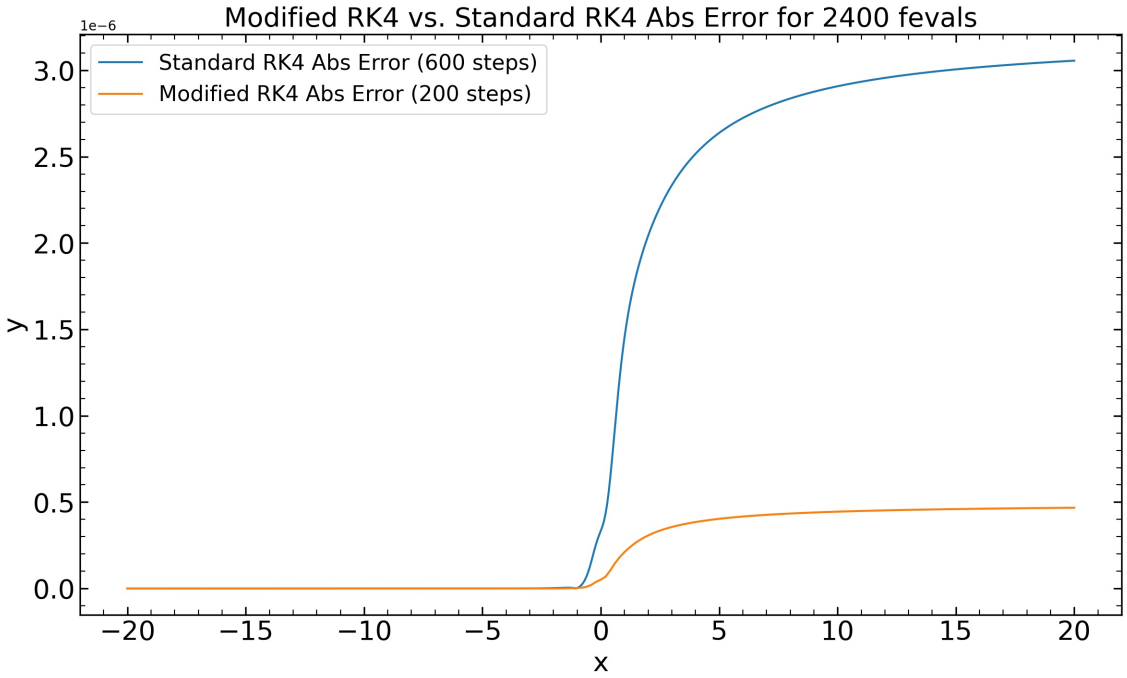

Figure 1: The absolute error in the modified RK4 stepper (orange) is less than the standard RK4 stepper (blue), especially at larger x-values.

Q2a) Write a program to solve for the decay of products of U238. You can use the ODE solver from scipy, but need to set up problem properly. Include all decay products in the chain. Assume you start from a sample of pure U238. What solver would you use for this problem?

The i^{th} nuclide's decay may be written as $\frac{dN_i}{dt} = \lambda_{i-1}N_{i-1} - \lambda_i N_i$, whilst the 0^{th} is $\frac{dN_0}{dt} = -\lambda_0 N_0$, and the last decay is simply $\frac{dN_{n-1}}{dt} = \lambda_{n-1}N_{n-1}$, given the first decay has no preceding nuclide and the last decay results in a stable nuclide. Taking $\lambda = \tau^{-1}$ for τ being the the mean lifetime of the nuclide we can compute its half-life as $t_{1/2} = \frac{\ln(2)}{\lambda}$, and hence replace λ_i with $\frac{\ln(2)}{t_{1/2,i}}$ in the previously stated equations.

Now that we have defined our decay process and calculated the half-lives from the slides in class in terms of Earth-years, we can numerically integrate using scipy. Given the question states we start with pure U238, we set this initial value to 1 and leave all other nuclides as 0. To choose the type of solver, we consider what we saw in class where the RK4 algorithm was very slow and required several orders of magnitude larger function evaluations compared to the Radau method. Therefore, we will use the Radau method here, and integrate over a time interval of comparable magnitude to the half-life of U238, as if we set this interval to be several orders of magnitude smaller, there will be little evolution of the system as the first nuclide in the chain will not have decayed substantially. In this instance, we will conservatively choose 10^{10} years, which is approximately two half-lives of U238.

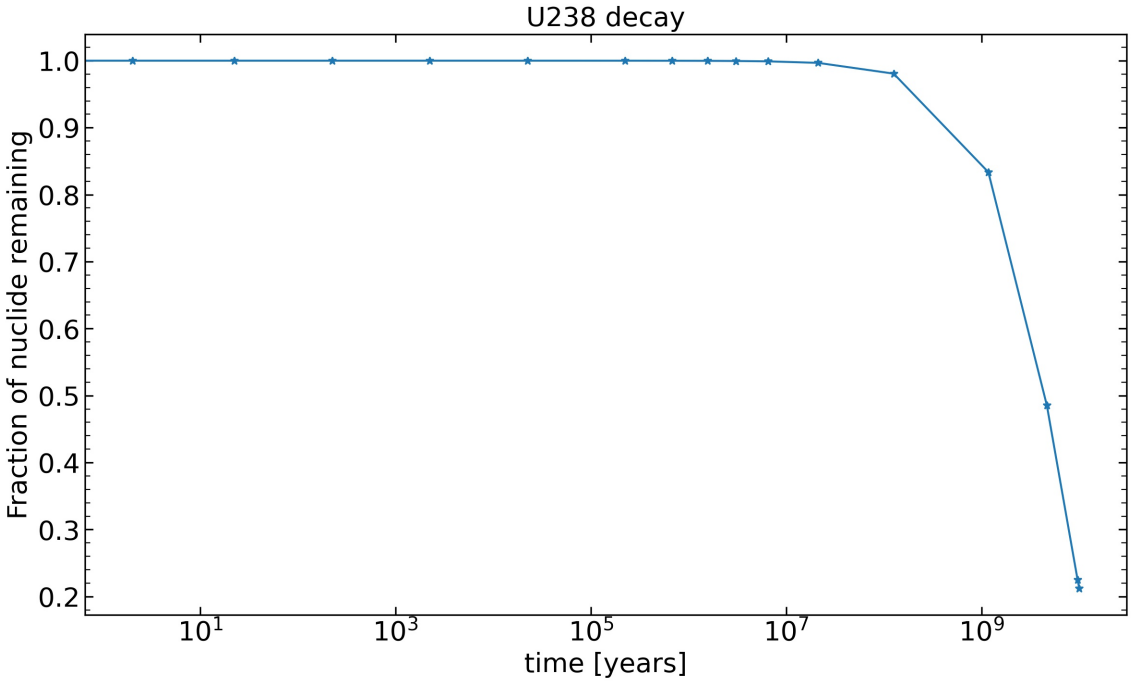

Figure 2: Decay of U238 over $\approx 2t_{1/2}$, using the scipy default of 16 points (number of nuclides + 1).

Q2b) Plot the ratio of Pb206 to U238 as a function of time over a region where it's interesting. Does this make sense analytically? (All half-lives are short compared to U238, so you can approximate the U238 decaying instantly to lead). Now plot the ratio of Thorium 230 to U234 over a region where that is interesting.

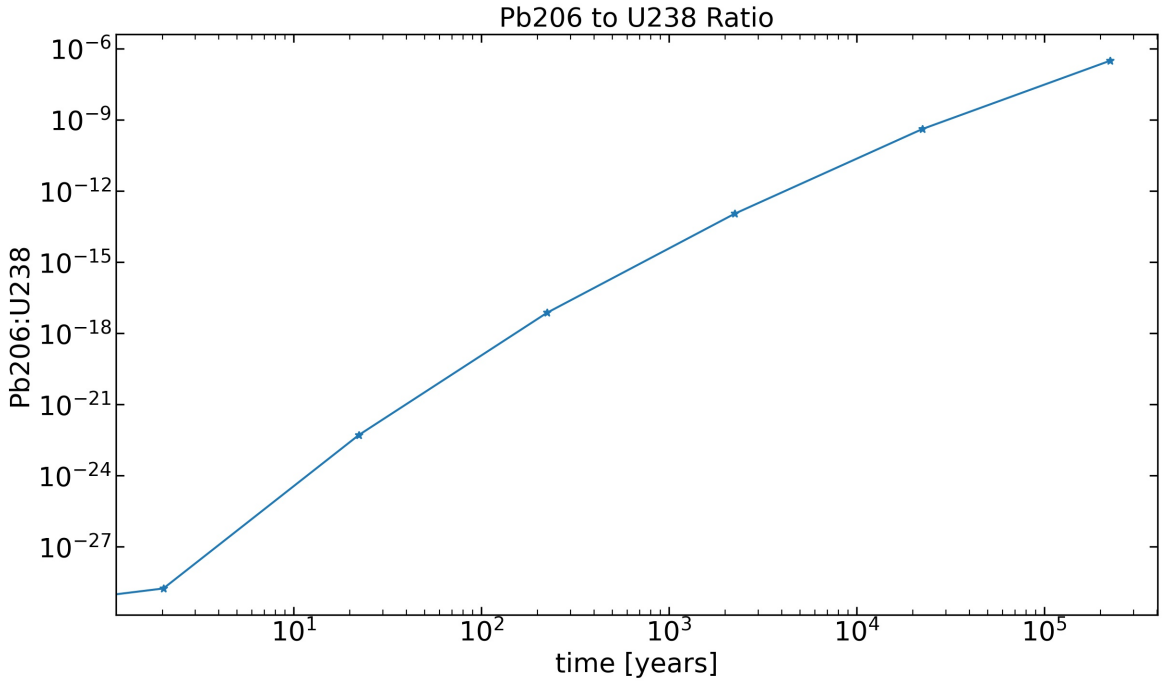

Figure 3: Pb206 is the last element in the chain, and therefore we can compute the ratio of Pb206 to U238 by dividing the last element by the first element as a function of time. We see the progressive increase in the ratio with time as more Pb206 is formed and more U238 decays, agreeing with our analytical expectation. Over an interval of $\approx 10^6$ years, the ratio jumps from $\approx 10^{-27}$ to $\approx 10^{-6}$.

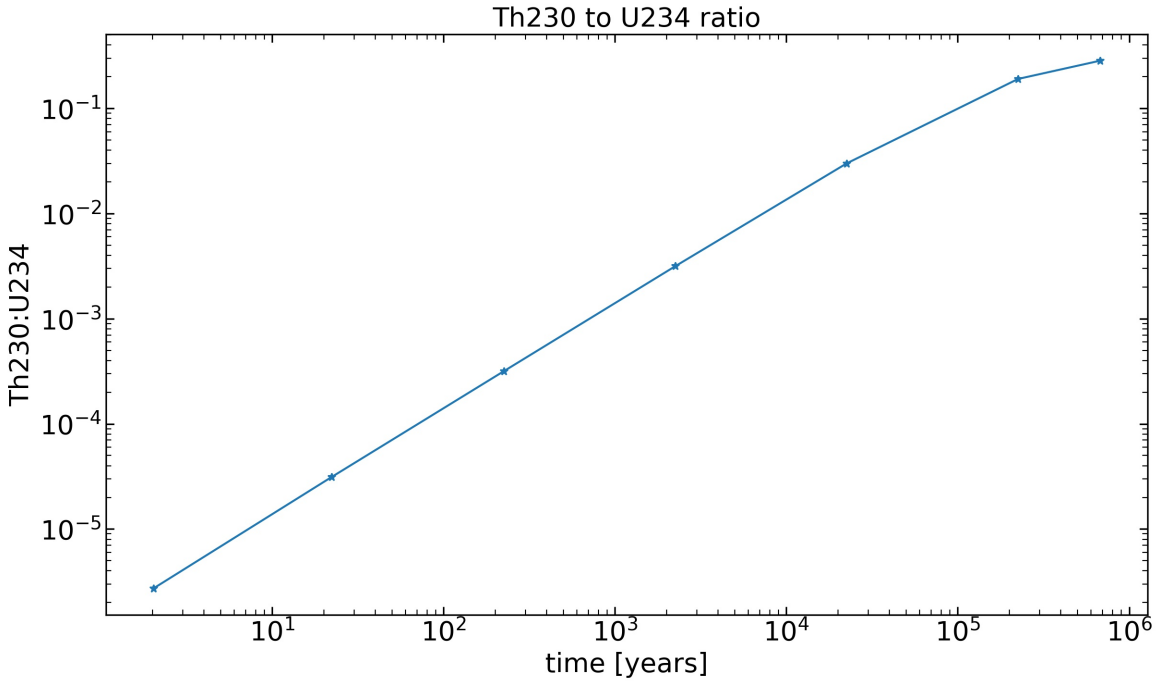

Figure 4: In this instance, the ratio evolves over a much smaller timescale, and earlier on, given the two nuclides are further up in the chain than Pb206. Over a period of $\approx 10^6$ years, the ratio increases from $10^{-6} \rightarrow 10^{-1}$.

Q3a) Look at file `dish_zenith.txt`, containing data for prototype telescope dish. The end result is that `dish_zenith.txt` contains the (x, y, z) positions in mm of a few hundred targets placed on the dish. The ideal telescope dish should be a rotationally symmetric paraboloid. We will try to measure the shape of the paraboloid, and see how well we did. For the dish pointing in the z-direction, we have a rotationally symmetric paraboloid, $z-z_0 = a((x-x_0)^2+(y-y_0)^2)$ where we need to solve for x_0, y_0, x_0, a. Show that we can pick a set of new parameters that make the problem linear. What are these new parameters and how do they relate to the old ones?

We can pick a set of new parameters by expanding out the formula for the parabola and rearranging as such,

$$z = a(x^2 + y^2) - 2ax_0x - 2ay_0y + (z_0 + ax_0^2 + ay_0^2)$$

$$\implies z = a(x^2 + y^2) + Cx + Dy + E \tag{4}$$

$$\text{for } \; a = a,\, C = -2ax_0,\, D = -2ay_0,\, E = z_0 + ax_0^2 + ay_0^2.$$

Therefore, given we now have a linear equation we can rewrite our paraboloid equation with these new parameters, which are related to the old parameters (x_0, y_0, z_0, a) by the following set of equations;

$$x_0 = \frac{C}{-2a},\ y_0 = \frac{D}{-2a},\ z_0 = E - \frac{C}{4a} - \frac{D}{4a},\ a = a. \tag{5}$$

Q3b) Carry out the fit, what are the best-fit parameters?

As we saw in class, we can find the best-fitting model by minimizing chi-square, defined as

$$\chi^2 = (d - A(m))^T N^{-1} (d - A(m)) \tag{6}$$

where d is our data (in this case z) and $A(m)$ is our model that is dependent on the parameters m (in this case we also have x, y as inputs). We take the result we derived in class for a linear least squares fit, where the best fit parameters m are given by,

$$m = (A^T N^{-1} A)^{-1} A^T N^{-1} d \tag{7}$$

where for the present we have assumed N is the identity matrix. We also saw in class that the parameter covariance matrix is given by $(A^T N^{-1} A)^{-1}$, so we may estimate the errors on our final parameters as well.

After defining our matrix A to be a nxm matrix, for n being the number of data points and m being the number of parameters (4), we can populate the matrix given the coordinates x, y from the dish data and carry out our linear least squares fit.

Therefore, our best-fit parameters (and their corresponding parameters from the original paraboloid equation) are;

$$\begin{aligned} a &= 0.00016670445477401355 \\ C &= 0.0004535990279795366 \\ D &= -0.01941155885263565 \\ E &= -1512.3118166739077 \\ x_0 &= -1.3604886221980108 \\ y_0 &= 58.22147608157856 \\ z_0 &= -1483.8813229442173 \end{aligned} \tag{8}$$

Q3c) Estimate noise in the data, and from that, uncertainty in a. Our target focal length was 1.5 m. What did we actually get, and what is the error bar? When calculating the error bar for the focal length, feel free to approximate using a first-order Taylor expansion.

We know from class that to get the noise from a large number of data points (which we have here), we can take the noise to be the square root of the variance of the residuals (as their magnitudes are relatively constant). Therefore, we can compute the noise from the variance of the residuals as;

$$\sqrt{\sigma^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - A(m_{BF}))^2} \quad \text{for} \quad m_{BF} = \text{best fit parameters}. \tag{9}$$

From here, we can compute the uncertainty in our best fit parameters as $(A^T N^{-1} A)^{-1}$, where $N = \sigma$ from above. We have already returned/computed the parameter covariance matrix $(A^T N^{-1} A)^{-1}$ for the case where $N = I_{nxn}$, and therefore simply dividing by our new N (given N is just a constant) will give us the new parameter covariance matrix. From here, we can obtain the uncertainty in a by taking the square root of the first (zeroth) diagonal entry (given our construction of the $A(m)$).

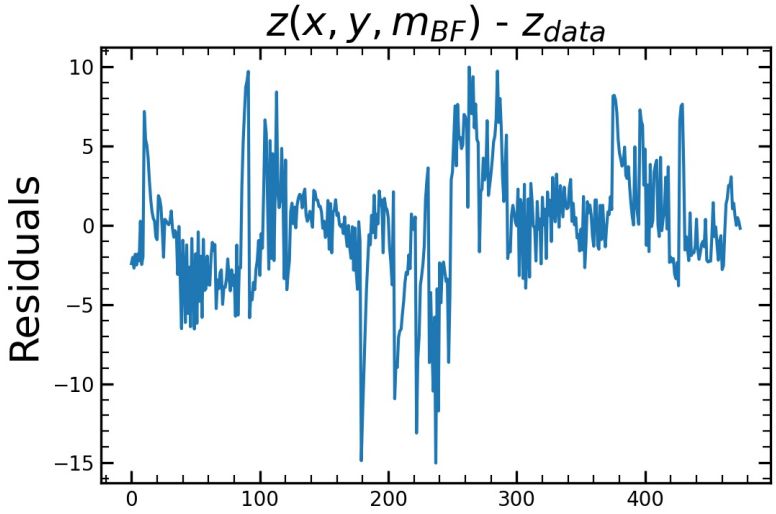

Figure 5: The residuals appear to be roughly constant in magnitude and random across the data. Therefore we can take the noise to be the RMS of the residuals.

Therefore, the uncertainty on a calculated in this manner is $\approx 8.8 * 10^{-9}$ mm.

We can find the focal length by considering a paraboloid (3D) going through $(0, 0)$, which implies $(x_0, y_0, z_0) = (0, 0, 0)$, and hence, $z = a(x^2 + y^2)$. Given the equation in the question $y = \frac{x^2}{4f}$, we can rewrite our equation for z in cylindrical coordinates as $z = \frac{R^2}{4f} \implies \frac{R^2}{4f} = aR^2 \implies f = \frac{1}{4a}$.

Therefore, focal length may be written in terms of a as, $f = \frac{1}{4a}$, where we have calculated a from our linear least squares fit above. We can propagate the uncertainty on a through this formula for f as,

$$\delta f = \sqrt{(\frac{df}{da}\delta a)^2}, \quad \text{where} \quad \frac{df}{da} = -\frac{1}{4a^2} \implies \delta f = \sqrt{\frac{(\delta a)^2}{16a^4}} = \frac{\delta a}{4a^2}. \tag{10}$$

Therefore, the focal length is $1.49966 \pm 8 * 10^{-5}$ m.