PHYS 512 Problem Set 3

Teophile Lemay, 281081252

1

The 4th order Runge-Kutta (RK4) method is a well defined algorithm for solving ODEs. For some first order ODE $\frac{\mathrm{d}y}{\mathrm{d}x} = f(x, y)$ and a step size h, RK4 is defined as

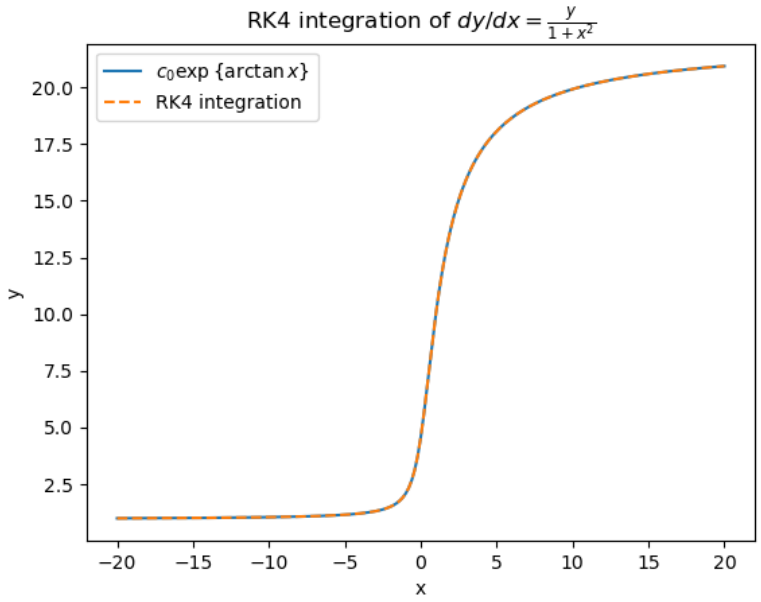
$$\begin{aligned} k_1 &= h \cdot f(y, x) \\ k_2 &= h \cdot f(y + \frac{1}{2}k_1, x + \frac{1}{2}h) \\ k_3 &= h \cdot f(y + \frac{1}{2}k_2, x + \frac{1}{2}h) \\ k_4 &= h \cdot f(y + k_3, x + h) \\ y(x + h) &= y(x) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned}$$

This algorithm is implemented directly in my "rk4_step" function which was used to evaluate

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{y}{1 + x^2}$$

as shown in figure 1.

Figure 1: basic RK4 integration of $\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{y}{1+x^2}$ $(N = 200)$



Each of the RK4 terms after k_1 can be described as a second order Taylor expansion using the previous term.

$$
\begin{aligned}
k_1 &= h \cdot f(y, x) \\
k_2 &= h \cdot f(y + \frac{1}{2}k_1, x + \frac{1}{2}h) = h \cdot \left(f(y,x) + \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}x}k_1 \right) \\
&= h \cdot \left(f(y,x) + \frac{h}{2}\frac{\mathrm{d}}{\mathrm{d}x}f(y,x) \right) \\
k_3 &= h \cdot f(y + \frac{1}{2}k_2, x + \frac{1}{2}h) = h \cdot \left(f(y,x) + \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}x}k_2 \right) \\
&= h \cdot \left(f(y,x) + \frac{h}{2}\frac{\mathrm{d}}{\mathrm{d}x}\left(f(y,x) + \frac{h}{2}\frac{\mathrm{d}}{\mathrm{d}x}f(y,x) \right) \right) \\
&= h \cdot \left(f(y,x) + \frac{h}{2}\frac{\mathrm{d}}{\mathrm{d}x}f(y,x) + \frac{h^2}{4}\frac{\mathrm{d}}{\mathrm{d}x}f(y,x) \right) \\
k_4 &= h \cdot f(y + k_3, x + h) = h \cdot \left(f(y,x) + \frac{\mathrm{d}}{\mathrm{d}x}k_3 \right) \\
&= h \cdot \left(f(y,x) + h\frac{\mathrm{d}}{\mathrm{d}x}\left(f(y,x) + \frac{h}{2}\frac{\mathrm{d}}{\mathrm{d}x}\left(f(y,x) + \frac{h}{2}\frac{\mathrm{d}}{\mathrm{d}x}f(y,x) \right) \right) \right) \\
&= h \cdot \left(f(y,x) + h\frac{\mathrm{d}}{\mathrm{d}x}f(y,x) + \frac{h^2}{2}\frac{\mathrm{d}^2}{\mathrm{d}x^2}f(y,x) + \frac{h^3}{4}\frac{\mathrm{d}^4}{\mathrm{d}x^4}f(y,x) \right) \\
y_{rk4}(x+h) &= y(x) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}
$$

Given the proper coefficients, RK4 is equivalent to a 4th order polynomial so

$$y(x+h) = y_{rk4}(x+h) + \frac{h^5}{5!}\frac{\mathrm{d}^4}{\mathrm{d}x^4}f(y,x) + \mathcal{O}(h^6)$$

If the step size is reduced to $h/2$, repeating the second order expansion on k_1', k_2', k_3', and k_4' (' designates the $h/2$ step) produces

$$y_{rk4}(x + h/2) = y(x) + \frac{1}{6}(k_1' + 2k_2' + 2k_3' + k_4')$$

$$y(x + h/2) = y_{rk4}(x + h/2) + \left(\frac{h}{2}\right)^5 \frac{\mathrm{d}^4}{\mathrm{d}x^4}f(y,x) + \mathcal{O}(h^6)\ .$$

A second $h/2$ step (') gives

$$y_{rk4'}(x+h) = y(x + h/2) + \frac{1}{6}(k_1' + 2k_2' + 2k_3' + k_4')$$

$$y(x+h) = y(x + h/2) + \frac{1}{6}(k_1' + 2k_2' + 2k_3' + k_4') + \left(\frac{h}{2}\right)^5 \frac{\mathrm{d}^4}{\mathrm{d}x^4}f(y,x) + \mathcal{O}(h^6)$$

$$y(x+h) = y_{rk4}(x + h/2) + \left(\frac{h}{2}\right)^5 \frac{\mathrm{d}^4}{\mathrm{d}x^4}f(y,x) + \mathcal{O}(h^6) + \frac{1}{6}(k_1' + 2k_2' + 2k_3' + k_4') + \left(\frac{h}{2}\right)^5 \frac{\mathrm{d}^4}{\mathrm{d}x^4}f(y,x) + \mathcal{O}(h^6)\ .$$

$$y(x+h) = y_{rk4}(x + h/2) + \frac{1}{6}(k_1' + 2k_2' + 2k_3' + k_4') + 2\left(\frac{h}{2}\right)^5 \frac{\mathrm{d}^4}{\mathrm{d}x^4}f(y,x) + \mathcal{O}(h^6)\ .$$

Thus, the leading error term for $y(x+h)$ after 2 RK4 steps of $h/2$ is $\frac{h^5}{16}\frac{\mathrm{d}^4}{\mathrm{d}x^4}f(y,x)$ compared to $h^5\frac{\mathrm{d}^4}{\mathrm{d}x^4}f(y,x)$ for a single step of length h. Using this result, I can make use of both step lengths to eliminate the 5^{th} order error term.

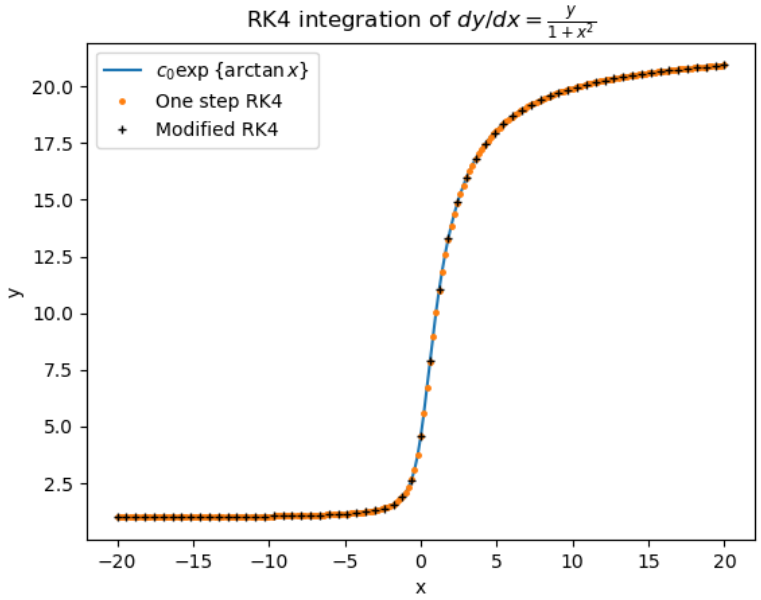
$$y_{h/2}(x+h) = y_{true}(x+h) - \frac{h^5}{16}\frac{\mathrm{d}^4}{\mathrm{d}x^4}f(y,x) + \mathcal{O}(h^6)$$

$$y_h(x+h) = y_{true}(x+h) - h^5 \frac{\mathrm{d}^4}{\mathrm{d}x^4} f(y,x) + \mathcal{O}(h^6)$$

$$\boxed{\frac{16y_{h/2}(x+h) - y_h(x+h)}{15} = y(x+h) + \mathcal{O}(h^6)}\,.$$

For each step H, this modified RK4 algorithm computes a one step RK4 3 times, requiring 12 function calls compared to the 4 required by a default RK4 implementation. Assuming that function evaluations are the most important complexity consideration, then the modified RK4 should be able to achieve greater accuracy than the one step RK4 while using $N/3$ points. Figure 2 shows a comparison of the expected result, basic RK4 integration with $N = 200$ steps, and my modified RK4 integration with $N_{h/2} = 66$ steps. For the given ODE, the root mean squared deviation (RMSD) for the modified RK4 integration is more than 20 times smaller than for the default RK4, with values of $RMSD_{h/2} \approx 1.04 \cdot 10^{-4}$ and $RMSD_h \approx 2.32 \cdot 10^{-3}$ respectively. All code for this question can be found in the file "Q1_runge_kutta.py".s

Figure 2: Modified RK4 integration of $\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{y}{1+x^2}$ $(N = 66)$



2

2.1 a)

A single stage exponential decay relates can be described by the differential equation

$$\frac{\mathrm{d}N}{\mathrm{d}t} = -\alpha N$$

with solution

$$N(t) = N_0 e^{-\frac{t}{\tau}\ln 2}$$

where τ is the half-life. Using this description, the decay chain for uranium-238 (figure 3) can be translated into a series of differential equations:

$$\frac{\mathrm{d}N_{U238}}{\mathrm{d}t} = -\frac{\ln 2}{\tau_{U238}} \tag{1}$$

$$\frac{\mathrm{d}N_{Th234}}{\mathrm{d}t} = \frac{\ln 2}{\tau_{U238}} N_{U238} - \frac{\ln 2}{\tau_{Th234}} N_{Th234} \tag{2}$$

$$\frac{\mathrm{d}N_{Pa234}}{\mathrm{d}t} = \frac{\ln 2}{\tau_{Th234}} N_{Th234} - \frac{\ln 2}{\tau_{Pa234}} N_{Pa234} \tag{3}$$

$$\frac{\mathrm{d}N_{U234}}{\mathrm{d}t} = \frac{\ln 2}{\tau_{Pa234}} N_{Pa234} - \frac{ln2}{\tau_{U234}} N_{U234} \tag{4}$$

$$\frac{\mathrm{d}N_{Th230}}{\mathrm{d}t} = \frac{\ln 2}{\tau_{U234}} N_{U234} - \frac{\ln 2}{\tau_{Th230}} N_{Th230} \tag{5}$$

$$\frac{\mathrm{d}N_{Ra226}}{\mathrm{d}t} = \frac{\ln 2}{\tau_{Th230}} N_{Th230} - \frac{\ln 2}{\tau_{Ra226}} N_{Ra226} \tag{6}$$

$$\frac{\mathrm{d}N_{Rn222}}{\mathrm{d}t} = \frac{\ln 2}{\tau_{Ra226}} N_{Ra226} - \frac{\ln 2}{\tau_{Rn222}} N_{Rn222} \tag{7}$$

$$\frac{\mathrm{d}N_{Po218}}{\mathrm{d}t} = \frac{\ln 2}{\tau_{Rn222}} N_{Rn222} - \frac{\ln 2}{\tau_{Po218}} N_{Po218} \tag{8}$$

$$\frac{\mathrm{d}N_{Pb214}}{\mathrm{d}t} = \frac{\ln 2}{\tau_{Po218}} N_{Po218} - \frac{\ln 2}{\tau_{Pb214}} N_{Pb214} \tag{9}$$

$$\frac{\mathrm{d}N_{Bi214}}{\mathrm{d}t} = \frac{\ln 2}{\tau_{Pb214}} N_{Pb214} - \frac{\ln 2}{\tau_{Bi214}} N_{Bi214} \tag{10}$$

$$\frac{\mathrm{d}N_{Po214}}{\mathrm{d}t} = \frac{\ln 2}{\tau_{Bi214}} N_{Bi214} - \frac{\ln 2}{\tau_{Po214}} N_{Po214} \tag{11}$$

$$\frac{\mathrm{d}N_{Pb210}}{\mathrm{d}t} = \frac{\ln 2}{\tau_{Po214}} N_{Po214} - \frac{\ln 2}{\tau_{Pb210}} N_{Pb210} \tag{12}$$

$$\frac{\mathrm{d}N_{Bi210}}{\mathrm{d}t} = \frac{\ln 2}{\tau_{Pb210}} N_{Pb210} - \frac{\ln 2}{\tau_{Bi210}} N_{Bi210} \tag{13}$$

$$\frac{\mathrm{d}N_{Po210}}{\mathrm{d}t} = \frac{\ln 2}{\tau_{Bi210}} N_{Bi210} - \frac{\ln 2}{\tau_{Po210}} N_{Bi210} \tag{14}$$

$$\frac{\mathrm{d}N_{Pb206}}{\mathrm{d}t} = \frac{\ln 2}{\tau_{Po210}} N_{Po210} \tag{15}$$

The values in the table of half-lives for the uranium-238 decay chain (in "integration.pdf" slides) were all converted to seconds and the system of differential equations was put together in a function called "decay_fun" which takes an initial value array y and returns an array of derivative values for each isotope by adding the half-life values to the equations above. The resulting system of differential equations was solved using SciPy's intial value problem ODE solver "scipy.integrate.solve_ivp". For this type of problem, I forced the SciPy solver to use an implicit method since the chain decay is a stiff system of equations with a very large range of decay times. Figure 3 shows the progress of the decay chain over 10 uranium-238 half-lives. As expected, on this extremely long time scale, uranium-238 shows obvious exponential decay, and the amount of lead-206, the final product, increases until it reaches a steady state once there is approximately no remaining uranium-238. All code used for question 2 can be found in the file "Q2_uranium_decay.py".

2.2 b)

Figure 4 shows the time evolution of the ratios of Pb-206 to U-238 and Th-230 to U-234. As expected, the ratio of the final product of the decay chain to U-238 looks exponential. This is to be expected as the half-life of U-238 is much greater than all other intermediate half-lives so the decay chain can be compared to one step over extremely long time intervals. The ratio of Th-230 to U-234 is more sensitive to smaller time scales. This is also expected since their half-lives are much closer to each other than that of U-238 to any member of the chain.

Figure 3: Uranium-238 decay chain results

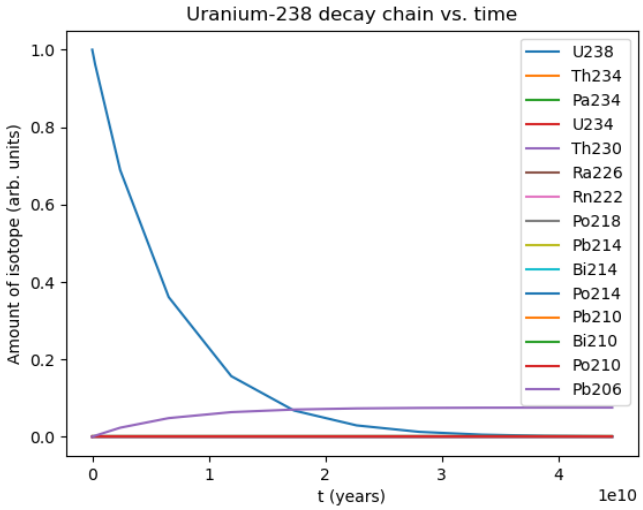
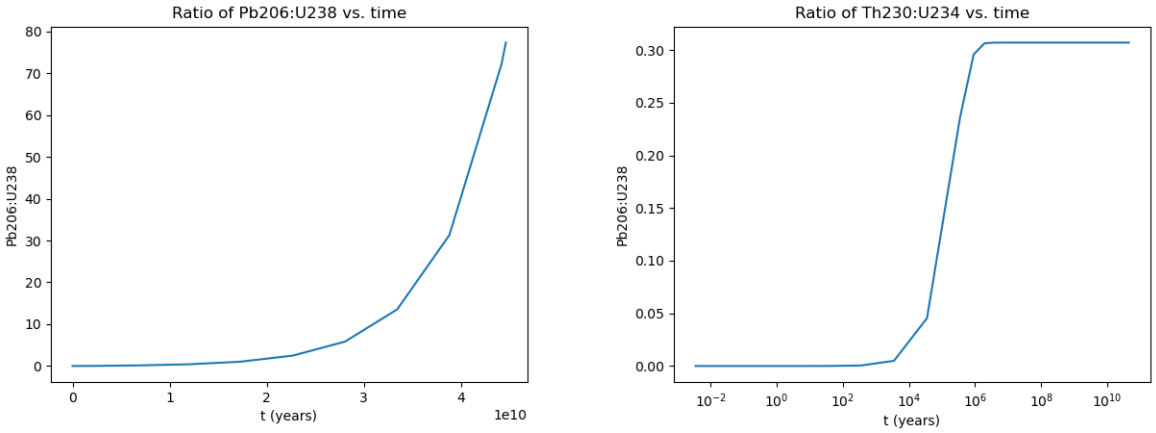


Figure 4: Ratios of Pb-206 to U-238 and Th-230 to U-234



3

x,y,z are locations (mm) of targets on dish. Dish shape is expected to be rotationally symmetric paraboloid.

3.1 a)

The equation for a rotationally symmetric paraboloid in 3 dimensions is

$$z - z_0 = a\left((x - x_0)^2 + (y - y_0)^2\right)$$

$$z = a\left((x - x_0)^2 + (y - y_0)^2\right) + z_0 \ .$$

In order to solve this equation with a linear least squares fit, I need some new set of parameters such that the equation can be written in the form

$$z = Am$$

where A is a matrix of functions of x and y without unknown parameters, and m is a vector of tuneable parameters to fit to the data. Expanding the quadratic terms gives

$$z = ax^2 - 2ax_0x + ax_0^2 + ay^2 - 2ay_0y + ay_0^2 + z_0$$

$$z = ax^2 + ay^2 + -2ax_0x - 2ay_0y + (ax_0^2 + ay_0^2 + z_0)$$

$$\mathbf{z} = \begin{bmatrix} 1 & \mathbf{x} & \mathbf{y} & \mathbf{x^2 + y^2} \end{bmatrix} \begin{pmatrix} ax_0^2 + ay_0^2 + z_0 \\ -2ax_0 \\ -2ay_0 \\ a \end{pmatrix} = \begin{bmatrix} 1 & \mathbf{x} & \mathbf{y} & \mathbf{x^2 + y^2} \end{bmatrix} \cdot \begin{pmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{pmatrix} = \mathbf{Am}$$

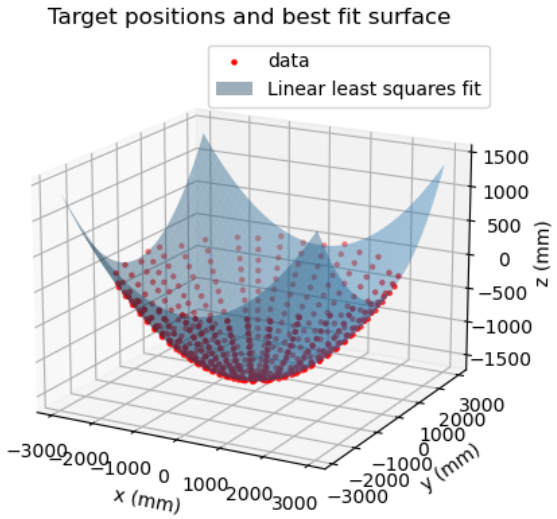
3.2 b)

The linear least squares fit was carried out as performed in class (same method as "polyfit_class.py"), with the corresponding A and m matrices described in part a). The best-fit parameters are

$$\begin{aligned} a &= 1.667 \cdot 10^{-4}\text{mm}^{-1} = 1.667 \cdot 10^{-1}\text{m}^{-1} \\ x_0 &= -1.360\text{mm} \\ y_0 &= 58.221\text{mm} \\ z_0 &= -1512.877\text{mm} \ . \end{aligned}$$

As shown in figure 5, using the best fit parameters in the paraboloid equation produces a close match to the target position data.

Figure 5: Target positions and best fit paraboloid



3.3 c)

Noise is the difference between the recorded data and the true values it should take on if perfect measurements were possible. Assuming that the data is noisy in the z values and that noise in x and y is negligible, I can estimate noise as the difference

$$\mathbf{n} \approx \mathbf{z} - \mathbf{Am}$$

$$\mathbf{N} = \mathbf{n} \otimes \mathbf{n} \ .$$

Error in the fit parameters can then be estimated as

$$\sigma_m^2 = \text{diag}\left(\mathbf{A}^T \mathbf{N}^{-1} \mathbf{A}\right)^{-1} \ .$$

Evaluating this equation gives $\sigma_a = 7 \cdot 10^{-17}$. This value is less than rounding error and is probably a very inaccurate estimate of the error for a.

The focal point F of a 3d paraboloid is related to it's depth D and radius at the rim R by

$$4FD = R^2$$

$$F = R^2/4D \ .$$

Since the dish is oriented such that it extends in the z axis from $z = z_0 \to z = 0$, the depth is $D = |z_0|$. Furthermore, at $y = y_0$, the radius of at the rim is derived from

$$0 - z_0 = a\left((x - x_0)^2 + (y_0 - y_0)^2\right) \ .$$

Since z_0 is negative

$$|z_0| = a\left((x - x_0)^2\right) = aR^2$$

$$R^2 = \frac{-z_0}{a} \ .$$

Thus

$$F = \frac{|z_0|}{4a|z_0|} = 1/4a$$

$$\boxed{F = \frac{1}{4 \cdot 1.667 \cdot 10^{-1}} = 1.4997 \pm 6 \cdot 10^{-13}\text{m}} \ .$$

This result is very close to the expected focal length. The expected value is not within the uncertainty of the calculated value, however, the error in a is probably extremely under-estimated so this is expected.