1. For `rk4_step()`, it requires 4 evaluations of $f(x)$. For `rk4_stepd()`, although it calls `rk4_step()` 3 times, we saved one function call to $f(x, y)$ by saving the value of k_1 at (x, y) when calling `rk4_step()` with step size h and reuse it when calling `rk4_step()` again with step size $h/2$. This means each `rk4_stepd()` requires 11 evaluations of $f(x)$. We round the number $200 \cdot 4/11$ to the nearest integer as the number of steps for `rk4_stepd()`. The output of the code is as follows,

```
Error using rk4_step: 0.00011776140531973092
Error using rk4_stepd: 1.7446376452210383e-05
```

   From the output, we can see `rk4_stepd()` is more accurate than `rk4_step()` at around the same number of function calls to $f(x, y)$.

2. a) Consider for the vector of remaining elements $\boldsymbol{y}(t)$ and a corresponding half life vector $\boldsymbol{t_{1/2}}$. The differential equation can be represented as

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} = M\boldsymbol{y}$$

   where M can be constructed base on the fact that

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = -\frac{\ln 2}{(t_{1/2})_i} y_i + \frac{\ln 2}{(t_{1/2})_{i-1}} y_{i-1}$$

   i.e. it decays away but also gaining the decayed products by the element upper in the chain. We used "Radau" solver for this problem.

   b) The ratio of Pb-206 over U-238 is shown in Figure 1. It is a monotonic increasing function and has the shape of a exponential function. If we assume U-238 decays to Pb-206 directly with some decay rate λ, we expect this ratio to be of the form

$$\frac{N(\text{Pb-268})}{N(\text{U-238})} = \frac{1 - \mathrm{e}^{-\lambda t}}{\mathrm{e}^{-\lambda t}} = \mathrm{e}^{\lambda t} - 1$$

   which is precisely the function we can see from Figure 1. The ratio of Th230 over U-234 is shown in Figure 2. The region we plotted is where U-234 is formed and then decays to Th-230, and the ratio reaches a stable value afterwards as the balance is formed.

3. a) Write

$$\begin{aligned} z = a((x - x_0)^2 + (y - y_0)^2) + z_0 &= ax^2 - 2ax_0x + ax_0^2 + ay^2 - 2ay_0y + ay_0^2 + z_0 \\ &= a(x^2 + y^2) - 2ax_0x - 2ay_0y + (ax_0^2 + ay_0^2 + z_0) \end{aligned}$$

   which is linear in $x^2 + y^2, x, y$. Suppose

$$z = m_1(x^2 + y^2) + m_2x + m_3y + m_4,$$

   i.e.

$$\begin{aligned} m_1 &= a, \\ m_2 &= -2ax_0, \\ m_3 &= -2ay_0, \\ m_4 &= ax_0^2 + ay_0^2 + z_0 \end{aligned}$$

   then

$$\begin{aligned} a &= m_1, \\ x_0 &= -\frac{m_2}{2a} = -\frac{m_2}{2m_1}, \\ y_0 &= -\frac{m_3}{2a} = -\frac{m_3}{2m_1}, \\ z_0 &= m_4 - ax_0^2 - ay_0^2 = m_4 - \frac{m_2^2}{4m_1} - \frac{m_3^2}{4m_1}. \end{aligned}$$

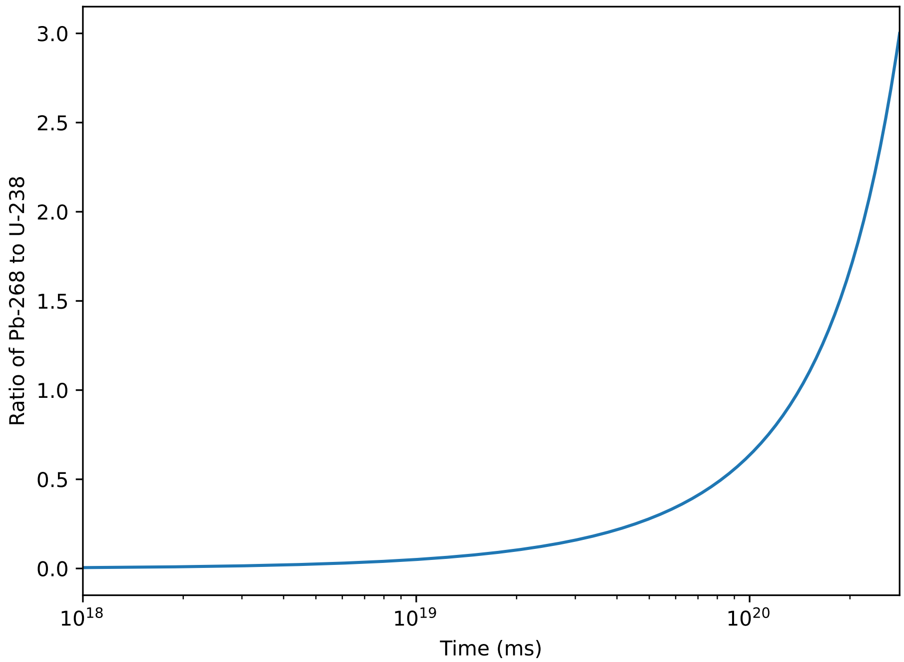

Figure 1: Ratio of Pb-206 over U-238 as a function of time.

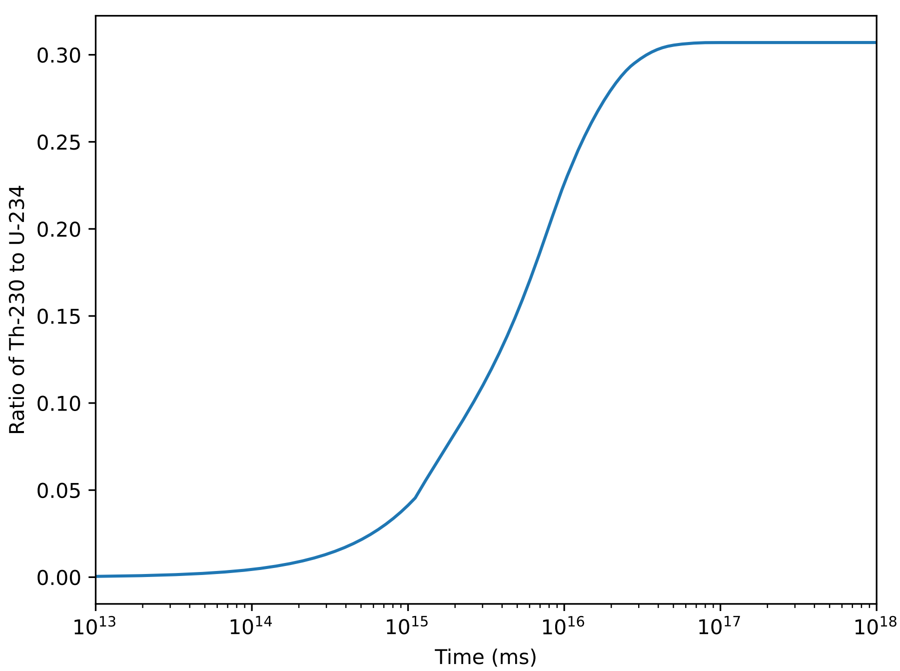

Figure 2: Ratio of Th230 over U-234 as a function of time for time between 1×10^{13} ms to 1×10^{18} ms.

b) We performed the linear fit of dependent variable z using independent variables x^2+y^2, x, y and we get

```
Fit result:
m = [ 1.66704455e-04  4.53599028e-04 -1.94115589e-02 -1.51231182e+03]
a = 0.00016670445477401355
x0 = -1.360488622198024
y0 = 58.221476081578636
z0 = -1512.8772100367885
```

c) The noise is estimated using the standard deviation between the sample values and fit values. This allows us to estimate the uncertainties on fitted parameters, including a. Since

$$f=\frac{1}{4a},$$

by (1st order) error propagation

$$\Delta f=\frac{\Delta a}{4a^2}.$$

Compute using Python, this gives us

```
Noise (mm): 3.768338648784727
Focal length (mm): 1499.6599841252162 0.5804077581892831
```

d) (Bonus) Let

$$x'=x-x_0,\ y'=y-y_0,\ z'=z-z_0$$

to fix the origin and let

$$x''=\cos\theta x'+\sin\theta y',\ y''=-\sin\theta x'+\cos\theta y',\ z''=z'$$

to correct possible angle between x, y and the principle axis. We have

$$\begin{aligned} z''&=ax''^2+by''^2\\ &=a\left[\cos^2\theta x'^2+\sin^2\theta y'^2+2\cos\theta\sin\theta x'y'\right]+b\left[\sin^2\theta x'^2+\cos^2\theta y'^2-2\cos\theta\sin\theta x'y'\right]\end{aligned}$$

which is in the form of

$$z=m_1x^2+m_2x+m_3y^2+m_4y+m_5xy+m_6.$$

We only care about a, b, we can see that using

$$\begin{cases} m_1=a\cos^2\theta+b\sin^2\theta,\\ m_3=a\sin^2\theta+b\cos^2\theta,\\ m_5=2a\cos\theta\sin\theta-2b\cos\theta\sin\theta,\end{cases}$$

we have 3 equations and 3 unknowns which allows us to solve for a, b, θ. First we note

$$m_1-m_3=(a-b)(\cos^2\theta-\sin^2\theta)=(a-b)\cos(2\theta)$$

and combine with

$$m_5=(a-b)\sin(2\theta),$$

we have

$$\theta=\frac{1}{2}\arctan\frac{m_5}{m_1-m_3}.$$

Then using

$$m_1+m_3=a+b$$

and the expression for m_5, we have

$$\begin{cases} a=\frac{1}{2}\left(m_1+m_3+\frac{m_5}{\sin(2\theta)}\right),\\ b=\frac{1}{2}\left(m_1+m_3-\frac{m_5}{\sin(2\theta)}\right).\end{cases}$$

This is implemented using Python and we get the result of

```
Focal length in x (mm): 1508.4884408086855
Focal length in y (mm): 1490.21912124064
```

This shows the dish is not perfectly round.