NENG 685 Fall 2017
HW 2 Due Oct. 25, 2017

Name: Kevin Choe

On homework:

- If you work with anyone else, document what you worked on together.
- Show your work.
- Always clearly label plots (axis labels, a title, and a legend if applicable).
- Homework should be done "by hand" (i.e. not with a numerical program such as MATLAB, Python, or Wolfram Alpha) unless otherwise specified. You may use a numerical program to check your work.
- If you use a numerical program to solve a problem, submit the associated code, input, and output (email submission is fine).

Do not write in the table to the right.

Problem	Points	Score
1	20	
2	10	
3	15	
4	20	
5	10	
6	25	
Total:	100	

1. (20 points) We often measure convergence by comparing one iteration to the previous iteration (rather than the solution, since we presumably don't know what it is). Imagine that you have software that gives the following solution vectors

$$x_{n-1} = \begin{pmatrix} 0.45 \\ 0.95 \\ 0.2 \\ -0.05 \\ 0.6 \end{pmatrix} \qquad x_n = \begin{pmatrix} 0.5 \\ 0.9 \\ 0.3 \\ -0.1 \\ 0.5 \end{pmatrix}$$

Calculate

a. The absolute and relative error using the 1 norm
b. The absolute and relative error using the 2 norm
c. The absolute and relative error using the infinity norm

Please see the attachment for calculation.

What is *least* restrictive (that is, what would cause the code to converge *first*)?

Please see the attachment for explanation.

Image that now $x_{n-1} = (0.49, 0.92, 0.4, -0.09, 0.51)^T$. Recalculate the convergence values for a)- c) above.

What do you observe? How does the **least** restrictive norm change? What does that mean about how you might select convergence criteria?

Please see the attachment for explanation.

2. (10 points) You have a piece of software that takes mesh spacing as an input variable. Imagine that you have the following relative error values for each mesh spacing (h) / number of mesh cells (N cells):

h	N cells	rel err
1.0	8	8.44660179e-03
0.5	16	2.30286448e-03
0.1	80	9.84273963e-05
0.05	160	2.48043656e-05
0.01	800	9.98488163e-07

One of the ways we characterize method performance is the order of convergence. We'd like to know how the error changes as we change the resolution of our discretization, in this case, mesh spacing.

Using a log-log plot, plot relative error as a function of

a. mesh spacing and
b. cell count.

Please see the attachment for the plots.

What is the functional relationship between error and mesh spacing / number of cells?

Please see the attachment for explanation.

3. (15 points) Consider the following four equally-spaced points on the interval $[x_0, x_3]$:

$$x_j = x_0 + jh \qquad j = 0, 1, 2, 3 \qquad h = (x_3 - x_0)/3\,.$$

Using the formula for the interpolating polynomial $P_3(x)$ you used in Question 1b of Homework 2, integrate $P_3(x)$ to derive the following Newton-Cotes formula, often referred to as **Simpson's three-eights rule**:

$$I(f(x)) \approx I_3(f(x)) = \frac{3h}{8}\Big[f(x_0) + 3f(x_0+h) + 3f(x_0+2h) + f(x_0+3h)\Big]$$

Note that you can substitute h in for the xs in P_3 because we are explicitly stating that the points are equally spaced (i.e. $x = x_0 + 3th$, where $t \in [0,1]$) $\therefore dx = 3hdt$, and x_0, x_1, and x_2 are modified accordingly).
For this problem, don't worry about an error term. Please see the attachment for calculation.

4. (20 points) Consider the following integral:

$$I = \int_2^4 \frac{x}{\sqrt{x^2-1}}dx\,.$$

(a) (6 points) Compute the integral exactly by hand. Please see the attachment for calculation.

(b) (8 points) Write a code that performs Composite Simpson's 3/8 rule to compute the integral. I advise something like
`I = CompSimp38(a,b,n)`
where `a` and `b` are the endpoints and `n` is the number of points to use in the integration (which must be divisible by 3!).

(c) (6 points) Experimentally determine the rate of convergence as a function of `h`. Specify the order of convergence.

5. (10 points) For $n = 100$ we will use this tridiagonal system of equations

$$\begin{aligned} 2x_0 - x_1 &= 0 \\ -x_{j-1}+2x_j - x_{j+1} &= j\,, \qquad j = 1,\dots,n-2 \\ -x_{n-2}+2x_{n-1} &= n-1 \end{aligned}$$

in a few different ways. You may use Python, MATLAB or another package or code language of your choice. Note: `NumPy` and `SciPy` are libraries that can be imported into Python and would be useful for this assignment.

(a) (2.5 points) Use built in Python or MATLAB commands to construct **A** and $\vec{b}$.

(b) (2.5 points) What is the condition number of **A**? What does this tell us? *(Hint: Use a Python built-in.)*

(c) (2.5 points) Solve this problem by explicitly inverting **A** and multiplying $\vec{b}$.

(a)-(c) are done in my Python script (HW2_P5), please run it and see A and b matrices in the Variable explorer. For (b) and (d), my condition number and plot are included in the attachment.

(d) (2.5 points) Use `scipy.linalg.solve` (Python) or the backslash operator (MATLAB) to solve the system. Please follow the instruction on the previous pg.

Plot both your explicit (c) and numerical (d) solutions on the same plot. Include axis labels, a title, and a legend. Please see the attachment.

6. (25 points) There are several fundamental alpha decay series (neptunium, uranium, thorium, and actinium). Here we'll look at the neptunium (which is the only one without the parent occurring naturally) as shown in the figure:

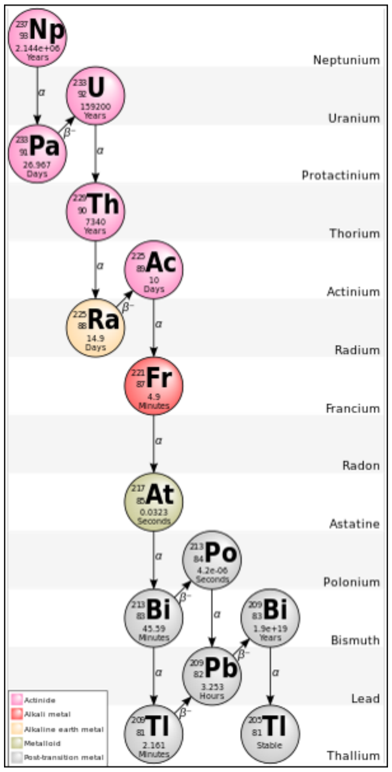


Figure 1: Neptunium Decay Series

This is a nice decay chain in the sense that there is only one non-linear term - the decay of ^{213}Bi, which can decay by β^- (97.8%) or α (2.2%). The population of each member, N, at any time t is given by the Bateman Equations:

$$\frac{dN_t(t)}{dt} = -\lambda_1 N_1(t)$$
$$\frac{dN_i(t)}{dt} = -\lambda_i N_i(t) + \lambda_{i-1} N_{i-1}(t)$$

which has the analytic solution of:

$$N_n(t) = \sum_{i=1}^{n} \left[N_i(0) \left(\prod_{j=1}^{n-1} \lambda_j \right) \left(\sum_{j=1}^{n} \left(\frac{e^{-\lambda_j t}}{\prod_{p=i,p\neq j}^{n} (\lambda_p - \lambda_j)} \right) \right) \right] \quad (1)$$

Find a solution to the system of ODEs for the neptunium series on a per atom basis. Plot the relative abundance of each isotope as a function of time spanning from 1 second to 1 day. Start with ^{221}Fr. Please see the attachment for explanation and plot.

Hint(s): In the HW directory, there is a python dictionary containing all of the relevant nuclear data in a file named nuclearData.txt. You can copy this directly into your code and use it - in fact it is recommended. Also, there is no reason to code up an ODE solver. Instead use Python's ODEINT (demonstrated) in class.

Comment on what happens if you start higher in the decay series. What about if you carry the decay to longer times?

If you carry the decay far enough in time, you start to see oscillations in the population. Comment on the cause of these oscillations.

BONUS (5 points): submit your code by providing read/clone access to an online version control repository where your code is stored (e.g. github or bitbucket). If you don't know what that means and want to learn about it, come talk to me or check out resources here: `http://software-carpentry.org/lessons.html` For Windows, you want to setup the Git GUI. NOTE: You will not be able to do this from AFIT's systems.

1. We often measure convergence by comparing on iteration to the previous iteration (rather than the solution, since we presumably don't know what it is). Imagine that you have software that gives the following solution vectors

$$X_{n-1} = \begin{pmatrix} 0.45 \\ 0.95 \\ 0.2 \\ -0.05 \\ 0.6 \end{pmatrix} \qquad X_n = \begin{pmatrix} 0.5 \\ 0.9 \\ 0.3 \\ -0.1 \\ 0.5 \end{pmatrix}$$

Let's figure out norms for each data sets!

norm 1: $\|\vec{X}\|_1 \equiv |X_1| + |X_2| + \cdots + |X_n|$.

$\|\vec{X}_{n-1}\|_1 \equiv |(0.45)| + (0.95)| + |(0.2)| + |(-0.05)| + (0.6) = \underline{2.25}$

norm 2: $\|\vec{X}\|_2 \equiv (|X_1|^2 + |X_2|^2 + \cdots + |X_n|^2)^{1/2}$

$\|\vec{X}_{n-1}\|_2 \equiv (|(0.45)|^2 + |(0.95)|^2 + |(0.2)|^2 + |(-0.05)|^2 + |(0.6)|^2)^{1/2} = \underline{1.23}$

norm ∞: $\|\vec{X}\|_\infty \equiv \max_{1 \le i \le n} |X_i|$

$\|\vec{X}_{n-1}\|_\infty \equiv 0.95$

Norm1:

$\|\vec{X}_n\|_1 \equiv |(0.5)| + |(0.9)| + |(0.3)| + |(-0.1)| + |(0.5)| = \underline{2.3}$

Norm2:

$\|\vec{X}_n\|_2 \equiv (|(0.5)|^2 + |(0.9)|^2 + |(0.3)|^2 + |(-0.1)|^2 + |(0.5)|^2)^{1/2} = \underline{1.19}$

norm ∞:

$\|\vec{X}_n\|_\infty \equiv 0.9$

a. Calculate the absolute and relative error using the 1 norm.

absolute error: $\underline{\|\vec{X}_n - \vec{X}_{n-1}\|_1} = \|(2.3) - (2.25)\| = \underline{0.05}$

relative error: $\dfrac{\|\vec{X}_n - \vec{X}_{n-1}\|_1}{\|\vec{X}_n\|_1} = \left\|\dfrac{(0.05)}{(2.3)}\right\| = \underline{0.02}$

b. Calculate the absolute and relative error using the 2 norm

absolute error: $\|\vec{X}_n - \vec{X}_{n-1}\|_2 = \|1.19 - 1.23\| = \underline{0.04}$

relate error: $\dfrac{\|\vec{X}_n - \vec{X}_{n-1}\|_2}{\|\vec{X}_n\|_2} = \left\|\dfrac{0.04}{1.19}\right\| = \underline{0.03}$

c. Calculate the absolute and relative error using the infinity norm.

absolute error: $\|\vec{X}_n - \vec{X}_{n-1}\|_\infty = \|0.9 - 0.95\| = \underline{0.05}$

relate error: $\dfrac{\|\vec{X}_n - \vec{X}_{n-1}\|_\infty}{\|\vec{X}_n\|_\infty} = \left\|\dfrac{0.05}{0.9}\right\| = \underline{0.05}$

What is least restrictive (what would cause the code to converge first)?: The method to adding all data induced the fastest convergence rate than the othe methods.

Image $x_{n-1} = \begin{pmatrix} 0.49 \\ 0.92 \\ 0.4 \\ -0.09 \\ 0.51 \end{pmatrix}$ Recalculate the convergence values.

For new x_{n-1}

norm 1: $\|\vec{x}_{n-1}\|_1 \equiv |0.49|+|0.92|+|0.4|+|-0.09|+|0.51| = 2.41$

norm 2: $\|\vec{x}_{n-1}\|_2 \equiv (|0.49|^2+|0.92|^2+|0.4|^2+|0.09|^2+|0.51|^2)^{1/2} = 1.23$

norm ∞ : $\|\vec{x}_{n-1}\|_\infty \equiv 0.92$

with $\vec{x}_n$ values we calculated previously, we can calculate,

norm1 absolute error : $\|\vec{x}_n - \vec{x}_{n-1}\|_1 = \|2.3 - 2.41\| = \underline{0.1}$

relative error : $\dfrac{\|\vec{x}_n - \vec{x}_{n-1}\|_1}{\|\vec{x}_n\|_1} = \left\|\dfrac{0.1}{2.3}\right\| = \underline{0.04}$

norm2 absolute error : $\|\vec{x}_n - \vec{x}_{n-1}\|_2 = \|1.19 - 1.23\| = \underline{0.04}$

relative error : $\dfrac{\|\vec{x}_n - \vec{x}_{n-1}\|_2}{\|\vec{x}_n\|_2} = \left\|\dfrac{0.04}{1.19}\right\| = \underline{0.03}$

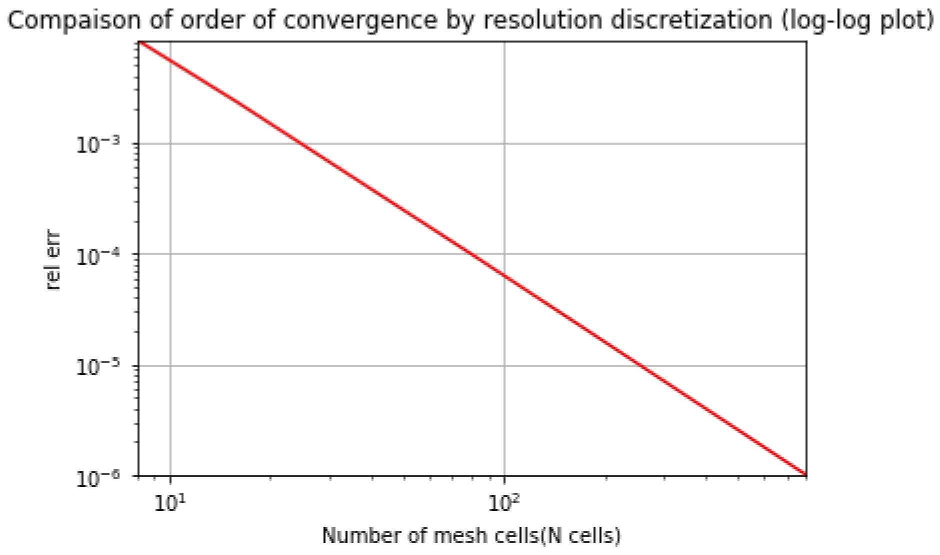
norm∞ absolute error : $\|\vec{x}_n - \vec{x}_{n-1}\|_\infty = \|0.9 - 0.92\| = \underline{0.02}$

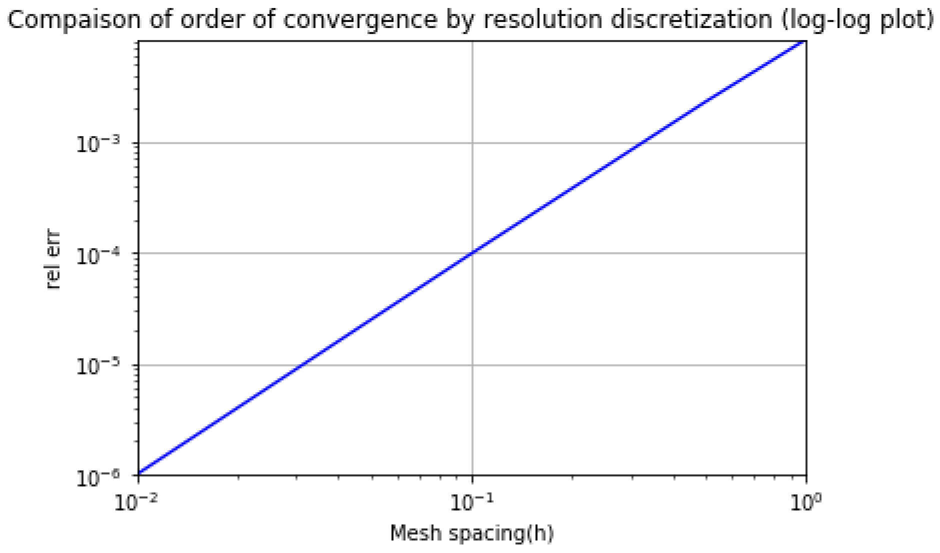
relative error : $\dfrac{\|\vec{x}_n - \vec{x}_{n-1}\|_\infty}{\|\vec{x}_n\|_\infty} = \left\|\dfrac{0.02}{0.9}\right\| = \underline{0.02}$

What do you observe? Convergence rate did not change for norm 2 and norm ∞. (they both have same absolute error and relative error as before with the previous $\vec{x}_{n-1}$ matrix). However I see that the norm 1 methods get some different values.

What does that mean about how you might select convergence criteria? With the norm 1 method, we can be more sensitive to recognize the difference between the two diffent data sets. Thus, this method has higher converging rate than the other two norms. If I want to go about more general (unbiased) outcome differences of data sets, I would chood norm 2. Norm ∞ calculates the least accurate differentiation between data sets. For example, if two different data sets have completely different data element distribution in range but have same highest data element, it will mess up its convergence calculation.

HW2 P.2 (a)(b) plots





What is the functional relationship between error and mesh spacing/number of cells?

For a given set of data(rel error in this case), the more number of cells(the higher resolutions) we create, the smaller mesh spacing the plot gets (increasing convergence rate). Also, the mesh spacing and number of mesh cells have one by one relationship, meaning that they are related to measured dataset linearly but not exponentially.

3. Consider the formula for the interpolating polynomial $P_3(x)$ integrate $P_3(x)$ to derive the Simpson's three-eights rule:

$x_j = x_0 + jh$; $j = 0,1,2,3$; $h = (x_3 - x_0)/3$ $\rightarrow$ $x_0 = x$; $x_1 = x_0 + h$; $x_2 = x_0 + 2h$; $x_3 = x_0 + 3$

$$P_3(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} f(x_0) + \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} f(x_1)$$

$$+ \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} f(x_2) + \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} f(x_3)$$

$$= \frac{(x-x_0-h)(x-x_0-2h)(x-x_0-3h)}{(\cancel{x_0}-\cancel{x_0}-h)(\cancel{x_0}-\cancel{x_0}-2h)(x_0-x_0-3h)} f(x_0) + \frac{(x-x_0)(x-x_0-2h)(x-x_0-3h)}{(x_0+h-x_0)(\cancel{x_0}+h-\cancel{x_0}-2h)(\cancel{x_0}+h-\cancel{x_0}-3h)} f(x_1)$$

$$+ \frac{(x-x_0)(x-x_0-h)(x-x_0-3h)}{(\cancel{x_0}+2h-\cancel{x_0})(\cancel{x_0}+2h-x_0-h)(\cancel{x_0}+2h-\cancel{x_0}-3h} f(x_2) + \frac{(x-x_0)(x-x_0-h)(x-x_0-2h)}{(\cancel{x_0}+3h-\cancel{x_0})(\cancel{x_0}+3h-\cancel{x_0}-h)(\cancel{x_0}+3h-\cancel{x_0}-2} f(x_3)$$

$$= f(x_0)\frac{(x-x_0-h)(x-x_0-2h)(x-x_0-3h)}{-6h^3} + \frac{(x-x_0)(x-x_0-2h)(x-x_0-3h)}{2h^3} f(x_1) + \frac{(x-x_0)(x-x_0-h)(x-x_0-3h)}{-2h^3} f($$

$$+ \frac{(x-x_0)(x-x_0-h)(x-x_0-2h)}{6h^3} f(x_3)$$

Let's try to integrate Lagrange terms separatly

$$\begin{cases} L_0 = \dfrac{(x-x_0-h)(x-x_0-2h)(x-x_0-3h)}{-6h^3} \rightarrow w_0 \\ L_1 = \dfrac{(x-x_0)(x-x_0-2h)(x-x_0-3h)}{2h^3} \rightarrow w_1 \\ L_2 = \dfrac{(x-x_0)(x-x_0-h)(x-x_0-3h)}{-2h^3} \rightarrow w_2 \\ L_3 = \dfrac{(x-x_0)(x-x_0-h)(x-x_0-2h)}{6h^3} \rightarrow w_3 \end{cases}$$

$w_{0\sim3}$ are calculated in the next couple pgs

$$\underline{\underline{w_2}} = \int_{x_0}^{x_3} L_2\,dx = \int_{x_0}^{x_3} \frac{(x-x_0)(x-x_0-h)(x-x_0-3h)}{-2h^3}\,dx$$

Let $x = x_0 + 3th \rightarrow x - x_0 = 3th \quad t \in [0,1]$

$dx = 3h\,dt$

$$w_2 = \frac{3h}{-2h^3}\int_0^1 (3th)(3th-h)(3th-3h)\,dt = -\frac{3}{2h^2}\int_0^1 3th\,(9t^2h^2 - 9th^2 - 3th^2 + 3h^2)\,dt$$

$$= -\frac{3}{2h^2}\int_0^1 (27t^3h^3 \underbrace{-27t^2h^3 - 9t^2h^3}_{-36t^2h^3} + 9th^3)\,dt = -\frac{3}{2h^2}\left[\frac{27t^4h^3}{4} - \frac{36t^3h^3}{3} + \frac{9t^2h^3}{2}\right]_0^1$$

$$= -\frac{3}{2h^2}\left(\frac{27h^3}{4} - \frac{48h^3}{4} + \frac{18h^3}{4}\right) = -\frac{3}{2h^2}\left(-\frac{3}{4}h^3\right) = \boxed{\frac{9h}{8}}$$

$$\underline{\underline{w_3}} = \int_{x_0}^{x_3} L_3\,dx = \int_{x_0}^{x_3} \frac{(x-x_0)(x-x_0-h)(x-x_0-2h)}{6h^3}\,dx.$$

Let $x = x_0 + 3th \rightarrow x - x_0 = 3th \quad t \in [0,1]$

$dx = 3h\,dt$

$$w_3 = \frac{3h}{6h^3}\int_0^1 3th\,(3th-h)(3th-2h)\,dt = \frac{1}{2h^2}\int_0^1 3th(9t^2h^2 - 6th^2 - 3th^2 + 2h^2$$

$$= \frac{1}{2h^2}\int_0^1 (27t^3h^3 - 18t^2h^3 - 9t^2h^3 + 6th^3)\,dt$$

$$= \frac{1}{2h^2}\left[\frac{27h^3}{4} - \frac{\cancel{18}^{6}h^3}{\cancel{3}} - \frac{\cancel{9}^{3}h^3}{\cancel{3}} + \frac{\cancel{6}^{3}h^3}{\cancel{2}}\right] = \frac{1}{2h^2}\left(\frac{27h^3 - 24h^3 - 12h^3 + 12h^3}{4}\right)$$

$$= \frac{1}{2h^2}\left(\frac{3h^3}{4}\right) = \boxed{\frac{3h}{8}}$$

Now put all $w_0 - w_3$ together with our points $f(x_0) - f(x_3)$ our equation becomes

$$I(P_3) \approx \left(f(x_0)\frac{3}{8}h + f(x_1)\frac{9h}{8} + f(x_2)\frac{9h}{8} + f(x_3)\frac{3h}{8}\right)$$

$$= \frac{3}{8}h\left(f(x_0) + 3f(x_0+h) + 3f(x_0+2h) + f(x_0+3h)\right) ✓$$

$$\underline{w_0} = \int_{x_0}^{x_3 = x_0+3h} L_0\,dx = \int_{x_0}^{x_3} \frac{(x-x_0-h)(x-x_0-2h)(x-x_0-3h)}{-6h^3}\,dx$$

Let $x = x_0 + 3th \rightarrow x - x_0 = 3th$: $t \in [0,1]$

$dx = 3h\,dt$

$$w_0 = 3h\int_0^1 \frac{(3th-h)(3th-2h)(3th-3h)}{-6h^3}\,dt = 3h\int_0^1 \frac{(9t^2h^2 \overset{-9th^2}{\boxed{-6th^2-3th^2}} + 2h^2)(3th-3h)}{-6h^3}\,dt$$

$$= \frac{3h}{-6h^3}\int_0^1 (27t^3h^3 - 27t^2h^3 + 6th^3 - 27t^2h^3 + 27th^3 - 6h^3)\,dt$$

$$= \frac{-1}{2h^2}\int_0^1 (27t^3h^3 - 54t^2h^3 + 33th^3 - 6h^3)\,dt$$

$$= \frac{-1}{2h^2}\left[\frac{27t^4h^3}{4} - \frac{54t^3h^3}{3} + \frac{33t^2h^3}{2} - 6th^3\right]_0^1$$

$$= -\frac{1}{2h^2}\left(\frac{27h^3}{4} - \frac{54h^3}{3} + \frac{33h^3}{2} - 6h^3\right)$$

$$= -\frac{1}{2h^2}\left(\frac{81h^3}{12} - \frac{216h^3}{12} + \frac{198h^3}{12} - \frac{72h^3}{12}\right) = -\frac{1}{2h^2}\left(-\frac{\overset{3}{\cancel{9}}h^3}{\cancel{12}_4}\right) = \boxed{\frac{3h}{8}}$$

$$\underline{w_1} = \int_{x_0}^{x_3} L_1\,dx = \int_{x_0}^{x_3} \frac{(x-x_0)(x-x_0-2h)(x-x_0-3h)}{2h^3}\,dx$$

Let $x = x_0 + 3th \rightarrow x - x_0 = 3th$ $t \in [0,1]$

$dx = 3h\,dt$

$$w_1 = 3h\int_0^1 \frac{3th(3th-2h)(3th-3h)}{2h^3}\,dt = \frac{3}{2h^2}\int_0^1 3th(9t^2h^2 - 9th^2 - 6th^2 + 6h^2)\,dt$$

$$= \frac{3}{2h^2}\int_0^1 (27t^3h^3 - 27t^2h^3 - 18t^2h^3 + 18th^3)\,dt = \frac{3}{2h^2}\int_0^1 (27t^3h^3 - 45t^2h^3 + 18th^3)\,dt$$

$$= \frac{3}{2h^2}\left[\frac{27t^4h^3}{4} - \frac{45t^3h^3}{3} + \frac{18t^2h^3}{2}\right]_0^1 = \frac{3}{2h^2}\left(\frac{27h^3}{4} - 15h^3 + 9h^3\right)$$

$$= \frac{3}{2h^2}\left(\frac{27h^3}{4} - \frac{24h^3}{4}\right) = \frac{3}{2h^2}\left(\frac{3h^3}{4}\right) = \boxed{\frac{9h}{8}}$$

4. consider the following integral:

$$I = \int_2^4 \frac{x}{\sqrt{x^2-1}}\,dx$$

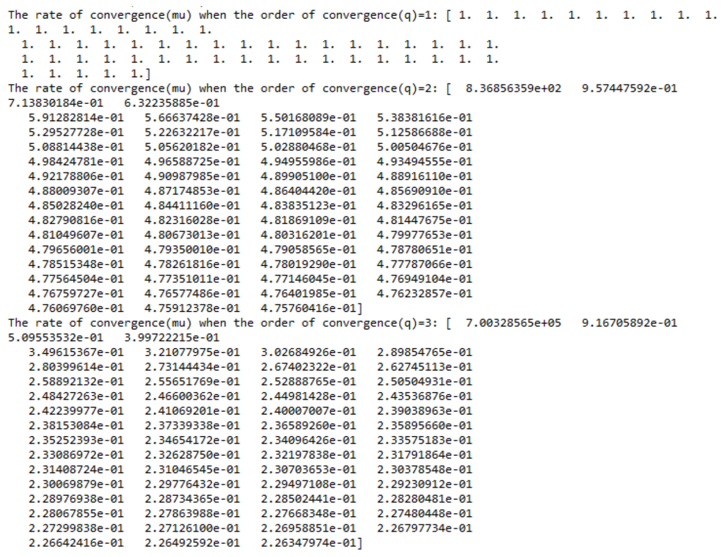
$$\begin{cases} u = x^2-1 \\ du = 2x\,dx \rightarrow x\,dx = \frac{1}{2}\,du \end{cases} \rightarrow I = \int_3^{15} \frac{1}{2\sqrt{u}}\,du = \frac{1}{2}\int_3^{15} u^{-1/2}\,du$$

Apply power rule: $\int x^a dx = \frac{x^{a+1}}{a+1}$, $a \neq -1$

$$= \frac{\cancel{1}}{\cancel{2}}\left[\frac{u^{1/2}}{\cancel{1/2}}\right]_3^{15} = \sqrt{15}-\sqrt{3} = \sqrt{3}(\sqrt{5}-1)$$

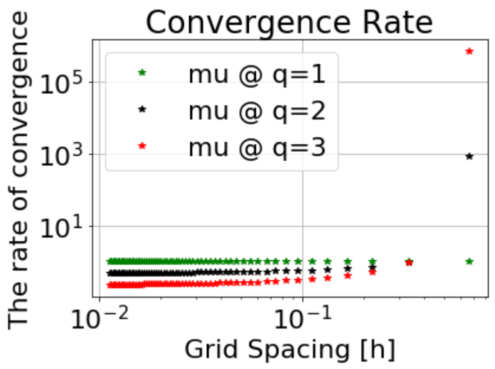
4.(c) The rate of convergence differentiated via different order of convergences

Data table



```
The rate of convergence(mu) when the order of convergence(q)=1: [ 1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.]
The rate of convergence(mu) when the order of convergence(q)=2: [  8.36856359e+02   9.57447592e-01
7.13830184e-01   6.32235885e-01
   5.91282814e-01   5.66637428e-01   5.50168089e-01   5.38381616e-01
   5.29527728e-01   5.22632217e-01   5.17109584e-01   5.12586688e-01
   5.08814438e-01   5.05620182e-01   5.02880468e-01   5.00504676e-01
   4.98424781e-01   4.96588725e-01   4.94955986e-01   4.93494555e-01
   4.92178806e-01   4.90987985e-01   4.89905100e-01   4.88916110e-01
   4.88009307e-01   4.87174853e-01   4.86404420e-01   4.85690910e-01
   4.85028240e-01   4.84411160e-01   4.83835123e-01   4.83296165e-01
   4.82790816e-01   4.82316028e-01   4.81869109e-01   4.81447675e-01
   4.81049607e-01   4.80673013e-01   4.80316201e-01   4.79977653e-01
   4.79656001e-01   4.79350010e-01   4.79058565e-01   4.78780651e-01
   4.78515348e-01   4.78261816e-01   4.78019290e-01   4.77787066e-01
   4.77564504e-01   4.77351011e-01   4.77146045e-01   4.76949104e-01
   4.76759727e-01   4.76577486e-01   4.76401985e-01   4.76232857e-01
   4.76069760e-01   4.75912378e-01   4.75760416e-01]
The rate of convergence(mu) when the order of convergence(q)=3: [  7.00328565e+05   9.16705892e-01
5.09553532e-01   3.99722215e-01
   3.49615367e-01   3.21077975e-01   3.02684926e-01   2.89854765e-01
   2.80399614e-01   2.73144434e-01   2.67402322e-01   2.62745113e-01
   2.58892132e-01   2.55651769e-01   2.52888765e-01   2.50504931e-01
   2.48427263e-01   2.46600362e-01   2.44981428e-01   2.43536876e-01
   2.42239977e-01   2.41069201e-01   2.40007007e-01   2.39038963e-01
   2.38153084e-01   2.37339338e-01   2.36589260e-01   2.35895660e-01
   2.35252393e-01   2.34654172e-01   2.34096426e-01   2.33575183e-01
   2.33086972e-01   2.32628750e-01   2.32197838e-01   2.31791864e-01
   2.31408724e-01   2.31046545e-01   2.30703653e-01   2.30378548e-01
   2.30069879e-01   2.29776432e-01   2.29497108e-01   2.29230912e-01
   2.28976938e-01   2.28734365e-01   2.28502441e-01   2.28280481e-01
   2.28067855e-01   2.27863988e-01   2.27668348e-01   2.27480448e-01
   2.27299838e-01   2.27126100e-01   2.26958851e-01   2.26797734e-01
   2.26642416e-01   2.26492592e-01   2.26347974e-01]
```

Plot



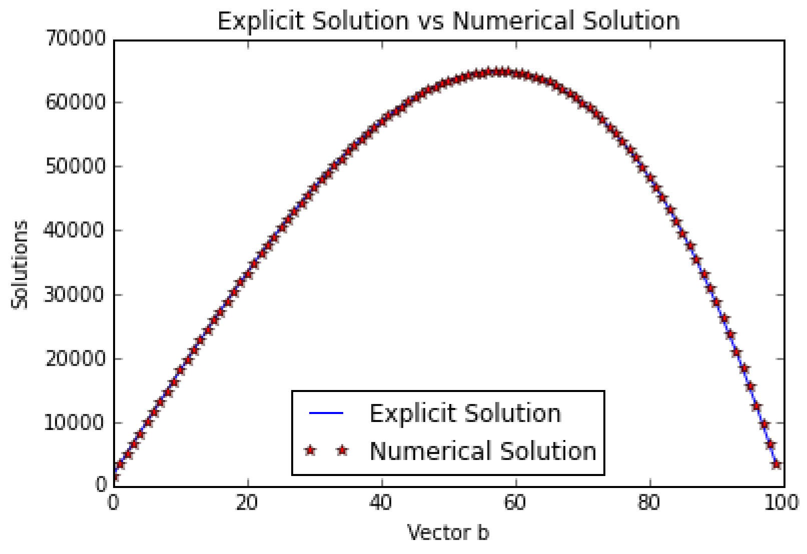
Comment: With the first order of convergence (q=1) the rate of convergence (mu) was not easily recognizable on the plot shown above, so I used the 2nd and 3rd orders to increase its magnitude. Also, I converted linlin plot to loglog for a better representation of this set up. As shown in the plot, we see that the more spacing reduces the real error quantity (mu) which is denoted as shown:

$$\lim_{k\to\infty} \frac{||\hat{x}^{(k+1)} - \vec{x}||_p}{||\hat{x}^{(k)} - \vec{x}||_p^q} = \mu$$

5.
(a) Done in Python; please refer to my python scrip attached on Github.
(b), (c), (d)

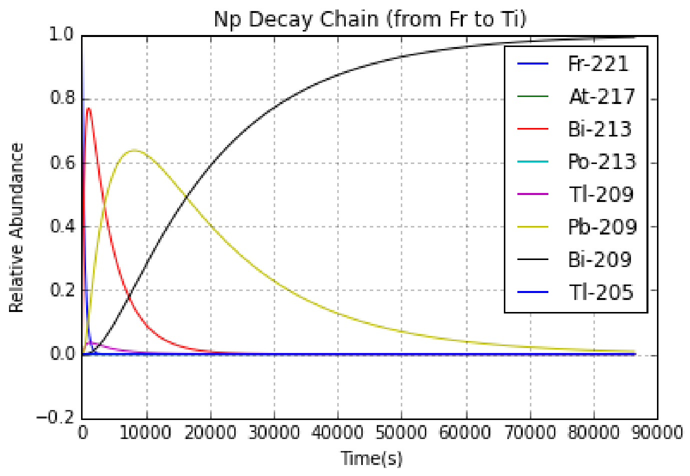
```
The condition number of A is: 4133.6429268
Display x matrix (explicit solution):
[  1650.   3300.   4949.   6596.   8240.   9880.  11515.  13144.  14766.
  16380.  17985.  19580.  21164.  22736.  24295.  25840.  27370.  28884.
  30381.  31860.  33320.  34760.  36179.  37576.  38950.  40300.  41625.
  42924.  44196.  45440.  46655.  47840.  48994.  50116.  51205.  52260.
  53280.  54264.  55211.  56120.  56990.  57820.  58609.  59356.  60060.
  60720.  61335.  61904.  62426.  62900.  63325.  63700.  64024.  64296.
  64515.  64680.  64790.  64844.  64841.  64780.  64660.  64480.  64239.
  63936.  63570.  63140.  62645.  62084.  61456.  60760.  59995.  59160.
  58254.  57276.  56225.  55100.  53900.  52624.  51271.  49840.  48330.
  46740.  45069.  43316.  41480.  39560.  37555.  35464.  33286.  31020.
  28665.  26220.  23684.  21056.  18335.  15520.  12610.   9604.   6501.
   3300.]

Display Scipy version x matrix (Numerical Solution):
[  1650.   3300.   4949.   6596.   8240.   9880.  11515.  13144.  14766.
  16380.  17985.  19580.  21164.  22736.  24295.  25840.  27370.  28884.
  30381.  31860.  33320.  34760.  36179.  37576.  38950.  40300.  41625.
  42924.  44196.  45440.  46655.  47840.  48994.  50116.  51205.  52260.
  53280.  54264.  55211.  56120.  56990.  57820.  58609.  59356.  60060.
  60720.  61335.  61904.  62426.  62900.  63325.  63700.  64024.  64296.
  64515.  64680.  64790.  64844.  64841.  64780.  64660.  64480.  64239.
  63936.  63570.  63140.  62645.  62084.  61456.  60760.  59995.  59160.
  58254.  57276.  56225.  55100.  53900.  52624.  51271.  49840.  48330.
  46740.  45069.  43316.  41480.  39560.  37555.  35464.  33286.  31020.
  28665.  26220.  23684.  21056.  18335.  15520.  12610.   9604.   6501.
   3300.]
```

Plot for (c), (d)

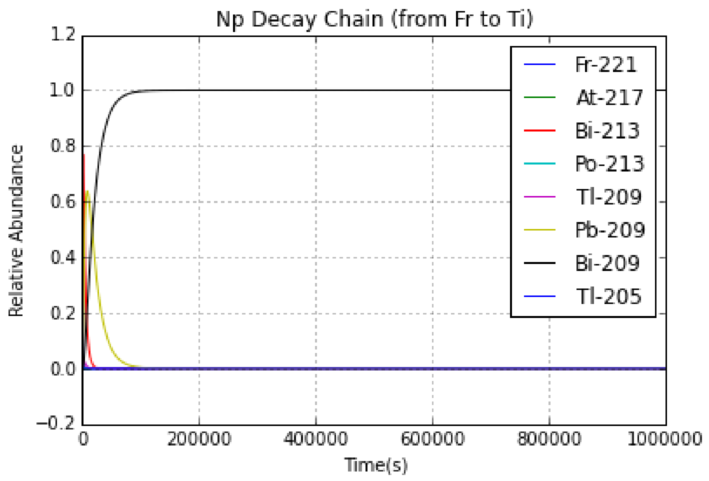


6. Plots

Time scale:1sec to 24hrs(1day)



Time scale:1sec to 10E5sec(appx. 11days)



Comment on what happens if you start higher in the decay series. What about if you carry the decay to longer times?

In this case, Fr-221 was the highest isotope that was started from the chain and it appears to have the shortest half-life amongst all others. On the other hand Bi-209, the second from the very last isotope in this chain, tends to have the longest half-life. This makes sense! Those isotopes in lower in the decay series are more stable, thus they get longer life span!

If you carry the decay far enough in time, you start to see oscillations in the population. Comment on the cause of these oscillations.

I did not include the plot in this document, but if you play with my python script, manipulating the time variable, I could see that some random picks appeared in the negative region at some point in time, interestingly enough when I run the script multiple times, with the same setting, I get different plotting each time. I think, it’s caused due to a systematic error.