

CP 1 - Modelling Radioactive Decay  
NPRE 247

Nalin Gadihoke

02/27/2018

# Contents

<b>1</b>	<b>Background</b>	<b>1</b>
1.1	Single element decay . . . . .	1
1.2	Chain decay . . . . .	1
<b>2</b>	<b>The assignment parameters</b>	<b>2</b>
<b>3</b>	<b>Theory</b>	<b>2</b>
3.1	Analytical Solutions . . . . .	2
3.2	Numerical Solutions . . . . .	3
<b>4</b>	<b>Modeling</b>	<b>5</b>
4.1	Analytical Solutions . . . . .	5
4.2	Numerical solutions . . . . .	5
4.2.1	. . . . .	6
4.2.2	. . . . .	7
4.2.3	. . . . .	7
4.3	Reading and writing to files . . . . .	8
<b>5</b>	<b>Summary/Conclusion of results</b>	<b>8</b>
<b>6</b>	<b>References/Sources of information</b>	<b>9</b>

Project was written in Latex and programming was done in Python.

## 1 Background

### 1.1 Single element decay

Radioactive decay is the process by which an unstable atomic nucleus emits radiation such as an alpha particle, beta particle or gamma particle with a neutrino. This process is a good example of exponential decay. Considering the case of  $N_A$  decaying to  $N_B$ .  $N_A \rightarrow N_B$ . In this case decay rate is proportional to number of atoms present  $N_A$ .

$$\frac{-dN_A}{dt} \propto N_A$$

Adding, a constant of proportionality or the decay constant, which is unique to every element, we get a general differential equation

$$\frac{dN_A}{dt} = -\lambda_A N_A \quad (1)$$

### 1.2 Chain decay

In the case of element  $N_A$  decaying to  $N_B$  decaying to  $N_C$  (stable),  $N_A \rightarrow N_B \rightarrow N_C$ , the first decay is represented by equation (1) however we need a new differential equation for the second decay of  $N_B \rightarrow N_C$ . Since the number of  $N_B$  increases as a result of activity from  $N_A$  and decreases due to its own decay into  $N_C$ , we can form a differential equation as follows

$$\frac{dN_B}{dt} = -\lambda_B N_B + \lambda_A N_A \quad (2)$$

The creation rate of the stable  $N_C$  atoms is simply the activity of  $N_B$ , considering its independent decay

$$\frac{dN_C}{dt} = \lambda_B N_B \quad (3)$$

These equations will be later solved in section 3.

## 2 The assignment parameters

The assignment asks us to consider a three component decay chain,  $N_A \rightarrow N_B \rightarrow N_C$ , and find solutions to the respective decay equations numerically and analytically. Further, we are to graph the results as specified by the assignment instructions. The given parameters of the assignment are displayed in Table 1.

Table 1: Initial Values

Parameter	Value	Description
$t_{half_A}$	1.1 hours	time for half of $N_A$ atoms to decay
$t_{half_B}$	9.2 hours	time for half of $N_B$ atoms to decay
$N_A(0)$	100	number of atoms of $N_A$ at $t = 0$
$N_B(0)$	0	number of atoms of $N_B$ at $t = 0$
$N_C(0)$	0	number of atoms of $N_C$ at $t = 0$
$t_{final}$	50 hours	time for full simulation

## 3 Theory

The theory section goes over the theoretical aspect of what was programmed.

### 3.1 Analytical Solutions

This approach included solving equations (1), (2) and (3) by integrating both sides to find equations for  $N_A(t)$ ,  $N_B(t)$  and  $N_C(t)$ .

$$N_A(t) = N_A(0)e^{-\lambda_A t} \quad (4)$$

Now, we can calculate the half life of  $N_A$  by setting  $N_A(t) = N(0)/2$  which results in

$$\frac{1}{2} = e^{-\lambda_A t}$$

$$t_{half_A} = \frac{\ln 2}{\lambda_A}$$

or, the decay constant can be represented as

$$\lambda_A = \frac{\ln 2}{t_{half_A}} \quad (5)$$

This formula is later used to calculate  $\lambda_A$ ,  $\lambda_B$  and  $\lambda_C$

Similarly,

solving (2) we get

$$N_B(t) = \frac{N_A(0)\lambda_A}{\lambda_B - \lambda_A}(e^{-\lambda_A t} - e^{-\lambda_B t}) \quad (6)$$

and solving (3) we get

$$N_C(t) = N_C(0) + \lambda_B N_B(0)(1 - e^{-\lambda_B t}) + \frac{N_A(0)}{\lambda_B - \lambda_A}(\lambda_B(1 - e^{-\lambda_A t}) - \lambda_A(1 - e^{-\lambda_B t})) \quad (7)$$

where values of  $N_A(0)$ ,  $N_B(0)$  and  $N_C(0)$  are taken from Table 1.  $\lambda_A$  and  $\lambda_B$  are calculated using (5) for which the values of  $t_{\text{half}}$  are taken from the same table

To find the equation for  $t_{\text{max}}$ , the time for  $N_B(t)$  to reach maximum value can be found by setting the value of  $\frac{dN_B}{dt} = 0$ . This means

$$\lambda_A e^{-\lambda_A t_{\text{max}}} = \lambda_B e^{-\lambda_B t_{\text{max}}}$$

$$t_{\text{max}} = \frac{\ln\left(\frac{\lambda_B}{\lambda_A}\right)}{\lambda_B - \lambda_A}$$

Using, values from Table 1, we calculate  $t_{\text{max}}$  for maximum  $N_B$  to be 3.828 seconds. To verify this,  $\frac{d^2 N_B}{dt^2}$  at  $t = t_{\text{max}}$  was calculated and was found to be  $< 0$ , proving  $t_{\text{max}}$  is the time for maximum  $N_B(t)$

### 3.2 Numerical Solutions

Here, the finite difference approach was used in computing the values for  $N_A(t + \delta t)$ ,  $N_B(t + \delta t)$  and  $N_C(t + \delta t)$  which are essentially the values of  $N_A(t)$ ,  $N_B(t)$  and  $N_C(t)$  for every time step given an initial  $N_A(0)$ ,  $N_B(0)$ ,  $N_C(0)$  and  $\delta t$ . By definition, a derivative is

$$\frac{df}{dt} = \lim_{\delta t \rightarrow 0} \frac{f(x + \delta t) - f(x)}{\delta t}$$

Applying this to (1), (2) and (3) we get

$$\frac{N_A(t + \delta t) - N_A(t)}{\delta t} = -\lambda_A N_A(t)$$

$$\frac{N_B(t + \delta t) - N_B(t)}{\delta t} = -\lambda_B N_B(t) + \lambda_A N_A(t)$$

$$\frac{N_C(t + \delta t) - N_C(t)}{\delta t} = \lambda_B N_B(t)$$

re-arranging,

$$N_A(t + \delta t) = -\lambda_A N_A(t) \delta t + N_A(t) \tag{8}$$

$$N_B(t + \delta t) = (-\lambda_B N_B(t) + \lambda_A N_A(t)) \delta t + N_B(t) \tag{9}$$

$$N_C(t + \delta t) = \lambda_C N_C(t) \delta t + N_C(t) \tag{10}$$

It is to be noted that three different values of  $\delta t$  were considered, 1 hour, 0.5 hours and 0.25 hours, and their impacts on the model will be discussed in the next section, along with modeling details and information on data points.

## 4 Modeling

Now, the programming and plotting part of the project is explained.

### 4.1 Analytical Solutions

The function *odeint* from the *integrate* library in python was used to solve the equations (1), (2) and (3) over 1500 time stamps between  $t = 0$  to  $t_{\text{final}}$ . The values for  $N_A(t)$ ,  $N_B(t)$  and  $N_C(t)$  were stored as a list containing three elements which were the corresponding arrays.

```
def model_b(N_0_list , t):
    na = N_0_list[0]
    nb = N_0_list[1]
    nc = N_0_list[2]
    dnadt = -lambda_a*na
    dnbdt = (-lambda_b*nb)+(lambda_a*na)
    dncdt = lambda_b*nb
    return [dnadt , dnbdt , dncdt]
n = odeint(model_b , N_0_list , t)
```

Array Nb was later used in figure 1 for plotting and the time for the maximum value of  $N_B$  in the array Nb was later used for plotting in figure 3. Detailed code can be found in part 1 of CP\_1.py, the code file.

### 4.2 Numerical solutions

This part utilized equations (8), (9) and (10). These were then programmed as a function *numerical* which accepts lists of  $N(0)$  and  $\lambda$  values and  $\delta t$  and outputs a list whose elements are the arrays  $N_A(t)$ ,  $N_B(t)$ ,  $N_C(t)$  and  $t$  (which contains the time values used for plotting). The core equations were modeled as follows between  $t = 0$  to  $t_{\text{final}}$  with  $\frac{t_{\text{final}}}{\delta t}$  time steps.

```
na[i] = (- lambda_list[0] * na[i-1] * t_del) + na[i-1]

nb[i] = ((-lambda_list[1]*nb[i-1]) + (lambda_list[0]*na[i-1]))*t_del
        + nb[i-1]

nc[i] = (lambda_list[1]*nb[i-1]*t_del) + nc[i-1]
```

#### 4.2.1

Values of  $N_B(t)$  were plotted against  $t$  in hours for three different values of  $\delta t$  - 1 hour (coarse), 0.5 hours (medium) and 0.25 hours (fine). The analytical solution array Nb was plotted too. The arrays were produced using the function described previously. Detailed code can be found in part 2 of CP\_1.py, the code file.  $t = 0$  to  $t_{\text{final}}$

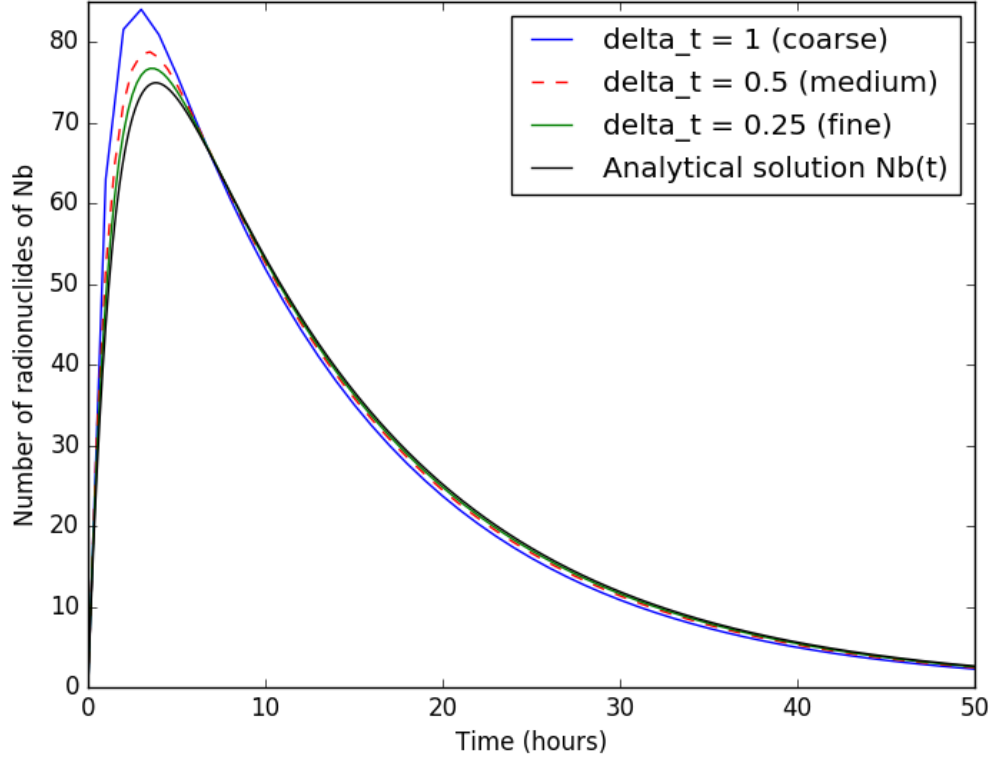


Fig. 1:  $N_B(t)$  vs Time, for different values of  $\delta t$



### 4.2.2

Numerical values of  $N_A(t)$ ,  $N_B(t)$ ,  $N_C(t)$  and  $N_{total}(t)$  were plotted against  $t$ . Here  $\delta t$  was considered to be 0.125 hours as it resulted the smoothest curves.  $t = 0$  to  $t_{final}$

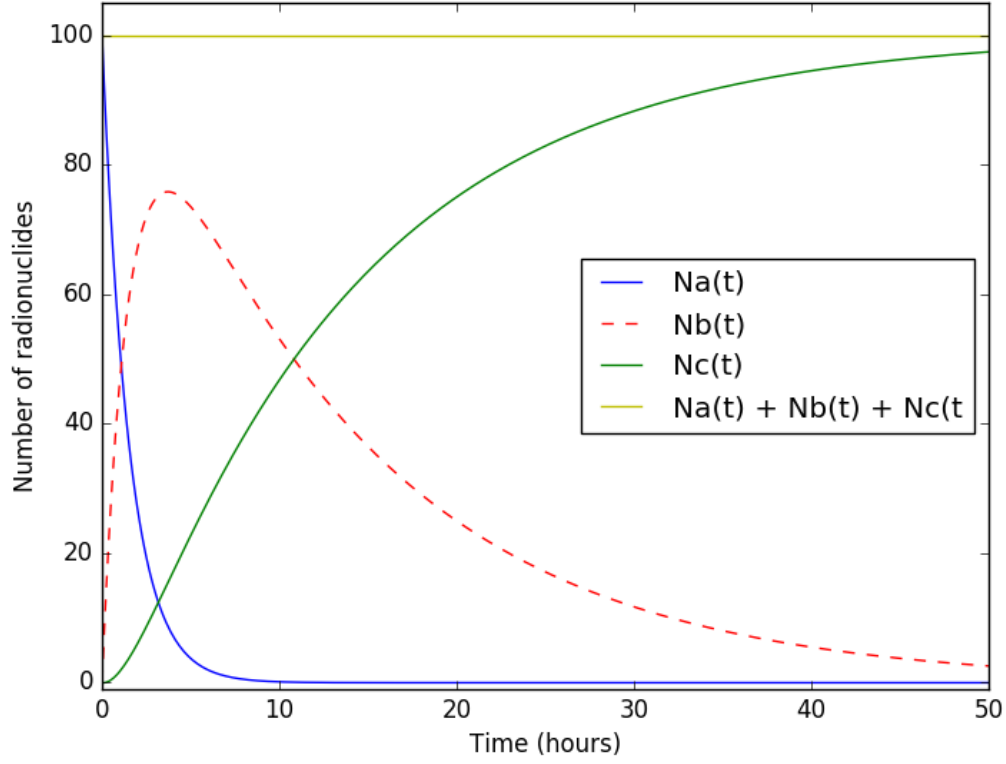


Fig. 2:  $N_A(t)$ ,  $N_B(t)$ ,  $N_C(t)$  and  $N_{total}$  vs Time

### 4.2.3

Using the *numerical* function iteratively, values of time to reach maximum  $N_B(t)$  were plotted against different values of  $1/\delta t$ .  $1/\delta t$  was plotted to have values between 1 to 310 in 310 time steps which means  $\delta t$  is between 1 and 0.0032258. The time of maximum  $N_B(t)$  using the analytical solution was also plotted.

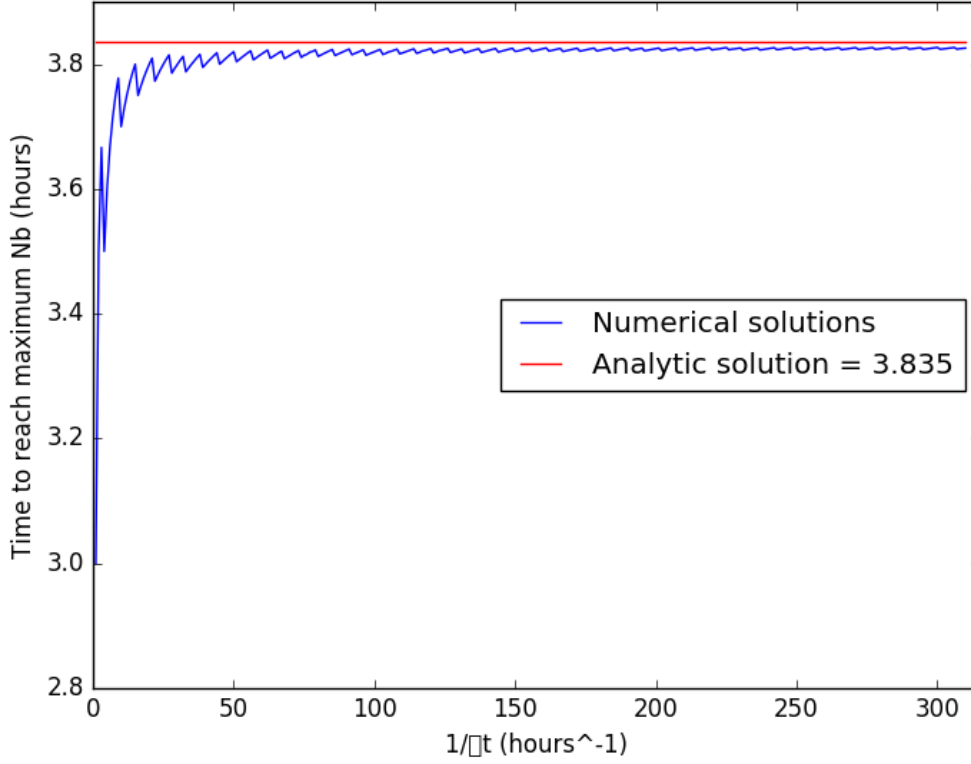


Fig. 3: Time to reach maximum  $N_B$  vs  $\frac{1}{\delta t}$

### 4.3 Reading and writing to files

The parameters were read into the python script from a .csv file. The results and necessary data needed to reproduce the results, in other words to reproduce the three figures were written into an excel file using the *writer* function. The columns of data have self-descriptive headings and the input parameters are included in the output file too.

## 5 Summary/Conclusion of results

1. From figure 1 we are able to deduce that for smaller and smaller values of  $\delta t$ , the numerical solution for  $N_B(t)$  approaches the analytical solution of  $N_B(t)$ . Furthermore, larger values of  $\delta t$  result in less smooth graph curves as compared to smaller values of  $\delta t$ .

2. Figure 2 demonstrates how  $N_A(t)$ ,  $N_B(t)$ ,  $N_C(t)$  and  $N_{total}(t)$  decay over 50 hours. It also shows how  $N_A$  decays considerably faster than  $N_B$ .
3. Figure 3 plots the analytical solution of time of maximum  $N_B(t)$ , which was found to be  $3.835seconds$ . We can clearly see how the numerical solution for the same approaches the analytical solutions for smaller and smaller values of  $\delta t$ , i.e. larger values of  $1/\delta t$

## 6 References/Sources of information

1. [www.sharelatex.com](http://www.sharelatex.com)
2. [www.stackoverflow.com](http://www.stackoverflow.com)
3. [www.wikipedia.com](http://www.wikipedia.com)
4. Mustafa Bakaç, Aslıhan Kartal Taşoğlu, Gizem Uyumaz, "Modeling radioactive decay", Procedia - Social and Behavioral Sciences, Volume 15, 2011.