Generations of Implanted Nuclear Atrophy (GINA) - Original Calculations

Preamble

This document is to go along with the python code it was written for. For this reason, relevant the code will be attached in code blocks. It is not updated to include references to the GINA program which is the C++ implementation.

Introduction

The rate of decay of a material is given by

$$\frac{dN(t)}{dt} = -\lambda N(t) = -\frac{N(t)}{\tau},\tag{1}$$

where N is the number of radioactive nuclei, λ is the decay constant and τ is the mean lifetime of the particles within the material. The decay relates to the half life $T_{1/2}$ by

$$\lambda T_{1/2} = \ln(2) \implies T_{1/2} = \tau \ln(2).\tag{2}$$

From Eq. 1, we can say

$$\frac{dN(t')}{N(t')} = -\frac{1}{\tau}dt' \implies \int_{N(t)}^{N(t+\Delta t)} \frac{dN(t')}{N(t')} = -\int_{t}^{t+\Delta t} \frac{1}{\tau}dt' \implies \ln\left[\frac{N(t+\Delta t)}{N(t)}\right] = -\frac{\Delta t}{\tau}.\tag{3}$$

We can re-write Eq. 3 in exponential form which gives

$$N(t+\Delta t) = N(t)e^{-\frac{\Delta t}{\tau}} = N(t)e^{-\frac{\Delta t}{T_{1/2}}\ln(2)} \implies N(\Delta t) = N(0)e^{-\frac{\Delta t}{T_{1/2}}\ln(2)}.\tag{4}$$

Analysis - Decays Over Time

RADIOACTIVE PARTICLE NUMBERS

Let the superscripts m, d and gd denote a mother, daughter and granddaughter (the nuclei the daughter nuclei decays into) nuclei respectively and T represent the total time that has passed. Consider a steady beam that is depositing radioactive particles onto a material at a constant rate of $R(t)$ (particles/second). Initially assume there are no radioactive particles on the material. That is

$$N_0^m = 0 \qquad\qquad N_0^d = 0 \qquad\qquad N_0^{gd} = 0\tag{5}$$

Over some small amount of time[1] Δt $(T = \Delta t)$, the total particles placed on the material will be

$$N_1^m = R(t)\Delta t.\tag{6}$$

None of the radioactive particles will have decayed yet so $N_1^d = 0$ and $N_1^{gd} = 0$. The mother nuclei may decay into the daughter nuclei at this point but the daughter nuclei will not have yet had a chance to decay and thus $N_2^{gd} = 0$.

---

[1]For sufficiently small Δt or $R(t)$, we can neglect the decays that happen between Δt time increments.

```
#[PYTHON CODE]#
#Sets the initial conditions.
N_m[0]=0
N_m[1]=R*del_t
N_d[0]=0
N_d[1]=0
N_gd[0]=0
N_gd[1]=0
```

After another small amount of time Δt $(T = 2\Delta t)$, the total radioactive particles on the material will increase by $R(t)\Delta t$ and decrease by some amount

$$N_2^d = D_1^m = N_1^m \left(1 - e^{-\frac{\Delta t}{T_{1/2}^m}\ln(2)}\right), \tag{7}$$

as the radioactive particles decay into the daughter nuclei. Thus, the total radioactive mother nuclei after $T = 2\Delta t$ is

$$N_2^m = 2N_1^m - N_1^m \left(1 - e^{-\frac{\Delta t}{T_{1/2}^m}\ln(2)}\right). \tag{8}$$

After yet another small amount of time Δt $(T = 3\Delta t)$, the total radioactive particles on the material will increase by $N_1^m = R(t)\Delta t$ and decrease by some amount

$$D_2^m = N_2^m \left(1 - e^{-\frac{\Delta t}{T_{1/2}^m}\ln(2)}\right), \tag{9}$$

as the radioactive particles decay into the daughter nuclei. At this point, the total number of mother nuclei will be

$$N_3^m = N_2^m + N_1^m - N_2^m \left(1 - e^{-\frac{\Delta t}{T_{1/2}^m}\ln(2)}\right) \tag{10}$$

$$= N_1^m + N_2^m e^{-\frac{\Delta t}{T_{1/2}^m}\ln(2)}. \tag{11}$$

At this point, the number of radioactive daughter nuclei will increase by ΔN_3^d and decrease by some amount based on the decay time. Thus at $T = 3\Delta t$ we will have

$$N_3^d = N_2^d + \Delta N_3^d - N_2^d \left(1 - e^{-\frac{\Delta t}{T_{1/2}^d}\ln(2)}\right) \tag{12}$$

$$= N_2^d + N_2^m \left(1 - e^{-\frac{\Delta t}{T_{1/2}^m}\ln(2)}\right) - N_2^d \left(1 - e^{-\frac{\Delta t}{T_{1/2}^d}\ln(2)}\right) \tag{13}$$

$$= N_2^m \left(1 - e^{-\frac{\Delta t}{T_{1/2}^m}\ln(2)}\right) + N_2^d e^{-\frac{\Delta t}{T_{1/2}^d}\ln(2)}. \tag{14}$$

At this stage, there will also be daughter nuclei that decay and so

$$N_3^{gd} = N_2^d \left(1 - e^{-\frac{\Delta t}{T_{1/2}^d}\ln(2)}\right). \tag{15}$$

For sufficiently large $T_{1/2}^{gd}$, we can ignore any further decays and still maintain a good approximation for our system. At this point, any possible increase/decrease in particles and all possible decays

are accounted for. Since we can represent the particle numbers at $T = 3\Delta t$ using terms for the previous time steps, we can generalize this to some $T = n\Delta t$ for $n \in \mathbb{N}$ giving

$$N_n^m = N_1^m + N_{n-1}^m e^{-\frac{\Delta t}{T_{1/2}^m} \ln(2)} \tag{16}$$

$$N_n^d = N_{n-1}^m \left(1 - e^{-\frac{\Delta t}{T_{1/2}^m} \ln(2)}\right) + N_{n-1}^d e^{-\frac{\Delta t}{T_{1/2}^d} \ln(2)} \tag{17}$$

$$N_n^{gd} = N_{n-1}^{gd} + N_{n-1}^d \left(1 - e^{-\frac{\Delta t}{T_{1/2}^d} \ln(2)}\right). \tag{18}$$

These relations can be proven by trivial induction.

```
#[PYTHON CODE]#
#Calculation of particle number
for n in range(2,size):
    N_m[n]=N_m[1]+N_m[n-1]*np.exp(-del_t*np.log(2)/HL_m)
    N_d[n]=N_m[n-1]*(1-np.exp(-del_t*np.log(2)/HL_m))+N_d[n-1]*np.exp(-del_t*np.log
        (2)/HL_d)
    N_gd[n]=N_gd[n-1]+N_d[n-1]*(1-np.exp(-del_t*np.log(2)/HL_d))
```

Addition. Instead of using m, g and gd for mother, daughter and granddaughter respectively, suppose we allow the superscript to denote the generation as a number, where $(0) \equiv m$, $(1) \equiv d$ and $(2) \equiv gd$ and so on such that (i) would represent the i-th generation. Using this scheme, this process can be generalized for i number of generations as

$$N_n^{(0)} = N_1^{(0)} + N_{n-1}^{(0)} e^{-\frac{\Delta t}{T_{1/2}^{(0)}} \ln(2)} \tag{19}$$

$$N_n^{(i)} = N_{n-1}^{(i-1)} \left(1 - e^{-\frac{\Delta t}{T_{1/2}^{(i-1)}} \ln(2)}\right) + N_{n-1}^{(i)} e^{-\frac{\Delta t}{T_{1/2}^{(i)}} \ln(2)}. \tag{20}$$

If we assume the lats stage has a near infinite half life, the limiting case will produce a final step to be equivalent to that of Eq. 18.

Important Relations

DECAY RATES - METHOD I

The values N_n^m, N_n^d and N_n^{gd} all represent a total number of particles of a specific nuclei. If we are interested in the decays D of the radioactive particles, we can represent them based on the number of particles are present. That is

$$D_n^m = N_n^m \left(1 - e^{-\frac{\Delta t}{T_{1/2}^m} \ln(2)}\right) \tag{21}$$

$$D_n^d = N_n^d \left(1 - e^{-\frac{\Delta t}{T_{1/2}^d} \ln(2)}\right). \tag{22}$$

These represent the today decays per unit Δt. The contamination is then defined as the number of decays the daughter is producing with respect to the number of decays the mother nuclei is producing.

$$C_n = \frac{D_n^d}{D_n^m} \times 100\% = \text{contamination } \%. \tag{23}$$

```
#[PYTHON CODE]#
#Calculation of decay rate (devided by del_t to make the units per second)
for n in range(0,size):
    D_m[n]=N_m[n]*(1-np.exp(-del_t*np.log(2)/HL_m))/del_t
    D_d[n]=N_d[n]*(1-np.exp(-del_t*np.log(2)/HL_d))/del_t
    if D_m[n] > 0:
        C[n]=D_d[n]/D_m[n]*100
    t[n]=n*del_t
```

DECAY RATES - METHOD II

An alternate, and nicer method of calculating the decay rate can be found from Eq. 1. By definition this is the decay rate, so using the total particle numbers we calculate in Eq. 16 - 18, we can determined the decay rate (note that we are taking the absolute value because we are interested in the decays per second) as

$$D_n^m = \frac{dN_n^m}{dt} = \frac{N_n^m \ln(2)}{T_{1/2}^m} \tag{24}$$

$$D_n^d = \frac{dN_n^d}{dt} = \frac{N_n^d \ln(2)}{T_{1/2}^d}. \tag{25}$$

These represent the today decays per second. Using this method, the contamination will still be calculated in the same way using Eq. 23.

```
#[PYTHON CODE]#
#Alternate calculation of decay rate
for n in range(0,size):
    D2_m[n]=N_m[n]*np.log(2)/HL_m
    D2_d[n]=N_d[n]*np.log(2)/HL_d
    if D2_m[n] > 0:
        C2[n]=D2_d[n]/D2_m[n]*100
```

Addition. Begin with the notation from the previous 'Addition'. The decay rate for any generation would simply be

$$D_n^{(i)} = \frac{dN_n^{(i)}}{dt} = \frac{N_n^{(i)} \ln(2)}{T_{1/2}^{(i)}}, \tag{26}$$

and the contamination of the original implant would be given by

$$C_n = \frac{\sum_{k=1}^{\infty} D_n^{(k)}}{D_n^{(0)}} \times 100\% = \text{contamination } \%. \tag{27}$$

BEAM CUTTOFF & PARTICLE LOSS

We will introduce a new variable T_c which will represent a cutoff of the beam that is implanting the radioactive particles. After this time, the system will reset and the original system will be

instantaneously[2] moved to a new location. So, when $T = iT_c$ for $i \in \mathbb{N}$, we will have our particle number values reset to zero, and thus

$$N_i^m = 0 \qquad\qquad N_i^d = 0 \qquad\qquad N_i^{gd} = 0. \tag{28}$$

This will of course occur at the n values such that $n = \frac{iT_c}{\Delta t}$. Let the total particles that have decayed be denoted by $\bar{N}$. The total decayed particles can be determined by

$$\bar{N}_n^m = N_n^d + N_n^{gd}. \tag{29}$$

The ratio of total particles implanted to the particles that have decayed is then simply

$$\frac{nN_1^m}{\bar{N}_n^m} = \frac{nN_1^m}{N_n^d + N_n^{gd}}. \tag{30}$$

The deposited implant loss B_n^ℓ (representing the total % of the implanted particles that have been decayed) is then the inverse

$$B_n^\ell = \frac{\bar{N}_n^m}{nN_1^m} \times 100\% = \frac{N_n^d + N_n^{gd}}{nN_1^m} \times 100\%. \tag{31}$$

The deposited implant health B^h is then the percentage of original implant that still exists and is given simply by

$$B_n^h = 100\% - B_n^\ell. \tag{32}$$

```python
#[PYTHON CODE]#
#Determines the implant loss and Health
for n in range(1,size):
    B_l[n] = 100*(N_d[n]+N_gd[n])/(n*N_m[1])
    B_h[n]=100-P_l[n]
```

Addition. Considering an arbitrary number of generations, the decayed particles would be given by

$$\bar{N}_n^{(i)} = \sum_{k=i+1}^{\infty} N_n^{(k)}, \tag{33}$$

to which it follows that the implant loss be given by

$$B_n^\ell = \frac{\bar{N}_n^{(0)}}{nN_1^{(0)}} \times 100\% = \frac{1}{nN_1^{(0)}} \sum_{k=1}^{\infty} N_n^{(k)} \times 100\%. \tag{34}$$

The implant health would still be given by

$$B_n^h = 100\% - B_n^\ell. \tag{35}$$

References

[1] Definition of Beta Decay: http://www2.lbl.gov/abc/wallchart/chapters/03/2.html

[2] Half life values: http://www.nndc.bnl.gov/chart/

---

[2]In reality this would not be instantaneous.