Radioactive Decay Graph

Kaylin Shanahan 2022

This python programme calculates the number of nucei in a given radioactive sample each day, over a twenty-day period, using an array operation, the results of which are displayed on a graph of number of nuclei. The given radioactive sample contains 10^6 unstable nuclei which have a half-life of 4.7 days

Radioactive decay can be described with the following equations:

The formula to calculate the number of radioactive nuclei remaining after time t is

$$N_t = N_0 e^{-\lambda t}$$

The formula to calculate the half-life of an isotope is

$$t_{half} = \frac{\ln(2)}{\lambda}$$

This formula can then be rearranged to calculate the decay constant

$$\lambda = -\frac{\ln(2)}{t_{half}}$$

Using these equations, we can calculate the decay constant and then graph the decay of the sample of 10^6 unstable nuclei with a half life of 4.7 days

- Given values of initial number of nuclei and half-life of the radioactive sample are inserted into the follwing equations:
  - Calculate the decay constant using the rearranged formula:

$$\lambda = -\frac{\ln(2)}{t_{half}}$$

- Calculate the the number of radioactive nuclei:

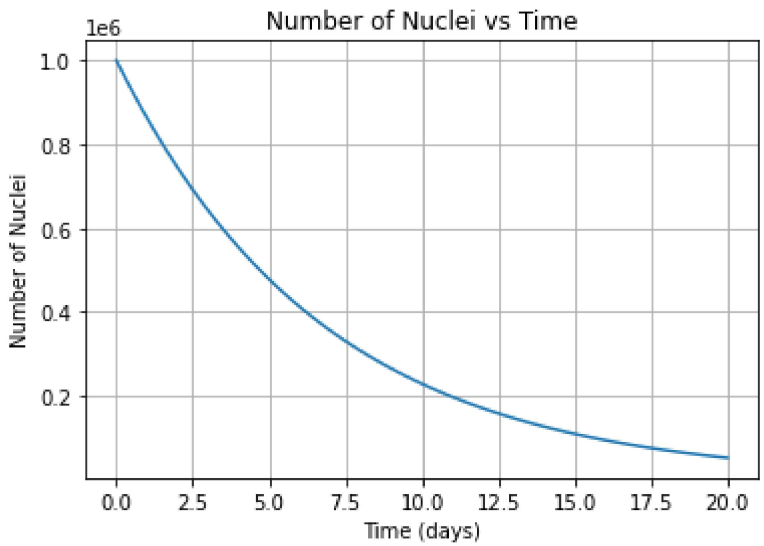
$$N_t = N_0 e^{-\lambda t}$$

- Output the decay constant per day
- Output a graph of number of nuclei as a function of time

In [47]:

```
# Numpy is needed for the natuaral log and exponential function
import numpy as np
# Matplotlib is needed to plat a graph of Number of Nuclei vs Time
import matplotlib.pyplot as plt
%matplotlib inline

# The initial amount of radioactive nuclei
N0 = 1000000
# The final amount of radioactive nuclei
Nf = 0
# The half-life of given radioactive sample
t_half = 4.7

# Calculate the decay constant
d = ((np.log (2)) / t_half)

# Output the decay constant
print("Decay Constant = {0:10.2e} per second".format(d))

# Define an array of x values, where x is time, t in days
x = np.linspace(0, 20)

# Calculate y values as a function of x, where y is number of nuclei, Nt
Nt = N0 * np.exp(-d*x)

# Plot the graph of number of nuclei as a function of time
plt.plot(x, Nt)
plt.title("Number of Nuclei vs Time")
plt.xlabel("Time (days)")
plt.ylabel("Number of Nuclei")
plt.grid()
plt.show()
```

```
Decay Constant =   1.47e-01 per second
```


Number of Nuclei vs Time
1e6
1.0
0.8
0.6
0.4
0.2
Number of Nuclei
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Time (days)