

## 57 Programmer en Python

### Programme à compléter

Cet exercice doit être fait après avoir fait l'exercice 56. L'objectif de cet exercice est de calculer et tracer, pas à pas, les nombres de noyaux radioactifs au cours du temps, dans un échantillon où un noyau radioactif se désintègre en un autre noyau, lui-même radioactif.

#### Fichiers Python

Programme à compléter  
Fiche d'accompagnement

[hatier-clic.fr/pct168](http://hatier-clic.fr/pct168)

#### Prérequis théoriques

- Loi de la décroissance radioactive

Le bismuth 210 vérifie la loi de décroissance radioactive. La désintégration du bismuth 210 crée le polonium 210, qui, en parallèle de sa création, se désintègre lui-même en plomb 206 stable.

Dans l'exercice 56 et à la question a de l'exercice 57, on exprime les variations du nombre de noyaux de chaque sorte pour une durée  $\Delta t$  donnée. Il s'agit ensuite, à la question b, de traduire cela en langage Python pour compléter le programme et obtenir les tracés.

## Programme à compléter

## Modules importés

Le module `math` fournit la fonction exponentielle.  
Le module `pylab` fournit les instructions graphiques utiles à l'exercice.

```

1  from math import *
2  from pylab import *
3
4  print("*****")
5  print("* Du bismuth 210 au plomb 206 *")
6  print("*   Exercice 57 p. 168   *")
7  print("*****Hatier 2020****")
8
9  ### Donnees
10 dt=0.1          # pas de temps en jours
11 tBi=5.01        # demi-vie du bismuth 210 en jours
12 tPo=138.4       # demi-vie du polonium 210 en jours
13 N0=1e6          # nombre de noyaux initial en jours
14 tmax=600        # duree maximale de calcul en jours
15 Ncalc=6000      # nombre de calculs
16
17 lBi=0.693/tBi   # calcul des constantes radioactives
18 lPo=0.693/tPo  # du bismuth 210 et du polonium 210
19
20 ### Initialisations
21 t=[0]
22 Nbi=[N0]
23 Npo=[0]
24 Npb=[0]
25 i=0
26
27 ### A MODIFIER : Remplissage des listes t, Nbi, Npo, Npb
28 while (i<Ncalc):
29     t.append(t[-1]+dt)
30     Nbi.append(...)
31     Npo.append(...)
32     Npb.append(...)
33     i=i+1
34
35 ### Graphique
36 plot(t,Nbi,label="Bismuth")
37 plot(t,Npo,label="Polonium")
38 plot(t,Npb,label="Plomb")
39 xlim(0,t[Ncalc])
40 ylim(0,N0)
41 xlabel("t (en j)")
42 ylabel("N")
43 legend()
44 show()
```

## Ne pas modifier : données et initialisations

L'énoncé n'amène pas nécessairement à modifier les données.

**Attention** Le séparateur décimal est le point, pas la virgule.

## À compléter : nombres de noyaux

Le nombre de noyaux de bismuth est connu grâce à la loi de décroissance radioactive. Les expressions des nombres de noyaux de polonium et de plomb sont à compléter à l'aide de la question **5c** de l'exercice **56** et de la question **a** de l'exercice **57**. La fonction `L.append(X)` ajoute `X` à la fin de la liste `L`.

## Ne pas modifier : fonctions graphiques

Les dernières lignes tracent le graphique.