

I. Problème: marche au hasard à une dimension

On modélise la marche aléatoire selon Ox d'une particule effectuant N pas aléatoires. A intervalles de temps régulier τ , la particule subit un choc à la suite duquel, aléatoirement, elle continue son chemin selon $+Ox$ ou $-Ox$. Entre deux chocs, la particule se déplace donc soit vers la droite soit vers la gauche d'une distance l (appelée libre parcours moyen).

1. Approche théorique:

D'un instant à l'autre, l'abscisse de la particule varie aléatoirement de $+l$ ou $-l$ avec une égale probabilité. On note $p(x, t)$ la probabilité qu'une particule se trouve en x à l'instant t .

1.a. Exprimer $p(x, t + \tau)$ en fonction de $p(x + l, t)$ et $p(x - l, t)$.

1.b. On donne le DL à l'ordre 2 en ϵ petit: $p(x + \epsilon, t) = p(x, t) + \epsilon \frac{\partial p}{\partial x}(x, t) + \frac{\epsilon^2}{2} \frac{\partial^2 p}{\partial x^2}(x, t)$.

Ecrire le DL à l'ordre 1 en τ de $p(x, t + \tau)$. Ecrire les DL à l'ordre 2 en l de $p(x + l, t)$ et de $p(x - l, t)$.

1.c. En déduire que $p(x, t)$ vérifie une équation de la forme $\frac{\partial p}{\partial t} = D \frac{\partial^2 p}{\partial x^2}$. Exprimer D en fonction de l et τ .

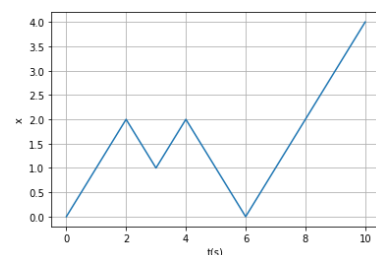
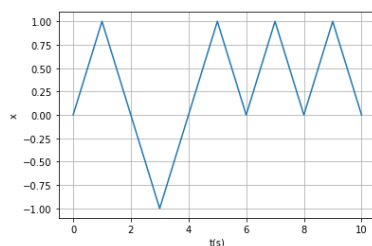
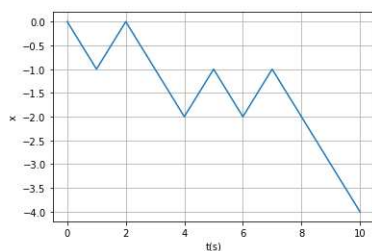
2. Simulation numérique de la marche au hasard:

A l'instant $t = 0$, la particule se trouve en $x = 0$. On cherche la position $x(t)$ de la particule aux instants $t_i = 0, \tau, 2\tau, \dots$. Avec le modèle étudié, les abscisses $x(t_i)$ valent $0, \pm l, \pm 2l, \pm 3l \dots$. On simule la marche au hasard de la particule sous python en utilisant la fonction `random.choice([a,b])` qui renvoie aléatoirement la valeur a ou b .

On donne le code suivant:

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import random
4 l=...
5 def position(N):
6     poslist=[...]
7     for i in range(N):
8         poslist.append(poslist[i]+random.choice([+l,-l]))
9     return poslist
10 N=.....
11 temps=[j for j in range(N+1)]
12 plt.plot(temps,position(N))
13 plt.xlabel('t(s)')
14 plt.ylabel('x')
15 plt.grid()
16 plt.show()
```

On exécute le code trois fois et on obtient:



2.a. Dédurre des courbes la valeur numérique de l (l'unité de x est arbitraire) et la valeur numérique de τ le pas de temps.

2.b. Expliquer la ligne 8 puis compléter les lignes 6 et 10. Ecrire ce que renvoie (numériquement) la fonction position pour chacune des courbes obtenues.

3. Les particules, au nombre de N_p , se trouvent initialement à l'origine O des coordonnées, puis chacune d'elles effectue une marche aléatoire de N pas comme dans le paragraphe précédent. On stocke les positions de toutes les particules à tous les instants dans un tableau noté T de $N + 1$ lignes et de N_p colonnes. L'élément de la ligne i et de la colonne j du tableau T se note $[i, j]$, il correspond à l'abscisse $x_j(t_i = i\tau)$ de la particule d'indice j (i prend les valeurs $0, 1, \dots, N$ et j prend les valeurs $0, 1, \dots, N_p - 1$).

$T[i, :]$ désigne la ligne i du tableau T , cette ligne contient les abscisses de toutes les particules à l'instant t_i .

$T[:, j]$ désigne la colonne j du tableau T , elle contient les abscisses de la particule d'indice j à tous les instants.

3.a. Exemple: on donne le tableau T ci-contre. Donner les valeurs de N_p et de N . Donner $x_0(t = 3\tau)$ pour la particule d'indice 0, $x_1(t = 2\tau)$ pour la particule d'indice 1 et $x_2(t = \tau)$ pour la particule d'indice 2.

$$\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 1 & 1 & -1 \\ \hline 2 & 0 & 0 \\ \hline 1 & 1 & -1 \\ \hline 2 & 0 & -2 \\ \hline \end{array}$$

3.b. On complète le code précédent pour obtenir le tableau T pour N_p particules et $N + 1$ pas de temps. Compléter la ligne 20 en expliquant la démarche.

17 $N_p, N = \dots, 1000$ # on choisira le nombre de particules dans la suite

18 $T = \text{np.zeros}((N+1, N_p))$ # on crée un tableau T de $N + 1$ lignes et N_p colonnes. Ce tableau créé contient uniquement des 0 pour le moment

19 for j in range(N_p):

20 ——— $T[:, j] = \text{positions}(\dots)$

3.c. On complète le code avec la fonction f . Expliquer l'opération réalisée par cette fonction.

21 def $f(l)$:

22 ——— $s = 0$

23 ——— for i in range(len(l)): # len(l) désigne le nombre de termes de la liste l

24 ——— ——— $s = s + l[i]**2$

25 ——— return $s/\text{len}(l)$

3.d. On complète encore le code avec les lignes suivantes:

26 $l1 = []$

27 for i in range($N+1$):

28 ——— $l1.append(f(T[i, :]))$

29 $D, a = 0.5, \dots$

30 $\text{plt.plot}(\text{temps}, l1)$

31 $\text{plt.plot}(\text{temps}, a * D * \text{np.array}(\text{temps}))$ # $\text{np.array}(\text{temps})$ est un vecteur contenant les mêmes termes que ceux présentes dans la liste temps , la transformation en vecteur de la liste est nécessaire pour pouvoir multiplier chacun des termes du vecteur par le nombre D

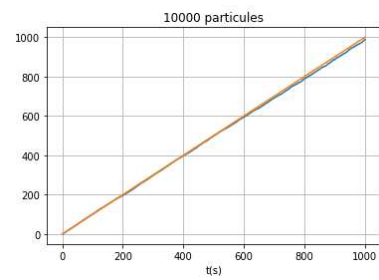
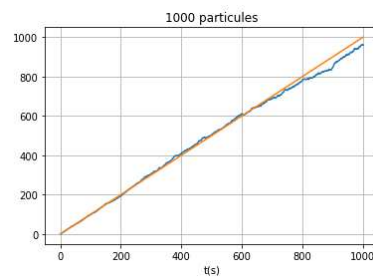
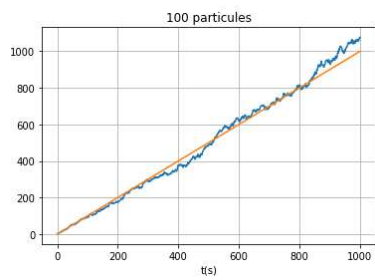
32 $\text{plt.xlabel}('t(s)')$

33 $\text{plt.title}('..... particules')$

34 $\text{plt.grid}()$

35 $\text{plt.show}()$

L'exécution du programme donne les courbes suivantes pour différentes valeurs de N_p (le nombre de particules).



Expliquer le contenu de la liste l1 (lignes 26 à 28).

Justifier (à l'aide des résultats de la question 1) la valeur numérique de D .

Préciser les abscisses et les ordonnées des courbes tracées lignes 30 et 31. Dédurre de ces deux courbes la valeur numérique de a .

Commenter l'évolution des graphes pour $N_p = 100, 1000$ et 10000 particules.