Complément chapitre th3 : marche au hasard

I. **Rappel : libre parcours moyen et vitesse quadratique moyenne**

Dans un fluide, les particules sont en mouvement incessant. Ces particules subissent de nombreuses collisions avec les particules qui les entourent et avec les parois du récipient qui les contient. Entre deux collisions, leur mouvement est rectiligne uniforme et à chaque collision, le vecteur vitesse est modifiée en sens, norme et direction.

On appelle *libre parcours moyen,*

Ordre de grandeur: dans l'air à température et pression ambiantes: $l = 100\ nm$

Les particules ont toutes des vitesses différentes mais on peut définir une vitesse particulière: la vitesse que possèdent le plus grand nombre de particules dans le fluide. On l'appelle *vitesse quadratique moyenne*, c'est la racine carrée de la moyenne de la vitesse au carré soit $v_q = \sqrt{< v^2 >}$.

Cas du GPM: d'après la statistique de Boltzman: l'énergie d'une particule est de $\frac{1}{2}k_BT$ par degré de liberté.

Ordre de grandeur: N_2 à $T = 293\ K$: $v_q = 510\ m/s$.

II. **Simulation du mouvement brownien d'une particule dans le plan** Oxy

1. Le modèle:

L'objectif est de modéliser le mouvement incessant (appelé mouvement brownien) d'une particule microscopique dans un fluide. Le principe de la modélisation repose sur le fait que:

- la particule subit aux instants $t = \tau$, $t = 2\tau$, $t = 3\tau$..., une collision qui modifie aléatoirement la direction de son vecteur vitesse.

- entre deux collisions, la particule a un mouvement rectiligne uniforme. On note l le libre parcours moyen, soit la distance moyenne parcourue par la particule entre deux collisions successives.

2. Simulation de la marche au hasard d'une particule dans le plan Oxy

2.a. *Principe de la simulation*

La simulation de la marche au hasard d'une particule consiste à:

- Initialiser sa position, choisir un nombre de collisions et le libre parcours moyen

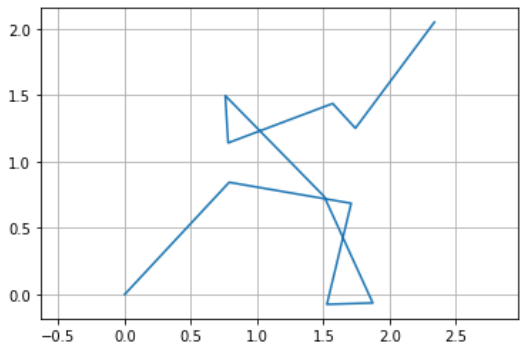
- Pour chaque collision : tirer au sort le déplacement, calculer la nouvelle position et la stocker.

2.b. *Le code proposé et les résultats*

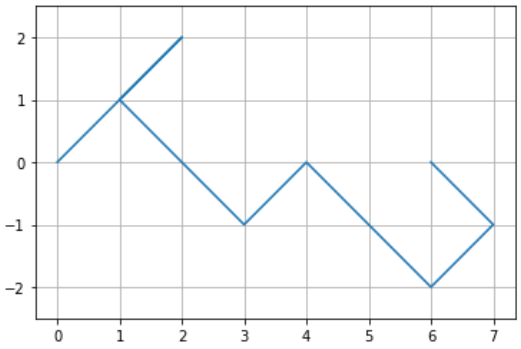
```
import matplotlib.pyplot as plt
import numpy as np
import random
```

```
l=....
def marche(N):
——x0,y0=0.0,0.0
——xlist,ylist=[x0],[y0]
——for i in range(N):
—— ——dx,dy=random.uniform(-l,+l),random.uniform(-l,+l)
—— ——xlist.append(xlist[i]+dx)
—— ——ylist.append(ylist[i]+dy)
——return xlist,ylist
x=marche(...)[0]
y=marche(...)[1]
plt.plot(x,y)
plt.axis('equal')
plt.grid()
plt.show()
```

L'exécution du code donne la courbe:



Quand on change random.uniform en random.choice([-l,+l]) on obtient:

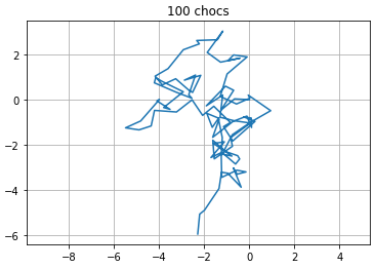


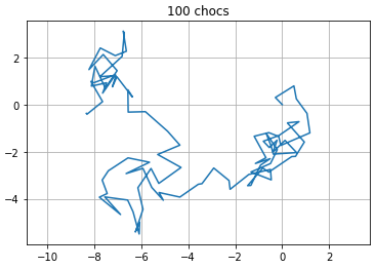
Expliquez la différence entre random.uniform et random.choice et donner les valeurs de l et de N.

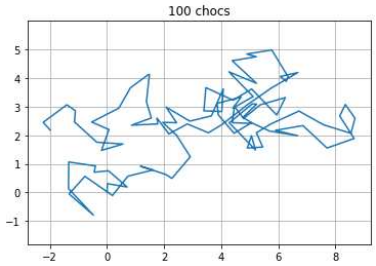
Compléter le code pour tracer les courbes $x(t)$ et $y(t)$.

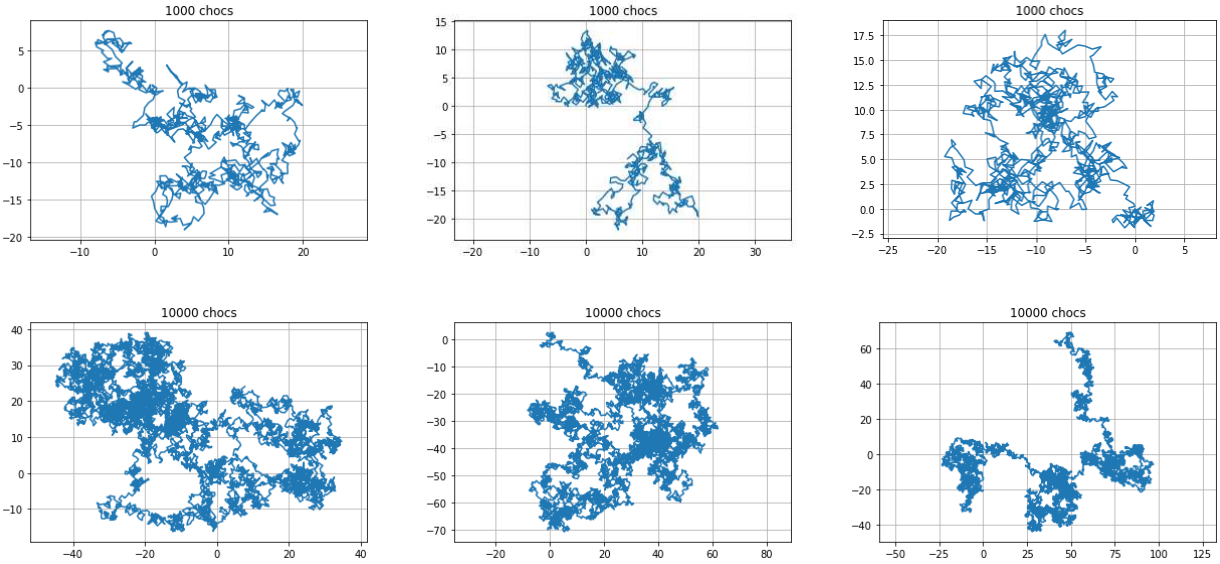
```
t=np.linspace(....,.....,.....)
plt.plot(.....,......)
plt.title('x(t)')
plt.show()
plt.plot(.....,......)
plt.title('y(t)')
plt.show()
```

On donne les résultats des simulations pour 100, 1000, 10 000 et 100 000 chocs









On note L la distance moyenne de diffusion pendant un temps Δt. On cherche la relation entre L et Δt. Cette relation n'est pas linéaire en effet:

2.c. *L'interprétation*

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

On a $p(x,t+\tau) =$

DL à l'ordre 1 de $p(x,t+\tau) =$

DL à l'ordre 2 de $p(x+l,t) =$

DL à l'ordre 2 de $p(x-l,t) =$

III. **Simulation de la marche au hasard de N particules selon** Ox

1. Principe de la simulation

Simuler la marche au hasard d'un grand nombre de particules et caractériser l'étalement spatial au cours du temps consiste à:

- Initialiser la position des particules, choisir un nombre de collisions, choisir un nombre de particules et le libre parcours moyen

- Pour chaque collision:

• pour chaque particule, tirer au sort le déplacement et calculer sa nouvelle position

• calculer la nouvelle valeur de $< x^2 >$ à stocker (valeur moyenne sur l'ensemble des particules des carrés des positions)

• tracer sur le même graphe en fonction du temps, les valeurs de $< x^2 >$

2. Le code proposé et les résultats

```
import matplotlib.pyplot as plt
import numpy as np
import random

def f(l):
    s=0
    for i in range(len(l)):
        s=s+l[i]**2
    return s/len(l)

lpm,Np,N=1,....,..... # lpm: libre parcours moyen, Np: nombre de particules et N: nombre de chocs

lx=np.zeros((Np))

lmoy=[0]

for j in range(N):

    for i in range(Np):

        lx[i]=lx[i]+random.choice([-lpm,+lpm])

    lmoy.append(f(l))

t=[i for i in range(len(lmoy))]

plt.plot(t,lmoy)
plt.xlabel('t(s)')
plt.ylabel('< x^2 >')
plt.title('Np particules')
plt.grid()
plt.show()
```

L'exécution du code pour différentes valeurs de N_p donne: