TP2 : Mouvement brownien

I. Marche aléatoire

Le botaniste écossais Robert Brown a observé pour la première fois en 1827 le mouvement aléatoire des particules en raison de l'agitation thermique. Il a cependant fallut attendre les travaux d'Einstein et Sutherland en 1905 pour qu'une première explication physique soit formulée.
Une particule brownienne se déplace dans un fluide sous l'action des collisions aléatoires avec les molécules de solvants. L'effet de ces collisions est de faire rebondir ou de pousser la particule brownienne dans une direction aléatoire.
Elle effectura donc un déplacement de longueur $v.\Delta t$ dans la direction $\pm\vec{e}_i$ où sens et direction sont tirés aléatoirement à chaque pas de temps Δt. La vitesse instantannée v de la particule est prise constante.

$$\vec{r}_{n+1} = \vec{r}_n \pm v\Delta t\ \vec{e}_i \text{ où } i = 1, 2 \text{ avec une probabilité } \tfrac{1}{2}.$$

L'objectif de ce TP est d'explorer numériquement ce type de trajectoire, appelée marche aléatoire, et d'obtenir quelques propriétés statistiques de ce mouvement.

I.1. Pourquoi considérer que la distribution des déplacements est aléatoire et de moyenne nulle?(1 point)

I.2 Une trajectoire est donnée par la suite x_n des positions de la particule aux instants $t_n = n\Delta t$, où Δt est le pas de temps de la simulation. Pour une marche aléatoire 1D

$$x_{n+1} = x_n + \xi v\Delta t \text{ où } \xi = \pm \text{ avec une probabilité } \tfrac{1}{2}$$

Il n'y a pas de mémoire des déplacement précédents.

En vous inspirant du TP1, compléter la fonction move_particle du module randomwalk.py afin de générer le mouvement Brownien d'une particule. (2 points)
Tracer sa position en fonction du temps. On pourra utiliser la fonction plot_curve. (2 points)
Un exemple d'implémentation vous est fourni dans le fichier TP2_example.py.
Vous commencerez par utiliser un pas de temps $\Delta t = 0.1$, un temps total $T = 10$ (soit 100 pas de temps), avant de faire des simulations plus longues.

II.1. Tracez les positions de 10 particules Browniennes en fonction du temps sur un même graphique. Vous pourrez vous inspirer du script TP2_example.py. Commentez (2 pt)

II.2. Tracez le nombre de particules en fonction de leur position pour différents temps de simulation T. On pourra utiliser une liste xi contenant la position de toutes les particules à l'instant t_i, et on réfléchira a l'ordre dans lequel on imbrique les boucles sur les particules et sur le temps. On prendra $N = 1000$ particules, $T = 1,\ 10,\ 100$ avec $\Delta t = 0.1$. On pourra utiliser la fonction plot_histogram pour faire un histogramme du nombre de particules en fonction de la position (à t fixé). Commentez. (3.5 points)

II.3. Le déplacement quadratique moyen $\sigma^2(t_k)$ est défini par la moyenne d'ensemble :

$$\sigma^2(t_k) = \left\langle (x_k - x_0)^2 \right\rangle \ . \tag{11}$$

On prendra $x_0 = 0$ pour simplifier. Afin d'évaluer cette moyenne, il y a deux possibilités :

- soit on génère N trajectoires jusqu'au temps t_k, et donc : $\sigma^2(t_k) = N^{-1} \sum_{i=1}^{N} (x_k^i)^2$.
- soit on fait tourner le programme une seule fois pendant un temps t_n, et on obtient $\sigma^2(t_k)$, pour $t_k < t_n$, en faisant "glisser" l'intervalle de temps t_k :

$$\sigma^2(t_k) = (n - k + 1)^{-1} \sum_{j=0}^{n-k} (x_{j+k} - x_j)^2 \,. \tag{12}$$

On complétera calculate_mean_squared_displacement afin de lui faire calculer le déplacement quadratique moyen, avec les deux méthodes. Tracez σ^2 en fonction du temps pour N=10^3 particules.
Vérifier que l'on obtient un comportement linéaire (3 pts)

III.1. En partant du script TP2_example.py, générez deux mouvement Brownien $x(t)$ et $y(t)$ qui seront les coordonnées d'une particule brownienne bi-dimensionnelle dont vous visualiserez la trajectoire 2D (à l'aide de la fonction plot_curve). Commentez.
Puis tracer pour différentes durées les trajectoires d'une dizaine de particules
Comment varie la taille typique des structures formées par les trajectoires des particules lorsque l'on augmente le temps T ?
On pourra prendre $T = 10, 10^2, 10^3, 10^4$, et tracer un cercle de rayon $v\Delta t\sqrt{\frac{2T}{\Delta t}}$ (3 points).

III.2. Comment s'effectue les déplacements élémentaire de votre code 2D ?
Pour générer des déplacements élémentaires selon les axes (0x) ou (Oy), remplacer la fonction move_particle du module randomwalk.py par une fonction qui génere une marche aléatoire 2D selon la loi initiale :

$$\vec{r}_{n+1} = \vec{r}_n \pm v\Delta t \ \vec{e}_i \text{ où } i = 1, 2 \text{ avec une probabilité } \tfrac{1}{2}. \text{ (2 points)}$$

III.3 Calculer et tracer le déplacement quadratique moyen σ^2 en fonction du temps pour 10^2 ou 10^3 particules. Vérifier que l'on obtient un comportement linéaire (1.5 pts)

Bonus : Explorer numériquement le cas d'un biais ou dérive, c'est à dire de probabilités non équiréparties.