Introduction à l'analyse numérique
Année : 2017-2018
Formation : L2 Mathématiques

TP3 : Mouvement Brownien et marches aléatoires

Le but de ce TP est d'étudier numériquement des phénomènes de marches aléatoires, qui sont - entre autres - un moyen d'approximer le mouvement brownien.

Le mouvement brownien est une description mathématique du mouvement aléatoire d'une "grosse" particule immergée dans un fluide et qui n'est soumise à aucune autre interaction que des chocs avec les "petites" molécules du fluide environnant. Il en résulte un mouvement très irrégulier de la grosse particule, qui a été décrit pour la première fois en 1827 par le botaniste Robert Brown en observant des mouvements de particules à l'intérieur de grains de pollen d'une espèce de fleur sauvage nord-américaine.

La description physique la plus élémentaire du phénomène est la suivante : entre deux chocs, la grosse particule se déplace en ligne droite avec une vitesse constante ; la grosse particule est accélérée lorsqu'elle rencontre une molécule de fluide ou une paroi.

Ce mouvement permet de décrire avec succès le comportement thermodynamique des gaz, ainsi que le phénomène de diffusion. Il est aussi très utilisé dans des modèles de mathématiques financières. (source Wikipedia). Pour visualiser : `https://www.canal-u.tv/video/cerimes/le_mouvement_brownien.10217`.

1 Marche aléatoire en une dimension

On considère une particule dans un tube très fin. Cette particule va se déplacer entre chaque pas de temps dt aléatoirement vers la gauche ou la droite.

1. Ecrire une fonction `pile_ou_face` qui renvoie aléatoirement les valeurs +1 ou -1.
```
function resultat=pile_ou_face()
// resultat= ???
endfunction
```
2. La position de la particule à l'instant initial t_1 est $x_1 = 0$. Sa position à l'instant $t_n = ndt$ est donnée par $x_n = x_{n-1}+$ `pile_ou_face()`. Ecrire une fonction qui prend un entier N en argument et envoie un tableau `position` donnant les positions de la particule aux instants successifs $t_1, \cdots, t_N$.

3. Afficher le résultat de la fonction précédente.

4. Ecrire une fonction qui, à partir du tableau `position` créé par la fonction précédente, renvoie la valeur minimale et la valeur maximale des positions de la particule.

5. Ecrire une nouvelle fonction qui calcule les positions de la particule jusqu'à ce que celle-ci atteigne la position -30 ou +30, ou que le nombre d'itérations dépasse 10 000.

2 Marche aléatoire en deux dimensions

La particule est sortie du tube et peut maintenant se déplacer dans quatre directions.

1. Ecrire une fonction `boussole` qui renvoie aléatoirement les valeurs [1,0], [-1,0], [0,-1] ou [0,1].

2. Pour représenter les positions successives de la particule, on va utiliser deux tableaux : `X` pour les abcisses et `Y` pour les ordonnées. A l'instant initial t_1 la position de la particule est $X(1) = 0, Y(1) = 0$. Ecrire une fonction qui prend un entier N en argument et envoie les tableaux `X` et `Y` donnant les positions de la particule aux instants successifs $t_1, \cdots, t_N$.

3. Afficher le résultat de la fonction précédente.

4. Ecrire une fonction qui calcule la distance (euclidienne) maximale de la particule, au cours de son parcours, par rapport à sa position initiale.

5. Appelez la fonction précédente 100 fois de suite, stochez les résultats et affichez-les.

3 Et s'il y a du courant ou des contraintes supplémentaires ?

1. Modifiez les programmes précédents pour imposer à la particule, en plus du mouvement aléatoire, une vitesse horizontale constante, et visualisez le résutat.

2. Autre possibilité : modifiez la fonction boussole pour changer les probabiliés de déplacement dans chaque direction, et comparez les trajectoires ainsi obtenues aux précédentes.

3. Vous pouvez aussi modifier le programme pour "empêcher" la particule d'aller au delà de la position $x = -1$, et visualiser l'influence de cette contrainte sur les trajectoires.