



Thème : Mouvements browniens



L'étude du mouvement brownien découvert par le biologiste anglais Robert Brown en 1828 fut faite par Albert Einstein (1905) puis développée par Norbert Wiener (1920) et Paul Levy (1948). Les implications de l'étude de ces trajectoires furent extrêmement fécondes et débouchèrent sur de nombreux sujets modernes des mathématiques, de la physique ou encore en finance. Les 4 parties sont indépendantes.

Exercice 1. ♦♦ Mouvement brownien dans le plan avec pas gaussiens

VAD37

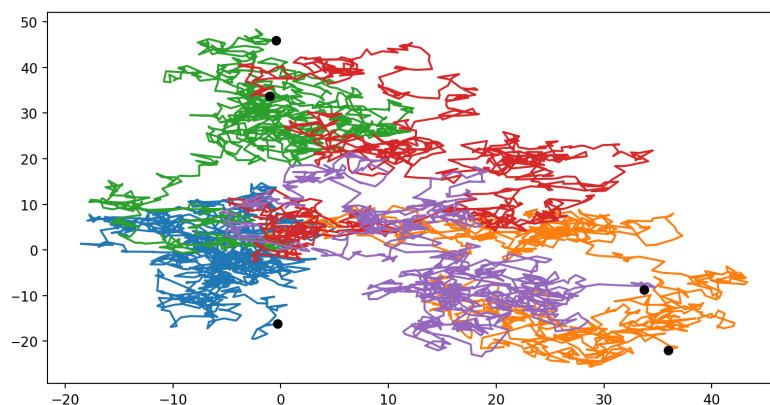
Une particule se déplace dans un plan muni d'un repère orthonormé. Elle part de l'origine O des coordonnées à l'instant 0.

Si à l'instant $t = k$, $k \in \mathbb{N}$, elle se situe au point de coordonnées (X_k, Y_k) , alors à l'instant $t = k + 1$ elle se trouve au point de coordonnées (X_{k+1}, Y_{k+1}) de sorte que $A_{k+1} = X_{k+1} - X_k$ et $B_{k+1} = Y_{k+1} - Y_k$ suivent la loi normale $\mathcal{N}(0, 1)$. On suppose les variables A_i et B_j mutuellement indépendantes.

Soit $\Phi_{m,\sigma}$ la fonction de répartition de la loi normale $\mathcal{N}(m, \sigma^2)$ et $\varphi_{m,\sigma}$ la densité continue sur $\mathbb{R}$ de cette loi.

On rappelle que $\Gamma(1/2) = \sqrt{\pi}$ et pour tout $x \in \mathbb{R}_+^*$, $\Gamma(x+1) = x\Gamma(x)$.

1. a) Quelle est la loi suivie par X_n ?
 b) Soit M_n le point de coordonnées (X_n, Y_n) .
 Exprimer, à l'aide de la fonction $\Phi_{0,1}$, la probabilité qu'à l'instant n la particule M_n se trouve dans le carré $C = [-1, 1]^2$.
 c) Que dire de cette probabilité lorsque $n \rightarrow +\infty$?
2. a) Sachant que la commande `rd.normal(0,1)` simule une loi normale centrée réduite, donner un programme python qui prend en argument n et renvoie la matrice ligne $[0 \quad X_1 \quad X_2 \quad \dots \quad X_n]$.
 b) Comment tracer une trajectoire de la particule entre les temps 0 et 100? Ci-dessous, quelques gribouillis trajectoires obtenues par le code.



3. Soit D_n la distance de M_n à l'origine : $D_n = \sqrt{X_n^2 + Y_n^2}$.
 a) Reconnaître la loi de $X_n^2/(2n)$, puis celle de D_n^2 . En déduire la loi de D_n et calculer son espérance.
 b) Quelle est la probabilité qu'à l'instant n la particule se trouve dans le disque de centre O et de rayon 1?

Exercice 2. ♦♦♦ Temps du premier retour à l'origine

RVA62

On admet les résultats suivants :

- Pour tout réel $x \in [0; 1[$, on a : $\frac{1}{\sqrt{1-x}} = \sum_{k=0}^{+\infty} \frac{\binom{2k}{k}}{4^k} x^k$.
- **Produit de Cauchy.** Soient $(x_n)_{n \in \mathbb{N}}$ et $(y_n)_{n \in \mathbb{N}}$ deux suites réelles positives. On pose, pour tout $n \in \mathbb{N}$,

$$z_n = x_0 y_n + x_1 y_{n-1} + x_2 y_{n-2} + \dots + x_{n-1} y_1 + x_n y_0 = \sum_{i=0}^n x_i y_{n-i}$$

Si les séries $\sum x_n$ et $\sum y_n$ sont absolument convergentes alors la série $\sum z_n$ converge et

$$\sum_{n=0}^{+\infty} z_n = \left(\sum_{k=0}^{+\infty} x_k \right) \left(\sum_{k=0}^{+\infty} y_k \right).$$

On se donne un espace probabilisé $(\Omega, \mathcal{A}, \mathbf{P})$. Sur cet espace, on considère une suite $(X_n)_{n \in \mathbb{N}^*}$ de variables aléatoires indépendantes et de même loi que X_1 , cette loi étant définie par

$$\mathbf{P}([X_1 = 1]) = \mathbf{P}([X_1 = -1]) = \frac{1}{2}.$$

On pose $S_0 = 0$ et, pour tout entier naturel n non nul, $S_n = \sum_{k=1}^n X_k = X_1 + X_2 + \dots + X_n$.

Par exemple, S_n pourrait représenter l'abscisse (aléatoire) au temps n d'une particule se déplaçant sur un axe et partie de l'origine au temps 0, qui saute à chaque instant d'une unité à gauche ou d'une unité à droite avec une égale probabilité.

On pose aussi, pour tout élément ω de Ω ,

$$R_\omega = \{n \in \mathbb{N}^* \mid S_n(\omega) = 0\} \quad \text{et} \quad T(\omega) = \begin{cases} \min R_\omega & \text{si } R_\omega \neq \emptyset \\ 0 & \text{si } R_\omega = \emptyset \end{cases}$$

Ainsi T pourrait être le temps d'attente (aléatoire) du premier retour à l'origine de la particule évoquée plus haut.

Pour tout entier naturel n , on note E_n l'événement $E_n = [T > n] \cup [T = 0]$.

1. Soit n un entier naturel non nul. On pose $A_n = [S_n = 0]$ et, pour tout entier naturel k tel que $0 \leq k \leq n-1$,

$$A_k = ([S_k = 0] \cap [S_{k+1} \neq 0] \cap [S_{k+2} \neq 0] \cap \dots \cap [S_n \neq 0]) = \left([S_k = 0] \cap \left(\bigcap_{i=k+1}^n [S_i \neq 0] \right) \right).$$

Ainsi, pour tout entier k tel que $0 \leq k \leq n$, A_k serait l'événement : "Pour la dernière fois avant l'instant n la particule est à l'origine à l'instant k ".

Pour tout entier k tel que $0 \leq k \leq n$, justifier l'égalité suivante

$$\mathbf{P}(A_k) = \mathbf{P}([S_k = 0]) \mathbf{P}(E_{n-k}).$$

2. En déduire l'égalité : $1 = \sum_{k=0}^n \mathbf{P}([S_k = 0]) \mathbf{P}(E_{n-k})$.

3. Établir que pour tout réel x de $[0, 1[$, $\frac{1}{1-x} = \left(\sum_{n=0}^{\infty} \mathbf{P}([S_n = 0]) x^n \right) \left(\sum_{n=0}^{\infty} \mathbf{P}(E_n) x^n \right)$.

4. Pour tout entier naturel n , calculer $\mathbf{P}([S_n = 0])$.

5. En déduire que, pour tout réel x de $[0, 1[$, on a $\sum_{n=0}^{+\infty} \mathbf{P}(E_n) x^n = \sqrt{\frac{1+x}{1-x}}$.

6. En remarquant que l'événement $[T = 0]$ est inclus dans E_n pour tout entier naturel n , montrer que l'on a :

$$\mathbf{P}([T = 0]) = 0.$$

Ainsi, presque sûrement, la particule citée en exemple, revient à l'origine.

7. *Preuve du premier résultat admis.*

On considère la fonction indéfiniment dérivable φ définie, pour tout réel x de $[0, 1[$, par : $\varphi(x) = \frac{1}{\sqrt{1-x}}$.

a) Pour tout réel x de $[0, 1[$ et tout entier naturel n , établir l'égalité

$$\varphi^{(n)}(x) = \frac{(2n)!}{4^n n!} (1-x)^{-\frac{2n+1}{2}}$$

où $\varphi^{(n)}$ désigne la dérivée n -ième de φ (avec, en particulier, $\varphi^{(0)} = \varphi$).

b) Pour tout entier naturel n et tout réel x de $[0, 1[$, justifier l'égalité suivante

$$\varphi(x) = \sum_{k=0}^n \frac{\binom{2k}{k}}{4^k} x^k + \int_0^x \frac{(x-t)^n}{n!} \varphi^{(n+1)}(t) dt.$$

c) Pour tout entier naturel n , prouver l'inégalité : $\binom{2n+2}{n+1} \leq 4^{n+1}$.

d) Pour tout couple (t, x) de réels tel que $0 \leq t \leq x < 1$, vérifier les inégalités : $0 \leq \frac{x-t}{1-t} \leq x$.

e) En déduire que, pour tout réel x de $[0, 1[$, on a $\lim_{n \rightarrow +\infty} \int_0^x \frac{(x-t)^n}{n!} \varphi^{(n+1)}(t) dt = 0$.

f) Pour tout réel x de $[0, 1[$, démontrer l'égalité $\frac{1}{\sqrt{1-x}} = \sum_{k=0}^{+\infty} \frac{\binom{2k}{k}}{4^k} x^k$.

Exercice 3. ♦♦ Mouvements browniens dans le plan

Une particule se déplace dans le plan à partir de l'origine $O = (0;0)$. À chaque étape, la particule se déplace d'une unité dans une direction choisie au hasard. On suppose les directions mutuellement indépendantes. Mathématiquement, on considère une suite de variables aléatoires $(\Theta_k)_{k \in \mathbb{N}}$ mutuellement indépendantes toutes de loi uniforme sur $[0; 2\pi]$. Si $M_n = (X_n, Y_n)$ désigne la position de la particule à l'étape $n \in \mathbb{N}$, on a alors

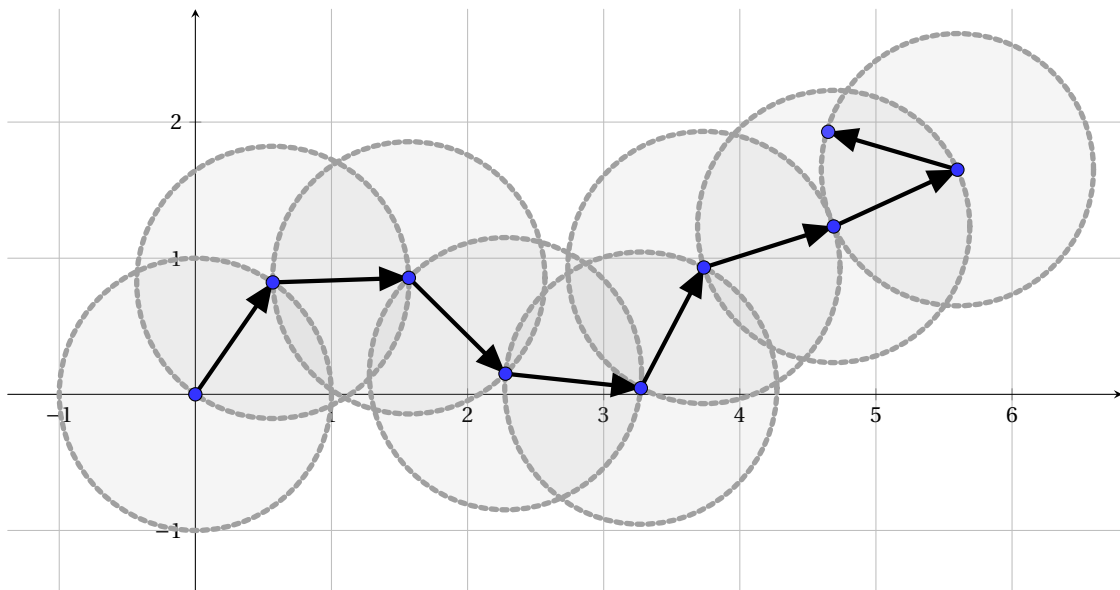
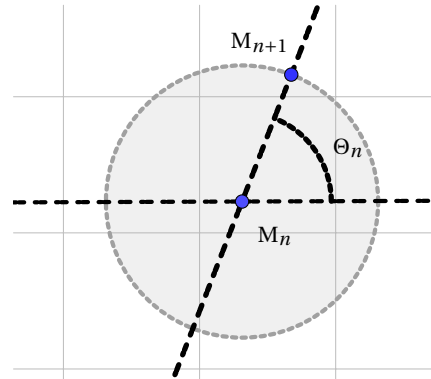
$$X_0 = Y_0 = 0$$

$$\text{et } X_n = \sum_{k=1}^n \cos(\Theta_k), \quad Y_n = \sum_{k=1}^n \sin(\Theta_k).$$

On pose aussi

$$D_n = \sqrt{X_n^2 + Y_n^2}.$$

Ci-dessous, un exemple des premières étapes d'une trajectoire.



1. Écrire un programme python qui prend en argument $n \in \mathbb{N}$ et renvoie une réalisation de la matrice aléatoire

$$A_n = \begin{bmatrix} X_0 & X_1 & \dots & X_n \\ Y_0 & Y_1 & \dots & Y_n \end{bmatrix}.$$

Afficher alors quelques trajectoires du pion.

2. En déduire un second programme qui prend en argument $n \in \mathbb{N}$ et renvoie une réalisation de la matrice aléatoire

$$B_n = [D_0 \quad D_1 \quad \dots \quad D_n].$$

3. Soit $n \in \mathbb{N}^*$. Calculer l'espérance et la variance de X_n et Y_n .
4. a) Que permet de conjecturer le code suivant? Prouver la conjecture.

Editeur

```
def truc(n):
    t=np.zeros(n+1)
    M=SimuA(n) # renvoie une simulation
                de A_n
    for i in range(1,n+1):
        t[i]=M[0,i]*M[1,i]
    return t
```

```
def machin(n):
    m=10000
    t=np.zeros(n+1)
    for i in range(m):
        t+=truc(n)
    return np.round(t/m,3)
#np.round renvoie une approximation
à 10-4)
```

```
>>> machin(20)
array([ 0.      , -0.001, -0.003,  0.009,  0.007,  0.013,  0.017,  0.029,
        0.025,  0.024,  0.049,  0.037,  0.026, -0.012, -0.017, -0.007,
       -0.029, -0.027, -0.08 , -0.153, -0.16  ])
```

b) Donner l'espérance de D_n^2 .

• *Approximations*

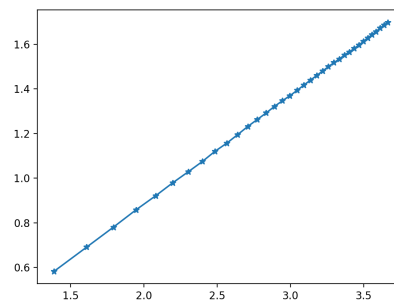
5. a) Écrire un programme `esperanceD` qui prend en argument n et renvoie une approximation de la matrice ligne

$$[E(D_0) \quad E(D_1) \quad E(D_2) \quad \dots \quad E(D_n).]$$

b) On admet qu'il existe deux réels α, β tels que $E(D_n) \simeq \beta n^\alpha$.

À l'aide du code python suivant, proposer une valeur simple pour α .

```
n=40
X=np.arange(n+1)[4:40]
Y=esperanceD(n)[4:40]
plt.plot(np.log(X), np.log(Y), '*--')
plt.show()
```



6. Justifier que pour n grand, les variables X_n et Y_n peuvent être approximées par des lois normales $\mathcal{N}(0, n/2)$.

7. Soient Z, T deux variables aléatoires de lois normales $\mathcal{N}(0, n/2)$ indépendantes. Vérifier que $\sqrt{Z^2 + T^2}$ est une variable aléatoire à densité dont une densité est g , définie sur $\mathbb{R}$ par

$$\forall t \in \mathbb{R}, \quad g(t) = \begin{cases} 0 & \text{si } t \leq 0 \\ \frac{2t}{n} e^{-t^2/n} & \text{sinon.} \end{cases}$$

8. On admet que la loi de D_n peut être approximer la loi de $\sqrt{Z^2 + T^2}$. Donner la valeur théorique de α et β . Vérifier à l'aide de python.

Exercice 4. ♦♦♦ Mouvement brownien sur une sphère

PY92

Dans la suite, on identifie tout vecteur de $\mathbb{R}^3$ avec sa matrice de $\mathcal{M}_{3,1}(\mathbb{R})$. On munit $\mathbb{R}^3$ de son produit scalaire canonique :

$$\forall x, y \in \mathbb{R}^3, \quad \langle x, y \rangle = {}^t x y.$$

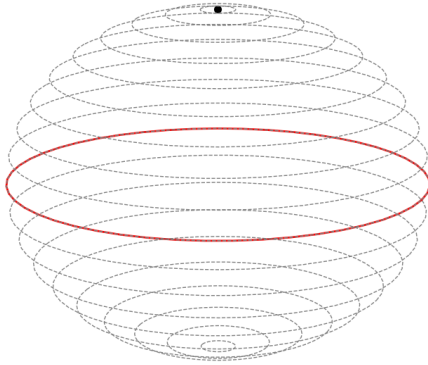
On note $\|\cdot\|$, la norme associée.

La sphère unité de $\mathbb{R}^3$ est définie par

$$\mathcal{S} = \{x \in \mathbb{R}^3 \mid \|x\| = 1\}.$$

Le pôle nord de la sphère est alors le point de coordonnées $x_p = (0; 0; 1)$.

Le code ci-contre permet de représenter la sphère unité et le pôle nord.



```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

axes = plt.axes(projection="3d")
# Le pôle nord
axes.plot([0],[0],[1],'.',color='black',
          linewidth=5)
# L'équateur
t = np.linspace(0,2*np.pi, 50)
axes.plot(np.cos(t),np.sin(t),np.zeros(50)
          ,color='red',linewidth=1)
# Les latitudes
r=np.linspace(0,2*np.pi,20)
for i in range(20):
    x = np.cos(r[i])*np.cos(t)
    y = np.cos(r[i])*np.sin(t)
    z = np.sin(r[i])*np.ones(50)
    axes.plot(x, y, z, '--',color='grey',
              linewidth=0.5)
plt.axis('off')
plt.show()
```

1. Les trois fonctions suivantes permettent de construire 3 matrices, lesquelles? On les notera $R_1(\theta)$, $R_2(\theta)$ et $R_3(\theta)$.

```
def Rotation1(theta):
    M=np.eye(3)
    M[1,1]=np.cos(theta)
    M[2,2]=M[1,1]
    M[2,1]=np.sin(theta)
    M[1,2]=-M[2,1]
    return M

def Rotation2(theta):
    M=np.eye(3)
    M[0,0]=np.cos(theta)
    M[1,1]=M[0,0]
    M[1,0]=np.sin(theta)
    M[0,1]=-M[1,0]
    return M

def Rotation3(theta):
    M=np.eye(3)
    M[0,0]=np.cos(theta)
    M[2,2]=M[0,0]
    M[2,0]=np.sin(theta)
    M[0,2]=-M[2,0]
    return M
```

2. Vérifier que les trois matrices obtenues sont orthogonales. En déduire que pour tout $x \in \mathbb{R}$, $\|R_i x\| = \|x\|$.
Les matrices $R_1(\theta)$, $R_2(\theta)$ et $R_3(\theta)$ correspondent à des matrices de rotation d'angle θ respectivement dans les plans (Oyz) , (Oxy) et (Oxz) .

Soit $h \in \mathbb{R}_+^*$. Soient les suites de variables aléatoires $(\Theta_i)_{i \in \mathbb{N}}$ de loi uniforme $\mathcal{U}([0; h])$ et $(N_i)_{i \in \mathbb{N}}$ de loi uniforme discrète sur $\{1; 2; 3\}$ définies sur un même espace probabilisé $(\Omega, \mathcal{A}, \mathbf{P})$. On suppose les variables $(\Theta_i, N_i)_{i \in \mathbb{N}}$ mutuellement indépendantes. On définit alors :

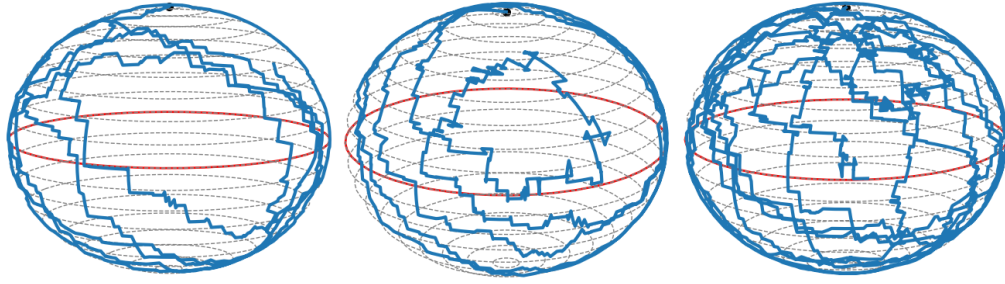
→ La suite de matrice aléatoire $(M_i)_{i \in \mathbb{N}}$ par :

$$\forall i \in \mathbb{N}, \quad M_i = R_{N_i}(\Theta_i).$$

→ La suite aléatoire $(X_i)_{i \in \mathbb{N}}$ de vecteurs de $\mathbb{R}^3$ par

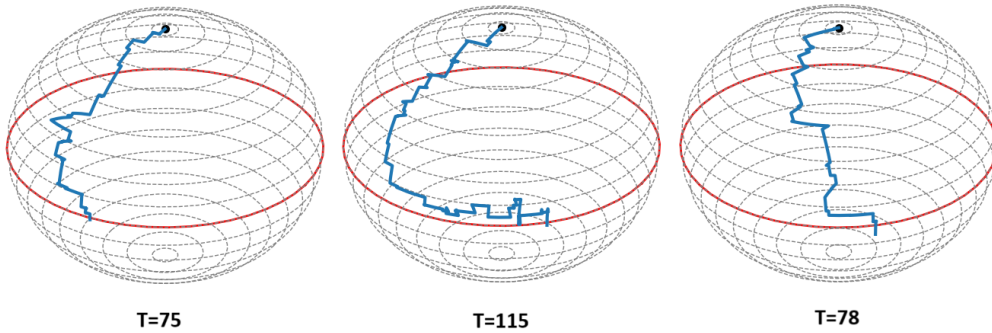
$$X_0 = x_p \quad \text{et} \quad \forall i \in \mathbb{N}, \quad X_{i+1} = M_i X_i.$$

3. Justifier que pour tout $i \in \mathbb{N}$, tout $\omega \in \Omega$, $X_i(\omega) \in \mathcal{S}$.
4. Écrire une fonction python qui prend en argument $h \in \mathbb{R}_+^*$ et renvoie une réalisation de M_1 .
5. En déduire un programme qui prend en argument n et renvoie une réalisation de X_n .
On pourra choisir $h = 0.1$.
6. Compléter le programme précédent pour afficher les $n+1$ premières réalisations de $X_0, X_1, \dots, X_n$ pour obtenir une "trajectoire".
Si x, y, z représentent 3 matrices lignes de même taille correspondent à des abscisses, des ordonnées et des hauteurs, alors la commande `axes.plot(x, y, z)` permet de représenter la trajectoire.
Ci-après, quelques exemples :



Si x , y , z représentent 3 matrices lignes de même taille correspondent à des abscisses, des ordonnées et des hauteurs, alors la commande `axes.plot(x, y, z)` permet de représenter la trajectoire.

7. Soit T , la variable aléatoire égale au rang pour lequel on traverse l'équateur pour la première fois. Simuler T et donner une estimation de son espérance.



8. Reprendre les questions précédentes où U_i suit une loi uniforme sur $[-h; h]$. Ci-dessous un exemple de trajectoire.



Exercice ??

p. ??

VA37

1.a) Par télescopeage $X_n = \sum_{k=1}^n A_k$. Les variables A_i , $i \in \mathbb{N}$ étant indépendantes et de même loi $\mathcal{N}(0, 1)$, on sait que $X_n \hookrightarrow \mathcal{N}(0, n)$.

Dans la suite, on note $\Phi_{0, \sqrt{n}}$, la fonction de répartition de X_n .

1.b) M est dans le carré voulu si et seulement si $-1 \leq X_n \leq 1$ et $-1 \leq Y_n \leq 1$. Par indépendance des variables X_n et Y_n (les variables A_i sont aussi indépendantes des variables B_j), il vient

$$\begin{aligned} \mathbf{P}(M_n \in C) &= \mathbf{P}(-1 \leq X \leq 1) \mathbf{P}(-1 \leq Y \leq 1) \\ &= \left(\Phi_{0, \sqrt{n}}(1) - \Phi_{0, \sqrt{n}}(-1) \right)^2 \\ &= \left(2\Phi_{0, \sqrt{n}}(1) - 1 \right)^2. \end{aligned}$$

En remarquant que $X_n/\sqrt{n}$ suit une loi normale centrée réduite et que pour tout réel x

$$\Phi_{0,1}(x) + \Phi_{0,1}(-x) = 1$$

On obtient la simplification

$$\Phi_{0, \sqrt{n}}(1) =$$

$$\mathbf{P}(X_n \leq 1) = \mathbf{P}\left(\frac{X_n}{\sqrt{n}} \leq \frac{1}{\sqrt{n}}\right) = \Phi_{0,1}\left(\frac{1}{\sqrt{n}}\right)$$

$$\text{d'où} \quad \mathbf{P}(M_n \in C) = \left(2\Phi_{0,1}\left(\frac{1}{\sqrt{n}}\right) - 1 \right)^2.$$

1.c) Par continuité de la fonction $\Phi_{0,1}$

$$\mathbf{P}(M_n \in C) \xrightarrow{n \rightarrow \infty} (2\Phi_{0,1}(0) - 1)^2 = 0.$$

2.a)

```
import numpy.random as rd
def simu_Xn(n):
    X=np.zeros(n+1)
    for i in range(1,n+1):
        X[i]=X[i-1]+rd.normal(0,1)
    return X
```

2.b)

```
n=2000
plt.plot(simu_Xn(n), simu_Xn(n))
plt.show()
```

3.a) La variable $W_n = X_n^2/(2n)$ prend ses valeurs dans $\mathbb{R}^+$. On a donc et $\mathbf{P}(W_n \leq x) = 0$ si $x < 0$ et pour $x \geq 0$:

$$\mathbf{P}(W_n \leq x) = \mathbf{P}\left(|X_n| \leq \sqrt{2nx}\right) = 2\Phi_{0, \sqrt{n}}(\sqrt{2nx})$$

On vérifie que la fonction de répartition de W_n est continue sur $\mathbb{R}$ et de classe $\mathcal{C}^1$ sur $\mathbb{R}^*$. La variable W_n est à densité et une densité est définie sur $\mathbb{R}$ par

$$f_{W_n}(x) = \begin{cases} \frac{2}{\sqrt{2nx}} \phi_{0, \sqrt{n}}(\sqrt{2nx}) = \frac{1}{\sqrt{x}\sqrt{\pi}} e^{-x} \text{ si } x > 0 \\ 0 \text{ sinon.} \end{cases}$$

Ainsi pour $x > 0$,

$$f_{W_n}(x) = \frac{1}{\Gamma(1/2)} x^{\frac{1}{2}-1} e^{-x}.$$

On reconnaît une densité d'une variable aléatoire qui suit la loi $\gamma(1/2)$. C'est aussi la loi de $Y_n^2/(2n)$. Comme $D_n^2/(2n) = X_n^2/(2n) + Y_n^2/(2n)$, avec $X_n^2/(2n)$ et $Y_n^2/(2n)$ indépendantes, on sait alors que :

$$D_n^2/(2n) \hookrightarrow \gamma(1/2 + 1/2) = \mathcal{E}(1).$$

Puis par transformation linéaire des lois exponentielles

$$D_n^2 \hookrightarrow \mathcal{E}\left(\frac{1}{2n}\right).$$

La variable D_n prend ses valeurs dans $\mathbb{R}^+$. Pour $x < 0$, $\mathbf{P}(D_n \leq x) = 0$ et pour $x \geq 0$,

$$\mathbf{P}(D_n \leq x) = \mathbf{P}\left(D_n^2 \leq x^2\right).$$

On vérifie que la fonction de répartition est continue sur $\mathbb{R}$ et de classe $\mathcal{C}^1$ sur $\mathbb{R}^*$. La variable D_n est à densité et une densité est définie que $\mathbb{R}$ par

$$f_{D_n}(x) = 2xf_{D_n^2}(x^2) = \begin{cases} \frac{x}{n} e^{-\frac{x^2}{2n}} \text{ si } x > 0 \\ 0 \text{ sinon.} \end{cases}$$

Par le théorème de transfert et sous réserve de convergence (absolue) de l'intégrale :

$$\mathbf{E}(D_n) = \mathbf{E}\left(\sqrt{D_n^2}\right) = \int_0^{+\infty} \sqrt{t} \times \frac{1}{2n} e^{-\frac{t}{2n}} dt.$$

Il y a bien convergence absolue et le changement de variable $u = \frac{t}{2n}$ donne :

$$\mathbf{E}(D_n) = \sqrt{2n} \int_0^{+\infty} \sqrt{u} e^{-u} du = \sqrt{2n} \Gamma(3/2)$$

Comme $\Gamma(3/2) = \frac{1}{2} \Gamma(1/2)$, il vient finalement :

$$\mathbf{E}(D_n) = \sqrt{\frac{n\pi}{2}}.$$

3.b) Le point M_n est dans le disque unité si et seulement si $[D_n^2 \leq 1]$ est réalisé. Ainsi, la connaissance de la fonction de répartition de D_n^2 donne :

$$\mathbf{P}(D_n \leq 1) = \mathbf{P}\left(D_n^2 \leq 1\right) = 1 - e^{-\frac{1}{2n}}.$$

Exercice 2

p. 1

RVA62

1. Pour $k = n$, la relation est directement vérifiée car

$$\mathbf{P}(E_0) = 1.$$

Soit $k \in \llbracket 0; n-1 \rrbracket$. D'après la définition de la probabilité conditionnelle

$$\mathbf{P}(A_k) = \mathbf{P}([S_k = 0]) \cdot \mathbf{P}_{[S_k=0]}(A_k)$$

Or on a aussi

$$\mathbf{P}_{[S_k=0]}(A_k) = \mathbf{P}_{[S_k=0]}\left(\bigcap_{i=k+1}^n [S_i \neq 0]\right).$$

Tout se passe comme si on recommençait au début à partir du temps k au lieu de 0. Ainsi si on pose

$$\tilde{S}_0 = S_k - S_{k1}, \quad \tilde{S}_1 = S_{k+1} - S_k, \dots, \tilde{S}_p = S_{k+p} - S_k$$

$\tilde{S}_p$ a la même loi que S_{k+p} et

$$\begin{aligned} \mathbf{P}_{[S_k=0]}(A_k) &= \mathbf{P}\left(\prod_{i=k+1}^n [\tilde{S}_{i-k} \neq 0]\right) \\ &= \mathbf{P}\left(\prod_{i=1}^{n-k} [S_i \neq 0]\right) \\ &= \mathbf{P}([T=0] \cup [T > n-k]) \\ &= \mathbf{P}(E_{n-k}) \end{aligned}$$

D'où le résultat.

2. Le système $(A_0, A_1, A_2 \dots A_n)$ est un système complet d'événements donc

$$\sum_{k=0}^n \mathbf{P}(A_k) = 1$$

D'où le résultat d'après la question précédente.

3. Soit $x \in [0; 1[$. Posons pour tout $k \in \mathbb{N}$.

$$x_k = \mathbf{P}(S_k = 0) x^k \quad \text{et} \quad y_k = \mathbf{P}(E_k) x^k.$$

On a pour tout $k \in \mathbb{N}$

$$|x_k| \leq x^k \quad \text{et} \quad |y_k| \leq x^k.$$

Par comparaison à une série géométrique convergente (la raison appartient à $] -1; 1[$) les séries $\sum x_k$ et $\sum y_k$ sont convergentes (et à termes positifs). D'après le résultat admis, la série $\sum z_n$ avec

$$z_k = \sum_{n=0}^{\infty} x_k y_{n-k}$$

est convergente et la somme vaut

$$\sum_{n=0}^{\infty} z_n = \sum_{k=0}^{\infty} x_k \cdot \sum_{k=0}^{\infty} y_k.$$

Or d'après la question précédente, on a aussi

$$z_n = \sum_{k=0}^n \mathbf{P}(S_k = 0) \mathbf{P}(E_{n-k}) x^{n-k} \cdot x^k = x^n$$

et par les séries géométriques

$$\sum_{n=0}^{\infty} z_n = \frac{1}{1-x}$$

D'où le résultat.

4. Posons Y_n le nombre de fois où la particule va à droite au bout de n étapes. En particulier $n - Y_n$ correspond au nombre de fois où la particule va à gauche. On a

$$\begin{aligned} S_n &= (-1) \times (n - Y_n) + Y_n \\ &= 2Y_n - n. \end{aligned}$$

De plus, Y_n suit une loi binomiale de paramètre (n, p) . D'où

$$\forall k \in [0; n], \quad \mathbf{P}(Y_k = k) = \binom{n}{k} \frac{1}{2^n}.$$

Ainsi

$$\begin{aligned} \mathbf{P}(S_n = 0) &= \mathbf{P}(2Y_n - n = 0) \\ &= \mathbf{P}\left(Y_n = \frac{n}{2}\right). \end{aligned}$$

Si n est impair, c'est impossible et

$$\mathbf{P}(S_n = 0) = 0.$$

Si n est pair

$$\mathbf{P}(S_n = 0) = \binom{n}{n/2} \cdot \frac{1}{2^n}.$$

5. En ne gardant que les termes d'indice pair

$$\begin{aligned} \sum_{n=0}^{+\infty} \mathbf{P}(S_n = 0) x^n &= \sum_{p=0}^{+\infty} \mathbf{P}(S_{2p} = 0) x^{2p} \\ &= \sum_{p=0}^{+\infty} \binom{2p}{p} \cdot \frac{1}{2^p} x^{2p} \\ &= \sum_{p=0}^{+\infty} \frac{\binom{2p}{p}}{4^p} (x^2)^p \end{aligned}$$

$$\sum_{n=0}^{+\infty} \mathbf{P}(S_n = 0) x^n = \frac{1}{\sqrt{1-x^2}}.$$

D'après le résultat admis avec $x^2 \in [0; 1[$. Enfin, par la question précédente

$$\begin{aligned} \sum_{n=0}^{+\infty} \mathbf{P}(E_n) x^n &= \frac{\sqrt{1-x^2}}{1-x} = \frac{\sqrt{1-x}\sqrt{1+x}}{(\sqrt{1-x})^2} \\ &= \sqrt{\frac{1+x}{1-x}}. \end{aligned}$$

6. Pour tout $n \in \mathbb{N}$

$$[T=0] \subset E_n$$

Par croissance de la probabilité

$$0 \leq \mathbf{P}(T=0) \leq \mathbf{P}(E_n).$$

Puis pour tout $x \in [0; 1[$

$$0 \leq \mathbf{P}(T=0) x^n \leq \mathbf{P}(E_n) x^n$$

par sommation

$$0 \leq \mathbf{P}(T=0) \sum_{n=0}^{+\infty} x^n \leq \sum_{n=0}^{+\infty} \mathbf{P}(E_n) x^n.$$

On simplifie

$$0 \leq \mathbf{P}(T=0) \cdot \frac{1}{1-x} \leq \sqrt{\frac{1+x}{1-x}}$$

$$\text{et} \quad 0 \leq \mathbf{P}(T=0) \leq \sqrt{(1+x)(1-x)} = \sqrt{1-x^2}.$$

Ce résultat étant valable pour tout $x \in [0; 1[$. On a par passage à la limite $x \rightarrow 1^-$

$$\mathbf{P}(T=0) = 0.$$

7.a) Procédons par récurrence

$$\mathcal{P}(n): \quad \varphi^{(n)}(x) = \frac{(2n)!}{4^n n!} (1-x)^{-\frac{2n+1}{2}}$$

- $\mathcal{P}(0)$ est directement vérifiée.
 → Soit $n \in \mathbb{N}$. Supposons $\mathcal{P}(n)$ vraie.

$$\begin{aligned}\varphi^{(n+1)}(x) &= \left(\varphi^{(n)}\right)'(x) \\ &= \frac{(2n)!}{4^n n!} \cdot \frac{2n+1}{2} (1-x)^{-\frac{2n+1}{2}-1} \\ &= \frac{(2n)!}{4^n n!} \frac{(2n+1)(2n+2)}{2 \cdot 2 \cdot (n+1)} (1-x)^{-\frac{2(n+1)+1}{2}} \\ &= \frac{(2(n+1))!}{4^{n+1} (n+1)!} (1-x)^{-\frac{2(n+1)+1}{2}}.\end{aligned}$$

$\mathcal{P}(n+1)$ est vérifiée.
 Pour tout $n \in \mathbb{N}$, $\mathcal{P}(n)$ est vraie.

7.b) C'est une application de la formule de Taylor avec reste intégral

$$\varphi(x) = \sum_{k=0}^n \frac{\varphi^{(k)}(0)}{k!} x^k + \int_0^x \frac{(x-t)^n}{n!} \varphi^{(n+1)}(t) dt$$

On obtient le résultat en remplaçant.

7.c) On peut procéder par récurrence mais il y a plus rapide avec la formule du binôme

$$\binom{2n+2}{n+1} \leq \sum_{k=0}^{2n+2} \binom{2n+2}{k} \leq (1+1)^{2n+2} = 4^{n+1}.$$

7.d) Soit $(x, t) \in \mathbb{R}^2$ avec $0 \leq t \leq x \leq 1$. Comme $x-t \geq 0$, $1-t \geq 0$, on a bien par quotient

$$\frac{x-t}{1-t} \geq 0.$$

Puis

$$x - \frac{x-t}{1-t} = \frac{x(1-t) - x + t}{1-t} = \frac{t(1-x)}{1-t}$$

et de nouveau, par quotient de facteurs positifs

$$\frac{t(1-x)}{1-t} \geq 0 \quad \text{puis} \quad x \geq \frac{x-t}{1-t}.$$

7.e) D'après ce qui précède pour $x \in [0; 1[$ et $t \in [0; x]$

$$\begin{aligned}& \left| \frac{(x-t)^n}{n!} \varphi^{(n+1)}(t) \right| \\ & \leq \frac{x^n (1-t)^n}{n!} \left| \varphi^{(n+1)}(t) \right| \\ & \leq \frac{x^n (1-t)^n}{n!} \frac{(2(n+1))!}{4^{n+1} (n+1)!} (1-t)^{-\frac{2n+3}{2}} \\ & \leq (n+1) \frac{x^n (1-t)^n}{4^{n+1}} \binom{2n+2}{n+1} (1-t)^{-n+3/2} \\ & \leq \frac{(n+1)x^n (1-t)^n}{(1-t)^n} \sqrt{1-t} \\ & \leq \frac{(n+1)x^n}{\sqrt{1-t}^3}.\end{aligned}$$

Par croissance de l'intégrale

$$\left| \int_0^x \frac{(x-t)^n}{n!} \varphi^{(n+1)}(t) dt \right| \leq (n+1)x^n \int_0^x \frac{1}{\sqrt{1-t}^3} dt$$

Or, par les croissances comparées, le membre de droite tend vers 0 lorsque n tend vers 0. Par encadrement, on a le

résultat demandé.

7.f) Il suffit de passer à la limite dans la relation de la question 7.b).

Exercice 3

p. 3

CVA60

1. On peut aussi définir les variables $(X_i)_{i \in \mathbb{N}}$, $(Y_i)_{i \in \mathbb{N}}$ par récurrence. Pour tout $i \in \mathbb{N}^*$

$$X_i = X_{i-1} + \cos(\Theta_i) \quad \text{et} \quad Y_i = Y_{i-1} + \sin(\Theta_i).$$

Un code possible est alors :

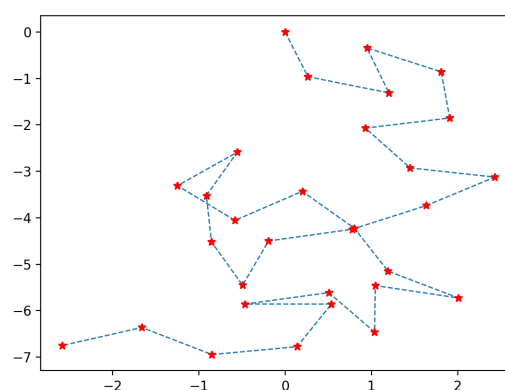
```
def simuAn(n):
    A=np.zeros([2,n+1])
    for i in range(1,n+1):
        theta=2*np.pi*rd.random()
        A[0,i]=A[0,i-1]+np.cos(theta)
        A[1,i]=A[1,i-1]+np.sin(theta)
    return A
```

On rappelle que la commande $A[i, :]$ permet de récupérer la ligne d'indice $i+1$ de la matrice A . On en déduit le code pour une trajectoire :

```
n=30
A=simuAn(n)
X=A[0,:]
Y=A[1,:]
plt.plot(X,Y)
plt.show()
```

Une seconde version pour un meilleur affichage :

```
n=30
A=simuAn(n)
X=A[0,:]
Y=A[1,:]
plt.plot(X,Y,'--',linewidth=1)
plt.plot(X,Y,'*',color='red')
plt.show()
```



2. Par exemple :

```
def simuDn(n):
    A=simuAn(n)
    B=np.zeros(n+1)
    for i in range(1,n+1):
        B[i]=np.sqrt(A[0,i]**2+A[1,i]**2)
    return B
```

3. Soit $\Theta \hookrightarrow \mathcal{U}([0; 2\pi])$. On a par la formule de transfert, l'existence de l'espérance de $\cos(\Theta)$ et l'égalité

$$\mathbf{E}(\cos(\Theta)) = \frac{1}{2\pi} \int_0^{2\pi} \cos(t) dt = 0.$$

Ensuite par linéarité de l'espérance, X_n admet une espérance et

$$\mathbf{E}(X_n) = \sum_{i=1}^n \mathbf{E}(\cos(\Theta_i)) = 0.$$

Le calcul est similaire pour Y_n , on trouve $\mathbf{E}(Y_n) = 0$.

• De nouveau, par la formule de transfert, on a l'existence du moment d'ordre 2 de $\cos(\Theta)$ et l'égalité

$$\begin{aligned} \mathbf{E}(\cos(\Theta)^2) &= \frac{1}{2\pi} \int_0^{2\pi} \cos(t)^2 dt \\ &= \frac{1}{2\pi} \int_0^{2\pi} \frac{\cos(2t) + 1}{2} dt = \frac{1}{2}. \end{aligned}$$

Ensuite, par indépendance des variables $(\cos(\Theta_i))_i$ (lemme des coalitions), on a

$$\mathbf{V}(X_n) = \sum_{i=1}^n \mathbf{V}(\cos(\Theta_i)) = \frac{n}{2}.$$

On trouve aussi $\mathbf{V}(Y_n) = n/2$.

4.a) La fonction python `truc` prend en argument n et renvoie une simulation de la matrice aléatoire

$$T = \begin{bmatrix} 0 & X_1 Y_1 & X_2 Y_2 & \dots & X_n Y_n \end{bmatrix}.$$

La fonction `machin` prend aussi en argument n et somme $m = 10000$ réalisation de la matrice ligne aléatoire T avant de diviser par m chaque coefficient. La matrice obtenue est alors une matrice ligne dont le coefficient d'indice i correspond à une moyenne arithmétique de m réalisations de la variable $X_i Y_i$. D'après la loi faible des grands nombres, le résultat obtenu est proche (avec une grande probabilité) de

$$T = \begin{bmatrix} 0 & \mathbf{E}(X_1 Y_1) & \mathbf{E}(X_2 Y_2) & \dots & \mathbf{E}(X_n Y_n) \end{bmatrix}.$$

D'après les résultats obtenus, on conjecture que

$$\forall i \in \mathbb{N}, \quad \mathbf{E}(X_i Y_i) = 0.$$

• Prouvons ce résultat :

$$\mathbf{E}(X_n Y_n) = \sum_{i=1}^n \sum_{j=1}^n \mathbf{E}(\cos(\Theta_i) \cos(\Theta_j)).$$

Distinguons deux cas :

→ Si $i \neq j$, alors les variables $\cos(\Theta_i)$ et $\cos(\Theta_j)$ sont indépendantes et

$$\mathbf{E}(\cos(\Theta_i) \cos(\Theta_j)) = \mathbf{E}(\cos(\Theta_i)) \mathbf{E}(\cos(\Theta_j)) = 0.$$

→ Si $i = j$, on applique la formule de transfert,

$$\mathbf{E}(\cos(\Theta_i) \cos(\Theta_i)) = \frac{1}{2\pi} \int_0^{2\pi} \cos(t) \sin(t) dt.$$

Or, on sait que

$$2 \cos(t) \sin(t) = \sin(t + t) = \sin(2t).$$

On en déduit alors que l'espérance est nulle.

Finalement,

$$\mathbf{E}(X_n Y_n) = 0.$$

Notons que par la formule de Huygens, on aurait aussi

$$\text{Cov}(X_i, Y_i) = \mathbf{E}(X_i Y_i) - \mathbf{E}(X_i) \mathbf{E}(Y_i) = 0.$$

4.b) Soit $n \in \mathbb{N}$

$$\begin{aligned} \mathbf{E}(D_n^2) &= \mathbf{E}(X_n^2) + \mathbf{E}(Y_n^2) \\ &= \mathbf{V}(X_n) + \mathbf{V}(Y_n) \end{aligned}$$

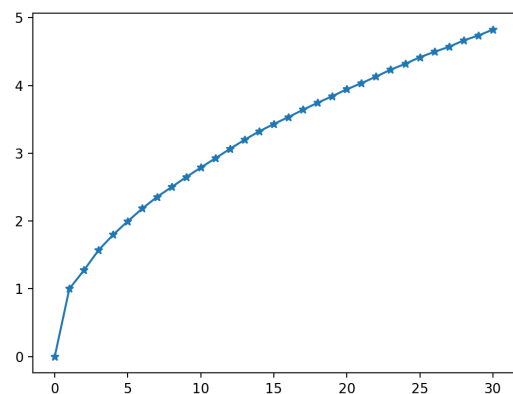
$$\mathbf{E}(D_n^2) = \frac{n}{2} + \frac{n}{2} = n.$$

5.a)

```
def esperanceD(n):
    m=10000
    B=np.zeros(n+1)
    for i in range(m):
        B+=simuDn(n)
    return B/m
```

Et pour l'affichage :

```
plt.clf()
B=esperanceD(n)
plt.plot(np.arange(n+1), B, 'x-')
plt.show()
```



5.b) Posons pour tout $k \in \mathbb{N}^*$

$$x_k = \ln(k) \quad \text{et} \quad y_k = \ln(\mathbf{E}(D_k)).$$

Le programme python trace les points (x_k, y_k) pour $k \in \llbracket 4; 40 \rrbracket$. On constate que les points se répartissent (approximativement) sur une droite. Or, si $\mathbf{E}(D_k) = \beta k^\alpha$ alors

$$\ln(\mathbf{E}(D_k)) \approx \alpha \ln(k) + \ln(\beta)$$

i.e $y_k = \alpha x_k + \ln(\beta)$.

On peut donc identifier α avec le coefficient directeur de la droite. Or on constate que ce coefficient est proche de $1/2$. On conjecture donc que

$$\alpha = \frac{1}{2}.$$

6. C'est une application du théorème central limite.

7.

8. Posons $S = \sqrt{Z^2 + T^2}$.

$$\begin{aligned} E(S) &= \int_{-\infty}^{+\infty} t g(t) dt \\ &= \frac{2}{n} \int_0^{+\infty} t^2 e^{-t^2/n} dt \end{aligned}$$

On peut calculer cette intégrale par double intégrations par parties. Donnons une autre méthode.

Effectuons le changement affine $u = \sqrt{\frac{2}{n}}t$, $du = \sqrt{\frac{2}{n}}dt$

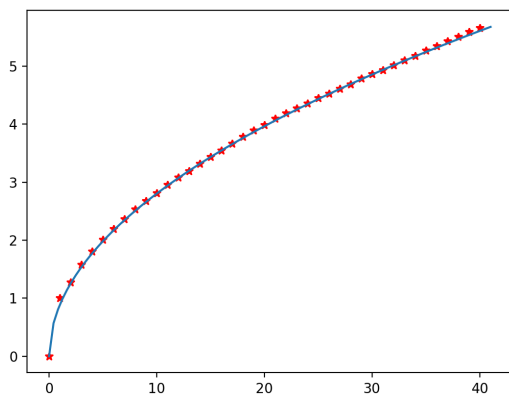
$$\begin{aligned} E(S) &= \sqrt{\frac{2}{n}} \int_0^{+\infty} \left(\frac{n}{2}u^2\right) e^{-u^2/2} du \\ &= \sqrt{\frac{n}{2}} \int_0^{+\infty} u^2 e^{-u^2/2} du \\ &= \sqrt{\frac{1}{2}} \cdot \frac{1}{2} \int_{-\infty}^{+\infty} u^2 e^{-u^2/2} du \quad (\text{parité}) \\ &= \sqrt{\frac{n}{8}} \cdot \sqrt{2\pi} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} u^2 e^{-u^2/2} du \\ &= \frac{\sqrt{n\pi}}{2} E(W^2) \quad \text{où } W \hookrightarrow \mathcal{N}(0,1) \\ E(S) &= \frac{\sqrt{n\pi}}{2}. \end{aligned}$$

On peut identifier

$$\alpha = \frac{1}{2} \quad \text{et} \quad \beta = \frac{\sqrt{\pi}}{2}.$$

Terminons par une vérification pythonesque :

```
plt.clf()
n=40
B=esperanceD(n)
plt.plot(np.arange(n+1),B,'*',color='red')
x=np.linspace(0,n+1,100)
y=np.pi**(1/2)/2*np.sqrt(x)
plt.plot(x,y,'-')
plt.show()
```



C'est bon!

Exercice 4

p. 4

PY92

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d #
    Fonction pour la 3D
plt.clf()
axes = plt.axes(projection="3d")

# Le pôle nord
axes.plot([0],[0],[1],',',color='black',
    linewidth=5)
# L'équateur
t = np.linspace(0,2*np.pi, 50)
axes.plot(np.cos(t),np.sin(t),np.zeros(50),
    color='red',linewidth=1)
# Les latitudes
r=np.linspace(0,2*np.pi,20)
for i in range(20):
    x = np.cos(r[i])*np.cos(t)
    y = np.cos(r[i])*np.sin(t)
    z = np.sin(r[i])*np.ones(50)
    axes.plot(x, y, z, '--',color='grey',
        linewidth=0.5)
plt.axis('off')
#plt.show()
```

```
import numpy.random as rd
def Rotation1(theta):
    M=np.eye(3)
    M[1,1]=np.cos(theta)
    M[2,2]=M[1,1]
    M[2,1]=np.sin(theta)
    M[1,2]=-M[2,1]
    return M
```

```
def Rotation2(theta):
    M=np.eye(3)
    M[0,0]=np.cos(theta)
    M[1,1]=M[0,0]
    M[1,0]=np.sin(theta)
    M[0,1]=-M[1,0]
    return M
```

```
def Rotation3(theta):
    M=np.eye(3)
    M[0,0]=np.cos(theta)
    M[2,2]=M[0,0]
    M[2,0]=np.sin(theta)
    M[0,2]=-M[2,0]
    return M
```

```
def rotation(h):
    p=rd.random()
    theta=rd.random()*h
    if p<(1/3):
        return Rotation1(theta)
    if p>(2/3):
        return Rotation3(theta)
    return Rotation2(theta)
```

```
h=0.1
n=2000
```

```

X=np.array([[0],[0],[1]])
x=np.zeros(n+1)
y=np.zeros(n+1)
z=np.zeros(n+1)
z[0]=1
for i in range(1,n+1):
    X=np.dot(rotation(h),X)
    x[i]=X[0,0]
    y[i]=X[1,0]
    z[i]=X[2,0]
axes.plot(x, y, z)
plt.show()

h=0.05

```

```

X=np.array([[0],[0],[1]])
x=[0]
y=[0]
z=[1]
while z[-1]>0:
    X=np.dot(rotation(h),X)
    x.append(X[0,0])
    y.append(X[1,0])
    z.append(X[2,0])

theta=(2*rd.random()-1)*h

```