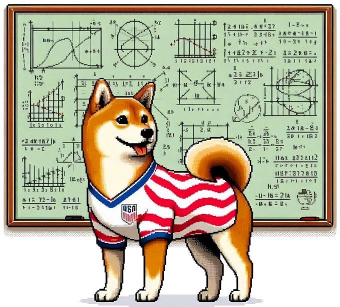

Préparation à l'oral

Math-Info : Math 2
Semaine du 8 Juin

Sujet OB-MATH2-2

Exercice de mathématiques

Soit X une v.a.d. finie qui prend comme valeurs $x_1 < \cdots < x_r$ avec les probabilités :

$$\mathbf{P}\left(X = x_i\right) = p_i \in]0,1[, \forall i \in [\![1, r]\!].$$

Soit $L_X : t \longmapsto \ln\left(\mathbf{E}\left(\mathrm{e}^{tX}\right)\right)$.

1. Montrer que L_X est définie et $\mathscr{C}^\infty$ sur $\mathbb{R}$.
2. Montrer que L'_X admet des limites en $\pm\infty$ et les déterminer.
3. Prouver l'existence de coefficients $c_0(X), c_1(X), c_2(X), c_3(X)$ tels que

$$L_X(t) = \sum_{k=0}^{3} \frac{c_k(X)}{k!} t^k + o\left(t^3\right)$$

4. Soit Y une v.a.d. finie indépendante de X. Montrer que $c_k(X+Y) = c_k(X) + c_k(Y)$.
5. Calculer $c_k(X)$ pour $k = 0, 1, 2, 3$.

Exercice d'informatique

1. Écrire une fonction `U(N, a)` de deux arguments, un entier naturel N et un nombre a, qui renvoie le vecteur $(1^a, 2^a, \cdots, N^a)$.
2. Faire tracer sur un même graphique n^a en fonction de n, pour n entier de 1 à 30 et pour a valant $0.0, -0.5$ et -1.0.
3. Observer dans la console ce que donnent les instructions :

```
import numpy as np
np.eye(4)
np.diag([1,2,3],k = -1)
```

En déduire une fonction `M(V)` d'argument un vecteur $V = (v_1, v_2, \cdots, v_{n-1}, v_n)$ qui renvoie la matrice :

$$\begin{pmatrix} 1 & -v_1 & 0 & \cdots & 0 & 0 & 0 \\ v_1 & 1 & -v_2 & \ddots & \ddots & \ddots & \vdots \\ 0 & v_2 & \ddots & \ddots & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 & -v_{n-2} & 0 & \vdots \\ \vdots & \ddots & 0 & v_{n-2} & 1 & -v_{n-1} & 0 \\ \vdots & \ddots & \ddots & 0 & v_{n-1} & 1 & -v_n \\ 0 & 0 & \cdots & 0 & 0 & v_n & 1 \end{pmatrix}$$

En utilisant la fonction `det` du module `numpy.linalg`, faire afficher le déterminant de cette matrice pour $\mathtt{V} = \mathtt{U}(n, a)$, pour $a = -0.5$ et $a = -1.0$ et pour n prenant ses valeurs dans $\{10, 50, 200, 1000\}$.

4. On **admet** que si l'on pose $a_n = \det(\mathrm{M}((v_1, v_2, \cdots, v_{n-1}, v_n)))$, la suite ainsi définie vérifie la récurrence :

$$a_0 = 1,\ a_1 = 1 + v_1^2 \quad \text{et} \quad \forall n \in \mathbb{N},\ n \geqslant 2,\ a_n = a_{n-1} + v_n^2 a_{n-2}$$

et cette suite est toujours croissante et à valeurs supérieures à 1. En déduire une fonction `A(V)` d'argument un vecteur $V = (v_1, v_2, \ldots, v_{n-1}, v_n)$ qui renvoie la valeur de a_n. Vérifier que l'on retrouve bien numériquement les mêmes valeurs qu'à la Question **3.**

5. Lorsque pour tout entier naturel non nul $n, v_n = 1/\sqrt{n}$, déterminer le plus petit entier $n_{\min}$ tel que, pour tout entier naturel n supérieur à $n_{\min}$, $a_n \geqslant 1000$.
Faire tracer a_n/n en fonction de n pour n variant de 1 à 10000 . Quelle conjecture peut-on faire lorsque n tend vers l'infini?
6. Vérifier numériquement sur les premières valeurs de n que :

   - ✗ si $v_n = 1$ pour tout n, alors $a_n = \left(1 + \frac{2}{\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^{n-1} + \left(1 - \frac{2}{\sqrt{5}}\right)\left(\frac{1-\sqrt{5}}{2}\right)^{n-1}$;
   - ✗ si $v_n = n$ pour tout n, alors $a_n = (n+1)\,!$.

Solution du sujet OB-MATH2-2

Math (Math II 2024, 2025)

1. Par le théorème de transfert, la série génératrice de X est en fait une somme finie d'exponentielles strictement positive. Plus précisément, pour tout $t \in \mathbb{R}$

$$\mathbf{E}\left(\mathrm{e}^{tX}\right) = \sum_{i=1}^{r} p_i e^{x_i t} > 0$$

On peut alors composer par le log. Les fonction $t \mapsto e^{x_i t}$ étant de classe $\mathscr{C}^\infty$ à valeurs strictement positives, la fonction L_X est bien définie et de classe $\mathscr{C}^\infty$ sur $\mathbb{R}$.

2. On a, pour tout $t \in \mathbb{R}$,

$$L'_X(t) = \frac{\sum_{i=1}^{r} x_i p_i e^{x_i t}}{\sum_{i=1}^{r} p_i e^{x_i t}} = \frac{e^{x_r t}}{e^{x_r t}} \left(\frac{x_r p_r + \sum_{i=1}^{r-1} x_i p_i e^{(x_i - x_1)t}}{p_r + \sum_{i=1}^{r-1} p_i e^{(x_i - x_1)t}} \right) \underset{t \to +\infty}{\longrightarrow} x_r$$

(car $(x_i - x_r) < 0$ si $1 \leq r - 1$).
De même en $-\infty$,

$$L'_X(t) = \frac{\sum_{i=1}^{r} x_i p_i e^{x_i t}}{\sum_{i=1}^{r} p_i e^{x_i t}} = \frac{x_1 p_1 + \sum_{i=2}^{r} x_i p_i e^{(x_i - x_1)t}}{p_1 + \sum_{i=2}^{r} p_i e^{(x_i - x_1)t}} \underset{t \to -\infty}{\longrightarrow} x_1.$$

3. L_X est de classe $\mathscr{C}^\infty$ au voisinage de 0. Elle possède donc un développement limité à tout ordre en 0, et en particulier d'ordre 3. On peut même affirmer que

$$\forall k \in \mathbb{N}, \qquad c_k(X) = L_X^{(k)}(0).$$

4. Si X et Y sont indépendantes, il en est de même pour e^{tX} et e^{tY} (et ce, pour tout $t \in \mathbb{R}$), il suit que

$$\forall t \in \mathbb{R}, \qquad L_{X+Y}(t) = \ln\left(\mathbf{E}(e^{t(X+Y)})\right) = \ln\left(\mathbf{E}(e^{tX})\mathbf{E}(e^{tY})\right) = \ln(\mathbf{E}(e^{tX})) + \ln(\mathbf{E}(e^{tY})) = L_X(t) + L_Y(t).$$

Par conséquent

$$c_k(X+Y) = L_{X+Y}^{(k)}(0) = (L_X + L_Y)^{(k)}(0) = L_X^{(k)}(0) + L_Y^{(k)}(0) = c_k(X) + c_k(Y).$$

5. On fait les différents calculs

$$\begin{aligned}
c_0(X) &= \ln\left(\sum_{i=1}^{r} p_i\right) = \ln(1) = 0 \\
c_1(X) &= \frac{\sum_{i=1}^{r} p_i x_i}{\sum_{i=1}^{r} p_i} = \mathbf{E}(X) \\
c_2(X) &= \frac{\left(\sum_{i=1}^{r} p_i x_i^2\right)\left(\sum_{i=1}^{r} p_i\right) - \left(\sum_{i=1}^{r} p_i x_i\right)^2}{\left(\sum_{i=1}^{r} p_i\right)^2} = \mathbf{E}(X^2) - \mathbf{E}(X)^2 = \mathbf{V}(X) \\
c_3(X) &= \frac{\left(\sum_{i=1}^{r} p_i x_i^3\right)\left(\sum_{i=1}^{r} p_i\right)^2 + 3\left(\sum_{i=1}^{r} p_i x_i^2\right)\left(\sum_{i=1}^{r} p_i x_i\right)\left(\sum_{i=1}^{r} p_i\right) - 2\left(\sum_{i=1}^{r} p_i x_i\right)^3}{\left(\sum_{i=1}^{r} p_i\right)^3} \\
&= \mathbf{E}(X^3) + 3\mathbf{E}(X^2)\mathbf{E}(X) - 2\mathbf{E}(X)^3.
\end{aligned}$$

Info (Math II 2024)

1.
```
def U(n,a):
    assert n != 0
    v = np.zeros(n)
    for i in range(n):
        v[i] = (i+1)**a
    return v
```

2.
```
N=30
X = np.arange(1, N+1)
for a in [0.0,)-0.5,-1]:
    plt.plot(X,U(30, N))
plt.show()
```

3.
```
np.eye(4)
np.diag([1,2,3],k = -1)

def M(v):
    n = len(v)
    res = np.eye(n+1)+ np.diag(v,k = -1) + np.diag(-v,k = +1)
    return res

L_a = [-0.5, -1] # Liste des a
L_n = [10,50,200,1000] # Liste des n
for a in L_a:
    for n in L_n:
        v = U(n,a)
        det = np.linalg.det(M(U(n,a)))
        print('det U(',n,',',a,')=',det)
```

4.
```
def A(v):
    n = len(v)
    a = 1
    b = 1+v[0]**2
    for k in range(2,n+1):
        a, b = b, b + v[k-1]**2*a
    return b
# On vérifie que l'on a bien A(v) = det(M(U(n,a)) avec les n et a de la Q3
```

5.
```
n = 1
a = -0.5
v = U(n,a)
while A(v) < 10**3:
    n += 1
    v = U(n,a)
print(n) # On trouve nmin = 1580
Y = list(range(1,10001))
Z = [A(U(n,-0.5))/n for n in Y]
plt.plot(Y,Z)
plt.show()
# Le résultat obtenu laisse penser que an/n tend evrs une limite proche de 0.6
```

6.
```
# Vérifications pour n=1,..., n=5
def f(n):
    return (1+2/np.sqrt(5))*((1+np.sqrt(5))/2)**(n-1) +
                            (1-2/np.sqrt(5))*((1-np.sqrt(5))/2)**(n-1)
```

```

for n in range(1,5):
    v = np.ones(n)
    print(A(v)-f(n))

def facto(n) :
    res=1
    for k in range(1,n+1):
        res=res*k
    return res

for n in range(1,5):
    v = U(n,1)
    print(A(v)-math.facto(n+1))
```