Oral de centrale 2 avec python

I Simulations de lois de probabiltés

Exercice 1 *(centrale Psi 2015 avec python: extrait) Soit N,$X_1,\ldots,X_i,\ldots$ sont des variables aléatoires définies sur un même espace probabilisé. On suppose que N suit une loi de Poisson de paramètre $\lambda=2$ et que $(X_n)_{n\in\mathbb{N}}$ est une famille de variables aléatoires mutuellement indépendantes de même loi: $X_i \hookrightarrow \mathcal{B}\left(100,\frac{1}{3}\right)$. Ecrire une fonction qui simule la valeur de $S=\sum_{i=0}^{N} X_i$. Donner une estimation de l'espérance de S.*

```
def S():
    import numpy.random as rd
    N=rd.poisson(2,1)
    X=rd.binomial(100,1/3,N[0])
    S=0
    for x in X:
        S+=x
    return S
```

Exercice 2 *(centrale extrait) On considère N étudiants souhaitant réussir à un examen. A chaque fois qu'un étudiant passe l'examen, il a une propbabilité $p\in]0,1[$ de réussite. Tous les passages de l'examen sont indépendants. Soit X_k le nombre de passage nécessaires au candidat n° k pour réussir l'examen. On pose $X=X_1+\cdots+X_N$ et $Y=\max(X_1,\ldots,X_N)$. A l'aide de python, calculer $E(X)$ et $E(Y)$. On prendra par exemple $N=3$ et $p=\frac{1}{3}$.*

Solution de l'exercice:

```
import numpy.random as rd
def X(N,p):
    x=0
    for i in range(N):
        x=x+rd.geometric(p)
    return x

def Y(N,p):
    y=1 # (car max(Xi)>=1)
    for i in range(N):
        xi=rd.geometric(p)
        if xi>y:
            y=xi
    return y

def E(Z,N,p,nb):# Z vaudra X ou Y
    S=0
    for i in range(nb):
        S=S+Z(N,p)
    return S/nb
E(Y,3,1/3,10000) # environ 9
E(Y,3,1/3,10000) # environ 5
```

II Algèbre linéaire

Exercice 3 *(Centrale Python 2016) On note H_n l'ensemble des matrices de $\mathcal{M}_n(\mathbb{R})$ à coefficients dans $\{-1,1\}$. dont les colonnes sont orthogonales pour le produit scalaire canonique et SH_n, l'ensemble des matrices symétriques de H_n.Soit A une matrice à coefficients dans $\{-1,1\}$. Montrer que $A\in H_n \Leftrightarrow A\times A^T=nI_n$.*
Que renvoie l'instruction:`[[x,y] for x in [0,1]for y in [0,1]]`?
Ecrire une suite d'instructions qui donne l'ensemble des matrices de H_4. Quel est le cardinal de H_4. Même question

avec SH_4.
Trouver $\{tr(A),\ A \in SH_4\}$. *Le démontrer sans python.*

Solution de l'exercice:

```
import numpy as np
L=[[x,y,z,t] for x in [-1,1] for y in [-1,1] for z in [-1,1] for t in [-1,1]]
M=[[L1,L2,L3,L4] for L1 in L for L2 in L for L3 in L for L4 in L]
M=np.array(M,dtype=int)
def In(n):
    I=np.zeros((n,n),dtype=int)
    for i in range(n):
        I[i,i]=1
    return I\newline{}# on peut aussi utilises eye ou diag
def est_dans_Hn(m):
    n=len(m)
    M=np.dot(m,np.transpose(m))
    P=n*In(n)
    res=True
    for i in range(n):
        for j in range(n):
            if M[i,j]!=P[i,j]:
                res=False
    return res
Liste_H4=[]
for m in M:
    if est_dans_Hn(m):
        Liste_H4.append(m)

def est_symetrique(m):
    res=True
    n=len(m)
    for i in range(n):
        for j in range(n):
            if m[i,j]!=m[j,i]:
                res=False
    return res

Liste_S4=[]
for m in Liste_H4:
    if est_symetrique(m):
        Liste_S4.append(m)
t=[]
for m in Liste_S4:
    t.append(np.trace(m))
t=set(t) # conversion en ensemble
```

On trouve que $tr(A) \in \{-4, 0, 4\}$.

Si $A \in S_4$, alors $A \times A^T = nI_n$ et $A = A^T$ donc $A^2 = nI_n$. On en déduit que $X^2 - n = (X - \sqrt{n})(X + \sqrt{n})$ est un polynôme annulateur de A scindé à racine simples A est diagonalisable et $sp(A) \subset \{-\sqrt{n}, \sqrt{n}\}$. On a donc $tr(A) = p\sqrt{n} + (n-p)(-\sqrt{n}) = (2p-n)\sqrt{n}$ avec $p \in [[0, n]]$. Comme $A \neq \lambda I_n$, on ne peut pas avoir $p = 0$ ou $p = n$ donc $p \in [[1, n-1]]$. Si $n = 4$; si $p = 1$, $tr(A) = -4$, si $p = 2$, $tr(A) = 0$, si $p = 3$, $tr(A) = 4$.

Exercice 4 *(Centrale Python 2016) On considère deux matrice* M_n *et* T_n *dans* $\mathcal{M}_n(\mathbb{R})$ *avec* $(M_n)_{i,j} = \inf(i, j)$ *et* T_n *triangulaire supérieure avec des* 1 *sur le triangle supérieur, diagonale comprise.*
Ecrire un script python permettant l'affichage de M_n *pour* $2 \leq n \leq 10$. *Calculer* $\det(M_n)$ *pour* $2 \leq n \leq 10$. *Conjecturer un résultat.*
Ecrire un script python permettant l'affichage de T_n *pour* $2 \leq n \leq 10$, *puis de* $T_n^T \times T_n$.

Montrer que M_n est diagonalisable et de valeurs propres strictement positives.
Montrer que $\max_{\lambda \in sp(M_n)} \lambda \geq \frac{n+1}{2}$.
Calculer M_n^{-1} en essayant de minimiser les calculs. Utiliser python pour le calcul de M_n^{-1}.

Solution de l'exercice:

```
import numpy as np
def Mat(n):
    M=np.zeros((n,n),dtype=int)
    for i in range(n):
        for j in range(n):
            M[i][j]=min(i+1,j+1)
    return M
for i in range(2,11):
    print(Mat(i)) #affichage des matrices
from numpy.linalg import det #importation de la fonction det de numpy.linalg
Mat(4)
liste_det=[]
for i in range(2,11):
    liste_det.append(det(Mat(i)))
print(liste_det) # les determinants valent 1
def T(n):
    M=np.zeros((n,n),dtype=int)
    for i in range(n):
        for j in range(n):
            if i<=j:
                M[i][j]=1
    return M
for i in range(2,11):
    print(T(i))
for i in range(2,11):
    M=T(i)
    print(np.dot(np.transpose(M),M)) # t(Tn)xTn=Mn pour 2<=n<=10
```

III Résolution d'équations

Exercice 5 *(centrale Psi 2015 avec python: extrait):*

Tester les commandes et expliquer

```
from scipy.optimize import fsolve
def f(x):
    return 2*x[0]**2+3*x[1]-11,3*x[0]-2*x[0]*x[1]-2
fsolve(f,[0,0])
fsolve(f,[1,1])
fsolve(f,[2,1])
```

Déterminer une matrice U orthogonale et une matrice S symétrique vérifiant $SU = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$.

Solution de l'exercice:

Pour chaque application de fsolve, on obtient une valeur approchée d'un couple de solution du système $\begin{cases} 2x^2+3y=11 \\ 3x-2xy=2 \end{cases}$

```
def f(x):
    e1=x[0]**2+x[2]**2-1
    e2=x[2]*x[3]+x[0]*x[1]
    e3=x[1]**2+x[3]**2-1
    e4=x[5]-x[6]
```

```
    e5=x[0]*x[4]+x[1]*x[5]-1
    e6=x[0]*x[6]+x[1]*x[7]-1
    e7=x[2]*x[4]+x[3]*x[5]    e8=x[2]*x[6]+x[3]*x[7]-1
    return e1,e2,e3,e4,e5,e6,e7,e8


fsolve(f,[0,0,0,0,0,0,0,0])
```

IV Intégration, équations différentielles, séries entières, graphiques

Exercice 6 *On pose* $f_p(x) = \int_0^1 \sqrt{x^2+t^p}dt$. *Représenter graphiquement les fonctions* f_i *sur un même graphique pour* $i \in [|0,5|]$.

Attention 1 *la difficulté ici est que la fonction argument de quad doit être une fonction de une variable.*

```
import matplotlib.pyplot as plt
from scipy.integrate import quad
X=np.linspace(0,2,100)
for i in range(1,7):
    Lf=[]
    for x in X:
        def g(t): # fonction d'une seule variable pour utiliser quad
                  #(on travaille dans l'environnmement des boucles for donc a i et x fixes)
            return np.sqrt(x**2+t**i)
        Lf.append(quad(g,0,1))
    plt.plot(X,Lf)
```

Exercice 7 *(Centrale Python 2016) On considère l'équation* $(1-x)y'' = y$ *sur* $]-\infty,1[$. *Montrer qu'il existe une et une seule solution* f *de cette équation vérifiant* $f(0)=0$ *et* $f'(0)=1$. *Représenter la fonction* f *sur l'intervalle* $[-2;0,95]$. *Soit* (a_n) *la suite définie par* $a_0=0$, $a_1=1$ *et* $\forall n \in \mathbb{N}\setminus\{0,1\}$, $a_n = \dfrac{n-2}{n}a_{n-1} + \dfrac{1}{n(n-1)}a_{n-2}$. *Représenter graphiquement* a_n *pour* $n \in [0,100]$. *Montrer que le rayon de convergence de* $\sum a_n x^n$ *est supérieur ou égal à* 1. *Représenter graphiquement la fonction* $x \mapsto \sum_{n=0}^{100} a_k x^k$ *.sur* $[-2;0,95]$. *Que constate-on? Démontrer le résultat.*

Solution de l'exercice:

On a $(E) = (1-x)y'' = y$ sur $]1,+\infty[\Leftrightarrow y'' - \dfrac{1}{(1-x)}y = 0$. D'après le théorème de Cauchy, 'il existe une et une seule solution f de cette équation vérifiant $f(0)=0$ et $f'(0)=1$.

```
"""representation de la solution f"""
def f(X,t): # fonction de l'equa diff X'=f(X,t)
    return [X[1],1/(1-t)*X[0]]
from scipy.integrate import odeint
import numpy as np
import matplotlib.pyplot as plt
liste_t=np.linspace(0,0.95,100)
sol=odeint(f,[0,1],liste_t) # X[0]=[0,1]
plt.plot(liste_t,sol[:,0]) # sol array 100,2: premiere col: valeurs de y, deuxieme: valeurs de y'
liste_t=np.linspace(0,-2,100) # liste des temps de 0 \U{e0} -2 (temps parcouru en sens inverse
sol=odeint(f,[0,1],liste_t) # X[0]=[0,1]
plt.plot(liste_t,sol[:,0])
"""representation des an"""
liste_n=[n for n in range(101)]
liste_a=[0,1]
a,b=0,1
for n in range(2,101):
```

```
    c=(n-2)/n*b+a/(n*(n-1))
    liste_a.append(c)
    a,b=b,c
plt.plot(liste_n,liste_a) # on constate 0<=an<=1
```

Montrons par récurrence sur n que $0 \leq a_n \leq 1$. C'est vrai pour $n = 0, 1$.

Soit $n \geq 2$. Supposons $0 \leq a_{n-2} \leq 1$ et $0 \leq a_{n-1} \leq 1$. On a $a_n = \dfrac{n-2}{n} a_{n-1} + \dfrac{1}{n(n-1)} a_{n-2} \leq \dfrac{n-2}{n} + \dfrac{1}{n(n-1)} \leq \dfrac{n-2}{n} + \dfrac{1}{n} < 1$ car $n - 1 \geq 1$ et $a_n \geq 0$ donc $0 \leq a_n \leq 1$. Posons $b_n = 1$. La série entière $\sum b_n x^n$ est de rayon 1 et $0 \leq a_n \leq b_n$ donc le le rayon de convergence de $\sum a_n x^n$ est supérieur ou égal à 1.

```
""" representation de S100(x)
def S100(x):
    S,p=0,1
    for i in range(101):
        S=S+liste_a[i]*p
        p=p*x
    return S
liste_S=[S100(t) for t in liste_t]
plt.plot(liste_t,liste_S)
plt.plot(liste_t,sol[:,0])
```

Pour tout $n \geq 0$, $n(n-1)a_n - (n-2)(n-1)a_{n-1} + a_{n-2} = 0$ donc si $|x| < 1$, $\sum_{n=2}^{+\infty} n(n-1)a_n x^{n-2} - \sum_{n=2}^{+\infty} (n-2)(n-1)a_{n-1}x^{n-2} + \sum_{n=2}^{+\infty} a_{n-2}x^{n-2} = 0$ (toutes les séries ont même rayon que $\sum a_n x^n$). Or $\sum_{n=2}^{+\infty} n(n-1)a_n x^{n-1} = x\sum_{n=2}^{+\infty} n(n-1)a_n x^{n-2} = xf''(x)$; $\sum_{n=2}^{+\infty} (n-2)(n-1)a_{n-1}x^{n-2} = x\sum_{n=2}^{+\infty} (n-2)(n-1)a_{n-1}x^{n-3} = x\sum_{n=3}^{+\infty} n(n-1)a_n x^{n-2} = xf'(x) - a_2 = xf'(x)$ car $a_2 = 0$.et $\sum_{n=2}^{+\infty} a_{n-2}x^{n-2} = f(x)$ donc $S(x)$ vérifie l'équation (E) et comme $S(0) = a_0 = 0$ et $S'(0) = a_1 = 1$, on en déduit que $S = f$.

V Tracé de surfaces

Exercice 8 *(centrale avec python 2017) Soit $M \in \mathcal{M}_n(\mathbb{R})$. On définit $\varphi_M : \mathbb{R}^n \to \mathbb{R}$ définie par $\varphi_M(X) = {}^tXMX$. On suppose que $M = \begin{pmatrix} -2 & 4 \\ -1 & 2 \end{pmatrix}$ et $X = \begin{pmatrix} x \\ y \end{pmatrix}$. Tracer la surface d'équation $z = \varphi_M(x, y)$ pour $(x, y) \in [-2, 2]^2$.*

Solution de l'exercice:

On a $\varphi_M(x, y) = {}^tXMX = (x, y)\begin{pmatrix} -2x + 4y \\ -x + 2y \end{pmatrix} = x(-2x + 4y) + y(-x + 2y) = (2x + y)(-x + 2y) = -2x^2 + 3xy + 2y^2$.

```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
ax=Axes3D(plt.figure())
def f(x,y):
    return (2*x+y)*(-x+2*y)
X=np.arange(-2,2,0.02)
Y=np.arange(-2,2,0.02)
X,Y=np.meshgrid(X,Y)
Z=f(X,Y)
ax.plot_surface(X,Y,Z)
plt.show()
```