Math ECG 2. 2022-2023

Mathématiques Appliquées - F. Gaunard
http://frederic.gaunard.com
ENC Bessières & Lycée Carnot, Paris 17e

Oral HEC

Suggestions d'exercices

Mardi 16 Mai - Roxane D.

Exercice avec préparation

Soit a un nombre strictement positif. On définit, pour tout entier naturel n:

$$P_n = \prod_{k=0}^{n} \frac{a+k}{2a+k}.$$

(1) **(Question de cours)**. Critères de convergence des séries à termes positifs.

(2) Écrire, en Python, une fonction d'en tête `def P(n, a):` qui prend en argument $a > 0$ et $n \in \mathbb{N}$ et renvoie la valeur de P_n.

(3) Montrer que, pour tout réel $u \in]-1; +\infty[$, on a:

$$\frac{u}{1+u} \leq \ln(1+u) \leq u.$$

(4) En déduire que

$$\exp\left(-\sum_{k=0}^{n} \frac{a}{a+k}\right) \leq P_n \leq \exp\left(-\sum_{k=0}^{n} \frac{a}{2a+k}\right)$$

(5) Montrer qu'alors la suite (P_n) est convergente et déterminer sa limite.

(6) À l'aide d'une comparaison série/intégrale, déterminer la nature de de la série de terme général P_n pour $a = 1$ et $a = 2$. On précisera la somme en cas de série convergente.

(7) Dans une urne, on dispose initialement une boule rouge et une boule noire (indiscernables au toucher). On tire ensuite une boule: si elle est noire, on arrête; si elle est rouge, on la remplace dans l'urne par deux boules rouges, et on recommence jusqu'à obtention de la boule noire. Que peut-on dire du nombre moyen de tirages effectués? Et si on était parti avec deux boules de chaque couleur?

(8) Comment illustrer la réponse à la toute dernière question en Python?

Solution.

(1) (Question de cours). Soient (u_n) et (v_n) deux suites de termes positifs. On dispose des critères suivants.

- Comparaison algébrique. On suppose que $u_n \leq v_n$ (à partir d'un certain rang).
  Si $\sum v_n$ converge, alors $\sum u_n$ converge.
  Si $\sum u_n$ diverge, alors $\sum v_n$ diverge.

- Comparaison par négligeabilité. On suppose que $u_n = o(v_n)$, $n \to +\infty$.
  Si $\sum v_n$ converge, alors $\sum u_n$ converge.
  Si $\sum u_n$ diverge, alors $\sum v_n$ diverge.

- Comparaison par équivalence. On suppose que $u_n \sim v_n, \quad n \to +\infty$.
  Alors $\sum u_n$ et $\sum v_n$ ont la même nature.

(2) On peut implémenter ce produit avec l'aide d'une boucle `for` ou, plus rapidement ici ce qui est pratique car le temps est limité et qu'il faut être synthétique à l'oral, à l'aide de la commande `np.prod` et d'une compréhension de liste. On propose le code suivant.

```
import numpy as np

def P(n, a):
    L=[(a+k)/(a+2*k) for k in range(n+1)]
    return np.prod(L)
```

(3) L'inégalité de droite se montre notamment par un argument de concavité du log (la courbe est au dessous de ses tangentes notamment celle en 0) qu'on ne reproduit pas ici. Montrons l'inégalité de gauche. On pose, pour $u > -1$,

$$h(u) = \ln(1+u) - \frac{u}{1+u}$$

La fonction h est clairement dérivable et

$$h'(u) = \frac{1}{1+u} - \frac{1}{(1+u)^2} = \frac{u}{(1+u)^2}.$$

Il est alors facile de dresser le tableau de variations de h:

u	-1		0		$+\infty$
$h'(u)$	‖	$-$	0	$+$	
h	‖ $+\infty$	$\searrow$	0	$\nearrow$	$+\infty$

On a bien que, pour tout $u > -1$, $h(u) \geq 0$ ou encore que $\ln(1+u) \geq \dfrac{u}{1+u}$ ce qui est bien ce qu'on voulait.

(4) Il est difficile de travailler avec des produits. La méthode classique est de passer au log pour travailler avec une somme. On a

$$P_n = \exp\left(\ln\left(\prod_{k=0}^{n} \frac{a+k}{2a+k}\right)\right) = \exp\left(-\sum_{k=0}^{n} \ln\left(1 + \frac{a}{a+k}\right)\right).$$

En appliquant l'encadrement précédent avec $u = \dfrac{a}{a+k}$, on obtient

$$\frac{\frac{a}{a+k}}{1+\frac{a}{a+k}} \leq \ln\left(1+\frac{a}{a+k}\right) \leq \frac{a}{a+k}$$

ou encore

$$\frac{a}{2a+k} \leq \ln\left(1+\frac{a}{a+k}\right) \leq \frac{a}{a+k}$$

En multipliant par -1, en passant à la somme et en composant par l'exponentielle croissante qui préserve les inégalités, on a bien

$$\exp\left(-\sum_{k=0}^{n}\frac{a}{a+k}\right) \leq P_n \leq \exp\left(-\sum_{k=0}^{n}\frac{a}{2a+k}\right)$$

(5) Pour $a > 0$ fixé, on peut écrire

$$\frac{a}{a+k} \sim \frac{a}{k}, \qquad \frac{a}{2a+k} \sim \frac{a}{k}$$

et la série $\sum a/k$ diverge (Riemann). Par critère d'équivalence (pour les SATP), les deux séries correspondantes divergent. Donc les deux sommes partielles

$$\sum_{k=0}^{n}\frac{a}{a+k} \underset{n\to+\infty}{\longrightarrow} +\infty, \qquad \sum_{k=0}^{n}\frac{a}{2a+k} \underset{n\to+\infty}{\longrightarrow} +\infty$$

et par composition avec l'exponentielle, on a

$$\exp\left(-\sum_{k=0}^{n}\frac{a}{a+k}\right) \underset{n\to+\infty}{\longrightarrow} 0, \qquad \exp\left(-\sum_{k=0}^{n}\frac{a}{2a+k}\right) \underset{n\to+\infty}{\longrightarrow} 0.$$

On en déduit, par théorème des gendarmes, que $P_n \to 0$.

(6) La comparaison série/intégrale (suggérée par l'énoncé) permet souvent des estimation relativement précises de sommes partielles. Travaillons pour commencer avec le terme de gauche de l'encadrement ci-dessus. Sans difficulté, on obtient, pour $k \geq 1$,

$$\frac{a}{a+k} \leq \int_{k-1}^{k}\frac{a}{a+t}\mathrm{d}t$$

Puis, par somme et Chasles

$$\sum_{k=0}^{n}\frac{a}{a+k} \leq 1+\sum_{k=1}^{n}\int_{k-1}^{k}\frac{a}{a+t}\mathrm{d}t = 1+\int_{0}^{n}\frac{a}{a+t}\mathrm{d}t = 1+\left[a\ln(a+t)\right]_0^n = 1+a\ln\left(\frac{a+n}{a}\right)$$

Il suit

$$\exp\left(-\sum_{k=0}^{n}\frac{a}{a+k}\right) \geq \exp\left(-1-a\ln\left(\frac{a+n}{a}\right)\right) = \frac{1}{e\left(1+\frac{n}{a}\right)^a},$$

ce qui est bien le minorant voulu. Le majorant s'obtient de la même manière. On a au final l'encadrement

$$\frac{1}{e\left(\frac{n}{a}+1\right)^a} \leq P_n \leq \frac{1}{\left(\frac{n+1}{2a}+1\right)^a}$$

Pour $a = 1$ l'encadrement précédent s'écrit

$$\frac{1}{e\,(n+1)} \leq P_n \leq \frac{2}{n+3}.$$

Or comme la série $\sum 1/[e(n+1)]$ diverge (Riemann), on peut conclure par comparaison (tout est positif) que $\sum P_n$ diverge.

Pour $a=2$, l'encadrement en revanche donne

$$\frac{1}{e\left(\frac{n}{2}+1\right)^2} \leq P_n \leq \frac{1}{\left(\frac{n+1}{4}+1\right)^2}$$

On peut alors utiliser le critère de comparaison; le majorant est équivalent au multiple de $\sum 1/n^2$ qui converge donc $\sum P_n$ converge. Mais on demande la somme... Donc on doit pouvoir calculer. Revenons alors à la définition de P_n dans ce cas

$$\begin{aligned} P_n &= \prod_{k=0}^{n} \frac{2+k}{4+k} = \prod_{k=2}^{n+2} \frac{k}{2+k} \\ &= \frac{2\times 3}{(n+3)\times(n+3)} \qquad \text{par télescopage} \\ &= 6\left(\frac{1}{n+3}-\frac{1}{n+4}\right). \end{aligned}$$

Ainsi, encore par télescopage,

$$\sum_{k=0}^{n} P_k = 6\sum_{k=0}^{n}\left(\frac{1}{k+3}-\frac{1}{k+4}\right) = 6\left(\frac{1}{3}-\frac{1}{n+4}\right) \underset{n\to+\infty}{\longrightarrow} 2.$$

☞ On peut en fait déterminer la nature de $\sum P_n$ en fonction de a. On laisse le soin au lecteur ou à la lectrice le plaisir de répondre avec cette question (on a tout ce qu'il faut pour!).

(7) Dans une urne, on dispose initialement a boule rouge et a boule noire (indiscernables au toucher). On tire ensuite une boule: si elle est noire, on arrête; si elle est rouge, on la remplace dans l'urne par deux boules rouges, et on recommence jusqu'à obtention d'une boule noire.
On va donc regarder ce qu'il se passe avec $a=1$ puis $a=2$.
Introduisons X la v.a correspondant au nombre de tirages effectués.

On peut obtenir la boule noire dès le premier tirage mais on peut aussi tirer une série arbitrairement longue de boules rouges consécutives (surtout qu'on en rajoute...) donc $X(\Omega)=\mathbb{N}^*$.

On introduit les évènements R_i (resp. N_i) "la boule obtenue au $i-$ème tirage est rouge (resp. noire)". Alors

$$P(X=1)=P(N_1)=\frac{a}{2a}=\frac{1}{2}=P_0.$$

Puis, pour $n\geq 1$, par la formule des probabilités composées

$$\begin{aligned} P(X>n) &= P(R_1\cap\ldots\cap R_n) = P(R_1)P_{R_1}(R_2)\ldots P_{R_1\cap..\cap R_{n-1}}(R_n) \\ &= \frac{a}{2a}\times\frac{a+1}{2a+1}\times\ldots\times\frac{a+n-1}{2a+n-1} \\ &= P_{n-1}. \end{aligned}$$

On utilise alors un résultat relativement classique sur une nouvelle formule pour la somme aprtielle qui définit l'espérance. On commence par observer que

$$[X=n]\cup[X>n]=[X>n-1]$$

donc, par incompatibilité

$$P(X=n)=P(X>n-1)-P(X>n)$$

On somme, ce qui sonne

$$\begin{aligned}
\sum_{k=1}^{n} kP(X=k) &= \sum_{k=1}^{n} j\left(P(X>k-1)-P(X>k)\right) \\
&= \sum_{k=1}^{n} kP(X>k-1)-\sum_{k=1}^{n} kP(X>k)=\sum_{j=0}^{n-1}(j+1)P(X>j)-\sum_{k=1}^{n} kP(X>k) \\
&= \sum_{j=0}^{n-1} jP(X>j)+\sum_{j=0}^{n-1} P(X>j)-\sum_{k=1}^{n} kP(X>k) \\
&= \sum_{j=0}^{n-1} P(X>j)-nP(X>n).
\end{aligned}$$

On sait que X admet une espérance si et seulement si la série $\sum nP(X=n)$ converge (absolument). Commençons par observer, d'après ce qui précède que

$$\begin{aligned}
\sum_{k=1}^{n} kP(X=k) &= \sum_{j=0}^{n-1} P(X>j)-nP(X>n) \\
&= 1+\sum_{j=1}^{n-1} P_{j-1}-nP_{n-1}=1-nP_n+\sum_{k=0}^{n} P_k
\end{aligned}$$

- Dans le cas où $a=1$. D'après tout ce qui précède (notamment l'encadrement des Questions $(4)-(5)$),
$$\sum_{k=1}^{n} kP(X=k)=1-nP_n+\sum_{k=0}^{n} P_k \geq 1-\frac{2n}{n+3}+\sum_{k=0}^{n} P_k \underset{n\to+\infty}{\longrightarrow} +\infty$$
donc X n'admet pas d'espérance.

- Dans le cas où $a=2$. L'encadrement obtenue aux Questions (4) et (5) permet de voir que nP_n tend vers 0 grâce au théorème des gendarmes. Mais alors,
$$\sum_{k=1}^{n} kP(X=k)=1-nP_n+\sum_{k=0}^{n} P_k \underset{n\to+\infty}{\longrightarrow} 1+\sum_{k=0}^{+\infty} P_k=1+2=3.$$
Ainsi, X admet une espérance et $E(X)=3$.

(8) On peut illustrer les derniers résultats en simulant la variable X. On réalise ensuite un $n-$échantillon de X (avec n grand, par exemple $n=1000$ ou $n=10000$) et la *moyenne* empirique de l'échantillon renvoie une *estimation* de l'espérance. On propose le code suivant

```python
import numpy as np
import numpy.random as rd

# simulation de X
def simul_X(a) :
    r= a # on commence avec a boules rouges
    x=1
    while rd.rand() > a/(r+a) : # tant qu'on pioche des rouges
        r= r+1 # on rajoute une rouge
        x = x+1 # un tirage de plus à faire
    return x
```

```
# estimation de l'espérance pour a=2
a=2
sample=[simul_X(a) for k in range(1000)]
print(np.mean(sample))
```

Exercice non préparé

Soient $n \in \mathbb{N}^*$ et $A \in \mathcal{M}_n(\mathbb{R})$ tels que
$$A\,{}^tAA\,{}^tAA = I.$$

(1) Montrer que A est symétrique.
(2) Déterminer A.

Solution.

(1) Par définition de l'inverse d'une matrice, il découle de la propriété que A est inversible et que (selon qu'on factorise par A à droite où à gauche)
$$A^{-1} = {}^tAA\,{}^tAA = A\,{}^tAA\,{}^tA.$$
En transposant la relation vérifiée par A, on a
$${}^tAA\,{}^tA\,{}^tAA = I$$
donc cette fois on a
$$\left({}^tA\right)^{-1} = A\,{}^tAA\,{}^tA$$
Ainsi
$$\left({}^tA\right)^{-1} = A^{-1}$$
Par unicité de l'inverse, il découle que $A = {}^tA$ ou encore que A est symétrique.

(2) La relation vérifiée par A est alors $A^5 = I$ donc $X^5 - 1$ est un polynôme annulateur de A et n'admet pour racine que 1 donc $\mathrm{Sp}(A) \subset \{1\}$. Or A est symétrique donc diagonalisable. Donc $\mathrm{Sp}(A) = \{1\}$ et $A = PIP^{-1}$ (pour une certaine matrice inversible P) et on peut conclure que $A = I$. Il n'y a donc que l'identité qui vérifie cette relation.

Mardi 16 Mai - Amazigh D.

Exercice avec préparation

(1) (**Question de cours**). Définition des valeurs propres d'une matrice. Cas particulier d'une matrice triangulaire.

(2) Soit $N \in \mathcal{M}_n(\mathbb{R})$ une matrice nilpotente, c'est à dire une matrice de taille n pour laquelle il existe un entier $p \geq 1$, tel que $N^p = 0$.
Déterminer le spectre de N.

Soit $M = \begin{pmatrix} 0 & a & b \\ c & 0 & d \\ e & f & 0 \end{pmatrix}$ une matrice de $\mathcal{M}_3(\mathbb{R})$ de coefficients diagonaux nuls.
On introduit les quantités $\gamma(M) = ac + df + be$ et $\delta(M) = bcf + ade$.

(3) Établir l'égalité $M^3 = \gamma(M)M + \delta(M)I_3$.

(4) Montrer que la matrice M est nilpotente si et seulement si $\gamma(M)$ et $\delta(M)$ sont nuls.

(5) En déduire l'écriture en Python d'une fonction d'en-tête `def test_nilpotente(M):` qui prend en argument une matrice M de taille 3 dont les coefficients diagonaux sont nuls et qui renvoie 1 ou 0 selon que la matrice M est nilpotente ou non.

(6) On suppose que a, b et d sont égaux à 1. Justifier qu'il existe une infinité de choix pour le triplet (c, e, f) de $\mathbb{R}^3$ pour lesquels la matrice M est nilpotente.

(7) À quelle question cette exercice permet-il de répondre ?

Solution.

(1) (Question de cours). Soit M une matrice de $\mathcal{M}_n(\mathbb{R})$. Par définition du cours,

$$\begin{aligned} \lambda \text{ valeur propre de } M &\iff M - \lambda I \text{ non inversible} \\ &\iff \operatorname{Ker}(M - \lambda I) \neq \{0\}. \end{aligned}$$

Si M est triangulaire, on obtient immédiatement (le système correspondant à la détermination du noyau est lui-même triangulaire) que ses valeurs propres sont exactement ses coefficients diagonaux.

(2) Soit $N \in \mathcal{M}_n(\mathbb{R})$ une matrice nilpotente. Il existe donc $p \neq 0$ tel que $M^p = 0$ et le polynôme X^p est annulateur de N. Si α est valeur propre de N alors $\alpha^p = 0$ et donc $\alpha = 0$. Il reste à montrer que 0 est bien valeur propre de N.

Comme $N^p = 0$ alors N est non inversible (sinon N^p le serait) donc 0 est valeur propre, ainsi 0 est la seule valeur propre de N. On peut conclure que

$$\operatorname{Sp}(N) = \{0\}.$$

(3) Le calcul donne

$$M^2 = \begin{pmatrix} 0 & a & b \\ c & 0 & d \\ e & f & 0 \end{pmatrix} \begin{pmatrix} 0 & a & b \\ c & 0 & d \\ e & f & 0 \end{pmatrix} = \begin{pmatrix} ac+be & bf & ad \\ ed & ac+df & cb \\ fc & ae & eb+df \end{pmatrix}$$

et

$$M^3 = \begin{pmatrix} ade+bcf & abe+a^2c+adf & b^2e+abc+bdf \\ ac^2+cdf+bce & ade+bcf & dac+d^2f+bde \\ be^2+def+ace & fbe+df^2+acf & ade+bcf \end{pmatrix}.$$

D'autre part,

$$\gamma(M)M+\delta(M)I_3 = (ac+df+be)\begin{pmatrix} 0 & a & b \\ c & 0 & d \\ e & f & 0 \end{pmatrix} + (bcf+ade)\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = M^3.$$

(4) Si $\gamma(M)$ et $\delta(M)$ sont nuls, alors $M^3=0$ donc M est nilpotente.
Réciproquement, si M est nilpotente alors, d'après ce qui précède M a pour unique valeur propre 0.
Si $\delta(M)\neq 0$ alors la relation

$$M^3=\gamma(M)M+\delta(M)I$$

donne un polynôme annulateur $P(X)=X^2-\gamma(M)X-\delta(M)$. Mais $P(0)=\delta\neq 0$. Or, toute valeur propre d'une matrice est racine de tout polynôme annulateur de cette même matrice. On a une contradiction. Il suit que $\delta(M)=0$.
Il suit que M vérifie alors

$$M^3=\gamma(M)M \iff M(M^2-\gamma(M)I)=0$$

Observons alors que si M est nilpotente, alors M^2 aussi (en effet $M^{2p}=0$ si $M^p=0$). Donc M^2 n'a que 0 pour valeur propre. Si $\gamma(M)\neq 0$, alors $\gamma(M)$ n'est pas valeur propre de M^2, donc $M^2-\gamma(M)I$ est inversible, En multipliant la relation ci-dessus par l'inverse de $M^2-\gamma(M)I$, on obtient alors $M=0$. Mais alors $\gamma(M)=0$ (tous les coefficients de M sont nuls), ce qui est absurde.
On a bien obtenu $\gamma(M)=\delta(M)=0$.

(5) Il suffit de faire calculer $\gamma(M)$ et $\delta(M)$ et de renvoyer 1 si les deux quantités sont nulles.

```
import numpy as np

def test_nilpotente(M) :
    gamma=M[0,1]*M[1,0]+M[0,2]*M[2, 0]+M[1,2]*M[2,1]
    delta = M[0,2]*M[1,0]*M[2, 1]+M[0,1]*M[1, 2]*M[2, 0]
    if gamma == 0 and delta == 0 :
        return 1
    return 0
```

(6) On suppose que a, b et d sont égaux à 1. Par la question précédente, M est nilpotente si et seulement si $ac+df+be=c+f+e=0$ et $bcf+ade=cf+e=0$, ce qui donne

$$\begin{cases} e &= -cf \\ c+f-cf &= 0 \end{cases} \iff \begin{cases} e &= -cf \\ c(1-f) &= -f \end{cases}$$

et, pour $f\neq 1$, le système donne

$$\begin{cases} e &= f^2/(1-f) \\ c &= -f/(1-f) \end{cases}.$$

Donc pour chaque $f\neq 1$ il y a une solution au système, ce qui en fait une infinité de triplets (c,e,f) de $\mathbb{R}^3$ pour lesquels la matrice M est nilpotente.

(7) Pour a, b et d égaux à 1 et $f \neq 1$ (et non nul afin que la matrice M ne soit pas triangulaire), la matrice

$$M = \begin{pmatrix} 0 & 1 & 1 \\ -f/(1-f) & 0 & 1 \\ f^2/(1-f) & f & 0 \end{pmatrix}$$

est non triangulaire et nilpotente d'après les calculs précédents. Comme elle est nilpotente, sa seule valeur propre est 0. Et comme le seul terme sur sa diagonale est 0, on a construit une matrice dont les valeurs propres sont ses éléments diagonaux mais qui n'est pas triangulaire, fournissant ainsi un contre-exemple à la réciproque du résultat rappelé en question de cours.

Exercice sans préparation

Soient $X_0, X_1, ..., X_n, ...$ une suite de variables aléatoires mutuellement indépendantes de même loi $\mathcal{N}(0,1)$ et $\theta \in \mathbb{R}^* \setminus \{-1, 1\}$.
On définit une nouvelle suite (Y_n) comme suit

$$Y_0 = X_0, \qquad \forall n \geq 0, \quad Y_{n+1} = \theta Y_n + X_{n+1}.$$

(1) Écrire une fonction en Python permettant de simuler Y_n.
(2) Déterminer la loi de Y_n.
(3) Comment vérifier le résultat précédent avec Python?

Solution.

(1) Sans difficulté

```
import numpy as np; import numpy.random as rd

def simul_Y(n, theta):
    Y=rd.normal(0,1)
    for k in range(n) :
        Y=theta*Y+rd.normal(0,1)
    return Y
```

(2) On pense au résultat de stabilité par somme des lois normales. Sauf que pur appliquer ce résultat il faut que les variables soient indépendantes, et toutes les deux de lois normales. Commençons par montrer par récurrence que les variables Y_{n+1} et X_{n+1} sont indépendantes, à l'aide du lemme des coalitions en montrant que, pour tout $n \in \mathbb{N}$, Y_n est fonction de $X_0, ..., X_n$.

- C'est clair, par définition de $Y_0 = X_0$ pour $n = 0$.
- Supposons que, pour un certain $n \in \mathbb{N}$, on ait $Y_n = \varphi(X_0, .., X_n)$. Alors,

$$Y_{n+1} = \theta Y_n + X_{n+1} = \theta\varphi(X_0, ..., X_n) + X_{n+1}$$

qui est bien une fonction de $X_0, ..., X_{n+1}$.

Par indépendantes mutuelles des X_i et lemme des coalitions, Y_{n+1} est donc indépendante de X_{n+1}.
Ensuite, on observe que $Y_1 = \theta X_0 + X_1$ est une loi normale de paramètre 0 et $\theta^2 + 1$. On montre donc par récurrence que

$$Y_n \hookrightarrow \mathcal{N}\left(0, \sum_{i=0}^{n} \theta^{2i}\right).$$

Il reste à montrer le caractère héréditaire de cette récurrence, celle-ci étant déjà initialisée. Supposons donc que, pour un certain $n \in \mathbb{N}$, on ait

$$Y_n \hookrightarrow \mathcal{N}\left(0, \sum_{i=0}^{n} \theta^{2i}\right)$$

Alors, d'une part

$$\theta Y_n \hookrightarrow \mathcal{N}\left(0, \theta^2 \sum_{i=0}^{n} \theta^{2i}\right),$$

puis, par stabilité par somme des lois normales indépendantes (θY_n est bien indépendante de X_{n+1} par ce qui précède et le lemme des coalitions), on a

$$Y_{n+1} = \theta Y_n + X_{n+1} \hookrightarrow N\left(0, 1 + \theta^2 \sum_{i=0}^{n} \theta^{2i}\right).$$

Or

$$1 + \theta^2 \sum_{i=0}^{n} \theta^{2i} = 1 + \sum_{i=0}^{n} \theta^{2i+2} = 1 + \sum_{j=1}^{n+1} \theta^{j} = \sum_{j=0}^{n+1} \theta^{2i}$$

ce qui termine cette récurrence. On a bien la loi de Y_n.

(3) Pour illustrer ce résultat, il faudrait simuler Y_n pour un certain n un grand nombre de fois et représenter l'histogramme de distribution des valeurs. La fonction de densité de la loi $\mathcal{N}\left(0, \sum_{i=0}^{n} \theta^{2i}\right)$ devrait en épouser le contour. Observant que

$$\sum_{i=0}^{n} \theta^{2i} = \frac{1 - \theta^{2n+2}}{1 - \theta^2},$$

on propose le code suivant, qui repose sur la fonction précédemment écrite

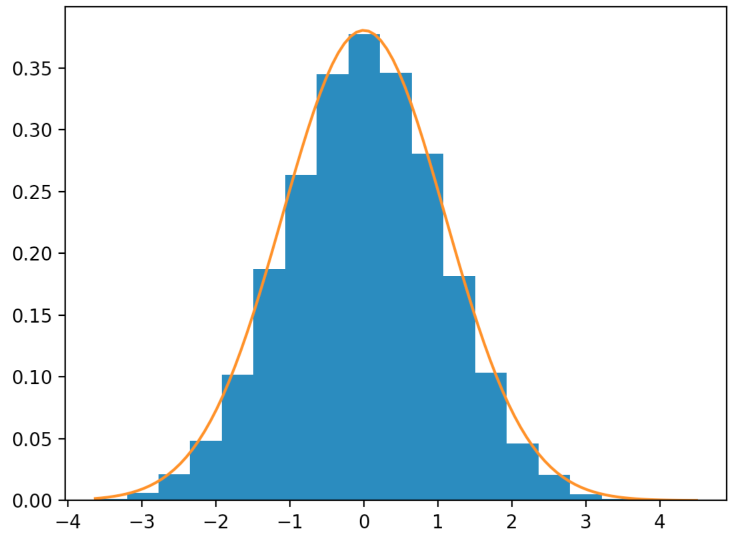
```python
N=10000
n=int(input("n=?")) # n choisi par utilisateur
theta = float(input("theta=?")) # theta choisi par utilisateur
sample=[simul_Y(n, theta) for k in range(N)]

def densite(x):
    v=(1-theta**(2*n+2))/(1-theta**2)
    return np.exp(-(x/v)**2/2)/np.sqrt(2*np.pi*v)

import matplotlib.pyplot as plt
x0=min(sample)
x1=max(sample)
X=np.linspace(x0, x1, 100)
plt.hist(sample, density=True, bins=np.linspace(x0, x1, 20))
Y=[densite(x) for x in X]
plt.plot(X,Y)
plt.show()
```

Ceci donne la figure suivante (avec $n = 12$ et $\theta = 0.3$)

Affichage Python



Mardi 16 Mai - Lucie P.

Exercice avec préparation

Soient $n \in \mathbb{N}^*$ et $E = \mathbb{R}_n[X]$ l'espace vectoriel des polynômes à coefficients réels de degré inférieur ou égal à n.

Soit φ l'application qui à tout polynôme P élément de E associe la matrice colonne $\begin{pmatrix} P(0) \\ P(1) \\ \vdots \\ P(n) \end{pmatrix}$.

(1) (**Question de cours**) : nombre de racines d'un polynôme $P \in \mathbb{R}_n[X]$.

(2) Montrer que φ est un isomorphisme de E dans $\mathcal{M}_{n+1,1}(\mathbb{R})$.

(3) On suppose dans cette question seulement que $n = 2$.

Montrer qu'il existe une matrice $A \in \mathcal{M}_3(\mathbb{R})$ que l'on précisera telle que pour tout $P \in E$ tel quel $P(X) = a + bX + cX^2$, on ait

$$\begin{pmatrix} P(0) \\ P(1) \\ P(2) \end{pmatrix} = A \begin{pmatrix} a \\ b \\ c \end{pmatrix}.$$

Montrer que A est inversible et déterminer son inverse.

(4) On revient au cas général.

(a) Montrer qu'il existe une matrice $V \in \mathcal{M}_{n+1}(\mathbb{R})$ que l'on précisera telle que pour tout $P \in E$ tel que $P(X) = \sum_{i=0}^{n} a_i X^i$, alors

$$\begin{pmatrix} P(0) \\ P(1) \\ \vdots \\ P(n) \end{pmatrix} = V \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix}.$$

Justifier que V est inversible.

(b) Pour $i \in \{0, 1, \ldots, n\}$, montrer qu'il existe un unique polynôme $L_i \in \mathbb{R}_n[X]$ tel que

$$\begin{cases} L_i(i) = 1 \\ L_i(j) = 0 \text{ pour tout} j \in \{0, 1, \ldots, n\} \setminus \{i\} \end{cases}$$

(c) On pose pour $i \in [\![0, n]\!]$, $L_i(X) = \sum_{k=0}^{p} nc_{k,i} X^k$ et l'on note $C_i = \begin{pmatrix} c_{0,i} \\ c_{1,i} \\ \vdots \\ c_{n,i} \end{pmatrix} \in \mathcal{M}_{n+1,1}(\mathbb{R})$.

Calculer VC_i et en déduire l'expression de V^{-1}.

(5) A l'aide de la Question 4., retrouver A^{-1}.

Solution.

(1) Un polynôme $P \in \mathbb{R}_n[X]$ **non nul** possède au plus n racines distinctes. Ainsi,

$$\forall P \in \mathbb{R}_n[X],\ (P = 0_{\mathbb{R}_n[X]} \Longleftrightarrow P \text{ possède au moins } n+1 \text{ racines distinctes})$$

(2) φ est linéaire de E dans $\mathcal{M}_{n+1,1}(\mathbb{R})$. Du résultat rappelé en question de cours, on déduit que $\mathrm{Ker}(\varphi) = \{0_E\}$ (sinon un polynôme dans le noyau aurait $n+1$ racines qui sont les entiers de 0

à n). Par théorème du rang $\mathrm{rg}(\varphi) = n+1-0 = n+1 = \dim(\mathcal{M}_{n+1,1}(\mathbb{R})$ et donc φ est injective et surjective donc bijective.

(3) Avec les notations de l'énoncé, on a

$$\begin{pmatrix} P(0) \\ P(1) \\ P(2) \end{pmatrix} = \begin{pmatrix} a \\ a+b+c \\ a+2b+4c \end{pmatrix} = A \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \qquad \text{où} \qquad A = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{pmatrix}$$

On a $A^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{3}{2} & 2 & -\frac{1}{2} \\ \frac{1}{2} & -1 & \frac{1}{2} \end{pmatrix}$.

(4) (a) La matrice suivante convient :

$$V = \begin{pmatrix} 1 & 0 & \cdots & & 0 \\ 1 & 1 & \cdots & & 1 \\ 1 & 2 & 2^2 & \cdots & 2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & n & n^2 & \dots & n^n \end{pmatrix} \in \mathcal{M}_{n+1}(\mathbb{R}).$$

On note $\mathcal{B}$ la base canonique de E et $\mathcal{B}'$ la base canonique de $\mathcal{M}_{n+1,1}(\mathbb{R})$.
On a $V = \mathrm{Mat}_{\mathcal{B},\mathcal{B}'}(\varphi)$ et φ est un isomorphisme, donc V est inversible.

(b) On note $\mathcal{B}' = (E_0, \dots, E_n)$. Soit $i \in [\![0, n]\!]$, par bijectivité de φ, il existe un unique polynôme $P \in E$ vérifiant $\varphi(P) = E_i$, que l'on note L_i (on a $L_i = \varphi^{-1}(E_i)$).

(c) Soit $i \in [\![0, n]\!]$. On a $VC_i = \begin{pmatrix} L_i(0) \\ L_i(1) \\ \vdots \\ L_i(n) \end{pmatrix} = E_i$. Donc $C_i = V^{-1}E_i$, donc C_i est la $(i+1)$-ième colonne de V^{-1}.

(5) $L_0 = 1 - \frac{3}{2}X + \frac{1}{2}X^2$, $L_1 = 2X - X^2$ et $L_2 = -\frac{1}{2}X + \frac{1}{2}X^2$ donc

$$C_0 = \begin{pmatrix} 1 \\ -\frac{3}{2} \\ \frac{1}{2} \end{pmatrix}, C_1 = \begin{pmatrix} 0 \\ 2 \\ -1 \end{pmatrix}, C_2 = \begin{pmatrix} 0 \\ -\frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$$

Exercice sans préparation

Deux urnes A et B contiennent initialement chacune deux boules numérotées 0 et 1. Leur contenu évolue lors d'une expérience aléatoire et est simulé par une liste Python.
Ainsi la ligne de code `A = [0, 1]` permet de voir qu'initialement, l'urne A contient une boule numérotée 0 et une autre numérotée 1.

On considère la fonction Python suivante qui permet de simuler une variable aléatoire X.

```
import numpy as np
import numpy.random as rd

def simulX():
    A = [0, 1]
    B = [0;1]
    Y = rd.randint(0, 2, [4, 2])
    k = 0
    while np.sum(A) > 0 and k < 4:
        i = Y[k, 0]
        j = Y[k, 1]
        c = A[i]
        A[i] = B[j]
        B[j] = c
        k = k + 1
    return k
```

Expliquer le protocole ainsi simulé et donner la loi de X.

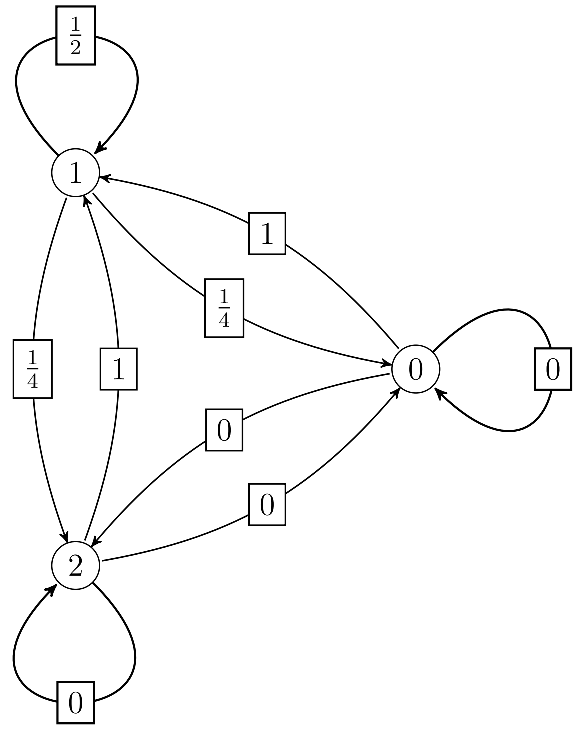
Solution.

À chaque étape on choisit une boule au hasard dans A et une boule au hasard dans B et on les change d'urnes. On effectue au maximum 3 étapes. Le résultat de la fonction donne

- le nombre d'échanges nécessaires pour que A contienne les deux boules numérotées 0 si cela se produit en trois étapes ou moins.
- 4 si au bout de trois échanges on n'a pas obtenu les deux boules 0 dans A

On a $X(\Omega) = \{1, 2, 3, 4\}$.

Notons Y_k la variable aléatoire égale au nombre de boules 0 dans l'urne A à l'issue de k échanges. En particulier : $Y_0 = 1$.
La suite (Y_k) est une chaîne de Markov dont on peut représenter le graphe probabiliste associé :



En s'aidant du graphe, on remarque :

- $[X = 1] = [Y_1 = 1]$ donc $P(X = 1) = \dfrac{1}{4}$.
- $[X = 2] = [Y_1 = 1] \cap [Y_2 = 2]$ donc $P(X = 2) = \dfrac{1}{2} \times \dfrac{1}{4} = \dfrac{1}{8}$ (formule des probabilités composées)
- $[X = 3] = ([Y_1 = 1] \cap [Y_2 = 1] \cap [Y_3 = 2]) \cup ([Y_1 = 0] \cap [Y_2 = 1] \cap [Y_3 = 2])$
  donc
  $$P(X = 3) = \frac{1}{2} \times \frac{1}{2} \times \frac{1}{4} + \frac{1}{2} \times 1 \times \frac{1}{4} = \frac{1}{8}$$
  (par incompatibilité et formule des probabilités composées).
- Par complémentarité, $P(X = 4) = \dfrac{1}{2}$.