DEVOIR SURVEILLE
MATHÉMATIQUES

Samedi 8 février 2025

(3 heures)

La présentation, la lisibilité, l'orthographe, la qualité de la rédaction, la clarté et la précision des raisonnements entrent pour une part importante dans l'appréciation des copies.

*Les candidats sont invités à **encadrer** dans la mesure du possible les résultats de leurs calculs.*

Si au cours de l'épreuve un candidat repère ce qui lui semble être une erreur d'énoncé, il le signalera sur sa copie et poursuivra sa composition en expliquant les raisons des initiatives qu'il sera amené à prendre.

La calculatrice est autorisée.

Cet énoncé comporte 4 pages.

Vous trouverez en fin du sujet une annexe avec des commandes Python.

Ce devoir comporte deux problèmes et un exercice indépendants les uns des autres.

L'exercice et le deuxième problème portent sur des problèmes d'évolution de population similaires sans être identiques : dans l'un, l'effectif total de la population peut varier, dans l'autre, il est fixé.

Cet exercice et ce problème sont indépendants même si les techniques utilisées sont similaires. Le premier est dans un esprit directement tourné vers une application numérique, le second dans un esprit plus abstrait propice à une programmation informatique.

Problème 1 : Loi à densité de Pareto

Partie 1 : Propriété d'une loi de probabilité.

On désigne par c un réel strictement positif et on considère la fonction f définie sur $\mathbb{R}$ par :

$$f(x) = \begin{cases} \frac{c}{x^{1+c}} & \text{si } x \geq 1 \\ 0 & \text{si } x < 1 \end{cases}$$

1) Montrer que f définit une densité de probabilité.

On considère dans la suite une variable aléatoire X de densité f et on note F sa fonction de répartition.

On dit que X suit la **loi de Pareto de paramètre c** .

2) Déterminer, pour tout réel x , l'expression de $F(x)$ en fonction de x et c .

3) Soit t un réel strictement supérieur à 1 .

a) Montrer que X admet une espérance si, et seulement si, $c > 1$, et dans ce cas là calculer cette espérance.

b) Déterminer, en distinguant les cas $x \geq 1$ et $x < 1$, la probabilité conditionnelle $P_{(X>t)}(X \leq tx)$.

c) On pose $Z = \frac{X}{t}$, calculer pour x réel $P_{(X>t)}(Z \leq x)$. Que remarque-t-on ?

Partie 2 : Simulation d'une variable suivant la loi de Pareto de paramètre c .

4) On pose $Z = \ln(X)$ et on admet que Z est une variable aléatoire définie sur le même espace probabilisé que X . On note H sa fonction de répartition.

a) Pour tout réel x , exprimer $H(x)$ à l'aide de la fonction F .

b) En déduire que Z suit une loi exponentielle dont on précisera le paramètre.

c) Soit $\lambda > 0$ et U suivant la loi $\mathcal{U}([0, 1])$, quelle est la loi suivie par $-\frac{1}{\lambda} \ln(1 - U)$? (*Justifier*)

d) En utilisant les résultats des questions précédentes, écrire une fonction Python d'en-tête `def simulX(c, n)` qui pour un entier n strictement positif renvoie une liste de n réalisations d'une variable aléatoire suivant la loi de Pareto de paramètre c .

Exercice.

On rappelle que, pour tout $p \in \mathbb{N}^*$, $\mathcal{M}_p(\mathbb{R})$ désigne l'ensemble des matrices à p lignes et p colonnes et à coefficients réels. Par convention, si $M \in \mathcal{M}_p(\mathbb{R})$ alors M^0 désigne la matrice identité de $\mathcal{M}_p(\mathbb{R})$.

Dans tout l'exercice, n désigne un entier naturel. On s'intéresse à une population d'oiseaux notée $\mathcal{P}$. Dans cette population, la moitié sont des oiselles (c'est-à-dire des femelles) et on modélise l'évolution de l'effectif de ces oiselles en fonction de n , le nombre d'années écoulées depuis un instant initial correspondant à $n = 0$.

5) On note les deux matrices :

$$A = \begin{pmatrix} 1 & 5 \\ \frac{3}{20} & 0 \end{pmatrix}, \quad P = \begin{pmatrix} 2 & 2 \\ \frac{1}{5} & -\frac{3}{5} \end{pmatrix}$$

a) Calculer $\det(P)$ puis P^{-1} .

b) Démontrer que A est diagonalisable et donner une matrice diagonale D telle que $A = P.D.P^{-1}$.

6) Les oiselles de $\mathcal{P}$ sont classées en deux catégories :

- les oiselles jeunes, âgées de moins d'un an. L'effectif des oiselles jeunes est noté j_n .
- les oiselles adultes, âgées d'au moins un an. L'effectif des oiselles adultes est noté a_n .

Une étude sur le terrain a permis de conclure que :

- chacune de ces oiselles donne naissance en moyenne à une oiselle pendant sa première année de vie et à 5 oiselles pendant sa deuxième année.
- 15% des oiselles survivent au delà de leur première année mais jamais au delà de leur seconde année.

On note pour $n \in \mathbb{N}$:

$$X_n = \begin{pmatrix} j_n \\ a_n \end{pmatrix} \text{ et } X'_n = P^{-1} \cdot X_n$$

a) Justifier que $X_{n+1} = A.X_n$ pour tout $n \in \mathbb{N}$.

b) En déduire une expression de X'_n en fonction de X'_0, n, P et D . On rédigera une démonstration par récurrence.

c) En supposant j_0 et a_0 non nuls, démontrer que j_n et a_n sont équivalents à des termes généraux de suites géométriques.

Problème 2 : Modèle de LESLIE conservatif.

Le modèle de LESLIE conservatif est un modèle d'évolution de la structure d'une population dont l'effectif global ne change pas au cours du temps. Chaque individu de la population est dans un certain état (en démographie ou en biologie, pour une population vivante, ce peut être l'âge ou le stade d'évolution, en physique nucléaire, pour une population de noyaux atomiques soumise à bombardement neutronique, ce peut être le type isotopique).

Le scénario est le suivant :

- Les états sont numérotés E_0 (*état fondamental*), $E_1, \dots, E_{K-1}$ (*états supérieurs*) et E_K (*état ultime*)
- A chaque pas de temps, pour chaque état E_k à l'exception de l'état ultime, une proportion $s_k \in]0, 1]$ de la population dans cet état passe à l'état suivant alors que la proportion $1 - s_k$ retourne à l'état fondamental.
- A chaque pas de temps, la population dans l'état ultime E_K repasse intégralement à l'état fondamental E_0 . En un sens, cela revient à considérer que $s_K = 0$.

On note, pour $n \in \mathbb{N}, k \in \{0, \dots, K\}, p_{n,k}$ l'effectif de la population dans l'état k à l'instant n ,

$$P_n = \begin{pmatrix} p_{n,0} \\ p_{n,1} \\ \vdots \\ p_{n,K} \end{pmatrix}$$

le vecteur de $\mathbb{R}^{K+1}$ codant la structure de la population à l'instant n .

On a donc, P_0 , la structure de la population à l'instant initial étant donnée, la relation de récurrence

$$\forall n \in \mathbb{N}, P_{n+1} = L.P_n$$

où

$$L = \begin{pmatrix} 1 - s_0 & 1 - s_1 & 1 - s_2 & \dots & 1 - s_{K-1} & 1 \\ s_0 & 0 & 0 & \dots & 0 & 0 \\ 0 & s_1 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & s_2 & \ddots & & \vdots \\ 0 & & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & s_{K-1} & 0 \end{pmatrix}$$

Partie A : Propriétés spectrales de la matrice L .

7) Quel est le rang de L ? 0 est-il une valeur propre de L ?

8) On note, pour $n \in \mathbb{N}$, $t_n = \sum_{k=0}^K p_{n,k}$ l'effectif total de la population. Vérifier que $t_n = (1 \dots 1) \cdot P_n$, calculer $(1 \dots 1) \cdot L$ et en déduire que $\forall n \in \mathbb{N}, t_n = t_0$.

- 9) a) Montrer qu'en tout généralité, une matrice carrée A et sa transposée A^T ont mêmes valeurs propres.
Indication : On pourra baser le raisonnement sur le fait qu'une matrice carrée est inversible si, et seulement si, sa transposée l'est.
- b) En déduire, sans calcul supplémentaire, que 1 est valeur propre de L .

10) On pose, pour $k \in \{1, \dots, K\}$, $\pi_k = \prod_{\ell=0}^{k-1} s_\ell$, $\pi_0 = 1$ et $\Pi = \begin{pmatrix} 1 \\ \pi_1 \\ \vdots \\ \pi_K \end{pmatrix}$.

Montrer que E_1 , l'espace propre de L associé à 1 est de dimension 1 et $E_1 = \text{Vect}(\Pi)$.

11) On rappelle qu'on a posé $s_K = 0$ et donc que $1 - s_K = 1$.

Montrer que $\lambda \in \mathbb{C}$ est valeur propre de L si, et seulement si, $\sum_{k=0}^K \pi_k \cdot (1 - s_k) \frac{1}{\lambda^{k+1}} = 1$

En déduire que si $\lambda \in \mathbb{C}$ est valeur propre de L alors $\sum_{k=0}^K \pi_k \cdot (1 - s_k) \frac{1}{|\lambda|^{k+1}} \geq 1$.

On admet qu'on a égalité dans l'inégalité précédente si, et seulement si, λ est réel positif.

12) Donner sens de variation et limites aux bornes du domaine de définition de la fonction F définie sur $]0, +\infty[$ par

$$\forall x > 0, F(x) = \sum_{k=0}^K \pi_k \cdot (1 - s_k) \frac{1}{x^{k+1}}$$

En déduire que : si $\lambda \in \mathbb{C}$, $\lambda \neq 1$, est valeur propre de L alors $|\lambda| < 1$.

Partie B : Etude asymptotique de la suite $(P_n)_{n \in \mathbb{N}}$

On suppose dans toute cette partie que L est diagonalisable sur $\mathbb{C}$ et on note $Q \in \mathcal{M}_{K+1}(\mathbb{C})$ une matrice inversible telle que $Q^{-1} \cdot L \cdot Q$ vaut D , une matrice diagonale. On suppose que la première colonne de Q est donnée par le vecteur Π défini à la question 10).

13) Démontrer, par une récurrence rédigée, que $\forall n \in \mathbb{N}, P_n = Q \cdot D^n \cdot Q^{-1} \cdot P_0$ et donner la forme de D^n

14) Décrire la limite lorsque $n \rightarrow +\infty$ de chaque entrée de la matrice D^n .

On signale que, par une légère extension de la notion de limite, on dit qu'une suite de nombres complexes $(z_n)_{n \in \mathbb{N}}$ «admet un nombre complexe ℓ pour limite lorsque $n \rightarrow +\infty$ » si la suite à valeurs réelles $(|z_n - \ell|)_{n \in \mathbb{N}}$ admet pour limite 0. Comme la notion de limite des suites réelles, cette notion de limite est «compatible» avec les opérations algébriques habituelles sur les limites : sommes/différences/produits.

15) Argumenter en faveur du résultat suivant : Lorsque $n \rightarrow +\infty$, chaque composante de P_n admet une limite (dans $\mathbb{C}$) et celle-ci est la composante correspondante du vecteur $P_\infty = \frac{t_0}{\sigma} \cdot \Pi$, où σ est le nombre, seul composante de la matrice 1×1 : $(1 \dots 1) \cdot \Pi$.

$$\text{C'est à dire : } (\sigma) = (1 \dots 1) \cdot \Pi \quad \text{et} \quad P_\infty = \frac{t_0}{\sigma} \cdot \Pi.$$

Partie C : Implémentation informatique

On cherche à développer les outils pour implémenter informatiquement dans un script Python les calculs liés au modèle d'évolution de LESLIE décrit précédemment.

16) Ecrire une fonction Python d'entête `Leslie(S)`, qui étant donné une liste Python S contenant les K valeurs $s_0, \dots, s_{K-1}$ retourne la matrice L sous forme de tableau Numpy de type `ndarray` de `shape (K+1, K+1)`.

17) Ecrire une fonction Python d'entête `VP(S)`, qui étant donné la liste Python S décrite en 16) retourne le vecteur Π définie en 10) sous forme d'un `ndarray` de `shape (K+1)`.

- 18) Pour n un entier Python quelconque représenté en machine par n , de type `int`, L une matrice carrée représentée par un tableau `L` Numpy de type `ndarray` de `shape` (K, K) , peut-on calculer L^n par l'instruction `L**n`? Sinon, que calcule cette instruction?
- 19) On propose l'extrait de script à analyser en fin de question.
- Expliquer ce que calcule la boucle contenant l'instruction `P = np.dot(L, P)`, comment, à partir de quelles données et ce que contient P en sortie de boucle.
 - Expliquer ce qui est calculé dans la variable `Pinfini`, comment, à partir de quelles données. Peut-on comparer les valeurs de P et de `Pinfini`?

```
L = Leslie(S)    # S est donné juste avant
P = P0           # P0 est donné juste avant
for k in range(50):
    P = np.dot (L, P)

vp, vecp = np.linalg.eig(L)
# la plus grande valeur propre en module est en vp[0]
# le vecteur propre correspondant est en vecp[:, 0]
t0 = P0.sum()
sigma = vecp[:, 0].sum()
Pinfini = t0/sigma*vecp[:, 0]
```

- 20) a) On a perdu tous nos scripts, toutes nos données patiemment récupérées (la liste `S` et `P0`) et on a juste retrouvé la valeur du vecteur `Pinfini`. Expliquer par des mots ou un bout de script comment retrouver : la population totale t_0 et les valeurs des s_k .
- b) La valeur retrouvée est :

```
Pinfini = [73.20644217+0.j 21.96193265+0.j 4.39238653+0.j 0.43923865+0.j]
```

Déterminer la population totale t_0 et les valeurs des s_k .

Annexe : Commandes Python

On suppose que le module `math` est importé via `from math import *`.

On suppose que le module `numpy` est importé via `import numpy as np`.

On suppose que le module `random` est importé via `import random as rd`.

Interprétation	Python
Transforme une liste L en tableau <code>numpy</code>	<code>np.array(L)</code>
Matrice nulle de taille $n \times p$	<code>np.zeros([n,p])</code>
Coefficient d'indice (i, j) de la matrice A	<code>A[i, j]</code>
Dimensions de la matrice A	<code>d = np.shape(A)</code>
- nombre de lignes	<code>d[0]</code>
- nombre de colonnes	<code>d[1]</code>
Pour A et B matrices de tailles compatibles :	
- addition et soustraction	<code>A + B , A - B</code>
- multiplication matricielle	<code>np.dot(A, B)</code>
Pour A une matrice :	
somme de tous les coefficients de A	<code>A.sum()</code>
Valeur absolue d'un réel x	<code>abs(x)</code>
Logarithme népérien de x	<code>log(x)</code>
Nombre entier aléatoire compris entre a et b inclus	<code>rd.randint(a, b)</code>
Nombre réel aléatoire compris entre 0 et 1	<code>rd.random()</code>