Concours ECRICOME Prépa, France

Session 2026, filière ECG

Mathématiques appliquées

Durée : 4 heures

Ce que traite le sujet

- Exercice 1
- Exercice 2
- Exercice 3

Sujet reproduit sans retouche à partir de la page suivante.

Résumé des parties, chapitres de cours liés et autres années :
prepahelp.com/annales/ecricome/ecg/2026/maths-appli

Prepa Help ne revendique aucun droit sur le sujet, propriété de l'organisateur du concours.

ecricome

CONCOURS D'ADMISSION 2026

prépas ECG/ECT

1

Mathématiques Appliquées

Série ECG

Mardi 14 avril 2026 de 8h00 à 12h00
Durée : 4 heures

Candidats bénéficiant de la mesure « Tiers-temps » : 8h00 – 13h20

L'énoncé comporte 11 pages.

INSTRUCTIONS

Tous les feuillets doivent être identifiables et numérotés par le candidat.

Aucun document n'est permis, aucun instrument de calcul n'est autorisé. La règle graduée est autorisée.

Conformément au règlement du concours, l'usage d'appareils communicants ou connectés est formellement interdit durant l'épreuve.

Les candidats sont invités à soigner la présentation de leur copie, à mettre en évidence les principaux résultats, à respecter les notations de l'énoncé et à fournir des raisonnements clairs, précis et concis. Le soin apporté à l'ensemble de la copie et la lisibilité entrent pour une bonne part dans l'évaluation de la copie. Le jury tiendra compte de la qualité rédactionnelle et de la maîtrise orthographique dans le barème de l'épreuve.

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

Ce document est la propriété d'ECRICOME, le candidat est autorisé à le conserver à l'issue de l'épreuve.

Le candidat dispose d'une annexe Python et SQL en pages 9, 10, 11 et 12.

Le concours ECRICOME PRÉPA est une marque déposée. Toute reproduction du sujet est interdite. Copyright ©ECRICOME - Tous droits réservés

Tournez la page s.v.p.

Dans les questions faisant intervenir des instructions en langage **`Python`**, *on prendra soin d'importer les bibliothèques nécessaires lors de leur première utilisation.*
Pour traiter les questions d'informatique, les candidats sont invités à ***se référer aux annexes fournies en fin de sujet****. Ils ne sont pas limités à l'utilisation des seules fonctions mentionnées dans ces annexes.*

EXERCICE 1

Soit n un entier naturel supérieur ou égal à 2.
Toutes les variables aléatoires intervenant dans cet exercice sont définies sur un espace probabilisé $(\Omega, \mathscr{A}, \mathbb{P})$.
Un particulier souhaite assurer un nouveau bien. Il consulte pour cela un site internet qui diffuse un classement de n compagnies d'assurances, de la mieux classée par les utilisateurs (compagnie n°1) à la moins bien classée (compagnie n°n). Ce classement ne varie pas d'année en année.

La première année, le particulier choisit une compagnie d'assurances au hasard, avec équiprobabilité, parmi les n compagnies proposées.
On note X la variable aléatoire égale au numéro, dans le classement, de la compagnie choisie.

L'année suivante, le particulier choisit au hasard de manière équiprobable une compagnie parmi toutes celles qui sont au moins aussi bien classées que celle qu'il avait choisie l'année précédente (il peut éventuellement être amené à conserver la même compagnie). On note alors Y la variable aléatoire égale au numéro, dans le classement, de la compagnie choisie pour cette seconde année.

1. Identifier la loi de la variable aléatoire X. Donner son espérance et sa variance.
2. Montrer que la loi conjointe du couple (X, Y) est donnée par :

$$\forall (j,k) \in [\![1,n]\!]^2, \qquad \mathbb{P}\big([X=j] \cap [Y=k]\big) = \begin{cases} \dfrac{1}{nj} & \text{si} \quad k \leqslant j, \\ 0 & \text{si} \quad k \geqslant j+1. \end{cases}$$

3. En déduire la valeur, en fonction de n, de la somme double suivante : $\displaystyle\sum_{k=1}^{n}\sum_{j=k}^{n}\frac{1}{j}$.
4. Pour tout entier naturel k de $[\![1,n]\!]$, exprimer $\mathbb{P}(Y=k)$ sous la forme d'une somme.
5. (a) Montrer que $\displaystyle\sum_{k=1}^{n}\sum_{j=k}^{n}\frac{k}{j} = \frac{1}{2}\sum_{j=1}^{n}(j+1)$.
   (b) Justifier que Y admet une espérance et montrer que l'espérance de Y vaut $\dfrac{n+3}{4}$.
6. (a) Justifier que l'espérance de la variable aléatoire XY existe et vérifie :

$$\boldsymbol{E}(XY) = \frac{1}{2n}\sum_{j=1}^{n} j(j+1).$$

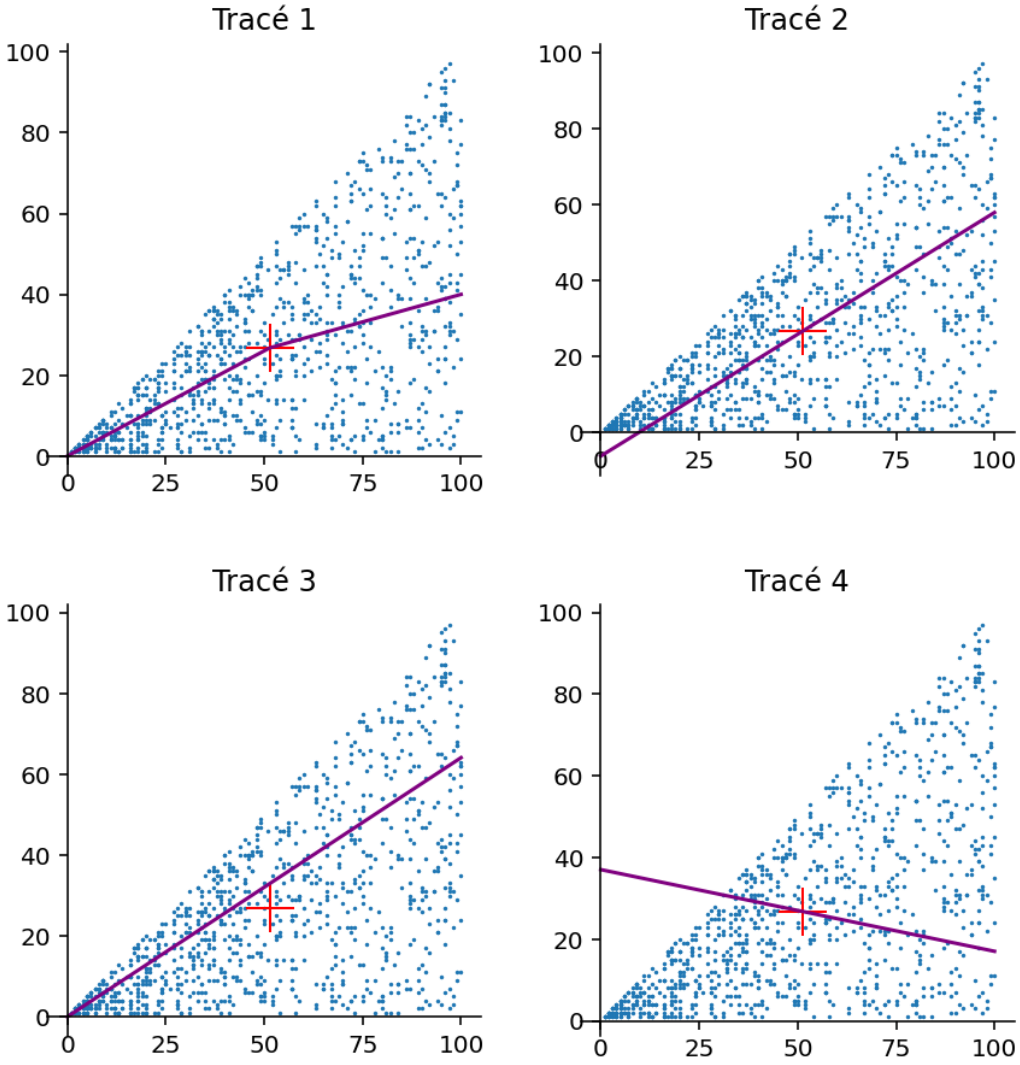
   (b) En déduire l'expression de $\boldsymbol{E}(XY)$ en fonction de n.
   (c) Déterminer la covariance du couple (X, Y) en fonction de n.
7. Les variables aléatoires X et Y sont-elles indépendantes ?
8. (a) Écrire une fonction en langage `Python` nommée `simulXY`, prenant en argument d'entrée l'entier n, et renvoyant une réalisation du couple de variables aléatoires (X, Y).
   (b) On considère une série statistique à deux variables (x, y).
   Expliquer ce que renvoie la fonction `Myst` suivante qui prend en argument d'entrée la série statistique (x, y) sous forme de tableau `Numpy` à deux lignes.

```
import numpy as np

def Myst(Tab):
    M=np.cov(Tab)
    R=M[0,1]/np.sqrt(M[0,0]*M[1,1])
    return(R)
```

(c) On a représenté ci-après plusieurs tracés figurant le nuage de points $(x_i, y_i)_{i\in[\![1,1000]\!]}$ correspondant à 1000 réalisations du couple (X, Y) pour $n = 100$ et le point moyen de la série statistique (i.e. le point de coordonnées $(\bar{x}, \bar{y})$) figuré par une croix rouge.
Parmi ces tracés figure celui de la droite d'ajustement affine par la méthode des moindres carrés.

Déterminer lequel de ces tracés correspond à la droite d'ajustement affine par la méthode des moindres carrés. Une réponse justifiée est attendue.



9. On note A_n l'événement : « le particulier choisit la même compagnie d'assurances lors des deux années considérées ».
   (a) Exprimer l'événement A_n à l'aide des variables aléatoires X et Y.
   (b) Montrer que : $\mathbb{P}(A_n) = \dfrac{1}{n}\displaystyle\sum_{j=1}^{n}\frac{1}{j}$.
   (c) Montrer que, pour tout entier naturel j non nul, $\dfrac{1}{j+1} \leqslant \ln(j+1) - \ln(j) \leqslant \dfrac{1}{j}$.
   (d) En déduire que $\dfrac{\ln(n+1)}{n} \leqslant \mathbb{P}(A_n) \leqslant \dfrac{1}{n} + \dfrac{\ln(n)}{n}$.
   (e) Montrer que $\mathbb{P}(A_n) \underset{n\to+\infty}{\sim} \dfrac{\ln(n)}{n}$.

Tournez la page s.v.p.

10. Pour répondre aux questions suivantes, on pourra se référer aux commandes SQL rappelées en annexe.
Dans une base de données, on dispose d'une table `adherent` dont chaque enregistrement correspond à un individu assuré, et qui recense les compagnies d'assurances choisies en 2025 et en 2026 par chaque assuré. Ses attributs sont les suivants :
   - `assure` (de type `INTEGER`) : un numéro permettant d'identifier l'assuré.
   - `id_2025` (de type `INTEGER`) : un code permettant d'identifier la compagnie d'assurances à laquelle l'assuré a souscrit en 2025.
   - `id_2026` (de type `INTEGER`) : un code permettant d'identifier la compagnie d'assurances à laquelle l'assuré a souscrit en 2026.

   On dispose également d'une table `compagnie` dont chaque enregistrement correspond à une compagnie et qui associe pour chaque compagnie son nom et son identifiant. Ses attributs sont les suivants :
   - `id` (de type `INTEGER`) : un code permettant d'identifier la compagnie d'assurances.
   - `nom` (de type `TEXT`) : le nom de la compagnie d'assurances.

   Le format de la table `adherent` est illustré dans le tableau ci-dessous.

assure	id_2025	id_2026
1003	3	3
1004	4	2
1005	1	1
1006	3	4
1007	2	4

   Le format de la table `compagnie` est illustré dans le tableau ci-dessous.

id	nom
1	Assurplus
2	Garantia
3	SecurVie
4	MutuelPro

   (a) Écrire une requête SQL permettant d'afficher le nombre d'assurés ayant conservé la même compagnie d'assurances entre 2025 et 2026 .

   (b) Expliquer ce que réalise la commande SQL suivante :

```
SELECT assure,  nom
FROM adherent INNER JOIN compagnie
ON adherent.id_2026 = compagnie.id;
```

   (c) Dans sa base de données, le site internet qui diffuse les avis sur les compagnies d'assurances dispose d'une table `avis`, dans laquelle chaque enregistrement correspond à un avis déposé par un utilisateur concernant une compagnie. La table `avis` est composée des attributs suivants :
   - `id` (de type `INTEGER`) : un numéro permettant d'identifier l'avis.
   - `usr` (de type `TEXT`) : le nom d'utilisateur de la personne ayant déposé l'avis.
   - `compagnie` (de type `INTEGER`) : un code permettant d'identifier la compagnie d'assurances évaluée dans l'avis.
   - `note` (de type `INTEGER`) : la note sur 10 attribuée par l'utilisateur à la compagnie évaluée.
   - `commentaire` (de type `TEXT`) : l'appréciation déposée par l'utilisateur sur la compagnie.

   Le format de la table `avis` est illustré dans le tableau ci-dessous.

id	usr	compagnie	note	commentaire
5001	alice92	1	8	Très bon service !
5002	karimBoss	2	6	Service correct.
5004	david123	3	7	Bon rapport qualité-prix.
5005	emma_p	2	5	Déçue par le service.
5007	alice92	3	8	Très professionnels.
5008	david123	4	9	Service impeccable.

   Écrire une requête SQL permettant d'obtenir, pour chaque compagnie d'assurances évaluée, la note moyenne que les utilisateurs lui ont attribuée. On affichera pour chaque enregistrement l'attribut `compagnie`, ainsi que l'attribut `note` correspondant désormais à la moyenne de toutes les notes obtenues par la compagnie.

   (d) On appelle `moyenne` la table issue de la requête décrite à la question précédente.
   Écrire une requête SQL permettant de classer les compagnies d'assurances, de la mieux notée à la moins bien notée.

EXERCICE 2

On considère la fonction f définie par

$$f(x) = \ln\left(\frac{1}{1-x}\right).$$

Partie I

1. Déterminer le domaine de définition de f.

Dans les questions qui suivent, on note I le domaine de définition de f.

2. Calculer les limites de f aux bornes de son domaine de définition.
3. Dresser le tableau de variations de f sur I.
4. (a) Déterminer le développement limité à l'ordre 2 de f en 0.
   (b) En déduire une équation de la tangente à la courbe représentative de f au point d'abscisse 0, ainsi que la position relative de la courbe par rapport à sa tangente au voisinage de ce point.
5. Représenter, dans un même repère orthonormé, l'allure de la courbe représentative de f et sa tangente au point d'abscisse 0.

Partie II

Pour tout entier naturel n non nul et tout réel x, on pose :

$$S_n(x) = \sum_{k=1}^{n} \frac{x^k}{k}.$$

6. (a) Donner la nature de la série $\displaystyle\sum_{k\geqslant 1} \frac{1}{k}$.
   (b) Étudier la monotonie de la suite $(S_n(1))_{n\in\boldsymbol{N}^*}$.
   (c) En déduire que $\lim\limits_{n\to+\infty} S_n(1) = +\infty$.
7. Montrer que, pour tout réel x de $]-1,1[$, la série $\displaystyle\sum_{k\geqslant 1} \frac{x^k}{k}$ converge absolument.
8. (a) Montrer que les suites $\big(S_{2n}(-1)\big)_{n\in\boldsymbol{N}^*}$ et $\big(S_{2n+1}(-1)\big)_{n\in\boldsymbol{N}^*}$ sont adjacentes.
   (b) En déduire que la suite $\big(S_n(-1)\big)_{n\in\boldsymbol{N}^*}$ est convergente.

   On admet que $\lim\limits_{n\to+\infty} S_n(-1) = \displaystyle\sum_{k=1}^{+\infty} \frac{(-1)^k}{k} = -\ln(2)$.
   (c) En utilisant la monotonie des suites $\big(S_{2n}(-1)\big)_{n\in\boldsymbol{N}^*}$ et $\big(S_{2n+1}(-1)\big)_{n\in\boldsymbol{N}^*}$, montrer que

   $$\forall n \in \mathbb{N}^*, \quad \Big|\ln(2) + S_{2n}(-1)\Big| \leqslant S_{2n}(-1) - S_{2n+1}(-1)$$

   (d) Montrer que, pour tout entier naturel n non nul, $\big|\ln(2) + S_n(-1)\big| \leqslant \dfrac{1}{n}$.
9. Dans cette question uniquement, x est un réel strictement supérieur à 1.
   (a) Déterminer la nature de la série $\displaystyle\sum_{k\geqslant 1} \frac{x^k}{k}$.
   (b) En déduire la limite de la suite $\big(S_n(x)\big)_{n\in\boldsymbol{N}^*}$.

Pour tout $x \in [-1,1[$, on pose $S(x) = \lim\limits_{n\to+\infty} S_n(x) = \displaystyle\sum_{k=1}^{+\infty} \frac{x^k}{k}$.

Tournez la page s.v.p.

Partie III

Dans toute cette partie, a désigne un réel fixé dans l'intervalle $]-1,1[$.
Pour tout entier naturel n non nul, on considère l'intégrale notée $R_n(a)$ définie par :

$$R_n(a) = \int_0^a \frac{(a-t)^n}{(1-t)^{n+1}}\,dt\,.$$

On considère à nouveau la fonction f définie à la **Partie I** ainsi que la suite $\left(S_n(x)\right)_{n\geqslant 1}$ et la fonction S définies à la **Partie II**.

10. (a) Déterminer en fonction de a les valeurs de $\displaystyle\int_0^a \frac{1}{1-t}\,\mathrm{d}t$ et $\displaystyle\int_0^a \frac{1}{(1-t)^2}\,\mathrm{d}t$.

    (b) Montrer que $f(a) = a + R_1(a)$.
    Indication : On pourra écrire $a - t = (a-1) + (1-t)$.

11. Montrer par intégration par parties que, pour tout entier naturel n non nul, $R_n(a) = \dfrac{a^{n+1}}{n+1} + R_{n+1}(a)$.

12. En utilisant les résultats des questions 10b et 11, montrer que, pour tout entier naturel n non nul,

$$f(a) = S_n(a) + R_n(a)\,.$$

    Indication : On pourra procéder par récurrence.

13. Dans cette question uniquement, on suppose que $a \in [0,1[$.

    (a) Montrer que pour tout réel t de $[0,a]$, $0 \leqslant \dfrac{a-t}{1-t} \leqslant a$.

    (b) Montrer que, pour tout entier naturel n non nul, $0 \leqslant R_n(a) \leqslant \dfrac{a^{n+1}}{1-a}$.

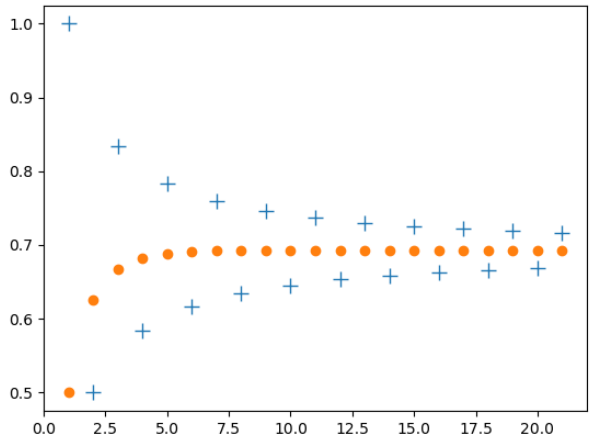
    (c) En déduire que $f(a) = S(a)$.

14. (a) En utilisant les résultats des questions précédentes, justifier que, pour tout entier naturel n non nul,

$$\left|\ln(2) - \sum_{k=1}^{n} \frac{1}{k2^k}\right| \leqslant \frac{1}{2^n}\,.$$

    (b) Écrire une fonction en langage `Python`, nommée `Val`, prenant en argument `eps` désignant le réel ε strictement positif, qui renvoie une valeur approchée de $\ln(2)$ à ε près.

    (c) Dans la figure suivante sont représentés les tracés des premiers termes des suites $\left(S_n\left(\dfrac{1}{2}\right)\right)_{n\in\boldsymbol{N}^*}$ et $\left(-S_n\left(-1\right)\right)_{n\in\boldsymbol{N}^*}$.
    Identifiez les suites dans cette figure.



EXERCICE 3

Partie I

On considère le problème de Cauchy $(\mathscr{C})$ suivant, d'inconnues $x : \mathbb{R} \longrightarrow \mathbb{R}$ et $y : \mathbb{R} \longrightarrow \mathbb{R}$ dérivables :

$$(\mathscr{C}) \qquad \forall t \in \mathbb{R}, \qquad \begin{cases} x'(t) &= x(t) + y(t) \\ y'(t) &= -x(t) - y(t) \\ x(0) &= 1 \\ y(0) &= 1 \end{cases}$$

Dans cette partie, on suppose que x et y sont solutions de $(\mathscr{C})$.
On pose, pour tout t réel, $s(t) = x(t) + y(t)$.

1. Déterminer une expression de la fonction s'.
2. En déduire une relation entre $x(t)$ et $y(t)$ pour tout réel t.
3. Déterminer l'unique couple de solutions du problème de Cauchy $(\mathscr{C})$.

Partie II

On considère les matrices $A = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}$, $J = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ et $\mathrm{P} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$.

4. Montrer que la matrice P est inversible et calculer son inverse.
5. Montrer que A possède une unique valeur propre que l'on déterminera.
6. La matrice A est-elle diagonalisable ?
7. Montrer que $J = P^{-1}AP$.

Dans la suite de cette partie, on considère le système différentiel $(\mathscr{S})$ défini par :

$$(\mathscr{S}) \qquad \forall t \in \mathbb{R}, \qquad \begin{cases} x'(t) &= x(t) + y(t) \\ y'(t) &= -x(t) - y(t) \end{cases}$$

Pour toutes fonctions x et y dérivables sur $\mathbb{R}$, on pose, pour tout réel t, $X(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$ et $X'(t) = \begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix}$.

On pose également, pour tout réel t, $Y(t) = P^{-1}X(t) = \begin{pmatrix} u(t) \\ v(t) \end{pmatrix}$ et $Y'(t) = \begin{pmatrix} u'(t) \\ v'(t) \end{pmatrix}$.

8. Déterminer les états d'équilibre du système $(\mathscr{S})$.
9. Montrer que x et y sont solutions de $(\mathscr{S})$ si et seulement si, pour tout réel t, $Y'(t) = JY(t)$.
10. Déterminer l'ensemble des solutions $Y(t) = \begin{pmatrix} u(t) \\ v(t) \end{pmatrix}$ du système différentiel $Y'(t) = JY(t)$.
11. En déduire les solutions du système différentiel $(\mathscr{S})$.

Tournez la page s.v.p.

Partie III

Soit n un entier naturel non nul.
Une matrice de $\mathscr{M}_n(\mathbb{R})$ est dite **nilpotente** quand il existe un entier naturel k tel que $M^k = 0_n$, où 0_n désigne la matrice nulle de $\mathscr{M}_n(\mathbb{R})$.
On admet que, pour toute matrice M **nilpotente** de $\mathscr{M}_n(\mathbb{R})$, il existe un entier naturel p non nul et inférieur ou égal à n tel que $M^{p-1} \neq 0_n$ et $\forall k \geqslant p$, $M^k = 0_n$. p est appelé **l'indice de nilpotence** de M.
On considère le système différentiel $(\mathscr{E})$ à n équations, d'inconnues $x_1, x_2, \ldots, x_n$ dérivables sur $\mathbb{R}$, défini par :

$$(\mathscr{E}) \qquad \forall t \in \mathbb{R}, \quad X'(t) = NX(t), \qquad \text{où} \quad X(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix}, \quad X'(t) = \begin{pmatrix} x_1'(t) \\ x_2'(t) \\ \vdots \\ x_n'(t) \end{pmatrix}$$

et N est une matrice nilpotente non nulle de $\mathscr{M}_n(\mathbb{R})$. On note p l'indice de nilpotence de N.

12. Écrire une fonction, en langage `Python`, nommée `Nil`, qui prend en entrée une matrice M et qui renvoie p lorsque la matrice M est nilpotente d'indice de nilpotence p et `0` sinon.
13. Montrer que la matrice N possède une unique valeur propre que l'on déterminera.
14. Justifier par l'absurde que la matrice N n'est pas diagonalisable.

Pour tout réel t, on définit la matrice $B(t)$ de $\mathscr{M}_n(\mathbb{R})$ par

$$B(t) = \sum_{k=0}^{p-1} \frac{t^k}{k!} N^k,$$

et la matrice colonne $X(t)$ de $\mathscr{M}_{n,1}(\mathbb{R})$ par

$$X(t) = B(t)\ X_0$$

où X_0 est une matrice colonne de $\mathscr{M}_{n,1}(\mathbb{R})$.

15. Justifier que, pour tout réel t,
$$X'(t) = \sum_{k=1}^{p-1} \frac{t^{k-1}}{(k-1)!} N^k\ X_0.$$
16. En déduire que $X(t)$ est l'unique solution du problème de Cauchy défini par le système différentiel $(\mathscr{E})$ et la condition initiale $X(0) = X_0$.
17. Recopier et compléter, en langage `Python`, la fonction suivante qui prend en entrée la matrice `N` et un réel `t` et qui renvoie $B(t)$.

```
def B(N,t):
    if Nil(N)==...:
        return("N n'est pas nilpotente")
    else:
        T=np.eye(len(N))
        S=T
        for k in range(....):
            T=.../...*np.dot(T,...)
            S=S+T
        return S
```

Annexe A - Fonctions `Python` utiles

Manipulation de listes. On suppose que `L` désigne une liste à `n` éléments.

- L'opérateur de concaténation `+`, appliqué entre deux listes, renvoie la liste obtenue en plaçant les éléments de la seconde liste à la suite de ceux de la première liste.
  Par exemple, `[1, 2, 5] + [4, 3]` renvoie la liste `[1, 2, 5, 4, 3]`.
- L'opérateur `*`, appliqué entre une liste `L` et un entier `n`, renvoie la liste obtenue en concaténant `n` fois la liste `L` avec elle-même.
  Par exemple, `[1, 4, 2]*3` renvoie la liste `[1, 4, 2, 1, 4, 2, 1, 4, 2]`.
- La fonction `len` prend en argument d'entrée une liste et renvoie le nombre d'éléments dans cette liste. Lorsqu'on applique cette fonction à un tableau `Numpy` à plusieurs lignes, elle renvoie le nombre de lignes du tableau.
- La commande `L.append(x)` permet d'inclure l'élément `x` à la fin de la liste `L`.
- Pour tout entier `i` entre `0` et `n-1`, la commande `L.pop(i)` retire de la liste `L` l'élément situé à la position `i`, et **renvoie** sa valeur.
  Par exemple, à l'issue des instructions

```
L = [5, 4, 8, 1]
a = L.pop(1)
```

  la liste `L` vaut `[5, 8, 1]` et la variable `a` vaut `4`.

La bibliothèque `numpy`.

- Exemple d'importation : `import numpy as np`.
- Les opérations `+`, `-`, `*`, `/`, `**`, lorsqu'elles sont possibles, peuvent être réalisées entre deux tableaux `Numpy` de tailles compatibles et agissent alors **coefficient par coefficient**.
- La fonction `np.eye` prend en argument d'entrée un entier n et renvoie la matrice identité sous la forme d'un tableau `Numpy`.
- La fonction `np.zeros` prend en arguments d'entrée une liste d'entiers $[n, p]$ et renvoie la matrice de n lignes et p colonnes dont tous les coefficients sont nuls sous la forme d'un tableau `Numpy`.
- Si M et N sont deux tableaux `Numpy` la commande `(M==N).all()` renvoie `True` si $M = N$ et `False` sinon.
- La fonction `np.dot` prend en argument d'entrée deux tableaux `Numpy` de nombres M et N et renvoie le tableau `Numpy` correspondant au produit matriciel NM lorsque celui-ci existe.
- La fonction `np.mean` prend en argument d'entrée un tableau `Numpy` de nombres, et renvoie la moyenne des éléments du tableau.
- La fonction `np.var` prend en argument d'entrée un tableau `Numpy` de nombres, et renvoie la variance des éléments du tableau.
- La fonction `np.std` prend en argument d'entrée un tableau `Numpy` de nombres, et renvoie l'écart-type des éléments du tableau.
- La fonction `np.cov` prend en argument une série statistique à deux variables (x, y) sous forme de tableau `Numpy` à deux lignes et renvoie, sous forme de tableau `Numpy`, la matrice suivante :

$$\begin{pmatrix} v_x & s_{x,y} \\ s_{x,y} & v_y \end{pmatrix},$$

  où v_x , v_y et $s_{x,y}$ désignent respectivement la variance de la série statistique x , la variance de la série statistique y et la covariance empirique de la série statistique double (x, y) .

La bibliothèque matplotlib.pyplot .

- Exemple d'importation : `import matplotlib.pyplot as plt`.
- La fonction `plt.plot` prend en arguments d'entrée deux listes de même longueur ou deux tableaux `Numpy` `x` et `y` à une ligne et de même longueur, et renvoie une figure constituée de la ligne brisée joignant les points du plan de coordonnées (x_i, y_i) , où x_i et y_i sont respectivement les coefficients des tableaux/listes `x` et `y`.
- La fonction `plt.show`, employée sans argument d'entrée, permet l'affichage d'une figure préalablement tracée, par exemple avec les fonctions `plt.plot`.

Tournez la page s.v.p.

Le module `numpy.random`.

- Exemple d'importation : `import numpy.random as rd`.
- La fonction `rd.randint` prend deux entiers n et p (avec $p > n$) en arguments d'entrée et renvoie une réalisation aléatoire de la loi uniforme discrète sur $[\![n, p-1]\!]$.

Le module `numpy.linalg`.

- Exemple d'importation : `import numpy.linalg as al`.
- La fonction `al.inv` prend un tableau `Numpy` M en argument d'entrée et renvoie l'inverse de M (au sens matriciel) sous la forme d'un tableau `Numpy` lorsqu'il existe et lève une erreur sinon.
- La fonction `al.matrix_power` prend un tableau `Numpy` M et un entier k en argument d'entrée et renvoie la matrice M^k sous la forme d'un tableau `Numpy` lorsque M est carrée et lève une erreur sinon.

Annexe B - Commandes SQL

La commande `ORDER BY`. La commande `ORDER BY`, placée en fin de requête et suivie d'un nom de colonne, permet de trier les résultats demandés dans l'ordre croissant des valeurs de la colonne spécifiée, ou dans l'ordre alphabétique s'il s'agit de données de type `TEXT`.
On peut ajouter le mot-clé `DESC` à la fin de la commande pour trier les données dans l'ordre décroissant.
Exemple : Une table `etablissement` contient les données suivantes concernant plusieurs établissements scolaires, leur nombre d'élèves et la ville où ils se situent.

identifiant	nom	nombre_eleves	ville
1	Edouard Herriot	887	Livry-Gargan
2	Diderot	220	Lyon
3	Edouard Herriot	808	Lyon
4	Louise Michel	653	Champigny-sur-Marne
5	Diderot	1200	Paris

- Pour afficher les données en triant les noms d'établissement dans l'ordre alphabétique, on peut utiliser la requête suivante :

```
SELECT * FROM etablissement
ORDER BY nom;
```

- Pour afficher les noms des établissements dans l'ordre décroissant de nombre d'élèves, on peut utiliser la requête suivante :

```
SELECT nom FROM etablissement
ORDER BY nombre_eleves DESC;
```

La fonction `COUNT()`. La fonction d'agrégation `COUNT()` permet de connaître le nombre d'enregistrements d'une table, vérifiant éventuellement une certaine condition.
Nous donnons ci-dessous deux exemples d'utilisation de la fonction `COUNT()`, en considérant une table nommée `ma_table` comportant deux colonnes `colonne_1` et `colonne_2`.

- La requête suivante renvoie le nombre total d'enregistrements dans `ma_table` :

```
SELECT COUNT(*) FROM ma_table;
```

- La requête suivante renvoie le nombre d'enregistrements de `ma_table` vérifiant la condition `cond` :

```
SELECT COUNT(*) FROM ma_table
WHERE cond;
```

La fonction d'agrégation `AVG()`. La fonction `AVG()` permet de calculer la moyenne des valeurs d'une colonne dans une table.
Par exemple, si on considère la table nommée `table` contenant les enregistrements suivants :

colonne_1	colonne_2	colonne_3	colonne_4
1	69	Lyon	4
2	31	Toulouse	8
3	54	Nancy	5
4	64	Saint-Jean-de-Luz	17
5	44	Nantes	6

alors la requête suivante

```
SELECT AVG(colonne_4)
FROM table
WHERE colonne_1 <= 3;
```

affiche la moyenne des valeurs de `colonne_4` des trois premiers enregistrements : `5.6667` c'est-à-dire $\frac{4+8+5}{3}$.

La commande `GROUP BY`. La commande `GROUP BY` permet de regrouper tous les enregistrements dont la valeur d'un attribut donné est identique, en appliquant une fonction à chaque groupe d'enregistrements.
Exemple : Une table `etablissement` contient les données suivantes concernant plusieurs établissements scolaires, leur nombre d'élèves et la ville où ils se situent.

identifiant	nom	nombre_eleves	ville
1	Pierre-Gilles de Gennes	598	Paris
2	Diderot	220	Lyon
3	Edouard Herriot	808	Lyon
4	Paul Valéry	525	Paris
5	Edouard Herriot	887	Livry-Gargan
6	Diderot	1200	Paris

On peut calculer le nombre d'établissements dans chaque ville à l'aide de la requête suivante :

```
SELECT ville, COUNT(*)
FROM etablissement
GROUP BY ville;
```

On obtient alors le résultat suivant :

ville	COUNT(*)
Livry-Gargan	1
Lyon	2
Paris	3

On peut de plus renommer les colonnes de cette nouvelle table :

```
SELECT ville, COUNT(*) AS nombre_etab
FROM etablissement
GROUP BY ville;
```

On obtient alors le résultat suivant :

ville	nombre_etab
Livry-Gargan	1
Lyon	2
Paris	3

Tournez la page s.v.p.

La commande `INNER JOIN`**.** La commande `INNER JOIN` permet de réaliser la jointure de deux tables. Cette commande retourne les enregistrements lorsqu'il y a au moins une ligne dans chaque table qui correspond à la condition.
Exemple : Une table `etablissement` contient pour plusieurs établissements scolaires leur identifiant, leur nom et la ville où ils se situent.

ident	nom	ville
1	Pierre-Gilles de Gennes	Paris
2	Diderot	Lyon
3	Edouard Herriot	Lyon
4	Paul Valéry	Paris
5	Edouard Herriot	Livry-Gargan
6	Diderot	Paris

Une seconde table `effectifs` contient pour plusieurs établissements scolaires leur identifiant et leur nombre d'élèves.

ident	eleves
1	598
2	220
3	808
4	525
5	887
6	1200

On peut alors créer une table contenant pour chaque établissement son nom et son nombre d'élèves :

```
SELECT nom, eleves
FROM etablissement INNER JOIN effectifs
ON etablissement.ident = effectifs.ident;
```

On obtient alors le résultat suivant :

nom	eleves
Pierre-Gilles de Gennes	598
Diderot	220
Edouard Herriot	808
Paul Valéry	525
Edouard Herriot	887
Diderot	1200

Travailler les chapitres de ce sujet

Prepa Help suit le programme d'ECG chapitre par chapitre : cours détaillé, méthodes, pièges, exercices et problèmes corrigés pas à pas, tirés des concours. L'essai gratuit donne accès aux chapitres offerts de ta filière, sans carte bancaire.

Chapitres liés à ce sujet

- Couples de variables aléatoires discrètes finies
- Variables aléatoires finies et lois usuelles
- Statistiques bivariées et régression
- Réels, sommes et produits
- Suites réelles
- Dérivation (et convexité en ECT1)
- Fonctions réelles : limites et continuité
- Éléments propres et réduction

Ouvrir l'essai gratuit

Scanne le code ou clique sur le bouton.

prepahelp.com/#essai

Une question sur le concours ou sur ta filière : **écrire sur WhatsApp**

Les autres années et épreuves : prepahelp.com/annales/ecricome/ecg