Samedi 7 Février 2026

Devoir Sur Table n°4 – Durée : 4h

L'utilisation de la calculatrice, des feuilles/notes de cours ou d'exercices est interdite.
La présentation, la rédaction, la clarté et la précision des raisonnements entreront dans l'appréciation de la copie.
Les résultats non encadrés/soulignés/surlignés ne seront pas pris en compte.

Exercice 1 - 3 questions sur les espaces vectoriels

1. On introduit les vecteurs de $\mathbb{R}^3$: $v_1 = (-1, 1, 0)$, $v_2 = (1, 0, 2)$, $v_3 = (1, 1, \alpha)$, où $\alpha \in \mathbb{R}$ est fixé.
   A quelle condition sur le réel α la famille (v_1, v_2, v_3) est-elle libre ?
2. On considère l'ensemble $S = \left\{u \in \mathbb{R}^{\mathbb{N}} \mid \forall n \in \mathbb{N},\ u_{n+2} = 4u_{n+1} - 4u_n\right\}$.
   (a) Montrer que S est un espace vectoriel.
   (b) Quelle est la forme générale d'une suite appartenant à S ?
   (c) En déduire une base de S composée de deux vecteurs. On justifiera qu'il s'agit bien d'une base.
3. Soit $n \in \mathbb{N}^*$ fixé. On considère l'ensemble $E = \left\{P \in \mathbb{R}_n[X] \mid \int_0^3 P(t)dt = 0\right\}$.
   (a) Montrer que E est un sous-espace vectoriel de $\mathbb{R}[X]$.
   (b) Pour tout $k \in [\![1, n]\!]$, montrer qu'il existe une unique constante c_k (que l'on déterminera) telle que le polynôme $X^k - c_k$ appartient à E.
   (c) En déduire une famille libre composée de n vecteurs de E.
   (d) Montrer que cette famille est en fait une base de E.

Exercice 2 : Tirages dans un grand nombre d'urnes

Dans cet exercice, $N \in \mathbb{N}^*$ et $n \in \mathbb{N}^*$ sont fixés. Pour $k \in [\![0, N]\!]$, on définit : $B_k = \int_0^1 t^k(1-t)^{N-k}dt$.

1. (a) Calculer les valeurs de B_N et B_{N-1} en fonction de N.
   (b) Justifier que pour tout $k \in [\![0, N]\!]$, $B_{N-k} = B_k$.
   (c) Etablir que : $\sum_{k=0}^{N} \binom{N}{k} B_k = 1$.
2. (a) Montrer la relation de récurrence : $\forall k \in [\![1, N]\!]$, $B_k = \frac{k}{N-k+1} B_{k-1}$.
   (b) En déduire, pour tout $k \in [\![0, N]\!]$, l'expression de B_k en fonction de N et de k.
   On fera apparaitre un coefficient binomial.

On dispose de n urnes numérotées de 1 à n.
Pour tout $j \in [\![1, n]\!]$, l'urne numéro j contient j boules blanches et $n - j$ boules noires.
On choisit l'une des urnes uniformément au hasard, puis on tire N boules avec remise dans cette urne.
On note X_n la variable aléatoire égale au nombre de boules blanches obtenues à l'issue des N tirages réalisés.

3. (a) Soit $j \in [\![1, n]\!]$. En admettant qu'on ait choisi l'urne numéro j, quelle serait la loi de X_n ?
   (b) Fort de ce constat, compléter le programme suivant pour que la fonction `tirage` simule une réalisation de la variable aléatoire X_n. On utilisera les instructions `rd.binomial` et `rd.randint`. **On recopiera l'intégralité du programme.**

```
import numpy.random as rd
def tirage(n,N) :
  J = ....
  X = ....
  return X
```

4. Démontrer que : $\forall k \in [\![0, N]\!],\ P(X_n = k) = \dfrac{1}{n}\displaystyle\sum_{j=1}^{n} \binom{N}{k} \left(\frac{j}{n}\right)^k \left(1 - \frac{j}{n}\right)^{N-k}$.

5. (a) Pour un réel $p \in [0, 1]$, rappeler la valeur de la somme $\displaystyle\sum_{k=0}^{N} k\binom{N}{k} p^k (1-p)^{N-k}$.

   (b) En déduire l'expression de l'espérance : $E(X_n) = \dfrac{N(n+1)}{2n}$.

6. Pour finir, on étudie le cas où le nombre n d'urnes est très grand.

   (a) Justifier que pour tout $k \in [\![0, N]\!]$, $\displaystyle\lim_{n\to+\infty} P(X_n = k) = \binom{N}{k} B_k$.

   (b) En déduire la valeur de $\displaystyle\lim_{n\to+\infty} P(X_n = k)$ en fonction de N seulement.
   Lorsque le nombre d'urnes est très grand, que peut-on donc dire de la variable aléatoire X_n ?

Problème

Partie I - Une suite d'intégrales

On considère la suite $(S_n)_{n\in\mathbb{N}}$ définie par : $\forall n \in \mathbb{N},\ S_n = \displaystyle\sum_{k=0}^{n} \frac{1}{k!}$. On cherche à déterminer sa limite.

1. Compléter le progamme suivant pour que l'appel de `valeur_S(n)` renvoie la valeur de S_n.
   On n'utilisera pas de fonction "factorielle". **On recopiera l'intégralité du programme.**

```
import numpy as np
def valeur_S(n) :
    S = ... ; a = ...
    for k in range ( ... ) :
      S = S + a
      a = a * ...
   return S
```

Pour tout $n \in \mathbb{N}$, on introduit : $I_n = \displaystyle\int_0^1 (1-t)^n e^t dt$.

2. Calculer la valeur de I_0.

Pour calculer I_1, I_2, etc, on propose le programme Python suivant :

```
def approx_I(n) :
  S = 0; N = 10000;
  for k in range(N) :
    S = S + (N-k)**n * np.exp(k/N)
  return S / (N**(n+1))
```

3. Expliciter, sous la forme d'une somme, la valeur calculée par ce programme.
   Démontrer que, si N est choisi très grand, cette valeur est une approximation de l'intégrale I_n.

L'instruction `print([approx_I(n) for n in range(4)])` affiche : `[ 1.718, 0.718, 0.436, 0.309 ]`.

4. Etablir le sens de variation de la suite $(I_n)_{n\in\mathbb{N}}$.

5. (a) Démontrer la relation de récurrence : $\forall n \in \mathbb{N},\ I_{n+1} = -1 + (n+1)I_n$.

   (b) A l'aide de cette relation, définir en Python une fonction nommée `valeur_I`,
   qui prend en entrée un entier n et renvoie la valeur exacte de I_n.

6. Montrer que pour tout $n \in \mathbb{N}$, $0 \leqslant I_n \leqslant \dfrac{e}{n+1}$. En déduire la valeur de $\displaystyle\lim_{n\to+\infty} I_n$.

7. Déterminer la valeur de $\displaystyle\lim_{n\to+\infty} (nI_n)$.

8. Montrer que $\forall n \in \mathbb{N},\ |S_n - e| \leqslant \dfrac{e}{(n+1)!}$. En déduire finalement $\displaystyle\lim_{n\to+\infty} S_n$.

   Indication : Poser $u_k = \dfrac{I_k}{k!}$ et calculer la somme $\displaystyle\sum_{k=0}^{n-1} (u_{k+1} - u_k)$ de deux façons différentes,
   l'une mettant en jeu I_n, l'autre mettant en jeu S_n.

9. On considère une autre suite similaire : $\forall n \in \mathbb{N},\ J_n = \displaystyle\int_0^2 t^n e^{-t/2} dt$.
   Montrer que $J_n = a_n \cdot I_n$, où a_n est un réel que l'on exprimera en fonction de n.

Partie II - Highscore

Soit $n \in \mathbb{N}^*$ un entier fixé.
Un jeu vidéo est composé de n niveaux consécutifs : Niveau 1, Niveau 2, ..., Niveau n (le dernier).

Un joueur se lance dans une partie, selon les modalités suivantes :

- Le joueur débute sa partie au Niveau 1. S'il parvient à terminer un niveau, il passe au niveau suivant.
- La partie s'arrête dès que le joueur échoue à l'un des niveaux, ou s'il parvient à terminer le Niveau n.

On note Z la variable aléatoire correspondant au nombre de niveaux que le joueur parvient à terminer durant sa partie. Pour tout $i \in [\![1, n]\!]$, on introduit l'évènement A_i = "Le joueur termine le Niveau i".

10. (a) Quel est, a priori, le support de la variable aléatoire Z ?
    (b) Pour chaque $k \in [\![0, n]\!]$, exprimer l'évènement $[Z = k]$ en fonction des évènements $(A_i)_{i \in [\![1,n]\!]}$.

Dans un premier temps, on considère que tous les niveaux sont d'une difficulté comparable.
Autrement dit, on fixe un réel $p \in [0, 1]$ et on fait l'hypothèse (H_1) suivante :

(H_1) : Pour tout $k \in [\![1, n]\!]$, Si le joueur atteint le Niveau k, la probabilité qu'il le termine est p.

11. Pour simuler une réalisation de la variable aléatoire Z, on propose d'abord le programme suivant :

```
import numpy.random as rd
def alea(n,p) :
  Z = 0
  for k in range(n) :
    if rd.random() < p :
      Z = Z +1
  return Z
```

    (a) Quelle loi de probabilité est simulée à l'aide de cette fonction ?
    Expliquer pourquoi ce programme ne correspond pas à la situation qui nous intéresse ici.
    (b) Proposer une autre fonction Python qui simule la variable Z de façon adéquate.
    Indication : Tant qu'on parvient à passer un niveau, et tant qu'il y a bien un prochain niveau, on peut passer au suivant...

12. Sous l'hypothèse (H_1), déterminer la loi de Z. On distinguera $P(Z = n)$ des autres probabilités.

13. Vérifier par le calcul que l'on a bien $\sum_{k=0}^{n} P(Z = k) = 1$.

On aimerait à présent modéliser une situation où les niveaux sont de difficulté croissante.
Dans toute la fin du problème, on fait l'hypothèse (H_2) suivante :

(H_2) : Pour tout $k \in [\![1, n]\!]$, Si le joueur atteint le Niveau k, la probabilité qu'il le termine est $\frac{1}{k}$.

14. Proposer une nouvelle fonction Python qui simule la variable aléatoire Z dans cette situation.

15. Sous l'hypothèse (H_2), établir que : $\forall k \in [\![0, n-1]\!]$, $P(Z = k) = \dfrac{k}{(k+1)!}$ et $P(Z = n) = \dfrac{1}{n!}$

16. (a) A l'aide du théorème de transfert, montrer que $E(Z + 1) = S_n$ (suite introduite en Partie I).
    En déduire l'expression de $E(Z)$ en fonction de S_n.
    (b) Si on suppose que le nombre total de niveaux disponibles est très grand,
    en moyenne, combien de niveaux le joueur parvient-il à terminer au cours de sa partie ?

17. On suppose $n \geqslant 3$.
    (a) Montrer que $E\Big((Z + 1)(Z - 1)\Big) = S_{n-3} + \dfrac{n^2 - 1}{n!}$
    (b) Etablir l'expression de $V(Z)$ en fonction de n, S_n et S_{n-3}.
    (c) Si on suppose que le nombre total de niveaux disponibles est très grand,
    que vaut (environ) la variance de Z ?