DM 14 (A)

A faire pour le Lundi 24 Février

La calculatrice est interdite. Durée : 4h

Exercice 1

Pour tout entier naturel n, on définit la fonction f_n de la variable réelle x par :

$$f_n(x) = x^n \exp\left(-\frac{x^2}{2}\right).$$

1. Justifier que $f_n(x)$ est négligeable devant $\dfrac{1}{x^2}$ au voisinage de $+\infty$.

2. Prouver la convergence de l'intégrale $\displaystyle\int_0^{+\infty} f_n(x)\,dx$.

3. On pose $I_n = \displaystyle\int_0^{+\infty} f_n(x)\,dx$.

   (a) A l'aide d'une intégration par parties portant sur des intégrales définies sur le segment $[0, A]$ avec $A \geq 0$, prouver que pour tout entier naturel n :

   $$I_{n+2} = (n+1)\,I_n.$$

   (b) En utilisant la loi normale centrée réduite, justifier que :

   $$I_0 = \sqrt{\frac{\pi}{2}}.$$

   (c) Donner la valeur de I_1.

   (d) Démontrer par récurrence que pour tout entier naturel n :

   $$I_{2n} = \sqrt{\frac{\pi}{2}}\frac{(2n)!}{2^n n!} \qquad \text{et} \qquad I_{2n+1} = 2^n n!.$$

4. Soit f la fonction définie pour tout réel x par :

   $$f(x) = \begin{cases} f_1(x) & \text{si } x \geq 0, \\ 0 & \text{si } x < 0. \end{cases}$$

   (a) Démontrer que f est une densité de probabilité.

   (b) Soit X une variable aléatoire réelle qui admet f pour densité de probabilité.

      i. Justifier que X admet une espérance $E(X)$, et préciser sa valeur

      ii. Justifier que X admet une variance $V(X)$, et préciser sa valeur.

5. On désigne par F et G les fonctions de répartitions respectives de X et de $Y = X^2$.

   (a) Exprimer $G(x)$ en fonction de $F(x)$ en distinguant les deux cas : $x < 0$ et $x \geq 0$.

   (b) En déduire que Y est une variable à densité puis déterminer une densité de Y.

   (c) Reconnaitre la loi de Y et donner la valeur de $E(Y)$ et $V(Y)$.

Exercice 2

Partie 1

On considère la matrice $A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$. On note f l'endomorphisme de $\mathbb{R}^4$ représenté par la matrice A dans la base canonique $\mathcal{C} = (e_1, e_2, e_3, e_4)$ de $\mathbb{R}^4$.

1. On considère les vecteurs suivants de $\mathbb{R}^4$:

$$u_1 = (-1, 1, 0, 1), \quad u_2 = (0, -1, 1, 0), \quad u_3 = (0, 1, 1, 0) \quad u_4 = (1, 0, 0, 1).$$

   On note $\mathcal{B} = (u_1, u_2, u_3, u_4)$.

   (a) Montrer que $\mathcal{B}$ est une base de $\mathbb{R}^4$.

   (b) Déterminer la matrice représentative de f dans la base $\mathcal{B}$.

   (c) En déduire une matrice P de $\mathscr{M}_4(\mathbb{R})$ inversible et une matrice T de $\mathscr{M}_4(\mathbb{R})$ triangulaire telles que $A = PTP^{-1}$.

2. (a) Calculer A^2, A^3, puis vérifier que $A^3 = 4A^2 - 4A$.

   (b) Montrer par récurrence que, pour tout entier naturel n non nul, il existe deux réels a_n et b_n tels que

$$A^n = a_n A^2 + b_n A$$

   vérifiant, pour tout entier naturel n non nul, $a_{n+1} = 4a_n + b_n$ et $b_{n+1} = -4a_n$.

3. (a) Montrer que, pour tout entier naturel n non nul,

$$a_{n+2} = 4a_{n+1} - 4a_n.$$

   (b) Déterminer, pour tout entier naturel n non nul, une expression de a_n en fonction de n.

   (c) En déduire, pour tout entier naturel n non nul, une expression de b_n en fonction de n.

4. Montrer que, pour tout entier naturel n non nul,

$$A^n = \begin{pmatrix} 2^{n-1} & 0 & 0 & 2^{n-1} \\ (n+1)2^{n-2} & 2^{n-1} & 2^{n-1} & (n-1)2^{n-2} \\ (n+1)2^{n-2} & 2^{n-1} & 2^{n-1} & (n-1)2^{n-2} \\ 2^{n-1} & 0 & 0 & 2^{n-1} \end{pmatrix}$$

Partie 2

Soient p un entier naturel non nul et G un graphe non pondéré orienté à p sommets. On note s_0, $s_1, \ldots, s_{p-1}$ les sommets de G.

5. (a) Rappeler la définition de la matrice d'adjacence du graphe G.

   (b) Soient n un entier naturel non nul, i un entier de $[\![1, p]\!]$ et j un entier de $[\![1, p]\!]$.
   Rappeler sans justification l'interprétation du coefficient situé à la ligne i et à la colonne j dans la matrice M^n, où M est la matrice d'adjacence du graphe G.

6. Dans cette question uniquement, on suppose que $p = 4$ et que la matrice d'adjacence du graphe G est la matrice

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

   étudiée dans la partie 1.

(a) Représenter les sommets et les arêtes du graphe G sous forme d'un diagramme.

(b) Le graphe G est-il connexe ? Justifier votre réponse.

(c) Soit n un entier naturel non nul.
Déterminer le nombre de chemins de longueur n menant du sommet s_3 au sommet s_0.

7. Dans cette question et les suivantes, on revient au cas général décrit au début de la partie 2.

   Soit s un sommet de G. On dit que le sommet t est un voisin de s quand (s,t) est une arête du graphe.

   Comme le graphe est orienté, si t est un voisin de s, alors s n'est pas forcément un voisin de t.

   On appelle liste d'adjacence du graphe G, une liste de p sous-listes telle que, pour tout entier k de $[\![0, p-1]\!]$, la sous-liste située à la position k contient tous les numéros des sommets voisins de s_k.

   Par exemple, la liste d'adjacence du graphe étudié à la question 6 est :

   ```
   L = [[0,3], [0,1,2], [0,1,2], [0,3]]
   ```

   Écrire une fonction en langage Python, nommée `matrice_vers_ligne`, prenant en entrée la matrice d'adjacence `A` d'un graphe G (définie sous forme de listes de listes) et renvoyant la liste d'adjacence de G.

8. On cherche à écrire une fonction en langage Python permettant d'obtenir la longueur du plus court chemin menant d'un sommet de départ s_i à chaque sommet du graphe G.

   On souhaite pour cela appliquer un algorithme faisant intervenir les variables suivantes :

   - Une liste `distances` à p éléments, où l'élément situé à la position k sera égal, à la fin de l'algorithme, à la longueur du plus court chemin menant du sommet de départ s_i au sommet s_k.
   - Une liste `a_explorer` contenant tous les sommets restant à traiter.
   - Une liste `marques` contenant tous les sommets déjà traités.

   Nous donnons ci-dessous la description de l'algorithme :

   - Initialisation des trois listes décrites ci-dessus :
     - Initialement, chaque élément de la liste `distances` est égal à p, à l'exception du sommet s_i, auquel on affecte la distance 0.
     - La liste `marques` ne contient initialement que le numéro du sommet de départ s_i.
     - La liste `a_explorer` ne contient initialement que le numéro du sommet de départ s_i.
   - Tant que la liste `a_explorer` n'est pas vide, on répète les opérations suivantes :
     - Nommer `s` le premier sommet de la liste `a_explorer`, et le retirer de cette liste.
     - Pour chaque voisin `v` du sommet `s` : si `v` n'est pas dans la liste `marques`, on l'ajoute à la fin de la liste `a_explorer`, et on lui affecte une distance égale à `distances[s]+1`.

   (a) On considère le graphe orienté G étudié à la question 6.
   Donner la valeur de la liste `distances` à l'issue de l'exécution de l'algorithme décrit ci-dessus, lorsqu'on l'applique au graphe G en choisissant s_1 comme sommet de départ.

   (b) Recopier et compléter la fonction suivante, prenant en entrée la liste d'adjcacence `L` du graphe G et le numéro `i0` du sommet de départ s_i, et renvoyant la liste `distances` après exécution de l'algorithme décrit ci-dessus.

```python
def parcours(L, i0):
    p = len(L)
    distances = .....
    distances[i0] = 0
    a_explorer = .....
    marques = .....
    while ..... :
        s = .....
        .....
        for v in ..... :
            if v not in marques :
                marques.append(v)
                .....
                .....
    return distances
```

(c) Modifier la fonction précédente pour qu'elle renvoie la liste de tous les sommets s pour lesquels il existe un chemin menant du sommet de départ s_i au sommet s.

Exercice 3

Soit n un entier naturel non nul.
Une urne contient n boules indiscernables au toucher et numérotées de 1 à n. On tire une boule au hasard dans l'urne. Si cette boule tirée porte le numéro k, on place alors dans une seconde urne toutes les boules suivantes : une boule numérotée 1, deux boules numérotées 2, et plus généralement pour tout $j \in [\![1, k]\!]$, j boules numérotées j, jusqu'à k boules numérotées k. Les boules de cette deuxième urne sont aussi indiscernables au toucher. On effectue alors un tirage au hasard d'une boule dans cette seconde urne.
Et on note X la variable aléatoire égale au numéro de la première boule tirée et on note Y la variable aléatoire égale au numéro de la deuxième boule tirée.

1. Reconnaître la loi de X et donner son espérance et sa variance.

2. Déterminer $Y(\Omega)$.

3. Soit $k \in [\![1, n]\!]$.

   (a) On suppose que l'événement $[X = k]$ est réalisé.
   Déterminer, en fonction de k, le nombre total de boules présentes dans la seconde urne.

   (b) Pour tout entier j de $[\![1, n]\!]$, exprimer $P_{[X=k]}(Y = j)$ en fonction de k et j.
   On distinguera les cas $j \leq k$ et $j \geq k + 1$.

4. (a) Déterminer deux réels a et b tels que, pour tout entier naturel k non nul,
$$\frac{1}{k(k+1)} = \frac{a}{k} + \frac{b}{k+1}.$$

   (b) En déduire que, pour tout élément j de $Y(\Omega)$,
$$P(Y = j) = \frac{2(n+1-j)}{n(n+1)}.$$

5. Justifier que Y admet une espérance et montrer que $E(Y) = \dfrac{n+2}{3}$.

6. Les variables X et Y sont-elles indépendantes ?

7. (a) Montrer que $E(XY) = \dfrac{(n+1)(4n+5)}{18}$.

   (b) En déduire que $\mathrm{Cov}(X,Y) = \dfrac{n^2-1}{18}$.

8. (a) Écrire une fonction en langage `Python`, nommée `seconde_urne`, prenant en entrée un entier naturel `k` non nul, et renvoyant une liste contenant 1 élément valant 1, 2 éléments valant 2, ..., j éléments valant j, ..., jusqu'à k éléments valant k.

   Par exemple, l'appel de `seconde_urne(4)` renverra `[1,2,2,3,3,3,4,4,4,4]`.

   (b) Recopier et compléter la fonction en langage `Python` suivante pour qu'elle prenne en entrée un entier naturel n non nul, et qu'elle renvoie une réalisation du couple de variables aléatoires (X,Y).

```python
import numpy.random as rd

def simul_XY(n):
    X = .....
    urne2 = seconde_urne( ..... )
    nb = len(urne2)
    i = rd.randint(0, nb)
    Y = .....
    return X,Y
```

   (c) On considère la fonction en langage `Python` suivante, prenant en entrée un entier naturel `n` non nul.

```python
def fonction(n):
    liste = [0]*n
    for i in range(10000):
        j = simul_XY(n)[1]
        liste[j-1] = liste[j-1] + 1/10000
    return liste
```

   Quelles valeurs les éléments de la liste renvoyée permettent-ils d'estimer ?

9. Dans toute cette question, on suppose $n = 20$. On simule 50 réalisations du couple de variables aléatoires (X,Y) à l'aide de la fonction `simul_XY` définie à la question 8.(b). On représente alors les valeurs obtenues sous forme d'un nuage de points, où les valeurs des réalisations de X sont représentées en abscisse et les valeurs des réalisations de Y en ordonnées. On trace également, sur la même figure, la droite de régression linéaire associée à ce nuage de points.

   (a) Déterminer par un calcul une valeur approchée des coordonnées du point moyen du nuage de points. Quel théorème de probabilités permet de justifier cette approximation ?

   (b) Parmi les figures représentées ci-dessous, en justifiant soigneusement votre réponse, indiquer celle qui correspond au nuage de points et à la droite de régression linéaire étudiés.

Figure 1

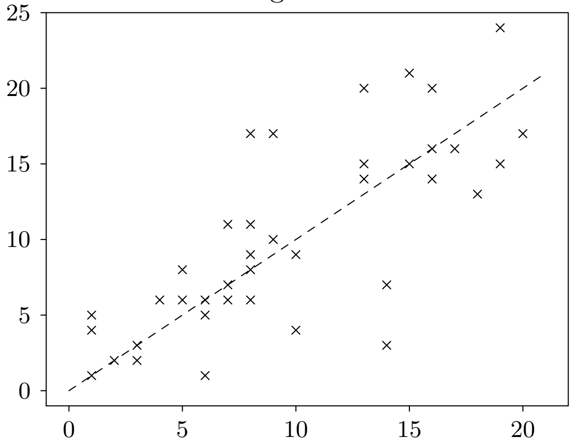

Figure 2

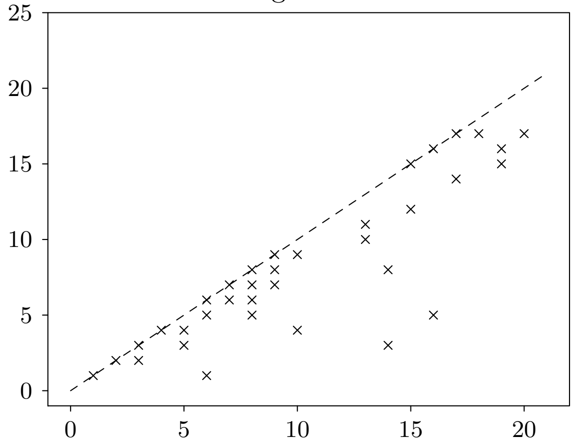

Figure 3

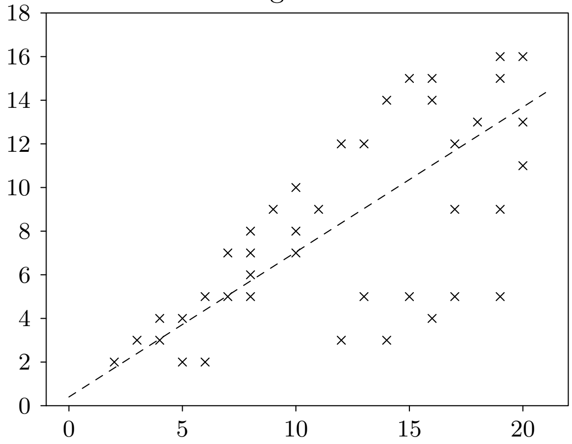

Figure 4