Ecricome 2023 – Mathématiques appliquées

Corrigé

Exercice 1

1. On voit immédiatement que $\boxed{X \text{ suit une loi uniforme sur } [\![1;n]\!]}$. On en déduit, d'après le cours, que $\boxed{E(X) = \dfrac{n+1}{2}}$ et $\boxed{V(X) = \dfrac{n^2-1}{12}}$.

2. Pour tout $k \geqslant 1$, si X prend la valeur k, alors la variable aléatoire Y peut prendre chacune des valeurs entre 1 et k (inclus). Étant donné que $X(\Omega) = [\![1;n]\!]$, on en déduit que $Y(\Omega) = \displaystyle\bigcup_{k\in[\![1;n]\!]} [\![1;k]\!]$, soit $\boxed{Y(\Omega) = [\![1;n]\!]}$.

3. (a) Dans la seconde urne, il y a 1 boule numérotée 1, 2 boules numérotées 2, ..., k boules numérotées k. Ainsi, le nombre total de boules dans l'urne est $\displaystyle\sum_{j=1}^{k} j$, soit $\boxed{\dfrac{k(k+1)}{2}}$.

   (b) Soit $j \in [\![1;n]\!]$. Si $j \geqslant k+1$, alors il n'y a aucune boule numérotée j dans la seconde urne (on suppose toujours que l'événement $[X=k]$ est réalisé). Donc $P_{[X=k]}(Y=j) = 0$.
   Si $j \leqslant k$, alors il y a j boules numérotées j dans la seconde urne (et $\dfrac{k(k+1)}{2}$ boules en tout d'après la question précédente). Par conséquent, $P_{[X=k]}(Y=j) = \dfrac{j}{\frac{k(k+1)}{2}} = \dfrac{2j}{k(k+1)}$.

   $$\boxed{\text{Conclusion : pour tout } (k,j) \in [\![1;n]\!]^2, \quad P_{[X=k]}(Y=j) = \begin{cases} \dfrac{2j}{k(k+1)} & \text{si } j \leqslant k \\ 0 & \text{sinon} \end{cases}}$$

4. (a) Soit $(a,b) \in \mathbb{R}^2$. Alors, pour tout $k \geqslant 1$, on a :

   $$\frac{a}{k} + \frac{b}{k+1} = \frac{a(k+1)+bk}{k(k+1)} = \frac{(a+b)k+a}{k(k+1)}$$

   On en déduit que c'est égal à $\dfrac{1}{k(k+1)}$ pour tout $k \geqslant 1$ si $\begin{cases} a+b &= 0 \\ a &= 1 \end{cases}$,

   c'est-à-dire si $\boxed{\begin{cases} a &= 1 \\ b &= -1 \end{cases}}$.

(b) Soit $j \in [\![1;n]\!]$. D'après la formule des probabilités totales avec le système complet d'événements $\Big([X=k]\Big)_{1\leqslant k\leqslant n}$, on a :

$$\begin{aligned}
P(Y=j) &= \sum_{k=1}^{n} P_{[X=k]}(Y=j)P(X=k)\\
&= \sum_{k=j}^{n} P_{[X=k]}(Y=j)P(X=k) \quad \text{car } P_{[X=k]}(Y=j)=0 \text{ si } k<j\\
&= \sum_{k=j}^{n} \frac{2j}{k(k+1)} \times \frac{1}{n} \quad \text{(questions 1 et 3.(b))}\\
&= \frac{2j}{n} \sum_{k=j}^{n} \frac{1}{k(k+1)}\\
&= \frac{2j}{n} \sum_{k=j}^{n} \left(\frac{1}{k} - \frac{1}{k+1}\right) \quad \text{(question précédente)}\\
&= \frac{2j}{n} \left(\frac{1}{j} - \frac{1}{n+1}\right) \quad \text{(somme télescopique)}\\
&= \frac{2j}{n} \times \frac{n+1-j}{j(n+1)}
\end{aligned}$$

Ce qui donne bien : $\boxed{P(Y=j) = \dfrac{2(n+1-j)}{n(n+1)}}$.

5. L'ensemble $Y(\Omega)$ étant fini, $\boxed{Y \text{ admet une espérance}}$. De plus,

$$\begin{aligned}
E(Y) &= \sum_{j=1}^{n} jP(Y=j)\\
&= \sum_{j=1}^{n} j\,\frac{2(n+1-j)}{n(n+1)}\\
&= \frac{2}{n(n+1)} \sum_{j=1}^{n} j(n+1-j)\\
&= \frac{2}{n(n+1)} \sum_{j=1}^{n} \Big((n+1)j - j^2\Big)\\
&= \frac{2}{n(n+1)} \left[(n+1)\sum_{j=1}^{n} j - \sum_{j=1}^{n} j^2\right]\\
&= \frac{2}{n(n+1)} \left[(n+1)\frac{n(n+1)}{2} - \frac{n(n+1)(2n+1)}{6}\right]\\
&= \frac{2}{n(n+1)} n(n+1) \left[\frac{n+1}{2} - \frac{2n+1}{6}\right]\\
&= 2\left[\frac{n+1}{2} - \frac{2n+1}{6}\right]
\end{aligned}$$

$$= 2\,\frac{3(n+1)-(2n+1)}{6}$$
$$= 2\,\frac{n+2}{6}$$

C'est-à-dire : $\boxed{E(Y) = \frac{n+2}{3}}$.

6. Si $n = 1$, alors X prend nécessairement la valeur 1, et Y aussi. On a alors $P((X = 1) \cap (Y = 1)) = 1 = P(X = 1)P(Y = 1)$, c'est-à-dire $P((X = i) \cap (Y = j)) = P(X = i)P(Y = j)$ pour tout $(i, j) \in X(\Omega) \times Y(\Omega)$. Les variables aléatoires X et Y sont donc indépendantes.
Maintenant, si $n \geqslant 2$, alors $P((X = 1) \cap (Y = n)) = 0$ (car $Y \leqslant X$), mais $P(X = 1)P(Y = n) = \frac{1}{n} \times \frac{2}{n(n+1)} = \frac{2}{n^2(n+1)} \neq 0$. Donc X et Y ne sont pas indépendantes.

Conclusion : les variables aléatoires X et Y sont indépendantes si et seulement si $n = 1$.

7. (a) L'espérance demandée est bien définie car X et Y sont à valeurs dans un ensemble fini. De plus, par le théorème de transfert :

$$\begin{aligned}
E(XY) &= \sum_{k=1}^{n}\sum_{j=1}^{n} kjP((X = k) \cap (Y = j)) \\
&= \sum_{k=1}^{n}\sum_{j=1}^{n} kjP_{[X=k]}(Y = j)P(X = k) \\
&= \sum_{k=1}^{n}\sum_{j=1}^{k} kjP_{[X=k]}(Y = j)P(X = k) \quad \text{car } P_{[X=k]}(Y = j) = 0 \text{ si } j > k \\
&= \sum_{k=1}^{n}\sum_{j=1}^{k} kj\frac{2j}{k(k+1)} \times \frac{1}{n} \quad \text{(questions 1 et 3.(b))} \\
&= \frac{2}{n}\sum_{k=1}^{n}\sum_{j=1}^{k} \frac{j^2}{k+1} \quad \text{(on simplifie et on factorise)} \\
&= \frac{2}{n}\sum_{k=1}^{n}\left(\frac{1}{k+1}\sum_{j=1}^{k} j^2\right) \\
&= \frac{2}{n}\sum_{k=1}^{n}\left(\frac{1}{k+1} \times \frac{k(k+1)(2k+1)}{6}\right) \\
&= \frac{2}{n}\sum_{k=1}^{n}\frac{k(2k+1)}{6} \\
&= \frac{1}{3n}\sum_{k=1}^{n} k(2k+1) \\
&= \frac{1}{3n}\left[2\sum_{k=1}^{n} k^2 + \sum_{k=1}^{n} k\right]
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{3n}\left[2\,\frac{n(n+1)(2n+1)}{6} + \frac{n(n+1)}{2}\right] \\
&= \frac{1}{3n} \times \frac{2n(n+1)(2n+1)+3n(n+1)}{6} \\
&= \frac{1}{3n} \times \frac{n(n+1)\Big[2(2n+1)+3\Big]}{6} \\
&= \frac{1}{3n} \times \frac{n(n+1)(4n+5)}{6}
\end{aligned}$$

Ce qui se simplifie en : $\boxed{E(XY) = \frac{(n+1)(4n+5)}{18}}$.

(b) Par la formule de König-Huygens, on a :

$$\begin{aligned}
Cov(X,Y) &= E(XY) - E(X)E(Y) \\
&= \frac{(n+1)(4n+5)}{18} - \frac{n+1}{2} \times \frac{n+2}{3} \quad \text{(questions 1, 5 et 7.(a))} \\
&= \frac{(n+1)(4n+5)}{18} - \frac{(n+1)(n+2)}{6} \\
&= \frac{(n+1)(4n+5) - 3(n+1)(n+2)}{18} \\
&= \frac{(n+1)\Big[(4n+5) - 3(n+2)\Big]}{18} \\
&= \frac{(n+1)(n-1)}{18}
\end{aligned}$$

Ce qui donne bien : $\boxed{Cov(X,Y) = \frac{n^2-1}{18}}$.

Remarque : on en déduit que $Cov(X,Y) = 0$ si et seulement si $n = 1$, ce qui est cohérent avec le résultat de la question 6.

8. (a) On peut faire une double boucle : une boucle sur `j` variant de 1 à `k`, à l'intérieur de laquelle se trouve une autre boucle destinée à rajouter à la suite de la liste le bon nombre de `j` :

```
def seconde_urne(k):
    L=[]
    for j in range(1,k+1):
        for i in range(j):
            L.append(j)
    return L
```

On peut éventuellement se passer de la seconde boucle en concaténant directement à `L` la liste contenant `j` fois l'élément `j` :

```
def seconde_urne(k):
    L=[]
    for j in range(1,k+1):
        L=L+[j]*j
    return L
```

(b) On commence par tirer X aléatoirement suivant une loi uniforme, puis on constitue la seconde urne, puis on prend un élément au hasard (avec équiprobabilité) dans cette seconde urne : ce sera Y.

```
import numpy.random as rd

def simul_XY(n):
    X=rd.randint(1,n+1)
    urne2=seconde_urne(X)
    nb=len(urne2)
    i=rd.randint(0,nb)
    Y=urne2[i]
    return X,Y
```

(c) La commande `simul_XY(n)[1]` renvoie une simulation de la variable aléatoire Y. Ainsi, le programme effectue 10000 simulations de Y, et à la fin, pour tout j, `liste[j-1]` contient $\dfrac{\text{nombre de fois où } Y \text{ a pris la valeur } j}{10000}$.

Ainsi, `fonction(n)` renvoie une liste contenant des estimations respectives de $P(Y=1),\dots,P(Y=n)$. Autrement dit, cette fonction renvoie une estimation de la loi de Y.

9. (a) D'après **la loi faible des grands nombres** (dont les hypothèses sont ici vérifiées, puisque X et Y admettent chacune un moment d'ordre 2), le point moyen du nuage de point aura approximativement comme coordonnées $(E(X),E(Y))$. Comme ici $n=20$, on en déduit, d'après les questions 1 et 5, que le point moyen aura approximativement comme coordonnées $\boxed{\left(\frac{21}{2},\frac{22}{3}\right)}$, soit $\boxed{\text{environ } (10{,}5\,,\ 7{,}3)}$.

(b) On procède par élimination :
- La figure 1 ne correspond à un nuage de points de simulations de (X,Y) puisqu'on est censé toujours avoir $Y \leqslant X$, ce qui n'est pas le cas sur la figure (on a des points d'abscisse 3 et d'ordonnée supérieure à 10).
- Les points de la figure 2 peuvent correspondre à des simulations de (X,Y). Par contre, la droite n'est pas la droite de régression linéaire. En effet, la droite de régression linéaire est censée passer par le point moyen du nuage, ce qui n'est pas le cas, ici.
- Les points de la figure 4 peuvent correspondre à des simulations de (X,Y). Par contre, la droite n'est pas la droite de régression linéaire. En effet, d'après la question 7.(b), on a $Cov(X,Y)>0$ pour $n=20$. La droite de régression linéaire devrait donc monter, et non descendre comme sur la figure.

Conclusion : la bonne figure est la figure 3.

Exercice 2

1. (a) La fonction $x \mapsto e^{x/2}$ est dérivable sur $\mathbb{R}$ — et donc sur $]0;+\infty[$ — comme composée de fonctions dérivables sur $\mathbb{R}$. De plus, la fonction racine carrée est dérivable et ne s'annule pas sur $]0;\infty[$.
Ainsi, $\boxed{f \text{ est dérivable sur }]0;+\infty[}$ comme quotient de fonctions dérivables.
De plus, pour tout $x > 0$, on a :

$$\begin{aligned} f'(x) &= \frac{\frac{1}{2}e^{x/2}\sqrt{x} - e^{x/2}\frac{1}{2\sqrt{x}}}{x} \\ &= \frac{e^{x/2}x - e^{x/2}}{2x\sqrt{x}} \quad \text{(on multiplie en haut et en bas par } 2\sqrt{x}) \\ &= \frac{(x-1)e^{x/2}}{2x\sqrt{x}} \end{aligned}$$

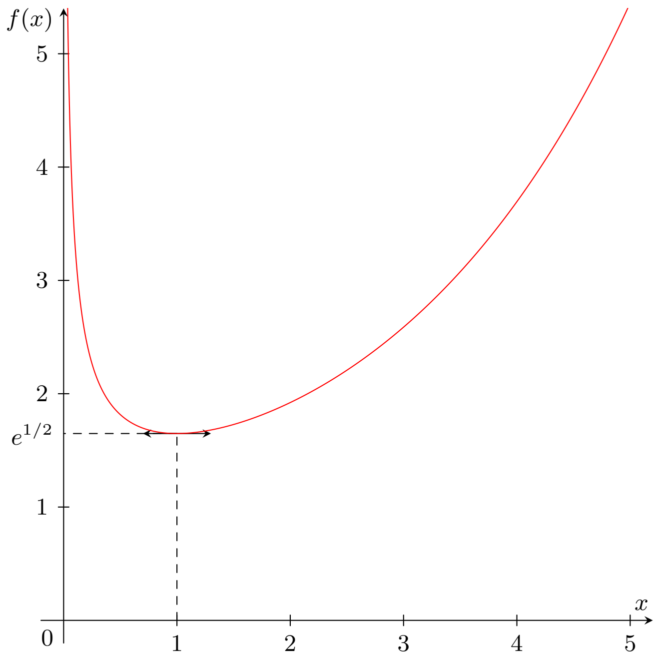
Ce qui donne bien : $\boxed{f'(x) = \dfrac{x-1}{2\sqrt{x}}f(x)}$.

(b) Commençons par chercher les limites. En 0 (à droite), on a $e^{x/2} \underset{x\to 0}{\longrightarrow} 1$ et $\sqrt{x} \underset{x\to 0^+}{\longrightarrow} 0^+$, donc $f(x) \underset{x\to 0^+}{\longrightarrow} +\infty$.
En $+\infty$, on a $\lim\limits_{x\to+\infty} \dfrac{e^{x/2}}{\sqrt{x}} = \lim\limits_{t\to+\infty} \dfrac{e^t}{\sqrt{2t}}$ (en posant $t = x/2$), soit $\lim\limits_{x\to+\infty} f(x) = \lim\limits_{t\to+\infty} \dfrac{e^t}{\sqrt{2}t^{1/2}}$. Or, $\lim\limits_{t\to+\infty} \dfrac{e^t}{t^{1/2}} = +\infty$ par croissances comparées. D'où $f(x) \underset{x\to+\infty}{\longrightarrow} +\infty$.

Maintenant, en ce qui concerne les variations, on a, pour tout $x > 0$, $e^{x/2} > 0$ et $\sqrt{x} > 0$, donc $f(x) > 0$. Ainsi, d'après la question précédente, $f'(x)$ est du signe de $x - 1$ pour tout $x > 0$. On en déduit le tableau de variations de f :

x	0		1		$+\infty$
$f'(x)$	‖	$-$	0	$+$	
$f(x)$	‖ $+\infty$	$\searrow$	$e^{1/2}$	$\nearrow$	$+\infty$

(c) D'après ce qui précède, l'allure de la courbe représentative de f est la suivante :



(d) Soit $n \geqslant 2$. La fonction f est continue (car dérivable) et strictement décroissante sur $]0;1]$. De plus, d'après les limites calculées à la question 1.(b) on a $f(]0;1]) = [e^{1/2}; +\infty[$, et donc $n \in f(]0;1])$ ($e < 4$, donc $e^{1/2} < 2$ et donc $e^{1/2} < n$). Par conséquent, d'après le théorème de la bijection, l'équation $f(x) = n$ admet une unique solution u_n sur $]0;1]$. De plus, u_n ne peut pas être égal à 1 car $f(1) = e^{1/2}$, qui est strictement inférieur à n. Donc $0 < u_n < 1$.
De même, f est continue et strictement croissante sur l'intervalle $]1;+\infty[$, avec $f(]1;+\infty[) =]e^{1/2}; +\infty[$ (qui contient n pour les mêmes raisons que ci-dessus). Donc, par le théorème de la bijection, l'équation $f(x) = n$ admet une unique solution v_n sur $]1;+\infty[$.

$\boxed{\text{Conclusion : l'équation } f(x) = n \text{ admet exactement 2 solutions } u_n \text{ et } v_n \text{ sur }]0;+\infty[\text{, qui vérifient } 0 < u_n < 1 < v_n.}$

2. (a) Pour tout $n \geqslant 2$, on a $n \leqslant n+1$, soit $f(v_n) \leqslant f(v_{n+1})$. Par stricte croissance de f sur $[1;+\infty[$, on en déduit que $v_n \leqslant v_{n+1}$. Par conséquent, $\boxed{\text{la suite } (v_n)_{n \geqslant 2} \text{ est croissante}}$.

(b) La suite $(v_n)_{n \geqslant 2}$ est croissante. Donc, par le théorème de la limite monotone, soit elle est majorée et dans ce cas elle converge vers une limite finie ℓ, soit elle ne l'est pas et dans ce cas elle diverge vers $+\infty$ lorsque n tend vers $+\infty$.
Par l'absurde, supposons qu'on soit dans le premier cas. Comme $v_n \geqslant 1$ pour tout $n \geqslant 2$, on a $\ell \geqslant 1$ (en passant à la limite lorsque n tend vers $+\infty$). Or, f est continue sur $[1;+\infty[$, donc $f(v_n) \underset{n \to +\infty}{\longrightarrow} f(\ell)$. Mais ceci est absurde car $f(v_n) \underset{n \to +\infty}{\longrightarrow} +\infty$ (étant donné que $f(v_n) = n$ pour tout $n \geqslant 2$). On ne peut donc pas être dans le premier cas.

$\boxed{\text{Conclusion : la suite } (v_n)_{n \geqslant 2} \text{ diverge vers } +\infty \text{ lorsque } n \text{ tend vers } +\infty.}$

3. (a) De même qu'à la question 2.(a), pour tout $n \geqslant 2$, on a $f(u_n) \leqslant f(u_{n+1})$. Par stricte décroissance de f sur $]0;1]$, on en déduit que $u_n \geqslant u_{n+1}$. Par conséquent, $\boxed{\text{la suite } (u_n)_{n\geqslant 2} \text{ est décroissante}}$.

   (b) La suite $(u_n)_{n\geqslant 2}$ est décroissante et minorée (par 0). Par le théorème de la limite monotone, on en déduit que $\boxed{\text{la suite } (u_n)_{n\geqslant 2} \text{ converge}}$.

   (c) Le principe est le même qu'à la question 2.(b). Supposons que u_n converge vers une limite finie ℓ strictement positive. Par continuité de f sur $]0;1]$, on a alors $f(u_n) \underset{n\to+\infty}{\longrightarrow} f(\ell)$. Or, ceci est absurde car $f(u_n) \underset{n\to+\infty}{\longrightarrow} +\infty$ (étant donné que $f(u_n) = n$ pour tout $n \geqslant 2$). Ainsi, la limite ℓ de $(u_n)_{n\geqslant 2}$ ne peut être strictement positive.
   $\boxed{\text{Conclusion : la suite } (u_n)_{n\geqslant 2} \text{ converge vers 0 lorsque } n \text{ tend vers } +\infty.}$

   (d) Pour tout $n \geqslant 2$, on a, par définition de u_n : $f(u_n) = n$, soit $\dfrac{e^{u_n/2}}{\sqrt{u_n}} = n$. Par conséquent (en élevant au carré) : $\dfrac{e^{u_n}}{u_n} = n^2$, ce qui donne $e^{u_n} = n^2 u_n$. Or, d'après la question précédente, $e^{u_n} \underset{n\to+\infty}{\longrightarrow} 1$. On en déduit que $\boxed{n^2 u_n \underset{n\to+\infty}{\longrightarrow} 1}$, et donc $\boxed{u_n \underset{n\to+\infty}{\sim} \dfrac{1}{n^2}}$.

4. (a) Il suffit d'appliquer l'algorithme décrit dans l'énoncé. Les deux cas `b=c` et `a=c` sont dans cet ordre par décroissance de f sur $]0;1]$.

```
import numpy as np

def approx_u(n,eps):
    a = 0
    b = 1
    while b-a > eps:
        c = (a+b)/2
        if np.exp(c/2)/np.sqrt(c) < n:
            b = c
        else:
            a = c
    return (a+b)/2
```

   Remarque : on teste : `approx_u(10,1e-6)` *donne* `0.01010179519653320`*, ce qui est bien en accord avec le résultat de la question précédente.*

   (b) Les erreurs d'arrondis s'accumulent. Donc pour avoir une erreur au plus `eps` sur la somme, il faut calculer chaque terme avec une précision de `eps/N`. D'où le programme :

```python
def sp(N,eps):
    somme=0
    for n in range(2,N+1):
        somme=somme+approx_u(n,eps/N)
    return somme
```

Exercice 3

Partie 1

1. (a) Montrons que la famille $\mathcal{B}$ est libre. Soit $(\lambda_1, \lambda_2, \lambda_3, \lambda_4) \in \mathbb{R}^4$. On a les équivalences :

$$\lambda_1 u_1 + \lambda_2 u_2 + \lambda_3 u_3 + \lambda_4 u_4 = 0 \iff \left\{\begin{array}{rcrcrcrcl} -\lambda_1 & & & & & + & \lambda_4 & = & 0 \\ \lambda_1 & - & \lambda_2 & + & \lambda_3 & & & = & 0 \\ & & \lambda_2 & + & \lambda_3 & & & = & 0 \\ \lambda_1 & & & & & + & \lambda_4 & = & 0 \end{array}\right.$$

$$\underset{\substack{L_2 \leftarrow L_2 + L_1 \\ L_4 \leftarrow L_4 + L_1}}{\iff} \left\{\begin{array}{rcrcrcrcl} -\lambda_1 & & & & & + & \lambda_4 & = & 0 \\ & - & \lambda_2 & + & \lambda_3 & + & \lambda_4 & = & 0 \\ & & \lambda_2 & + & \lambda_3 & & & = & 0 \\ & & & & & & 2\lambda_4 & = & 0 \end{array}\right.$$

$$\underset{L_3 \leftarrow L_3 + L_2}{\iff} \left\{\begin{array}{rcrcrcrcl} -\lambda_1 & & & & & + & \lambda_4 & = & 0 \\ & - & \lambda_2 & + & \lambda_3 & + & \lambda_4 & = & 0 \\ & & & & 2\lambda_3 & + & \lambda_4 & = & 0 \\ & & & & & & 2\lambda_4 & = & 0 \end{array}\right.$$

$$\iff \left\{\begin{array}{rcl} \lambda_1 & = & 0 \\ \lambda_2 & = & 0 \\ \lambda_3 & = & 0 \\ \lambda_4 & = & 0 \end{array}\right.$$

Par conséquent, la famille $\mathcal{B}$ est libre. De plus, c'est une famille de 4 vecteurs de $\mathbb{R}^4$ et $\dim(\mathbb{R}^4) = 4$.

$\boxed{\text{Conclusion : la famille } \mathcal{B} \text{ est une base de } \mathbb{R}^4.}$

(b) On calcule :

- $A \begin{pmatrix} -1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, donc $f(u_1) = 0$.
- $A \begin{pmatrix} 0 \\ -1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, donc $f(u_2) = 0$.

- $A\begin{pmatrix}0\\1\\1\\0\end{pmatrix}=\begin{pmatrix}0\\2\\2\\0\end{pmatrix}=2\begin{pmatrix}0\\1\\1\\0\end{pmatrix}$, donc $f(u_3)=2u_3$.
- $A\begin{pmatrix}1\\0\\0\\1\end{pmatrix}=\begin{pmatrix}2\\1\\1\\2\end{pmatrix}=\begin{pmatrix}0\\1\\1\\0\end{pmatrix}+2\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$, donc $f(u_4)=u_3+2u_4$.

Par conséquent, $\boxed{\text{la matrice de } f \text{ dans la base } \mathcal{B} \text{ est } \begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&2&1\\0&0&0&2\end{pmatrix}}$.

(c) La matrice A est la matrice de f dans la base $\mathcal{C}$ et la matrice triangulaire ci-dessus est la matrice de f dans la base $\mathcal{B}$.

$$\boxed{\begin{array}{l}\text{Par la formule de changement de base, on a donc } A=PTP^{-1}, \text{ où } T\\ \text{est la matrice triangulaire } T=\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&2&1\\0&0&0&2\end{pmatrix} \text{ et } P \text{ est la matrice de}\\ \text{passage de la base } \mathcal{C} \text{ à la base } \mathcal{B}, \text{ soit } P=\begin{pmatrix}-1&0&0&1\\1&-1&1&0\\0&1&1&0\\1&0&0&1\end{pmatrix}.\end{array}}$$

2. (a) On calcule :

$$\boxed{A^2=\begin{pmatrix}1&0&0&1\\1&1&1&0\\1&1&1&0\\1&0&0&1\end{pmatrix}\begin{pmatrix}1&0&0&1\\1&1&1&0\\1&1&1&0\\1&0&0&1\end{pmatrix}=\begin{pmatrix}2&0&0&2\\3&2&2&1\\3&2&2&1\\2&0&0&2\end{pmatrix}}$$

puis

$$\boxed{A^3=A^2A=\begin{pmatrix}2&0&0&2\\3&2&2&1\\3&2&2&1\\2&0&0&2\end{pmatrix}\begin{pmatrix}1&0&0&1\\1&1&1&0\\1&1&1&0\\1&0&0&1\end{pmatrix}=\begin{pmatrix}4&0&0&4\\8&4&4&4\\8&4&4&4\\4&0&0&4\end{pmatrix}}$$

et enfin :

$$\boxed{4A^2-4A=\begin{pmatrix}8&0&0&8\\12&8&8&4\\12&8&8&4\\8&0&0&8\end{pmatrix}-\begin{pmatrix}4&0&0&4\\4&4&4&0\\4&4&4&0\\4&0&0&4\end{pmatrix}=\begin{pmatrix}4&0&0&4\\8&4&4&4\\8&4&4&4\\4&0&0&4\end{pmatrix}=A^3}$$

(b) On montre par récurrence, pour tout $n\in\mathbb{N}^*$, la propriété $\mathcal{P}(n)$: « il existe deux réels a_n et b_n tels que $A^n=a_nA^2+b_nA$ ».

Pour $n=1$, on a $A^1 = A = 0A^2 + 1A$. Donc, en posant $a_1 = 0$ et $b_1 = 1$, $\mathcal{P}(1)$ est vérifiée.
Maintenant, soit $n \in \mathbb{N}^*$ quelconque fixé. Supposons $\mathcal{P}(n)$ et montrons $\mathcal{P}(n+1)$:
Par hypothèse de récurrence, il existe $(a_n, b_n) \in \mathbb{R}^2$ tel que $A^n = a_nA^2 + b_nA$. D'où

$$\begin{aligned}
A^{n+1} &= AA^n \\
&= A(a_nA^2 + b_nA) \\
&= a_nA^3 + b_nA^2 \\
&= a_n(4A^2 - 4A) + b_nA^2 \quad \text{(question précédente)} \\
&= 4a_nA^2 - 4a_nA + b_nA^2 \\
&= (4a_n + b_n)A^2 - 4a_nA \\
&= a_{n+1}A^2 + b_{n+1}A
\end{aligned}$$

En posant $a_{n+1} = 4a_n + b_n$ et $b_{n+1} = -4a_n$. Ce qui montre la propriété $\mathcal{P}(n+1)$.

Conclusion : par récurrence, pour tout $n \in \mathbb{N}^*$, il existe deux réels a_n et b_n tels que $A^n = a_nA^2 + b_nA$. De plus, comme on l'a vu dans la démonstration, on a les relations $a_{n+1} = 4a_n + b_n$ et $b_{n+1} = -4a_n$.

3. (a) Pour tout $n \in \mathbb{N}^*$, on a, d'après la question précédente (appliquée à $n+1$) : $a_{n+2} = 4a_{n+1} + b_{n+1}$. Or, $b_{n+1} = -4a_n$. D'où $\boxed{a_{n+2} = 4a_{n+1} - 4a_n}$.

(b) La suite $(a_n)_{n\in\mathbb{N}^*}$ vérifie une relation de récurrence linéaire d'ordre 2, de polynôme caractéristique $X^2 - 4X + 4$, soit $(X-2)^2$. Ce polynôme admet une racine double, qui est 2. Donc il existe deux réels λ et μ tels que, pour tout $n \geqslant 1$, $a_n = (\lambda n + \mu)2^n$.
De plus, on sait que $a_1 = 0$ (cf initialisation de la récurrence à la question 2.(b)) et $a_2 = 4a_1 + b_1 = 4 \times 0 + 1 = 1$. Ce qui donne le système $\left\{\begin{array}{rcrcl} 2\lambda & + & 2\mu & = & 0 \\ 8\lambda & + & 4\mu & = & 1 \end{array}\right.$. On le résout :

$$\left\{\begin{array}{rcrcl} 2\lambda & + & 2\mu & = & 0 \\ 8\lambda & + & 4\mu & = & 1 \end{array}\right. \iff \left\{\begin{array}{rcl} \lambda & = & -\mu \\ -8\mu + 4\mu & = & 1 \end{array}\right.$$
$$\iff \left\{\begin{array}{rcl} \lambda & = & -\mu \\ -4\mu & = & 1 \end{array}\right.$$
$$\iff \left\{\begin{array}{rcl} \lambda & = & -\mu \\ \mu & = & \frac{-1}{4} \end{array}\right.$$
$$\iff \left\{\begin{array}{rcl} \lambda & = & \frac{1}{4} \\ \mu & = & \frac{-1}{4} \end{array}\right.$$

Conclusion : pour tout $n \in \mathbb{N}^*$, on a $a_n = \left(\frac{1}{4}n - \frac{1}{4}\right)2^n = \frac{1}{4}(n-1)2^n$, ou encore $a_n = (n-1)2^{n-2}$.

(c) Pour tout $n \geqslant 2$, on a, d'après la question 2.(b) (appliquée à $n-1$) : $b_n = -4a_{n-1}$. On en déduit, d'après la question précédente : $b_n = -4 \times \frac{1}{4}(n-2)2^{n-1} = -(n-2)2^{n-1}$. De plus, cette expression est encore valable pour $n = 1$ puisque $b_1 = 1$ (cf initialisation de la récurrence à la question 2.(b)) et $-(1-2)2^{1-1} = 1$ également.

$\boxed{\text{Conclusion : pour tout } n \in \mathbb{N}^*, \text{ on a } b_n = -(n-2)2^{n-1} = (2-n)2^{n-1}.}$

4. Pour tout $n \in \mathbb{N}^*$, on a, d'après la question 2.(b) :

$$\begin{aligned} A^n &= a_n A^2 + b_n A \\ &= a_n \begin{pmatrix} 2 & 0 & 0 & 2 \\ 3 & 2 & 2 & 1 \\ 3 & 2 & 2 & 1 \\ 2 & 0 & 0 & 2 \end{pmatrix} + b_n \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 2a_n + b_n & 0 & 0 & 2a_n + b_n \\ 3a_n + b_n & 2a_n + b_n & 2a_n + b_n & a_n \\ 3a_n + b_n & 2a_n + b_n & 2a_n + b_n & a_n \\ 2a_n + b_n & 0 & 0 & 2a_n + b_n \end{pmatrix} \end{aligned}$$

Or, d'après les 2 questions précédentes, on a :

$$\begin{aligned} 2a_n + b_n &= 2(n-1)2^{n-2} - (n-2)2^{n-1} \\ &= (n-1)2^{n-1} - (n-2)2^{n-1} \\ &= \big((n-1) - (n-2)\big)2^{n-1} \\ &= 2^{n-1} \end{aligned}$$

et

$$\begin{aligned} 3a_n + b_n &= 3(n-1)2^{n-2} - (n-2)2^{n-1} \\ &= 3(n-1)2^{n-1} - 2(n-2)2^{n-2} \\ &= \big(3(n-1) - 2(n-2)\big)2^{n-2} \\ &= (n+1)2^{n-2} \end{aligned}$$

D'où finalement :

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

ce qui peut se réécrire :

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

Partie 2

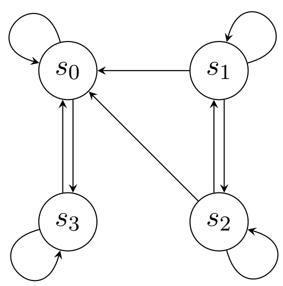
5. (a) C'est du cours (de première année).

$\boxed{\text{La matrice d'adjacence de } G \text{ est la matrice } M = (m_{i,j})_{0\leqslant i,j\leqslant p-1} \text{ telle que pour tout } (i,j)\in [\![0;p-1]\!]^2,\ m_{i,j} \text{ est égal au nombre d'arêtes allant du sommet } s_i \text{ au sommet } s_j.}$

(b) En faisant attention au décalage d'indice (les sommets sont numérotés à partir de 0, mais les lignes et colonnes à partir de 1, ici), on en déduit que :

$\boxed{\text{le coefficient situé à } 2^{\text{ème}} \text{ ligne et à la } j^{\text{ème}} \text{ colonne de } M \text{ est le nombre d'arêtes partant du sommet } s_1 \text{ et arrivant au sommet } s_{j-1}.}$

6. (a) D'après la matrice, le graphe est le suivant :



(b) D'après la question 2.(a), on a :

$$\begin{aligned} I_4 + A + A^2 + A^3 &= \begin{pmatrix} 1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1 \end{pmatrix} + \begin{pmatrix} 1&0&0&1\\1&1&1&0\\1&1&1&0\\1&0&0&1 \end{pmatrix} + \begin{pmatrix} 2&0&0&2\\3&2&2&1\\3&2&2&1\\2&0&0&2 \end{pmatrix} + \begin{pmatrix} 4&0&0&4\\8&4&4&4\\8&4&4&4\\4&0&0&4 \end{pmatrix} \\ &= \begin{pmatrix} 8&0&0&7\\12&8&7&5\\12&7&8&5\\7&0&0&8 \end{pmatrix} \end{aligned}$$

On constate que cette matrice a des coefficients nuls. Par conséquent, $\boxed{\text{le graphe orienté } G \text{ n'est pas connexe}}$.

Remarques :

— *On peut le voir sur le schéma de la question précédente : lorsqu'on est sur un des sommets s_0 ou s_3, on est obligé d'aller en s_0 ou en s_3. Ainsi, par exemple, il n'est existe aucun chemin partant du sommet s_0 et arrivant en s_1 (ou s_2).*

— *Si on oublie l'orientation (i.e qu'on considère les arêtes à double-sens), alors le graphe devient connexe.*

(c) Soit $n \in \mathbb{N}^*$. Le nombre de chemins de longueur n menant du sommet s_3 au sommet s_0 est le coefficient à la $4^{\text{ème}}$ ligne et $1^{\text{ère}}$ colonne de A^n. D'après la question 4, il y en a donc $\boxed{2^{n-1}}$.

7. On part avec une liste de listes vides `L`. Ensuite, on passe en revue chaque coefficient de `A` et on met à jour la liste `L` :

```
def matrice_vers_liste(A):
    n=len(A)
    L=[[] for i in range(n)]
    for i in range(n):
        for j in range(n):
            if A[i][j]==1:
                L[i].append(j)
    return L
```

8. (a) On applique l'algorithme décrit dans l'énoncé :
   - Initialement, on a : `distances=[4,0,4,4]`, `a_explorer=[1]`, et enfin `marques=[1]`.
   - On prend `s=1` le premier élément de `a_explorer` (qui devient alors la liste vide). Pour chaque voisin `v` de `s` (i.e `v=0`, `v=1`, `v=2`), on met à jour `distances`, `a_explorer` et `marques` comme expliqué dans le dernier point de l'algorithme. Après l'avoir fait pour `v=0`, `v=1` et `v=2`, on se retrouve avec `distances=[1,0,1,4]`, `a_explorer=[0,2]`, et enfin `marques=[1,0,2]`.
   - On recommence. On prend `s=0` le premier élément de `a_explorer` (qui devient alors la liste `[2]`). Pour chaque voisin `v` de `s` (i.e `v=0` et `v=3`), on met à jour `distances`, `a_explorer` et `marques`. Après l'avoir fait pour `v=0` et `v=3`, on se retrouve avec `distances=[1,0,1,2]`, `a_explorer=[2,3]`, et enfin `marques=[1,0,2,3]`.
   - On recommence. On prend `s=2` le premier élément de `a_explorer` (qui devient alors la liste `[3]`). Pour chaque voisin `v` de `s` (i.e `v=0`, `v=1`, `v=2`), on met à jour `distances`, `a_explorer` et `marques`. Ici, rien ne change car les trois voisins 0, 1 et 2 appartiennent à la liste `marques`. On se retrouve alors avec `distances=[1,0,1,2]`, `a_explorer=[3]`, et enfin `marques=[1,0,2,3]`.
   - On recommence. On prend `s=3` le premier élément de `a_explorer` (qui devient alors la liste vide). Pour chaque voisin `v` de `s` (i.e `v=0` et `v=3`), on met à jour `distances`, `a_explorer` et `marques`. Ici non plus, rien ne change car les deux voisins 0, et 3 appartiennent à la liste `marques`. On se retrouve alors avec `distances=[1,0,1,2]`, `a_explorer=[]`, et enfin `marques=[1,0,2,3]`.
   - La liste `a_explorer` est vide. On a fini.

Conclusion : après application de l'algorithme décrit dans l'énoncé, on a `distances=[1,0,1,2]`.

   Remarque : l'algorithme décrit dans l'énoncé est l'algorithme de Dijkstra.

   (b) On traduit en Python les différentes étapes de l'algorithme. Le programme complété est le suivant :

```
def parcours(L,i0):
    p = len(L)
    distances = [p]*p
    distances[i0] = 0
    a_explorer = [i0]
    marques = [i0]
    while a_explorer != []:
        s = a_explorer[0]
        del a_explorer[0]
        for v in L[s]:
            if v not in marques:
                marques.append(v)
                a_explorer.append(v)
                distances[v] = distances[s]+1
    return distances
```

(c) Après avoir appliqué l'algorithme, la liste de tous les sommets s pour lesquels il existe un chemin menant du sommet de départ s_i au sommet s est simplement la liste `marques` (c'est la liste de tous les sommets qu'on a visités lors de notre exploration du graphe, en partant de s_i).

Conclusion : pour renvoyer cette liste, il suffit de changer la dernière ligne de la fonction en `return marques`.