MATHÉMATIQUES APPLIQUÉES - Ecricome 2023

Proposition de corrigé par David Meneu

Lycée Champollion - Grenoble, pour Major-Prépa

Exercice 1

1. Lors du premier tirage, le numéro obtenu est un entier aléatoire compris entre 1 et n avec équiprobabilité puisque chaque numéro figure sur une seule boule.

   La variable aléatoire X suit donc la loi uniforme sur $[\![1\,;n]\!]$: $\forall k \in [\![1\,;n]\!],\ \mathbf{P}(X=k)=\dfrac{1}{n}$.

   Le cours donne par ailleurs l'espérance et la variance de X : $\quad E(X)=\dfrac{n+1}{2}$ et $V(X)=\dfrac{n^2-1}{12}$.

2. D'après la description de l'expérience, la variable aléatoire Y peut prendre toute valeur entière de 1 à k, où k est lui-même un entier qui peut prendre toutes les valeurs de 1 à n, donc :

$$Y(\Omega)=[\![1\,;n]\!].$$

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

   a) On suppose que l'événement $[X=k]$ est réalisé. Le nombre total de boules présentes dans la seconde urne est donc :

$$1+2+\cdots+k=\sum_{i=1}^{k} i=\frac{k(k+1)}{2},$$

   d'après la formule du cours pour cette somme classique (souvent appelée *somme de Gauss*).

   b) Pour tout $j \in [\![1\,;k]\!]$, toujours sachant que $[X=k]$ est réalisé, il y a j boules portant le numéro j dans l'urne, donc :

$$\text{si } 1 \leqslant j \leqslant k, \quad \mathbf{P}_{[X=k]}(Y=j)=\frac{j}{\frac{k(k+1)}{2}}=\frac{2j}{k(k+1)}.$$

   Par contre si $j \geqslant k+1$, sachant que $[X=k]$ est réalisé il n'y a aucune boule portant le numéro j dans la deuxième urne, donc :

$$\text{si } j \geqslant k+1, \quad \mathbf{P}_{[X=k]}(Y=j)=0.$$

4. a) On cherche ici deux réels a et b tels que :

$$\forall k \in \mathbb{N}^*,\quad \frac{1}{k(k+1)}=\frac{a}{k}+\frac{b}{k+1} \iff \frac{1}{k(k+1)}=\frac{a(k+1)+bk}{k(k+1)} \iff \frac{1}{k(k+1)}=\frac{(a+b)k+a}{k(k+1)}.$$

   Par identification des coefficients de mêmes degrés au numérateur, on en déduit que a et b vérifient les relations :

$$\begin{cases} a+b &= 0 \\ a &= 1 \end{cases} \iff \begin{cases} a &= 1 \\ b &= -1 \end{cases}, \quad \text{soit : } \forall k \in \mathbb{N}^*,\ \frac{1}{k(k+1)}=\frac{1}{k}-\frac{1}{k+1}.$$

b) Soit $j \in Y(\Omega) = [\![1\,;n]\!]$; on obtient $\mathbf{P}(Y=j)$ grâce à la formule des probabilités totales, appliquée avec le système complet d'événement $\big([X=k]\big)_{1\leqslant k\leqslant n}$:

$$\mathbf{P}(Y=j) = \sum_{k=1}^{n} \mathbf{P}(X=k)\times \mathbf{P}_{[X=k]}(Y=j) = \sum_{k=j}^{n} \frac{1}{n}\times\frac{2j}{k(k+1)} \quad \text{car } \mathbf{P}_{[X=k]}(Y=j) = 0 \text{si } k<j$$

$$= \frac{2j}{n}\sum_{k=j}^{n}\frac{1}{k(k+1)} = \frac{2j}{n}\sum_{k=j}^{n}\Big(\frac{1}{k}-\frac{1}{k+1}\Big) = \frac{2j}{n}\Big(\frac{1}{j}-\frac{1}{n+1}\Big)$$

$$\mathbf{P}(Y=j) = \frac{2}{n} - \frac{2j}{n(n+1)} = \frac{2(n+1)-2j}{n(n+1)} = \frac{2(n+1-j)}{n(n+1)}.$$

5. La variable aléatoire Y est finie, donc elle admet une espérance qui vaut :

$$E(Y) = \sum_{j=1}^{n} j\mathbf{P}(Y=j) = \frac{2}{n(n+1)}\sum_{j=1}^{n} j(n+1-j) = \frac{2}{n(n+1)}\Big((n+1)\sum_{j=1}^{n} j - \sum_{j=1}^{n} j^2\Big)$$

$$= \frac{2}{n(n+1)}\Big((n+1)\times\frac{n(n+1)}{2} - \frac{n(n+1)(2n+1)}{6}\Big) = \frac{2}{n(n+1)}\times n(n+1)\Big(\frac{n+1}{2}-\frac{2n+1}{6}\Big)$$

$$= 2\times\frac{3n+3-2n-1}{6} = \frac{n+2}{3}.$$

item Il est assez évident que les variables aléatoires X et Y ne sont *pas* indépendantes, puisque la valeur prise par X détermine toujours la valeur maximale possible pour Y.

Plus concrètement, on obtient facilement un contre-exemple dans le cas général, comme celui-ci : $\mathbf{P}\big([X=1]\cap[Y=2]\big) = 0$ car si $X=1$, la deuxième urne ne contient qu'une boule numérotée 1, et pourtant $\mathbf{P}(X=1)\times\mathbf{P}(Y=2) = \dfrac{1}{n}\times\dfrac{2(n-1)}{n(n+1)} \neq 0$ dès que $n\geqslant 2$.

Remarque : il y a en fait un seul cas où X et Y sont bien indépendantes, lorsque $n=1$: dans ce cas la première urne ne contient que la boule 1 et X prend forcément la valeur 1 ; la deuxième urne contient alors uniquement une boule 1 également, et Y est encore constante égale à 1.

6. a) Puisque X et Y sont des variables aléatoires finies, $E(XY)$ existe et vaut :

$$E(XY) = \sum_{k=1}^{n}\sum_{j=1}^{n} kj\mathbf{P}\big([X=k]\cap[Y=j]\big) = \sum_{k=1}^{n}\sum_{j=1}^{k} kj\frac{1}{n}\frac{2j}{k(k+1)}$$

$$= \frac{2}{n}\sum_{k=1}^{n}\frac{1}{k+1}\sum_{j=1}^{k} j^2 = \frac{2}{n}\sum_{k=1}^{n}\frac{1}{k+1}\times\frac{k(k+1)(2k+1)}{6}$$

$$= \frac{1}{3n}\sum_{k=1}^{n}(2k^2+k) = \frac{1}{3n}\Big(\frac{n(n+1)(2n+1)}{3}+\frac{n(n+1)}{2}\Big)$$

$$= \frac{1}{3}(n+1)\times\Big(\frac{2n+1}{3}+\frac{1}{2}\Big) = \frac{(n+1)(4n+5)}{18}$$

b) Le couple (X,Y) admet alors une covariance qui vaut :

$$\mathrm{Cov}(X,Y) = E(XY) - E(X)E(Y) = \frac{(n+1)(4n+5)}{18} - \frac{(n+1)(n+2)}{6} = \frac{n+1}{18}(4n+5-3n-6)$$

$$= \frac{(n+1)(n-1)}{18} = \frac{n^2-1}{18}.$$

7. a) On peut proposer ici deux rédactions possibles de la fonction `seconde_urne` :

```
def seconde_urne(k) :
    L = [1]
    for j in range(2,k+1):
        L = L + [j]*j
    return L
```

On peut aussi proposer une version avec une double boucle `for` et la méthode `append` :

```
def seconde_urne(k):
    L = [1]
    for j in range(2,k+1):
        for i in range(j):
            L.append(j)
    return L
```

b) La fonction `Python` suivante prend en entier un entier naturel n non nul, et renvoie une réalisation du couple de variables aléatoires (X, Y) :

```
import numpy.random as rd

def simul_XY(n):
    X = rd.randint(1,n+1)   # simulation de la loi uniforme sur [|1;n|]
    urne2 = seconde_urne(X)
    nb = len(urne2)
    i = rd.randint(0,nb)
    Y = urne2[i]
    return X,Y
```

Commentaires :

- C'est bien la valeur de X (notée k dans la question 3.) qui détermine le contenu de la deuxième urne.
- Il faut bien comprendre que l'entier aléatoire i est choisi au hasard avec équiprobabilité, dans un ensemble d'entiers tous distincts compris entre 0 et `len(nb)`-1, qui correspond au nombre total de boules contenues dans la deuxième urne.
  Cet entier i est donc seulement un indice correspondant à un unique emplacement dans la liste `urne2` dont le contenu respecte bien le nombre de boules pour chaque numéro présent : la valeur de Y à rendre, est donc bien l'élément de `urne2` à l'emplacement `i`, c'est-à-dire `urne2[i]`.

c) On recopie ici la fonction `Python` de l'énoncé pour la commenter ensuite :

```
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
```

La fonction initialise donc une liste de taille $n = \text{Card}\big(Y(\Omega)\big)$, pleine de zéros au départ.
Ensuite, on réalise 10000 simulations successives du couple (X, Y) pour cette même valeur de n, et `j` ne conserve que la deuxième composante (numéro 1) du couple, à savoir la simulation de Y.
L'élément de la variable `liste` d'indice $j-1$ (décalage rendu nécessaire par le fait que la numérotation des indices de listes en `Python` commence à 0 alors que $Y(\Omega) = [\![1\,;n]\!]$), est alors incrémenté de $1/10000$.
Pour tout $j \in [\![1\,;n]\!]$, l'élément `L[j-1]` de la liste finale rendue par la fonction, contient donc la fréquence d'apparition de j comme valeur de Y. Avec une taille d'échantillon de cet ordre de grandeur, on peut considérer que les éléments successifs de la liste rendue permettent d'estimer les probabilités $\big(\mathbf{P}(Y=j)\big)_{1\leqslant i\leqslant n}$.

8. Dans cette question, on suppose $n = 20$.

a) La loi faible des grands nombres assure que la moyenne des valeurs prises par les simulations de X, sera proche de $E(X) = \dfrac{n+1}{2} = 11,5$. Il en sera de même pour la moyenne des valeurs prises par Y sera proche de $E(Y) = \dfrac{n+2}{3} = \dfrac{22}{3} \approx 7,33$.

Le point moyen du nuage de points a alors pour coordonnées approximatives $(11,5\,;7,33)$.

b) On peut procéder par éliminations dans cette question :

- La figure 4 peut être assez rapidement écartée : la droite tracée sur le graphique n'est pas cohérente avec l'allure générale du nuage de points pour en être sa droite de régression linéaire. Rappelons ici que le coefficient directeur de cette droite est égal au coefficient de corrélation linéaire du nuage de points, qui est lui-même du même signe que la covariance de X et Y, donc positif ici.
- On peut également éliminer la figure 2 au regard du fait que la droite tracée est entièrement au-dessus du nuage de points, ce qui n'est pas cohérent avec la définition de la droite de régression linéaire : on peut très facilement sur ce graphique, tracer une droite qui approche mieux le nuage qui aura certains points au-dessus de cette droite.
- La figure 1 doit aussi être éliminée, cette fois pour des raisons de cohérences des univers-images considérés : comme ici $n = 20$ alors $X(\Omega) = Y(\Omega) = [\![1\,;20]\!]$, et le nuage de points considéré ici contient des points dont l'ordonnée, qui correspond aux valeurs de Y, dépassent strictement cette valeur.

On conclut donc que c'est la figure 3 qui correspond au nuage de points et à la droite de régression linéaire étudiés : aucun des reproches faits aux trois graphiques précédent ne s'applique à ce graphique, la position de la droite de régression linéaire est cohérente au sein du nuage, les contraintes entre les valeurs de l'abscisse et de l'ordonnée de chaque point sont bien respectées.

Exercice 2

On considère la fonction f définie sur $]0\,;+\infty[$ par : $\quad \forall x \in]0\,;+\infty[, \quad f(x) = \dfrac{e^{\frac{x}{2}}}{\sqrt{x}}$.

1. a) La fonction f est bien dérivable sur $]0\,;+\infty[$ comme quotient de deux fonctions dérivables sur cet intervalle, et :

$$\forall x \in]0\,;+\infty[, \quad f'(x) = \frac{\frac{1}{2}e^{\frac{x}{2}}\sqrt{x} - e^{\frac{x}{2}}\cdot\frac{1}{2\sqrt{x}}}{(\sqrt{x})^2} = \frac{1}{x} \times \frac{e^{\frac{x}{2}}.(\sqrt{x})^2 - e^{\frac{x}{2}}}{2\sqrt{x}} = \frac{x-1}{2x} \times \frac{e^{\frac{x}{2}}}{\sqrt{x}} = \frac{(x-1)}{2x} f(x).$$

b) Sur le domaine $]0\,;+\infty[$, il est clair qu'on a toujours $2x > 0$ et $f(x) = \dfrac{e^{\frac{x}{2}}}{\sqrt{x}} > 0$, donc $f'(x)$ est sur cet intervalle, du signe de $x - 1$.

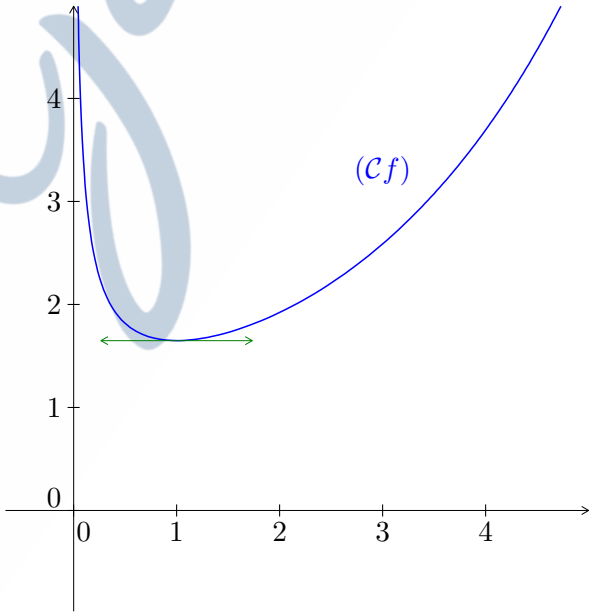
On en déduit donc que f est strictement décroissante sur $]0\,;1]$, puis strictement croissante sur $[1\,;+\infty[$.

Calcul des limites aux bornes du domaine :

- $\lim\limits_{x\to 0} e^{\frac{x}{2}} = e^0 = 1$ par continuité de l'exponentielle sur $\mathbb{R}$, et $\lim\limits_{x\to 0} \sqrt{x} = 0^+$, donc par quotient $\lim\limits_{x\to 0} f(x) = +\infty$.
- On a directement $\lim\limits_{x\to+\infty} \dfrac{e^{\frac{x}{2}}}{\sqrt{x}} = +\infty$: c'est un résultat de croissances comparées du type $\lim\limits_{x\to+\infty} \dfrac{e^{\beta x}}{x^\alpha}$ avec $\beta = \alpha = \frac{1}{2} > 0$.

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

c) On trace l'allure de la courbe représentative de f en tenant compte des propriétés obtenues : la courbe admet une tangente horizontale au point d'abscisse 1 (et d'ordonnée $f(1) = e^{1/2} = \sqrt{e}$ compris entre 1 et 2), ainsi qu'une asymptote verticale au voisinage de 0.



d) Soit n un entier supérieur ou égal à 2 : comme on l'a déjà dit, $1 < e^{1/2} < 2$ (car $1 < e < 4$), donc $n > e^{1/2}$, et :

- Sur $]0\,;1]$, la fonction f est continue (car dérivable), strictement croissante et n appartient à l'intervalle $[f(1)\,;\lim\limits_{x\to 0} f(x)[= [e^{1/2}\,;+\infty[$.

  D'après le théorème de la bijection, l'équation $f(x) = n$ admet donc une unique solution u_n sur $]0\,;1]$ (avec $u_n < 1$ puisque $f(1) < n$).
- Sur $]1\,;+\infty[$, la fonction f est continue, strictement croissante et n appartient à l'intervalle $[f(1)\,;\lim\limits_{x\to+\infty} f(x)[= [e^{1/2}\,;+\infty[$.

  D'après le même théorème, l'équation $f(x) = n$ admet une unique solution v_n sur $]1\,;+\infty[$.

Pour tout entier $n \geqslant 2$, l'équation $f(x) = n$ admet donc exactement 2 solutions u_n et v_n sur $]0\,;+\infty[$, telles que $0 < u_n < 1 < v_n$.

2. a) Pour tout entier $n \geqslant 2$, l'inégalité évidente : $n < n+1$ se réécrit aussi $f(v_n) < f(v_{n+1})$.

Or tous les termes de la suite $(v_n)_{n\geqslant 2}$ appartiennent à l'intervalle $[1\,;+\infty[$ sur lequel la fonction f est strictement croissante : on en déduit que $\forall n \geqslant 2,\ v_n < v_{n+1}$ (les antécédents sont rangés dans le même ordre que les images), ce qui prouve que la suite $(v_n)_{n\geqslant 2}$ est (strictement) croissante.

b) Suppposons ici que la suite $(v_n)_{n\geqslant 2}$ converge vers une limite $\ell \geqslant 1$: alors par continuité de f sur $[1\,;+\infty[$, on a :

$$\lim_{n\to+\infty} f(v_n) = f(\ell) \iff \lim_{n\to+\infty} n = f(\ell),$$

ce qui est évidemment absurde puisque $f(\ell)$ est un réel, tandis que $\lim\limits_{n\to+\infty} n = +\infty$.

3. a) Selon un raisonnement analogue à celui de la question 2.a) :

$$\forall n \geqslant 2, \quad n < n+1 \iff f(u_n) < f(u_{n+1}) \iff u_n > u_{n+1},$$

puisque tous les termes de la suite $(u_n)_{n\geqslant 2}$ appartiennent à l'intervalle $]0\,;1[$, sur lequel la fonction f est strictement décroissante. La suite $(u_n)_{n\geqslant 2}$ est donc bien strictement décroissante.

b) On vient de voir que la suite $(u_n)_{n\geqslant 2}$ est décroissante ; or elle est aussi minorée par 0, puisque : $\forall n \geqslant 2,\ 0 < u_n < 1$ (voir la question 1.d)).

La suite $(u_n)_{n\geqslant 2}$ est donc convergente, d'après le théorème de limite monotone ; notons ℓ sa limite, qui est un nombre réel compris entre 0 et 1.

c) Si on suppose $\ell > 0$, alors de façon analogue à ce qui a été fait en 2.b), on peut écrire :

$$\lim_{n\to+\infty} f(u_n) = f(\ell) \in \mathbb{R} \iff \lim_{n\to+\infty} n = f(\ell) \in \mathbb{R},$$

ce qui est à nouveau absurde. Ainsi donc $\ell \geqslant 0$, mais $\ell > 0$ est impossible : on conclut que

$$\ell = 0 = \lim_{n\to+\infty} u_n.$$

d) Examinons plus précisément l'équation que u_n est le seul réel de $]0\,;1[$ à vérifier :

$$\forall n \geqslant 2, \quad f(u_n) = n \iff \frac{e^{\frac{u_n}{2}}}{\sqrt{u_n}} = n \iff e^{\frac{u_n}{2}} = n\sqrt{u_n} \iff e^{u_n} = n^2 u_n.$$

Ainsi, par continuité de l'exponentielle sur $\mathbb{R}$:

$$\lim_{n\to+\infty} n^2 u_n = \lim_{n\to+\infty} e^{u_n} = e^0 = 1.$$

Ce résultat s'écrit aussi : $\lim\limits_{n\to+\infty} \dfrac{u_n}{\frac{1}{n^2}} = 1$, et donne donc l'équivalent :

$$u_n \underset{n\to+\infty}{\sim} \frac{1}{n^2}.$$

4. a) L'algorithme de dichotomie est depuis longtemps un incontournable des sujets de l'ex-voie ECE, qui a été repris tel quel dans le programme de Mathématiques Appliquées ; il est toujours indispensable de le maîtriser parfaitement !

```
import numpy as np

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

Comme la fonction f est strictement décroissante sur $]0\,;1[$:

si $f(c) < n$, alors $f(c) < f(u_n) \iff c > u_n$ et u_n appartient à l'intervalle $[a\,;c]$: il convient donc d'actualiser b en lui donnant la valeur de c.
Sinon, $u_n \in [c\,;b]$ et c'est a qui est actualisée, on lui donne la valeur de c.
L'algorithme s'arrête lorsque pour la première fois, $b-a \leqslant \varepsilon$: cela signifie que a et b encadrent u_n à ε près, et un dernier calcul du milieu de l'intervalle fournit bien *une* valeur approchée de u_n à ε près.

b) L'algorithme de calcul d'une somme est assez facile à écrire grâce à une boucle `for`. Il y a cependant ici une vraie subtilité à laquelle il fallait prendre garde :

L'algorithme de dichotomie précédent calcule seulement une valeur approchée de chaque terme u_n à la précision ε, c'est-à-dire que la valeur rendue à chaque fois contient potentiellement une erreur d'approximation comprise entre $-\varepsilon$ et $+\varepsilon$.
Comme on ajoute ces valeurs approchées, les erreurs d'approximation s'ajoutent aussi ! Concrètement puisqu'il y a $N-1$ termes dans cette somme, l'erreur d'approximation finale pour la valeur de $\displaystyle\sum_{n=2}^{N} u_n$ sera comprise entre $-(N-1)\varepsilon$ et $+(N-1)\varepsilon$.
Il conviendra donc, dans chaque appel à la fonction `approx_u`, de demander plutôt une précision $\frac{\varepsilon}{N}$ pour que le résultat rendu soit bien une valeur approchée de la somme voulue à ε près.

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

Exercice 3

Partie 1

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

1. Soient les vecteurs

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

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

a) La famille $\mathscr{B}$ est constituée de 4 vecteurs dans l'espace vectoriel $\mathbb{R}^4$ qui est de dimension 4 : il suffit donc de prouver que $\mathscr{B}$ est libre, pour que ce soit une base de $\mathbb{R}^4$.

On considère donc 4 réels a, b, c, d tels que :

$$a.u_1 + b.u_2 + c.u_3 + d.u_4 = 0_{\mathbb{R}^4} \iff \begin{cases} -a & & & + & d & = & 0 \\ a & - & b & + & c & = & 0 \\ & & b & + & c & = & 0 \\ a & & & + & d & = & 0 \end{cases}$$

$$\iff \begin{cases} -a & & & + & d & = & 0 \\ & - & b & + & c & + & d & = & 0 \quad L_2 \leftarrow L_2 + L_1 \\ & & b & + & c & & & = & 0 \\ & & & & & & 2d & = & 0 \quad L_4 \leftarrow L_4 + L_1 \end{cases}$$

$$\iff \begin{cases} -a & & & & & + & d & = & 0 \\ & - & b & + & c & + & d & = & 0 \\ & & & & 2c & + & d & = & 0 \quad L_3 \leftarrow L_3 + L_2 \\ & & & & & & 2d & = & 0 \end{cases}$$

Le système est échelonné et de Cramer, il admet bien pour unique solution $d = c = b = a = 0$: la famille $\mathscr{B}$ est bien libre, c'est une base de $\mathbb{R}^4$.

b) Pour déterminer la matrice de f dans la base $\mathscr{B}$, il s'agit de calculer les images $f(u_1), f(u_2), f(u_3), f(u_4)$ et de les exprimer **en fonction des vecteurs** u_1, u_2, u_3, u_4.
La matrice A permet d'obtenir ces images via un simple calcul matriciel :

- $A \begin{pmatrix} -1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, donc $f(u_1) = (0, 0, 0, 0) = 0.u_1 + 0.u_2 + 0.u_3 + 0.u_4$.
- $A \begin{pmatrix} 0 \\ -1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, donc $f(u_2) = (0, 0, 0, 0) = 0.u_1 + 0.u_2 + 0.u_3 + 0.u_4$ également.
- $A \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 2 \\ 0 \end{pmatrix}$, donc $f(u_3) = (0, 2, 2, 0)$: on reconnaît que $f(u_3) = 2.u_3$.

- $A\begin{pmatrix}1\\0\\0\\1\end{pmatrix}=\begin{pmatrix}2\\1\\1\\2\end{pmatrix}$, donc $f(u_4)=(2,1,1,2)$ qui se décompose facilement en : $f(u_4)=u_3+2.u_4$.

On en déduit la matrice représentative de f dans la base $\mathscr{B}=(u_1,u_2,u_3,u_4)$:

$$\begin{array}{c} \begin{matrix} f(u_1) & f(u_2) & f(u_3) & f(u_4) \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 2 \end{pmatrix} \end{array} \begin{matrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{matrix}$$

c) En notant T la matrice précédente (qui est bien triangulaire comme son nom le suggère) : la formule de changement de base assure que :

$$A=PTP^{-1},$$

où P est la matrice de passage de la base canonique $\mathscr{C}$ à la base $\mathscr{B}$: on l'obtient explicitement en écrivant les coordonnées des vecteurs u_1,u_2,u_3,u_4 en colonne, soit $P=\begin{pmatrix}-1&0&0&1\\1&-1&1&0\\0&1&1&0\\1&0&0&1\end{pmatrix}$.

2. a) Les calculs matriciels donnent : $A^2=\begin{pmatrix}2&0&0&2\\3&2&2&1\\3&2&2&1\\2&0&0&2\end{pmatrix}$ et $A^3=\begin{pmatrix}4&0&0&4\\8&4&4&4\\8&4&4&4\\4&0&0&4\end{pmatrix}$.

On remarque qu'en effet : $4A^2-4A=\begin{pmatrix}8&0&0&8\\12&8&8&4\\12&8&8&4\\8&0&0&8\end{pmatrix}-\begin{pmatrix}4&0&0&4\\4&4&4&0\\4&4&4&0\\4&0&0&4\end{pmatrix}=\begin{pmatrix}4&0&0&4\\8&4&4&4\\8&4&4&4\\4&0&0&4\end{pmatrix}=A^3$.

b) Montrons par récurrence que la propriété $\mathcal{P}(n)$: "il existe deux réels a_n et b_n tels que $A^n=a_n.A^2+b_n.A$", est vraie pour tout $n\in\mathbb{N}^*$.

I. Pour $n=1$: $A^1=A$ s'écrit évidemment $0.A^2+1.A$, donc $\mathcal{P}(0)$ est vraie avec $a_1=0$ et $b_1=1$.

H. Supposons $\mathcal{P}(n)$ vraie pour un certain $n\in\mathbb{N}^*$, et montrons qu'alors $\mathcal{P}(n+1)$ est encore vraie, soit : il existe deux réels a_{n+1} et b_{n+1} tels que $A^{n+1}=a_{n+1}.A^2+b_{n+1}.A$.

$$\begin{aligned}A^{n+1}=A^n\times A &\overset{H.R.}{=}(a_n.A^2+b_n.A)\times A=a_n.A^3+b_n.A^2 \overset{2.a)}{=} a_n.(4A^2-4A)+b_n.A^2\\ &=(4a_n+b_n).A^2+4a_n.A,\end{aligned}$$

donc si on pose $\begin{cases}a_{n+1}&=4a_n+b_n\\b_{n+1}&=-4a_n\end{cases}$, alors $\mathcal{P}(n+1)$ est vraie si $\mathcal{P}(n)$ l'est.

C. La propriété est initialisée et héréditaire : elle est donc vraie pour tout $n\in\mathbb{N}^*$, d'après le théorème de récurrence.

L'étape d'hérédité a bien fait apparaître les relations de récurrence demandées pour les deux suites $(a_n)_{n\in\mathbb{N}^*}$ et $(b_n)_{n\in\mathbb{N}^*}$.

3. a) En utilisant les relations de récurrence précédemment obtenues, on peut directement écrire :

$$\forall n\in\mathbb{N},\quad a_{n+2}=4a_{n+1}+b_{n+1}=4a_{n+1}-4a_n.$$

b) La suite $(a_n)_{n\in\mathbb{N}^*}$ vérifie donc une relation de récurrence linéaire d'ordre 2, d'équation caractéristique : $x^2 = 4x - 4 \iff x^2 - 4x + 4 = 0 \iff (x-2)^2 = 0$.

Cette équation admet donc une unique solution $r_0 = 2$: d'après le cours (de ECG1 !) il existe deux réels λ et μ tels que :

$$\forall n \in \mathbb{N}^*, \quad a_n = (\lambda + \mu.n)2^n.$$

On obtient les valeurs de λ et μ en écrivant cette formule générale aux deux premiers rangs $n = 1$ et $n = 2$, sachant que $a_1 = 0$ et que $a_2 = 4a_1 + b_1 = 1$:

$$\begin{cases} (\lambda + \mu.1)2^1 &= a_1 \\ (\lambda + \mu.2)2^2 &= a_2 \end{cases} \iff \begin{cases} 2\lambda + 2\mu &= 0 \\ 4\lambda + 8\mu &= 1 \end{cases} \iff \begin{cases} \mu = -\lambda \\ -4\lambda = 1 \end{cases} \iff \begin{cases} \lambda = -\frac{1}{4} \\ \mu = -\lambda = \frac{1}{4} \end{cases}$$, donc

pour tout entier $n \in \mathbb{N}^*$:

$$a_n = \big(-\tfrac{1}{4} + \tfrac{1}{4}n\big)2^n = (n-1)2^{n-2}.$$

c) La première relation de récurrence de la question 3.a) par exemple, permet d'en déduire b_n :

$$\forall n \in \mathbb{N}^*,\ a_{n+1} = 4a_n + b_n \iff b_n = n2^{n-1} - 4(n-1)2^{n-2} = (2n - 4n + 4)2^{n-2} = (-2n+4)2^{n-2}.$$

4. Il ne reste plus qu'à réaliser le calcul explicite de $A^n = a_n.A^2 + b_n.A$ avec les expressions explicites obtenues : pour tout $n \in \mathbb{N}^*$,

$$A^n = \begin{pmatrix} 2a_n + b_n & 0 & 0 & 2a_n + b_n \\ 3a_n + b_n & 2a_n + b_n & 2a_n + b_n & a_n \\ 3a_n + b_n & 2a_n + b_n & 2a_n + b_n & a_n \\ 2a_n + b_n & 0 & 0 & 2a_n + b_n \end{pmatrix} = \begin{pmatrix} 2^{n-1} & 0 & 0 & 2^{n-1} \\ (n+1)2^{n-2} & 2^{n-1} & 2^{n-1} & (n-1)2^{n-2} \\ (n+1)2^{n-2} & 2^{n-1} & 2^{n-1} & (n-1)2^{n-2} \\ 2^{n-1} & 0 & 0 & 2^{n-1} \end{pmatrix}.$$

Quelques calculs détaillés : $2a_n + b_n = (2n-2)2^{n-2} + (-2n+4)2^{n-2} = 2 \times 2^{n-2} = 2^{n-1}$,
et $3a_n + b_n = (3n-3)2^{n-2} + (-2n+4)2^{n-2} = (n+1)2^{n-2}$.

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.

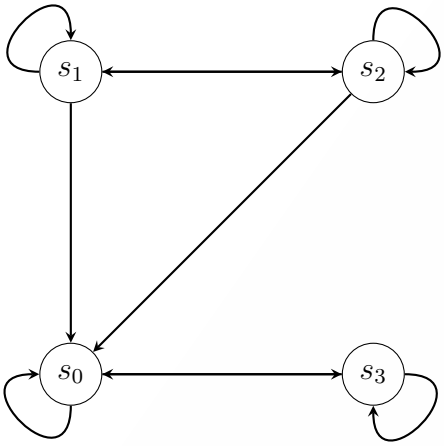
5. a) Par définition, la matrice d'adjacence du graphe G est la matrice carrée d'ordre p, dont chaque coefficient d'indice (i,j) vaut 1 s'il existe une arête directe du sommet s_{i-1} au sommet s_{j-1}, et 0 sinon.


En mathématiques les coefficients d'une matrice sont numérotés à partir de l'indice 1, alors qu'ici les sommets sont numérotés à partir de l'indice 0 (en prévision de l'implémentation en `Python` : il y a donc forcément un décalage d'indice à gérer !

b) Soient $n \in \mathbb{N}^*$, $(i,j) \in [\![1\,;p]\!]^2$.
Toujours d'après le cours, le coefficient d'indices (i,j) de la matrice M^n est égal au nombre de chemins de longueur n dans le graphe G, depuis le sommet s_{i-1} vers le somme s_{j-1}.

6. Dans cette question, 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) Le graphe considéré a donc 4 sommets s_0, s_1, s_2, s_3 :



b) On sait d'après le cours, que ce graphe d'ordre 4 est connexe si et seulement si la matrice $I_4 + A + A^2 + A^3$ a des coefficients tous strictement positifs.
Or il est facile de voir que I_4, A, A^2, A^3 ont des coefficients nuls aux mêmes indices, concrètement aux indices $(1,2), (1,3), (3,2), (3,3)$, ce sera donc aussi le cas de leur somme.

Le graphe G n'est donc pas connexe.

c) Soit $n \in \mathbb{N}^*$: le nombre de chemins de longueur n menant du sommet s_3 au sommet s_0 est égal au coefficient de A^n d'indices $(4,1)$: il est donc égal à 2^{n-1}, d'après le résultat de la question 4.

7. La fonction ci-dessous réalise le travail demandé : elle parcourt la matrice d'adjacence ligne par ligne : à la ligne i, les coefficients égaux à 1 correspondent à des voisins du sommet s_{i-1} et sont ajoutés en tant que tels à la liste des voisins de ce sommet, puis on passe à la ligne suivante. Le résultat rendu est bien une liste de listes.

```
import numpy as np

def matrice_vers_liste(A):
    n,p = np.shape(A)   # format de A : on doit avoir n=p
    L = []
    for i in range(n):
        voisins = []
        for j in range(n):
            if A[i][j]==1:
                voisins.append(j)
        L.append(voisins)
    return L
```

8. a) Puisque la liste `distances` doit contenir la liste des plus courts chemins de s_1 à chacun des sommets s_i du graphe, on aura logiquement :

`distances = [1,0,1,2]`

à la fin de l'algorithme : en effet s_0 est directement voisin de s_1, de même que s_2 ; s_1 est à une distance nulle de lui-même, et le chemin le plus court de s_1 à s_3 est : $s_1 \to s_0 \to s_3$.

b) La fonction suivante réalise les instructions de l'algorithme décrites ci-dessus :

```
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) Il suffit ici de modifier ce que renvoie la fonction : on peut demander à Python de ne renvoyer que les indices `k` tels que `distances[k]` ne soit plus égal à p à la fin de la boucle : ce sera le signe que l'élément `distances[k]` a été modifié, donc qu'on a trouvé un chemin de s_i à s_k.
Le plus simple est de réaliser une définition par compréhension de cette liste à la dernière ligne :

```
def sommets(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 [k for k in range(p) if distances[k] < p]
```

Dans notre exemple, cette fonction va rendre la liste de tous les sommets `[1,2,3,4]` lorsque `i0`$= 1$ ou 2 puisque tous les sommets du graphe G sont accessibles depuis les sommets s_1 et s_2. Par contre, elle ne rendra que la liste `[0,3]` pour `i0`$= 0$ ou 3.