DEVOIR MAISON # 1
à rendre le Jeudi 12/09/2019

Consignes

La qualité de la rédaction, la clarté et la précision des raisonnements, entreront pour une part importante dans l'appréciation de la copie. Les abréviations, sigles ou phrases nominales sont à proscrire. La numérotation des exercices (et des questions) doit être respectée et mise en évidence. Les résultats doivent être encadrés proprement.
Vous avez la possibilité de rendre un devoir pour deux, mais les écritures doivent alors apparaître en parts égales. Il est rappelé également que recopier une correction sur internet est complètement inutile pour tout le monde.

Exercice 1 Soient $A = \begin{pmatrix} 2 & 0 & 1 \\ 1 & 1 & 0 \\ -1 & 1 & 3 \end{pmatrix}$ et $P = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}$.

1 — Déterminer $\mathrm{Ker}(A - \lambda I_3)$ pour tout $\lambda \in \mathbf{R}$.

2 — Étudier l'inversibilité de la matrice P et calculer $T = P^{-1}AP$.

3 — Exprimer A en fonction de T, P, P^{-1} puis A^n en fonction de T^n, P et P^{-1} pour tout $n \in \mathbf{N}$.

4 — Calculer T^n pour tout entier $n \in \mathbf{N}$. *Indication : On pourra utiliser le binôme de Newton.*

5 — En déduire A^n pour tout $n \in \mathbf{N}$.

Exercice 2 Soit E l'ensemble des suites réelles $(u_n)_{n\in\mathbf{N}}$ telles que pour tout $n \in \mathbf{N}^*$,

$$(n+1)u_{n+1} - (n+2)u_n + u_{n-1} = 0.$$

1 — Écrire un programme qui demande les valeurs de u_0 et u_1 et renvoie la valeur de u_{100}.

2 — Montrer que E est un sous-espace vectoriel de l'espace vectoriel des suites réelles. On admet que la dimension de E est finie égale à 2.

3 — Soit a et b les deux suites définies par : $\forall n \in \mathbf{N}, \quad a_n = 1$ et $b_n = \sum_{i=0}^{n} \frac{1}{i!}$.
Montrer que (a, b) est une famille libre de E. En déduire une base de E

4 — Soit p un entier fixé et $F_p = \{(u_n)_{n\in\mathbf{N}} \in E,\ u_p = 0\}$. Montrer que F_p est un sous-espace vectoriel de E et donner une base.

Exercice 3 Un échiquier est un plateau de 8 lignes et 8 colonnes. Ces lignes et ces colonnes seront, dans cet exercice, numérotées de 0 à 7. Une position de l'échiquier est un couple $[i, j]$ d'entiers compris entre 0 et 7 inclus, avec i le numéro de ligne et j le numéro de colonne.
Un cavalier placé sur l'échiquier se déplace en bougeant de 2 cases dans une direction (verticale ou horizontale) et de 1 case perpendiculairement. Si le cavalier est loin des bords de l'échiquier, il a 8 possibilités de déplacements, mais il en a moins s'il est près des bords.

1 — Illustrer sur un brouillon les deux cas énoncés précédemment.

2 — Écrire une fonction `valide` prenant en argument deux entiers relatifs i et j et vérifiant que le couple $[i, j]$ est bien une position de l'échiquier *i.e.* `valide` doit renvoyer un booléen.

3 — Écrire une fonction `coupsSuivants` prenant en argument une position $[i, j]$ et renvoyant la liste des positions que peut atteindre un cavalier placé en $[i, j]$ en un seul coup.

4 — Écrire une fonction `cavalier` prenant en argument une position $[a, b]$ et renvoyant un tableau T carré d'ordre 8 telle que $T[i, j]$ est le nombre minimum de coups nécessaires à un cavalier placé en position $[a, b]$ pour arriver à la position $[i, j]$. *Indication : On pourra commencer par voir l'échiquier comme une liste de de taille 64.*

Correction

Solution (Exercice 1)

1 — On cherche donc à résoudre en $X \in \mathfrak{M}_{n,1}(\mathbf{K})$ le système $(-\lambda I_3) = 0$. Échelonnons la matrice associée

$$\begin{pmatrix} 2-\lambda & 0 & 1 \\ 1 & 1-\lambda & 0 \\ -1 & 1 & 3-\lambda \end{pmatrix} = \begin{pmatrix} -1 & 1 & 3-\lambda \\ 1 & 1-\lambda & 0 \\ 2-\lambda & 0 & 1 \end{pmatrix} \qquad L_1 \leftrightarrow L_3$$

$$= \begin{pmatrix} -1 & 1 & 3-\lambda \\ 0 & 2-\lambda & 3-\lambda \\ 0 & 2-\lambda & 1+(3-\lambda)(2-\lambda) \end{pmatrix} \qquad L_2 \leftrightarrow L_2 + L_1, \quad L_3 \leftarrow L_3 + (2-\lambda)L_1$$

$$= \begin{pmatrix} -1 & 1 & 3-\lambda \\ 0 & 2-\lambda & 3-\lambda \\ 0 & 0 & (\lambda-2)^2 \end{pmatrix} \qquad L_3 \leftrightarrow L_3 - L_2$$

Donc :

$$X = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathrm{Ker}(A - \lambda I_3) \iff \left\{ \begin{array}{rcl} -x + y + (3-\lambda)z & = & 0 \\ (2-\lambda)y + (3-\lambda)z & = & 0 \\ (\lambda-2)^2 z & = & 0 \end{array} \right. .$$

Ainsi, si $\lambda = 2$, on obtient comme système : $\left\{ \begin{array}{rcl} -x+y+z & = & 0 \\ z & = & 0 \end{array} \right.$ donc l'ensemble des solutions est $\boxed{\mathrm{Vect}\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}}$.

Sinon, l'unique solution est $\boxed{\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}}$.

2 — En utilisant l'algorithme du pivot de Gauß, on trouve que P est inversible d'inverse $\boxed{P^{-1} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -1 & 0 \\ -1 & 1 & 1 \end{pmatrix}}$. Un calcul de produit matriciel montre que $\boxed{T = P^{-1}AP = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix}}$.

3 — Ainsi $A = PTP^{-1}$ et montrons par récurrence sur $n \in \mathbf{N}$ que $A^n = PT^nP^{-1}$.

■ ***Initialisation.*** Pour $n = 0$ la propriété est immédiate.
■ ***Hérédité.*** Supposons la propriété vraie au rang $n \in \mathbf{N}$, alors $A^{n+1} = A^nA = PT^nP^{-1}PTP^{-1} = PT^{n+1}P^{-1}$. La propriété est donc vraie pour tout $n \in \mathbf{N}$ par principe de récurrence.

4 — On peut séparer en deux la matrice T : $T = 2I_3 + N$ où $N = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$. On vérifie facilement que $N^2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$, puis que $N^3 = 0$, et que $2I_3$ commute avec N. Donc, d'après le binôme de Newton matriciel :

$$\mathrm{T}^n = \binom{n}{0}2^n\mathrm{I}_3 + \binom{n}{1}2^{n-1}\mathrm{I}_3\mathrm{N} + \binom{n}{2}2^{n-2}\mathrm{I}_3\mathrm{N}^2 + 0.$$

Après calculs, on trouve finalement :

$$\boxed{\mathrm{T}^n = \begin{pmatrix} 2^n & n2^{n-1} & n(n-1)2^{n-3} \\ 0 & 0 & 2^n \end{pmatrix}.}$$

5 — Puis $\mathrm{A}^n = \mathrm{P}\begin{pmatrix} 2^n & n2^{n-1} & n(n-1)2^{n-3} \\ 0 & 0 & 2^n \end{pmatrix}\mathrm{P}^{-1} = \dots$

Solution (Exercice 2)

1 —

```python
def suite_u(u0,u1,n):
    u = u0
    v = u1
    for k in range(2,n+1):
        S = (1/k)*((k+1)*v-u)
        u = v
        v = S
    return S
```

Ou bien de manière récursive

```python
def suite_u_rec(u0,u1,n):
    if n==0:
        return u0
    if n==1:
        return u1
    return 1/(n)*((n+1)*suite_u_rec(u0,u1,n-1)-suite_u_rec(u0,u1,n-2))
```

2 — Montrons qu'il s'agit d'un sous-espace vectoriel de l'ensemble des suites à valeurs réelles. Soient $(u_n)_{n\in\mathbf{N}}$ et $(v_n)_{\mathrm{N}\in\mathbf{N}}$, deux suites à valeurs réelles, $\lambda, \mu \in \mathbf{R}$. Alors, pour tout entier $n \in \mathbf{N}$:

$$\begin{aligned}&(n+1)(\lambda u_{n+1} + \mu v_{n+1}) - (n+2)(\lambda u_n + \mu v_n) + (\lambda u_{n-1} + \mu v_{n-1}) \\ &= \lambda((n+1)u_{n+1} - (n+2)u_n + u_{n-1}) + \mu((n+1)v_{n+1} - (n+2)v_n + v_{n-1}) = 0.\end{aligned}$$

De plus, l'ensemble est inclus dans l'ensemble des suites réelles, et la suite nulle est solution. Donc $\boxed{\text{E est un espace vectoriel.}}$

3 — On commence par vérifier que a, b sont deux éléments de E.
Ensuite, soient $\lambda, \mu \in \mathbf{R}$. Alors supposons que pour tout entier $n \in \mathbf{N}$, $\lambda + \mu\sum_{i=0}^{n}\frac{1}{i!} = 0$. En choisissant $n = 0$ et $n = 1$ on forme le système $\lambda + \mu = 0, \lambda + 2\mu = 0$. Puis on déduit facilement $\lambda = \mu = 0$. $\boxed{\text{La famille } (a, b) \text{ est libre}}$. Nous avons une famille libre de cardinal deux, dans un espace vectoriel de dimension deux donc $\boxed{(a, b) \text{ est une base.}}$

4 — Notons E_p l'espace vectoriel des suites qui s'annulent au rang p. Alors $\mathrm{F}_p = \mathrm{E} \cap \mathrm{E}_p$. On vérifie sans peine que E_p est un sous-espace vectoriel de E donc F_p est une intersection de deux sous-espaces vectoriels, c'est un sous-espace vectoriel.
Cherchons une base. Cette fois-ci, les deux suites a, b ne sont pas dans F_p. Mais comme $\mathrm{F}_p \subset \mathrm{E}$, tout élément de F_p s'écrit sous la forme $\lambda a + \mu b$ avec $\lambda, \mu \in \mathbf{R}$. Or, $\lambda a + \mu b \in \mathrm{F}_p$ si et seulement si

$$\lambda + \mu\sum_{i=0}^{p}\frac{1}{i!} = 0.$$

Notons $a_p = \sum_{i=0}^{p}\frac{1}{i!} \neq 0$, alors $\mu = -\frac{1}{a_p}\lambda$. Donc dit autrement $\mathrm{F}_p \subset \mathrm{Vect}\left(a - 1/a_p b\right)$. Inversement, la suite $a - 1/a_p b$ s'annule au rang p donc est dans F_p, et est non nulle. Donc $\boxed{\text{une base de } \mathrm{F}_p \text{ est donnée par la famille } (a - 1/a_p b).}$

■■ Solution (Exercice 3)

1 —

2 —

```python
def valide(i,j):
    return i in range(8) and j in range(8)
```

3 —

```python
def coupsSuivants(pos):
    i,j = pos
    cs = []
    for dp in [[2,1],[2,-1],[-2,1],[-2,-1],
               [1,2],[1,-2],[-1,2],[-1,-2]]:
        ni = i + dp[0]
        nj = j + dp[1]
        if valide(ni,nj) :
            cs.append([ni,nj]) #append ajoute l'élément en fin
    return cs
```

4 —

```python
def cavalier(pos):
    L = 64*[None]
    oldpos = [pos]
    n = 0
    while None in L:
        newpos = []
        for p in oldpos :
            if L[8*p[0]+p[1]] == None :
                L[8*p[0]+p[1]] = n
            newpos.extend(coupsSuivants(p))
        oldpos = newpos
        n += 1
    #Transformation de la grosse liste en échiquier
    T = []
    for i in range(8):
            #Création de la ligne i
        T.append(L[8*i:8*i+8])
    return T
```