# DS 3 mathématiques

BCPST 1B 2018-2019

- Durée : 3 heures.
- Documents et calculatrice non autorisés.
- Une importance est accordée à la clarté, à la concision et à la précision de la rédaction.

## Étude d'une famille de suites

Dans tout le problème, pour tout $m \in \mathbb{R}$, pour tout $(a,b) \in \mathbb{R}^2$, on pose

$$S_m(a,b) = \{(u_n)_{n\in\mathbb{N}} \text{ suites réelles } | u_0 = a, u_1 = b, \forall n \in \mathbb{N}, u_{n+2} = u_{n+1} + m \cdot u_n\}.$$

1. Soit $m \in \mathbb{R}$. On considère l'équation d'inconnue complexe $x$, $E_m : x^2 - x - m = 0$. Déterminer le nombre de solutions réelles de $E_m$ en fonction de $m$.
2. (INFO) Écrire une fonction `nb_sol(m)` qui renvoie le nombre de solutions réelles de l'équation $x^2 - x - m = 0$.
3. Résoudre dans $\mathbb{C}$ : $E_1$, $E_{-\frac{1}{3}}$, $E_{-\frac{1}{4}}$. Dans le cas de solutions complexes non réelles donner également une écriture exponentielle des solutions.
4. Soit la suite $(\alpha_n)_{n\in\mathbb{N}}$ définie par : $\alpha_0 = 2, \alpha_1 = 3, \forall n \in \mathbb{N}, \alpha_{n+2} = \alpha_{n+1} - \frac{1}{4}\alpha_n$.
   (a) (INFO) On considère les deux fonctions suivantes écrites en Python :

```
def alpha(n) :
    u0=2
    u1=3
    for i in range(n) :
        u1=u1-0.25*u0
        u1=u0
    return u0
```

```
def alphabis(n) :
    u0=2
    u1=3
    for i in range(n) :
        u=u1
        u1=u-0.25*u0
        u0=u
        return u
```

   Déterminer la valeur de `alpha(3)` et de `alphabis(3)`.

   (b) (INFO) Écrire une fonction `alphater(n)` qui prend en argument un entier $n$ et qui renvoie la valeur de $\alpha_n$.

   (c) Déterminer pour tout $n \in \mathbb{N}$, une expression de $\alpha_n$ en fonction de $n$.

   (d) Déterminer $a, b, m$ des réels tels que $(\alpha_n)_{n\in\mathbb{N}}$ est un élément de $S_m(a,b)$.
5. Soit la suite $(\beta_n)_{n\in\mathbb{N}}$ définie par : $\beta_0 = 0, \beta_1 = 4, \forall n \in \mathbb{N}, \beta_{n+2} = \beta_{n+1} - \frac{1}{3}\beta_n$.
   (a) (INFO) Écrire une fonction `beta(n)` qui renvoie la valeur de $\beta_n$. On utilisera la formule de récurrence.

   (b) Déterminer pour tout $n \in \mathbb{N}$, une expression de $\beta_n$ en fonction de $n$.

   (c) Montrer que $S_{-\frac{1}{4}}(0,4)$ contient comme unique élément la suite $(\beta_n)_{n\in\mathbb{N}}$. On pourra considérer une suite quelconque $(u_n)_{n\in\mathbb{N}}$ de $S_{-\frac{1}{4}}(0,4)$ et montrer par une récurrence double que pour tout $n \in \mathbb{N}, u_n = \beta_n$.
6. On définit la suite de Lucas $(L_n)_{n\in\mathbb{N}}$ par : $L_0 = 2, L_1 = 1, \forall n \in \mathbb{N}, L_{n+2} = L_{n+1} + L_n$.
   (a) Montrer que pour tout $n \in \mathbb{N}$, on a $L_n = \left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{1-\sqrt{5}}{2}\right)^n$.

   (b) (INFO) Déduire de la question 6a une fonction `Lucas(n)` qui renvoie la valeur de $L_n$.

   (c) Soient $a, b, m$ des réels. Montrer que si $(L_n)_{n\in\mathbb{N}}$ est un élément de $S_m(a,b)$ alors $a = 2, b = 1, m = 1$.

7. Soient $m, a, b$ des réels et $(u_n)_{n\in\mathbb{N}}$ une suite de $S_m(a,b)$. Pour tout $n \in \mathbb{N}$, on souhaite simplifier $\sum_{k=0}^{n} u_k$.
   (a) (INFO) Écrire une fonction `suite(a,b,m,n)` qui prend en arguments trois réels $a, b, m$ et un entier $n$ et qui renvoie la valeur de $u_n$ où $(u_n)_{n\in\mathbb{N}}$ est un élément de $S_m(a,b)$.
   (b) (INFO) On déduit de la question précédente la fonction `somme(a,b,m,n)` définie par

```
def somme(a,b,m,n) :
    S=0
    for i in range(n+1) :
        S=S+suite(a,b,m,n)
    return S
```

   Quelle est la valeur de retour de cette fonction ? Modifier celle-ci pour que la valeur de retour de cette fonction soit $\sum_{k=0}^{n} u_k$, où $(u_n)_{n\in\mathbb{N}}$ est un élément de $S_m(a,b)$.
   (c) Cas 1 : $m = 0$.
      i. Calculer $\sum_{k=0}^{0} u_k, \sum_{k=0}^{1} u_k, \sum_{k=0}^{2} u_k$.
      ii. Pour tout $n \geq 2$, simplifier $\sum_{k=0}^{n} u_k$.
   (d) Cas 2 : $m \neq 0$. Pour tout $n \in \mathbb{N}$, simplifier $\sum_{k=0}^{n} u_k$. On pourra remarquer que

$$\forall k \in \mathbb{N}, u_k = \frac{1}{m}\left(u_{k+2} - u_{k+1}\right).$$

   (e) Pour tout $n \in \mathbb{N}$, en déduire $\sum_{k=0}^{n} \alpha_k, \sum_{k=0}^{n} \beta_k, \sum_{k=0}^{n} L_k$.
8. Dans cette question, on cherche à simplifier diverses sommes faisant intervenir la suite $(L_n)_{n\in\mathbb{N}}$ définie à la question 6.
   (a) Pour tout $n \in \mathbb{N}$, simplifier $\sum_{k=0}^{n} L_{2k}$ et $\sum_{k=0}^{n} L_{2k+1}$.
   (b) Montrer que pour tout $n \in \mathbb{N}$, on a

$$\left\{\begin{array}{lcl} L_{2n} & = & L_n^2 - 2\cdot(-1)^n \\ L_{2n+1} & = & L_n L_{n+1} - (-1)^n \end{array}\right.$$

   (c) (INFO) Écrire en python une fonction correspondant au pseudo-code suivant :

```
liste_lucas(n) :
    Si n=0 :
        retourner [2]
    si n=1 :
        retourner [2,0]
    sinon :
        L=[2,0]
        pour  i allant de 0 à n-1 (inclus) :
            si i est pair et est égal à 2k :
                S= L[k]**2-2 (-1)**n
            si i est impair et est égal à 2k+1 :
                S=L[k]*L[k+1]-(-1)**n
            ajouter S à la fin de L
        retourner L
```

   (d) Pour tout $n \in \mathbb{N}$ simplifier $\sum_{k=0}^{n} L_k^2$ et $\sum_{k=0}^{n} L_k L_{k+1}$. On pourra s'aider des questions précédentes.
   (e) (INFO) Écrire une fonction `somme1(n)` qui renvoie la valeur de $\displaystyle\sum_{0\leq i,j\leq n} L_i L_j$. On pourra utiliser la fonction `Lucas` de la question 6b.
   (f) (INFO) Écrire une fonction `somme2(n)` qui renvoie la valeur de $\displaystyle\sum_{0\leq i\leq j\leq n} L_i L_j$. On pourra utiliser la fonction `Lucas` de la question 6b.
   (g) Pour tout $n \in \mathbb{N}$, simplifier $\displaystyle\sum_{0\leq i,j\leq n} L_i L_j$ et $\displaystyle\sum_{0\leq i\leq j\leq n} L_i L_j$. On pourra s'aider de résultats de la question 7.