DS3

3h00

- Les calculatrices sont interdites durant les cours, TD et *a fortiori* durant les DS de mathématiques.
- Si vous pensez avoir découvert une erreur, indiquez-le clairement sur la copie et justifiez les initiatives que vous êtes amené·e ·s à prendre.
- Une grande attention sera apportée à la clarté de la rédaction et à la présentations des solutions. (Inscrivez clairement en titre le numéro de l'exercice, vous pouvez aussi encadrer les réponses finales.)
- Vérifiez vos résultats.
- Le résultat d'une question peut être admis et utilisé pour traiter les questions suivantes en le signalant explicitement sur la copie.

Exercice 1. Soit $\lambda \in \mathbb{R}$. On considère le système suivant

$$(S_\lambda) \quad \left\{ \begin{array}{ccccccc} x & + & y & + & z & = & \lambda x \\ x & + & y & + & z & = & \lambda y \\ x & + & y & + & z & = & \lambda z \end{array} \right.$$

1. Échelonner le système.
2. Déterminer le rang de S_λ en fonction de λ
3. Déterminer Σ l'ensemble des réels λ pour lequel ce système n'est pas de Cramer.
4. Pour $\lambda \in \Sigma$, résoudre S_λ
5. Quelle est la solution si $\lambda \notin \Sigma$.

Exercice 2. On définit deux suites $(u_n)_{n\in\mathbb{N}}$ et $(v_n)_{n\in\mathbb{N}}$ par

$$u_0 = 0 \quad v_0 = 1 \quad \text{et} \quad \text{pour tout } n \in \mathbb{N},\ u_{n+1} = 2u_n - 4v_n \quad \text{et} \quad v_{n+1} = u_n + 4v_n.$$

1. INFO Ecrire une fonction Python qui prend en argument un entier n et retourne les valeurs de u_n et v_n.
2. Montrer que pour tout $n \in \mathbb{N}$
$$u_{n+2} = 6u_{n+1} - 12u_n$$
3. Calculer les racines de $X^2 - 6X + 12$ et les mettre sous formes exponentielles.
4. En déduire la valeur de u_n en fonction de n.

Exercice 3. Soit $\omega = e^{\frac{2i\pi}{7}}$. On considère $A = \omega + \omega^2 + \omega^4$ et $B = \omega^3 + \omega^5 + \omega^6$

1. Calculer $\frac{1}{\omega}$ en fonction de $\overline{\omega}$
2. Montrer que pour tout $k \in [\![0, 7]\!]$ on a
$$\omega^k = \overline{\omega}^{7-k}.$$
3. En déduire que $\overline{A} = B$.
4. Justifier que $\sin\left(\frac{2\pi}{7}\right) - \sin\left(\frac{\pi}{7}\right) > 0$.
5. Montrer alors que la partie imaginaire de A est strictement positive.
6. Prouver par récurrence que pour tout $q \neq 1$, et tout $n \in \mathbb{N}$: on a :
$$\sum_{k=0}^{n} q^k = \frac{1 - q^{n+1}}{1 - q}.$$
7. Montrer alors que $\displaystyle\sum_{k=0}^{6} \omega^k = 0$. En déduire que $A + B = -1$.
8. Montrer que $AB = 2$.
9. En déduire la valeur exacte de A.

Exercice 4. Soient z, z' deux nombres complexes.

1. Rappeler les valeurs de $z\overline{z}$, $|z\overline{z}|$ en fonction de $|z|$ et $|z'|$. Rappeler la formule reliant $z, \overline{z}$ et $Re(z)$.
2. Montrer que $|z - z'|^2 = |z|^2 - 2Re(zz') + |z'|^2$
3. On suppose dans cette question et la suivante que $|z| < 1$ et $|z'| < 1$. Montrer que
$$\overline{z}z' \neq 1$$

4. Montrer que

$$1-\left|\frac{z-z'}{1-\overline{z}z'}\right|^2=\frac{(1-|z'|^2)(1-|z|^2)}{|1-\overline{z}z'|^2}$$

5. Soit $(z_n)_{n\in\mathbb{N}}$ une suite de nombres complexes vérifiant : $|z_0|<1, |z_1|<1$ et pour tout $n\in\mathbb{N}$:

$$z_{n+2}=\frac{z_n-z_{n+1}}{1-\overline{z_n}z_{n+1}}$$

Montrer que pour tout $n\in\mathbb{N}$, $|z_n|<1$ et que $\overline{z_n}z_{n+1}\neq 1$, et donc que $(z_n)_{n\in\mathbb{N}}$ est bien définie pour tout $n\in\mathbb{N}$.
On pourra utiliser les deux questions précédentes dans une récurrence double

INFORMATIQUE

Exercice 5. 1. Dire ce qu'affiche la console avec les scripts suivants

```
#script 1
L=[2*k for k in range(1,5)]
print(L)
```

```
#script 2
L=[k for k in range(1,5)]
print(L+L)
```

2. Ecrire une fonction `max_list` qui prend en argument une liste de flottants et retourne la position du maximum de cette liste. Si plusieurs valeurs réalisent ce maximum, la fonction retournera le premier indice.

   Exemple `max_list([1,5,3,5,0,2]` retournera la valeur 1

3. Que fait la fonction mystère suivante :

```
def mystere(L):
    Lt=[]
    n=len(L)
    while len(Lt)<n:
        i_m=max_list(L)  #fonction de la question 2
        M=L.pop(i_m)
         Lt.append(M) #Li.append(x) ajoute l'element x a la liste Li
    return(Lt)
```

4. Compléter (en la recopiant sur votre copie) la fonction suivante qui prend en argument une liste de flottants L et un flottant a et retourne une liste contenant tous les éléments de L supérieur ou égal à a

```
def sous_liste_sup(L,a):
    Lsup=[]
    for el in L:
        if .... :
            Lsup.....
    return(Lsup)
```