DS6

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. On note $\mathcal{C}_f$ la courbe représentative de la fonction f définie par $f(x) = \dfrac{e^x}{e^{2x}+1}$.

1. (a) Donner le tableau de variations avec les limites aux bornes de la fonction f.
   (b) Donner l'équation de la tangente en 0 à la courbe représentative de f.
   (c) La fonction f est-elle injective de $\mathbb{R}$ dans $\mathbb{R}$? Est-elle surjective de $\mathbb{R}$ dans $\mathbb{R}$? On justifiera correctement les réponses.
   (d) Calculer $f(\mathbb{R})$.
2. (a) Montrer que f est bijective de $\mathbb{R}^+$ sur un ensemble F à préciser. On appelle g la restriction de f à ces ensembles, c'est-à-dire :
$$g \left| \begin{array}{ccc} \mathbb{R}^+ & \to & F \\ x & \mapsto & \dfrac{e^x}{e^{2x}+1} \end{array} \right.$$
   (b) Calculer $g^{-1}(\frac{1}{4})$

Exercice 2. On considère $S = \{z \in \mathbb{C} \mid |z| = 2\}$.

1. Rappeler la nature géométrique de S.

   Soit $f : \mathbb{C} \to \mathbb{C}$ la fonction définie par $f(z) = \frac{2z+1}{z+1}$. Déterminer D_f le domaine de définition de f. Est-elle bien définie pour tous les points de S ?
2. (a) Soit $(z_1, z_2) \in \mathbb{C}^2$, montrer que
$$|z_1 + z_2|^2 = |z_1|^2 + 2\mathfrak{Re}(z_1\overline{z_2}) + |z_2|^2$$
   (b) Montrer que pour tout z dans l'ensemble de définition de f,
$$\left|f(z) - \frac{7}{3}\right|^2 = \frac{|z|^2 + 8\mathfrak{Re}(z) + 16}{9(|z|^2 + 2\mathfrak{Re}(z) + 1)}$$
   (c) On note S_2 le cercle de centre le point d'affixe $7/3$ et de rayon $\frac{2}{3}$. Montrer que $f(S) \subset S_2$

Exercice 3. On considère les points $A(1, 2, -1)$, $B(-2, 4, 0)$ et $C = (2, 2, -2)$

1. Déterminer une représentation paramétrique de (AB).
2. Déterminer une équation cartésienne du plan P qui passe par les points A, B et C.
3. En fonction de $m \in \mathbb{R}$, déterminer l'intersection de (AB) avec le plan $\mathcal{P}_m$ représentée par l'équation cartésienne
$$mx + y + z + 1 = 0$$
4. A quelle(s) condition(s) sur m, a-t-on $C \in P_m$?
5. A quelle(s) condition(s) sur m, P_m et P sont-ils parallèles ?
6. Ecrire une fonction Python nommée `dans_le_plan`, qui prend en argument un flottant `m` et une liste de flottants de taille 3, `M`, correspondant aux coordonnées d'un point M de l'espace et qui renvoie `True` si $M \in P_m$ et `False` sinon.

Exercice 4. Pour tout $n \in \mathbb{N}^*$, on note $E_n = [\![1, n]\!]$. On note $S_{n,p}$ le nombre de surjections de E_n sur E_p.

1. Calculer $S_{n,p}$ si $p > n$.
2. Justifier grâce au cardinal qu'une surjection de E_n dans E_n est une bijection. En déduire $S_{n,n}$.
3. Déterminer $S_{n,1}$.
4. Combien y-a-t-il d'applications de E_n dans E_2 ? Parmi ces applications lesquelles ne sont pas surjectives ? En déduire $S_{n,2}$.
5. Soit f une surjection de E_{p+1} dans E_p, justifier que tous les éléments de E_p ont exactement un antécédent sauf un qui en a exactement deux. En déduire que $S_{p+1,p} = \frac{p}{2}(p+1)!$

   On suppose désormais que $0 < p \leq n$.
6. Montrer que $\displaystyle\sum_{k=0}^{p} \binom{p}{k}(-1)^k = 0$
7. Montrer que pour tout (k, q) tel que $0 \leq q \leq k \leq p$
$$\binom{p}{k}\binom{k}{q} = \binom{p}{q}\binom{p-q}{k-q}.$$
8. (a) En déduire que, si $0 \leq q < p$, alors
$$\sum_{k=q}^{p} \binom{p}{k}\binom{k}{q}(-1)^k = 0.$$
   (b) Que vaut la somme précédente quand $q = p$?
9. Montrer que pour tout entier q de E_p le nombre d'applications de E_n dans E_p ayant un ensemble d'images à q éléments est égal à $\binom{p}{q}S_{n,q}$.
10. En déduire que pour tout $(n, p) \in \mathbb{N}^2$
$$p^n = \sum_{q=1}^{p} \binom{p}{q} S_{n,q}.$$
11. Montrer que pour tout (k, q) tel que $0 \leq q \leq k \leq p$ on a :
$$\sum_{k=1}^{p}(-1)^k \binom{p}{k} k^n = \sum_{q=1}^{p} \left(\sum_{k=q}^{p}(-1)^k \binom{p}{k}\binom{k}{q} \right) S_{n,q}.$$
   On pourra sommer la relation obtenue à la question précédente en veillant à changer le nom des variables puis intervertir la somme double.
12. A l'aide des questions précédentes (8, 10, 11 notamment), en déduire que
$$S_{n,p} = (-1)^p \sum_{k=1}^{p}(-1)^k \binom{p}{k} k^n.$$

Exercice 5 (Matrices de Hadamard).

Une matrice carrée $M \in \mathcal{M}_n(\mathbb{R})$ est dite de **Hadamard** si tous ses coefficients appartiennent à $\{-1, 1\}$ et tel que

$$M^T M = nI_n$$

Par exemple, la matrice suivante :

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

est une matrice de Hadamard. Dans cet exercice, nous considèrerons des matrices sous forme de tableau `numpy`.

On considère que le module `numpy`, permettant de manipuler des tableaux à deux dimensions, est importé via `import numpy as np`.

1. Écrire une fonction `Transposition(M)` qui prend en argument une matrice carrée M et renvoie sa transposée (on n'utilisera pas la fonction `transpose` de la bibliothèque `numpy`).
2. Écrire une fonction `Egalite(M,N)` qui prend en argument deux matrices M et N et qui renvoie `True` si elles sont égales et `False` sinon. Attention, le `==` entre deux tableaux `M`, `N` numpy ne renvoie pas un booléen : Si les deux tableaux ne sont pas de la même taille, alors il renvoie une erreur, sinon il renvoie un tableau numpy de booléens dont l'entrée (i,j) est la valeur `M[i][j]== N[i][j]`
3. Écrire une fonction `Hadamard(M)` qui prend en argument une matrice carrée M et renvoie `True` si c'est une matrice de Hadamard et `False` sinon.

   On peut montrer (on ne demande pas de le faire) que si M est une matrice de Hadamard symétrique, alors la matrice $\begin{pmatrix} M & M \\ M & -M \end{pmatrix}$ est encore une matrice de Hadamard.

   Par exemple si $M = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, on obtient $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$. Cette construction est dûe à Sylvester.
4. Écrire une fonction `Symetrique(M)` qui prend en argument une matrice M et qui renvoie `True` si M est symétrique et `False` sinon.
5. Écrire une fonction `Duplication(M)` qui prend en argument une matrice M qui teste si c'est une matrice de Hadamard symétrique et qui renvoie $\begin{pmatrix} M & M \\ M & -M \end{pmatrix}$ si elle l'est et qui renvoie une matrice nulle de même taille que $\begin{pmatrix} M & M \\ M & -M \end{pmatrix}$ sinon.

Annexe : Rappel Python

Pour une matrice M à n lignes et p colonnes, les indices vont de 0 à $n-1$ pour les lignes et de 0 à $p-1$ pour les colonnes.

Python	**Interprétation**
`M[i,j]`	Coefficient d'indice (i,j) de la matrice M
`np.zeros((n,p))`	Matrice à n lignes et p colonnes remplie de zéros
`np.eye(n)`	Matrice identité de taille n
`T = np.shape(M)`	Dimensions de la matrice M
`T[0]` ou `np.shape(M)[0]`	Nombre de lignes
`T[1]` ou `np.shape(M)[1]`	Nombre de colonnes
`np.dot(M,N)`	Produit de la matrice MN