

## MATHÉMATIQUES

*Le candidat dispose d'environ trente minutes pour préparer les deux exercices qui suivent.*

*A l'issue de cette préparation, il sera convié à exposer ses résultats au tableau.*

*L'ordre d'exposition des exercices est laissé à son libre choix.*

---

**Exercice 1.** Soit  $\mathcal{C}$  la courbe d'équation  $x^2 + 2y^2 + 5x - 3y = 5$ . Déterminer l'équation de la tangente à  $\mathcal{C}$  en  $(1, 1)$ .

**Exercice 2.** On note  $A$  la matrice suivante

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

1. Justifier que  $A$  est diagonalisable.
2. Déterminer les valeurs propres de  $A$ .
3. Que renvoie la fonction trace définie dans le script ? Corriger cette fonction.
4. Tester la fonction `fonct1` sur la matrice  $A$  et un entier « grand » ( $n = 50$  par exemple). Cette fonction renvoie une approximation d'un élément propre de la matrice  $A$ , lequel ? sous quelles hypothèses ?
5. Soit  $p \in \mathbb{N}^*$ . L'appel de la fonction `numpy.random.rand(p)` renvoie aléatoirement la représentation d'un élément  $U$  de  $\mathbb{R}^p$ . On admet que la probabilité que  $U$  appartienne à un sous-espace vectoriel donné de  $\mathbb{R}^p$  distinct de  $\mathbb{R}^p$  est nulle. Soit  $\mathcal{B}$  une base de  $\mathbb{R}^n$ . On note  $X_1, \dots, X_p$  les coordonnées de  $U$  dans la base  $\mathcal{B}$ . Prouver que

$$P \left( \bigcap_{i=1}^p (X_i \neq 0) \right) = 1$$

6. Soit  $C$  une matrice diagonalisable telle que  $C$  possède une valeur propre  $\lambda$  de module maximal de multiplicité 1. Proposer une fonction écrite en Python prenant en argument une telle matrice  $C$  et un entier  $n$  puis renvoyant une approximation d'un vecteur propre de  $C$  associé à  $\lambda$ . Justifier l'algorithme utilisé.

Code Python fourni :

```
# Matrice A
A = array([[0,1,1],[1,0,1],[1,1,0]])

def trace(A) :
    """Fonction prenant en argument un tableau représentant une matrice carrée
    et renvoyant sa trace."""
    s = 0
    for i in range (len(A)) :
        s += A[i][0]
    return s

def fonct1(A,n) :
    """Fonction prenant en argument un tableau représentant une matrice et un entier"""
    B = la.matrix_power(A,n)
    return trace(B.dot(A))/trace(B)
```