

## Examen

Durée 2h. Documents fournis sur le site du cours autorisés. Autres documents et accès à internet interdits. La barème est à titre indicatif et peut encore changer. Il faut que vos programmes soient **exécutables** par la commande `python3 nomdefichier.py` et qu'ils **affichent leurs résultats**. Notamment, il ne sera pas accepté que les résultats ne soient accessibles que par le débogueur de votre environnement de développement.

### Exercice 1. Programmation élémentaire (6 points)

Une conjecture mathématique célèbre, prouvée en 2002 après plus que 150 ans, est que *les seules puissances d'entiers consécutives sont*  $8 = 2^3$  et  $9 = 3^2$ . C'est-à-dire, il n'y a pas de solution de l'équation  $x^a + 1 = y^b$  (avec  $x, y, a$  et  $b$  des entiers  $> 1$ ) autre que  $x = b = 2$  et  $a = y = 3$ . Réalisez un programme qui vérifie la conjecture pour  $1 < x < 100$ ,  $1 < y < 100$ ,  $1 < a < 10$ ,  $1 < b < 10$ . Votre programme affichera toutes les solutions qu'il trouve et rien d'autre.

### Exercice 2. Graphisme et recherche des zéros (8 points)

On définit la fonction  $J_0(x)$  (une des *fonctions de Bessel* qui sont utiles, par exemple, pour décrire les oscillations d'une membrane circulaire) par la série infinie

$$J_0(x) = \lim_{N \rightarrow \infty} \sum_{k=0}^N \frac{(-1)^k}{(k!)^2} \left(\frac{x}{2}\right)^{2k}.$$

1. Réalisez une fonction `J0(x, N=20)` qui renvoie une approximation à  $J_0(x)$  en prenant en compte les premiers  $N + 1$  termes de la somme ci-dessus.

*Indication* : Pour le calcul de la factorielle, vous pouvez vous servir de la fonction `scipy.special.factorial(k)` que vous importerez de la bibliothèque `scipy.special`.

2. Tracez le graphe de votre fonction pour  $x$  compris entre 0 et 10. Comparez avec le graphe de la fonction préfabriquée `scipy.special.j0(x)`.
3. Réalisez un programme qui trouve tous les zéros de  $J_0$  entre 0 et 10 avec une précision numérique d'au moins  $10^{-3}$ . Utilisez une méthode du cours appropriée. Pour le calcul, vous pouvez utiliser soit votre propre fonction `J0` (et éventuellement `scipy.special.factorial`) soit la fonction `scipy.special.j0`, mais pas d'autres éléments de la bibliothèque `scipy.special`.

*Indication* : Du graphe de 2. vous pouvez obtenir des premières approximations aux zéros.

### Exercice 3. Calcul matriciel et algèbre linéaire (6 points)

On cherche la solution  $\vec{x}$  du système d'équations linéaires  $A\vec{x} = \vec{b}$ , où  $A$  est une matrice  $n \times n$  et  $\vec{b}$  un vecteur à  $n$  composantes. Soit la décomposition QR de  $A$  donnée par les matrices  $Q$  et  $R$ .

1. Par un commentaire dans votre fichier de copie, montrez que le vecteur des inconnues  $\vec{x}$  vérifie le système d'équations linéaires  $R\vec{x} = Q^T\vec{b}$ , où  $Q^T$  est la transposée de  $Q$ .
2. Rendez-vous compte que cette propriété permet de calculer  $\vec{x}$  avec un algorithme différent de celui de Gauss. Réalisez un programme qui implémente cet algorithme pour trouver  $\vec{x}$  dans

$$\begin{pmatrix} 1.5 & 4 & 2 \\ 3 & -2 & 0 \\ 1 & 1 & 5 \end{pmatrix} \vec{x} = \begin{pmatrix} 7 \\ 5 \\ -0.5 \end{pmatrix}.$$

*Indication* : Il convient de se servir de la fonction du cours `decomp_QR` pour le calcul de la décomposition QR, et/ou du code du cours pour la solution d'un système linéaire dont la matrice des coefficients est triangulaire supérieure.