

Évaluation TP3 bis

Durée : 1h15

Les calculatrices et téléphones portables sont interdits.
Seule la feuille avec les commandes python et vos codes de TP sont autorisés.

Exercice 1

Soit $f : [-3, 3] \rightarrow \mathbb{R}$. On se donne $n+1$ points d'interpolations notés $x_k = -3 + \frac{6k}{n}$ pour $k = 0, \dots, n$ et on souhaite faire une interpolation polynomiale, c'est à dire approcher f par une fonction polynomiale $p(x) = \sum_{j=0}^n p_j x^j$. Si $n = N$, on a vu en cours qu'il existait un unique polynôme P tel que $f(x_k) = P(x_k)$. On s'intéresse au cas où on dispose de beaucoup plus de données $n \geq N$. Dans ce cas, on a plus d'équations que de degrés de libertés et on va résoudre le problème des moindres carrés.

Le problème s'écrit : trouver p_0^*, \dots, p_n^* réalisant le minimum de $\|Ap - F\|_2$ où $p = (p_0, \dots, p_n)$ est le vecteur des coefficients du polynôme P , $F = (f(x_0), \dots, f(x_n))$ et $A \in M_{n,N}$ est la matrice de coefficients $A_{i,j} = x_i^j$. La solution du problème au moindre carrés s'écrit $A^t Ap = A^t F$ où A^t est la transposée de A .

Dans ce qui suit, on choisit $f(x) = \frac{1}{1+x^2}$. On choisit $N = 5$ et n pourra prendre les valeurs 5, 10, 15, 20.

1. Ecrire un programme résolvant ce problème pour tout n, N avec $n \geq N$.
2. Reprendre vos codes du TP interpolation pour représenter les polynômes d'interpolation de la fonction f pour $n = N = 5, 10, 15$.
3. Comparer avec le polynôme d'interpolation au sens des moindres carrés de degré $N = 5$ pour $n = 10, 15, 20$ données.

Exercice 2

Soit $A \in M_n(\mathbb{R})$ la matrice définie par

$$A = \begin{pmatrix} c_1 & c_2 & c_3 & c_4 & \dots & c_n \\ c_n & c_1 & c_2 & c_3 & \ddots & \vdots \\ c_{n-1} & \ddots & \ddots & \ddots & \ddots & 0 \\ c_{n-2} & \ddots & \ddots & \ddots & \ddots & c_3 \\ \vdots & \ddots & c_{n-1} & c_n & c_1 & c_2 \\ c_2 & \dots & c_{n-2} & c_{n-1} & c_n & c_1 \end{pmatrix}$$

avec $c_k = \frac{1}{2^{k-1}}$

1. On choisit $n = 40$: utiliser les boites noires de numpy pour calculer la plus grande et la plus petite valeur propre de A en module (celles ci serviront de valeur de référence)
2. Programmer la méthode de la puissance et enregistrer les valeurs approchées de la plus grande valeur propre.
3. Illustrer la vitesse de convergence de la méthode.

4. Programmer la méthode de la puissance inverse et enregistrer les valeurs approchées de la plus petite valeur propre de A .
5. Illustrer la vitesse de convergence de la méthode de la puissance inverse.