TP n° 1 — Interpolation polynomiale et base canonique

Introduction

Le but de ce TP est d'explorer une méthode directe afin d'interpoler une fonction par un polynôme.

Soit :

- $a = x_0 < x_1 < \ldots < x_n = b$ une subdivision d'un intervalle $[a\,;\,b]$;
- f une fonction dont on connaît les valeurs $y_i = f(x_i)$, pour i allant de 0 à n.

Objectif

On cherche un polynôme P de degré au maximum égal à n et tel que :

$$y_i = P(x_i), \text{ pour } 0 \text{ allant de } 1 \text{ à } n$$

On travaille dans la base canonique $(1, x, \ldots, x^n)$ et on cherche P sous la forme :

$$P(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

Questions

1. On note :

- V la matrice de Vandermonde associée à $(x_0, \ldots, x_n)$;
- A la matrice colonne associée aux inconnues $a_0, \ldots, a_n$;
- Y la matrice colonne associée aux y_i, pour i allant de 0 à n.

Après avoir traduit l'égalité $y_i = f(x_i)$, pour i allant de 0 à n, écrire le système matriciel :

$$(S) : VA = Y$$

2. Résoudre ce système et récupérer P en l'évaluant pour x appartenant à $[a\,;\,b]$.

3. Tracer sur un même graphique une fonction f ainsi que son polynôme associé.

Attendus

Chaque binôme devra rendre :

- Un compte rendu succint ;
- Un code (Python) qui fonctionne.

Indications et contenu souhaitable :

1. Besoin d'imports : math, matplotlib.pyplot, numpy.

2. Entrées : a, b, n et f (par exemple $f = sin$ sur $[0\,; 2\pi]$, $n = 10$, $20\ldots$).

3. Utiliser la fonction Vandermonde pour remplir la matrice V.

4. Résolution du système (S) et récupération de A avec « np.linalg.solve ».

5. Créer une fonction qui évalue le polynôme P en x réel donné. *Possibilité* d'optimisation du code avec l'algorithme de Horner.

6. Passer à l'afficahge en choisissant le nombre de points (au-dessus de 500) pour tracer f et P.

7. Calculer l'écart maximum entre f et P en ces points et l'afficher.