1 Interpolation et approximation polynômiale

1.1 Introduction

Soit $f : [a, b] \to \mathbb{R}$ une fonction continue. On se donne $n+1$ couples (x_i, y_i), pour $i = 0, \ldots, n$, tels que $y_i = f(x_i)$, $x_0 = a$ et $x_n = b$. Le but de l'interpolation polynômiale consiste ouver un polynôme $P_n \in \mathbb{R}_n[X]$ qui coincide avec f aux noeuds $(x_i)_{0 \leq i \leq n}$, c'est â dire $\forall i \in \{0, \ldots, n\}, P_n(x_i) = f(x_i)$.

1.2 Interpolation de Lagrange

L'un des outils de base de l'interpolation polynômiale est la méthode d'interpolation de Lagrange. Le polynômiale d'interpolation de Lagrange est défini par :

$$P_n(x) = \sum_{j=0}^{n} f(x_j) L_j(x) = \sum_{j=0}^{n} y_j L_j(x)$$

tel que

$$\forall j \in \{0, \ldots, n\}, \qquad L_j(x) = \prod_{i=0, i \neq j}^{n} \frac{x - x_i}{x_j - x_i}$$

1.2.1 Application 1 :

Écrire une fonction *Lagrange* qui a pour paramètres d'entrées :

✓ $x = \begin{bmatrix} x_0, & \ldots, & x_n \end{bmatrix}$: tel que x_i sont les points d'évaluation.

✓ $y = \begin{bmatrix} y_0, & \ldots, & y_n \end{bmatrix}$: tel que $y = f(x)$.

✓ n le degré du polynôme d'interpolation.

✓ $c \in [a, b]$ le point d'interpolation.

et qui retourne en sortie la valeur du polynôme d'interpolation de Lagrange $P_n(c)$.

1.2.2 Application 2 :

À partir des données suivantes :

$$x = \begin{bmatrix} 0 & 1 & 2 \end{bmatrix}$$

et

$$y = \begin{bmatrix} 1 & 2 & 5 \end{bmatrix}$$

- ✓ Déterminer de façon analytique, selon la méthode de Lagrange, le polynôme d'interpolation $P_2(x)$ de degré deux.
- ✓ Proposer un code Matlab, permettant de construire et de tracer les éléments de base de Lagrange $L_0(x)$, $L_1(x)$ et $L_2(x)$.
- ✓ Tracer, sur la même figure, le polynôme $P_2(x)$ et les points d'interpolation.
- ✓ Moyennant la fonction *Lagrange* définie précédemment, calculer le polynôme d'interpolation.
- ✓ Tracer, sur la même figure, le polynôme obtenu et les points d'interpolation.
- ✓ Interpréter.

1.2.3 Application 3 :

On considére la fonction $f(x) = \sin(x)$ définie sur l'intervalle $[0, 3\pi]$.

- ✓ Pour des points équidistants, calculer le polynôme d'interpolation de Lagrange $P_n(x)$ de la fonction $f(x)$ moyennant la fonction *Lagrange* définie précédemment, pour $n = 2, 4, 8, 12, 18$.
- ✓ Tracer la graphe de fonction $f(x)$ ainsi que de son polynôme d'interpolation $P_n(x)$.
- ✓ Évaluer l'erreur d'interpolation $E_n(x) = \max_{x\in[0,3\pi]} |f(x) - P_n(x)|$.
- ✓ Visualiser le graphe de $E_n(x)$ en fonction de n.
- ✓ Interpréter les résultats obtenus.

1.2.4 Application 4 :

Afin d'étudier le Phénomène de Runge, on considére la fonction $g(x) = \frac{1}{1+x^2}$ définie sur l'intervalle $[-5, 5]$.

- ✓ Pour des points équidistants, calculer le polynôme d'interpolation de Lagrange $P_n(x)$ de la fonction $g(x)$ moyennant la fonction *Lagrange* définie précédemment, pour $n = 2, 4, 7, 11, 14$.
- ✓ Tracer la graphe de fonction $g(x)$ ainsi que de son polynôme d'interpolation $P_n(x)$.

✓ Évaluer l'erreur d'interpolation $E_n(x) = \max_{x\in[-5,5]} |g(x) - P_n(x)|$.

✓ Visualiser le graphe de $E_n(x)$ en fonction de n.

✓ Refaire les questions précédentes pour les points de Tchebychev définies par :

$$\forall i \in \{0, \ldots, n\} \qquad x_i = \left(\frac{b+a}{2}\right) + \left(\frac{b-a}{2}\right) \cos\left(\frac{(2(n-i)+1)\pi}{2n}\right)$$

✓ Interpréter les résultats obtenus.

1.3 Interpolation de Newton

On s'intéresse à appliquer une méthode plus pratique pour le calcul du polynôme d'interpolation, c'est la méthode de Newton définie par :

$$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \ldots + f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

où :

- Le système suivant

  $$\left\{ \quad 1, \quad (x - x_0), \quad (x - x_0)(x - x_1), \quad (x - x_0)(x - x_1)\ldots(x - x_{n-1}) \quad \right\}$$

  est appelée base de Newton.

- $f[x_0, x_1, \ldots, x_n]$ est le coefficient de x^n du polynôme d'interpolation, qui peut être calculé â partir du table des différences divisées, suivante :

x_0	$f(x_0)$				
x_1	$f(x_1)$	$f[x_0,x_1]$			
x_2	$f(x_2)$	$f[x_1,x_2]$	$f[x_0,x_1,x_2]$		
.					
.					
.					
x_n	$f(x_n)$	$f[x_{n-1},x_n]$	$f[x_{n-2},x_{n-1},x_n]$	………	$f[x_0,\ldots,x_n]$

1.3.1 Application 1 :

- Écrire une fonction $Coefficient$ qui a pour paramètres d'entrées :

  ✓ $x = \left[\begin{array}{ccc} x_0, & \ldots, & x_n \end{array}\right]$: tel que x_i sont les points d'évaluation.

  ✓ $y = \left[\begin{array}{ccc} y_0, & \ldots, & y_n \end{array}\right]$: tel que $y = f(x)$.

  ✓ n le degré du polynôme d'interpolation.

  et qui retourne en sortie les coefficients de polynôme $P_n(c)$.

- Écrire une fonction $Newton$ qui a pour paramètres d'entrées :

  ✓ $x = \left[\begin{array}{ccc} x_0, & \ldots, & x_n \end{array}\right]$: tel que x_i sont les points d'évaluation.

  ✓ n le degré du polynôme d'interpolation.

  ✓ $c \in [a, b]$ le point d'interpolation.

et qui retourne en sortie la valeur du polynôme d'interpolation de Newton $P_n(c)$.

1.3.2 Application 2 :

À partir des données suivantes :

$$x = \left[\begin{array}{cccc} 4 & 6 & 8 & 10 \end{array}\right]$$

et

$$y = \left[\begin{array}{cccc} 1.59 & 1.817 & 2 & 2.1544 \end{array}\right]$$

✓ Déterminer de façon analytique, selon la table des différences divisées, le polynôme d'interpolation $P_3(x)$ de degré trois.

✓ Proposer un code Matlab, permettant de construire et de tracer les éléments de base de Newton.

✓ Tracer, sur la même figure, le polynôme $P_3(x)$ et les points d'interpolation.

✓ Moyennant les fonctions $Coefficient$ et $Newton$ définies précédemment, calculer le polynôme d'interpolation.

✓ Tracer, sur la même figure, le polynôme obtenu et les points d'interpolation.

✓ Interpréter.

1.3.3 Application 3 :

On considére la fonction $f(x) = \sin(x)$ définie sur l'intervalle $[0, 3\pi]$.

- ✓ Pour des points équidistants, calculer le polynôme d'interpolation de Newton $P_n(x)$ de la fonction $f(x)$ moyennant les fonctions $Coefficient$ et $Newton$ définies précédemment, pour $n = 2, 4, 8, 12, 18$.
- ✓ Tracer la graphe de fonction $f(x)$ ainsi que de son polynôme d'interpolation $P_n(x)$.
- ✓ Évaluer l'erreur d'interpolation $E_n(x) = \max_{x\in[0,3\pi]} |f(x) - P_n(x)|$.
- ✓ Visualiser le graphe de $E_n(x)$ en fonction de n.
- ✓ En utilisant les fonctions : *tic* et *toc*, déterminer le temps d'exécution.
- ✓ Interpréter les résultats obtenus.
- ✓ Refaire les questions de la section 1.2.4.

1.4 Interpolation d'Hermite

En plus des valeurs de la fonction f aux $(n+1)$ points distincts x_i, on fixe les valeurs de sa derivée f' en ces mêmes points, on cherche donc un polynôme $P_N(x)$ avec $N = 2n+1$, tel que :

$$\forall i \in \{0, \ldots, n\}, \begin{cases} P_N(x_i) = f(x_i) \\ P_N'(x_i) = f'(x_i) \end{cases}$$

où $P_N(x)$ est défini par :

$$P_N(x) = \sum_{i=0}^{n} f(x_i) H_i(x) + \sum_{i=0}^{n} f'(x_i) K_i(x)$$

avec

$$K_i(x) = (x - x_i) L_i^2(x)$$

$$H_i(x) = [1 - 2(x - x_i) c_i] L_i^2(x)$$

et

$$c_i = \sum_{j=0, i\neq j}^{n} \frac{1}{x_i - x_j}$$

1.4.1 Application 1 :

À partir des données suivantes :

$$x = \begin{bmatrix} 0 & 1 & 2 \end{bmatrix}$$

,

$$y = \begin{bmatrix} 1 & 2 & 5 \end{bmatrix}$$

et

$$y^{'} = \begin{bmatrix} 1 & 1 & 3 \end{bmatrix}$$

- ✓ Écrire un algorithme Matlab permettant l'implémentation de la méthode de Hermite. Déterminer ce polynôme.
- ✓ Tracer, sur la même figure, le polynôme obtenu et les points d'interpolation.