1 Introduction à l'interpolation

1.1 Définition

Definition 1.1. Soit un nuage de points (exemple: un ensemble discret de points du graphe d'une fonction). Interpréter ce nuage de points correspond à chercher un polynôme de degré $N-1$ qui passe par chacun de ces points.

- Comment le construire ?
- $P_{N-1} \in \mathbb{P}_{N-1}[x]$
- $P_{N-1}(x_i) = y_i$

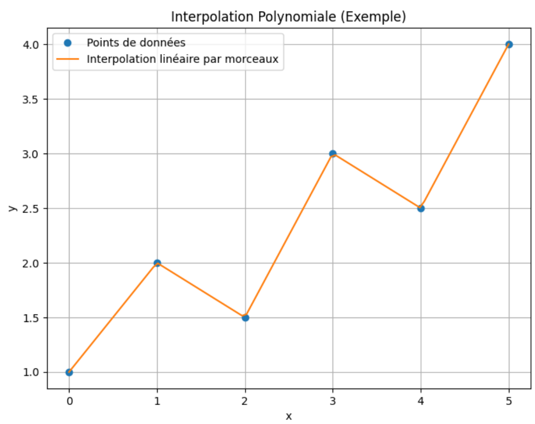

Figure 1: Illustration de l'interpolation polynomiale.

1.2 Motivations

- La solution d'un problème est fournie par une formule représentative : noyau de la chaleur (ex: convolution) et on cherche la solution en un nombre de points. $\implies$ on approche alors la fonction par un polynôme i.e. chercher le polynôme de degré "bas" proche de la fonction.
- La solution d'un problème n'est connue qu'à travers ses valeurs en un nombre fini de points et on souhaite l'évaluer partout. $\implies$ Interpolation.
- On peut utiliser l'interpolation dans :
  - la résolution numérique
  - la résolution numérique des Équations Différentielles Ordinaires (EDO)
  - la visualisation scientifique

Definition 1.2. Un tel polynôme est appelé **polynôme interpolateur de Lagrange** de degré $N-1$ de ces points.

1.3 Exemples d'interpolation

1.3.1 Théorème: Polynôme interpolateur de degré 1

Theorem 1.3. Soient (x_1, y_1) et (x_2, y_2) deux points distincts de $\mathbb{R}^2$. Il existe une unique droite D passant par ces deux points.

$$(x, y) \in D \iff (x - x_1)(y_2 - y_1) - (y - y_1)(x_2 - x_1) = 0$$

Si de plus, $x_1 \neq x_2$, il existe un unique polynôme de degré 1 (i.e., $P \in \mathbb{P}_1[x]$) tel que $y = P(x)$.
avec

$$P_1(x) = \frac{(x - x_2)}{(x_1 - x_2)} y_1 + \frac{(x - x_1)}{(x_2 - x_1)} y_2$$

pour des abscisses x_1, x_2 distinctes.

Example 1.4. Montrons que $M(x, y)$ est sur la droite (M_1M_2) si et seulement si les vecteurs $\overrightarrow{M_1M}$ et $\overrightarrow{M_1M_2}$ sont colinéaires. Soient $M_1(x_1, y_1)$, $M_2(x_2, y_2)$ et $M(x, y)$.

$$M \in (M_1M_2) \Leftrightarrow \overrightarrow{M_1M} // \overrightarrow{M_1M_2}$$

$$\Leftrightarrow \det(\overrightarrow{M_1M}, \overrightarrow{M_1M_2}) = 0$$

$$\Leftrightarrow \begin{vmatrix} x - x_1 & x_2 - x_1 \\ y - y_1 & y_2 - y_1 \end{vmatrix} = 0$$

$$\Leftrightarrow (x - x_1)(y_2 - y_1) - (x_2 - x_1)(y - y_1) = 0$$

Si $x_1 \neq x_2$, alors on peut réécrire l'équation de la droite sous la forme $y = ax + b$:

$$\Longrightarrow y - y_1 = \frac{(y_2 - y_1)}{(x_2 - x_1)}(x - x_1)$$

$$\Leftrightarrow y = P_1(x)$$

Remark 1.5. On a l'écriture équivalente de P_1 :

$$P_1(x) = \frac{x - x_2}{x_1 - x_2} y_1 + \frac{x - x_1}{x_2 - x_1} y_2 = \frac{y_1 - y_2}{x_1 - x_2} x + \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1} = a_1 x + a_0$$

c'est l'écriture dans la base $(1, x)$ de $\mathbb{P}_1[x]$ (base canonique).

- $P_1(x) = \frac{x - x_2}{x_1 - x_2} y_1 + \frac{x - x_1}{x_2 - x_1} y_2$

$$= \underbrace{\frac{x - x_2}{x_1 - x_2}}_{\ell_1(x)} y_1 + \underbrace{\frac{x - x_1}{x_2 - x_1}}_{\ell_2(x)} y_2$$

C'est l'écriture dans la base (ℓ_1, ℓ_2) de $\mathbb{P}_1[x]$ (base de Lagrange).

Remark 1.6. $\ell_1(x_1) = 1, \ell_1(x_2) = 0$
$\ell_2(x_1) = 0, \ell_2(x_2) = 1$

- $P_1(x) = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$ C'est l'écriture dans la base $(1, x - x_1)$ de $\mathbb{P}_1[x]$ (base de Newton).

1.3.2 Exemple: méthode de calcul employée

Chercher le polynôme interpolateur de Lagrange aux points $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

Méthode 1. $(x_1 \neq x_2, x_2 \neq x_3, x_1 \neq x_3)$

P_2 sera un polynôme de degré 2 :

$$P_2(x) = a_0 + a_1 x + a_2 x^2$$

Comme $P_2(x_i) = y_i$, pour $i = 1, 2, 3$, on a le système d'équations linéaires:

$$\begin{cases} P_2(x_1) = y_1 \\ P_2(x_2) = y_2 \\ P_2(x_3) = y_3 \end{cases} \implies \begin{cases} a_0 + a_1 x_1 + a_2 x_1^2 = y_1 \\ a_0 + a_1 x_2 + a_2 x_2^2 = y_2 \\ a_0 + a_1 x_3 + a_2 x_3^2 = y_3 \end{cases}$$

Matriciellement :

$$\begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \implies \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{pmatrix}^{-1}}_{H^{-1}} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

Matrice de Vandermonde mal-conditionnée mais facile à construire.

Remark 1.7. Pour 2 points :

$$H = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \end{pmatrix} \implies H^{-1} = \frac{1}{x_2 - x_1} \begin{pmatrix} x_2 & -x_1 \\ -1 & 1 \end{pmatrix}$$

si $x_1 \neq x_2$.

Méthode 2. Base de Newton

$$P_2(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2)$$

$$\begin{cases} P_2(x_1) = y_1 \implies a_0 = y_1 \\ P_2(x_2) = y_2 \implies a_0 + a_1(x_2 - x_1) = y_2 \\ P_2(x_3) = y_3 \implies a_0 + a_1(x_3 - x_1) + a_2(x_3 - x_1)(x_3 - x_2) = y_3 \end{cases}$$

$$\implies \begin{cases} a_0 = y_1 \\ a_1 = \frac{y_2 - y_1}{x_2 - x_1} \\ a_2 = \frac{y_3 - y_1 - \frac{y_2 - y_1}{x_2 - x_1}(x_3 - x_1)}{(x_3 - x_1)(x_3 - x_2)} = \frac{\frac{y_3 - y_1}{x_3 - x_1} - \frac{y_2 - y_1}{x_2 - x_1}}{(x_3 - x_2)} = \frac{\frac{y_3 - y_1}{x_3 - x_1} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_2} \end{cases}$$

Cette construction est différentielle et facile à mettre à jour quand on rajoute un point supplémentaire (on rajoute uniquement une ligne).

On a donc :

$$a_0 = y_1, \quad a_1 = \frac{y_2 - y_1}{x_2 - x_1}, \quad a_2 = \frac{\frac{y_3 - y_1}{x_3 - x_1} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_2}$$

Le polynôme P_2 s'écrit donc :

$$P_2(x) = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + \frac{\frac{y_3 - y_1}{x_3 - x_1} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_2}(x - x_1)(x - x_2)$$

	a_0	a_1	a_2
x_1	y_1		
x_2	y_1	$\frac{y_2 - y_1}{x_2 - x_1}$	
x_3	y_1	$\frac{y_2 - y_1}{x_2 - x_1}$	$\frac{\frac{y_3 - y_1}{x_3 - x_1} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_2}$

Table 1: Tableau des coefficients pour la base de Newton.

Construction facile et différentielle par différences divisées : ajout d'un terme.

Méthode 3. Base de Lagrange

$$P_2(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}y_1 + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}y_2 + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}y_3$$

$$\begin{aligned} P_2(x) &= \sum_{i=1}^{3} \left(\prod_{\substack{j=1 \\ j\neq i}}^{3} \frac{x-x_j}{x_i-x_j} \right) y_i \\ &= \ell_1(x)y_1 + \ell_2(x)y_2 + \ell_3(x)y_3 \end{aligned}$$

Remark 1.8. Pour deux points (x_1, y_1) et (x_2, y_2), le polynôme interpolateur de Lagrange de degré 1 est :

$$P_1(x) = \frac{x-x_2}{x_1-x_2}y_1 + \frac{x-x_1}{x_2-x_1}y_2$$

2 Polynôme interpolateur de Lagrange

2.1 Définitions et propriétés

2.1.1 Théorème: Existence et unicité

Theorem 2.1 (Existence et unicité). Soient $x_1, \ldots, x_n$ des réels deux à deux distincts et $y_1, \ldots, y_n$ des réels quelconques. Il existe un unique polynôme $P \in \mathbb{P}_{n-1}[x]$ (i.e. de degré au plus $n-1$) tel que $P(x_i) = y_i, \forall i = 1, \ldots, n$.

On dit que P est le **polynôme interpolateur de Lagrange** aux points $(x_1, y_1), \ldots, (x_n, y_n)$.

Preuve: Soit l'application linéaire $\Phi : \mathbb{P}_{n-1}[x] \to \mathbb{R}^n$ définie par

$$P \mapsto \begin{pmatrix} P(x_1) \\ \vdots \\ P(x_n) \end{pmatrix}$$

Montrons que Φ est injective. Si $\Phi(P) = 0$, alors $P(x_i) = 0$ pour tout $i = 1, \ldots, n$. Donc P a n racines distinctes $x_1, \ldots, x_n$. Comme P est un polynôme de degré au plus $n-1$ avec n racines, il s'ensuit que $P \equiv 0$. Donc Φ est injective.

Comme $\mathbb{P}_{n-1}[x]$ et $\mathbb{R}^n$ sont deux espaces vectoriels de même dimension n, une application linéaire injective est aussi bijective, donc un isomorphisme d'espaces vectoriels. La bijectivité de Φ assure l'existence et l'unicité du polynôme interpolateur.

Definition 2.2. Si f est une fonction continue sur $[a, b] \to \mathbb{R}$, et $x_1, \ldots, x_n \in [a, b]$ sont n points deux à deux distincts, alors l'unique polynôme $P \in \mathbb{P}_{n-1}[x]$ tel que $P(x_i) = f(x_i)$, pour $i = 1, \ldots, n$ est appelé **polynôme d'interpolation de Lagrange** de f aux points $x_1, \ldots, x_n$.

2.2 Estimation de l'erreur d'interpolation

2.2.1 Théorème: Erreur d'interpolation

Theorem 2.3 (Erreur d'interpolation). Soient $a < b$, $f : [a, b] \to \mathbb{R}$ une fonction continue, et $x_1, \ldots, x_n$ n points deux à deux distincts dans $[a, b]$. Soit P_n le polynôme d'interpolation de Lagrange de f aux points x_i. Si f est de classe $\mathcal{C}^n$ sur $[a, b]$, alors pour tout $x \in [a, b]$, il existe $\xi \in [a, b]$ tel que :

$$f(x) - P_n(x) = \frac{f^{(n)}(\xi)}{n!} \underbrace{\omega_n(x)}_{=\prod_{i=1}^{n}(x-x_i)}$$

où $\omega_n(x) = (x - x_1)\cdots(x - x_n)$.

Corollaire: Si $|f^{(n)}(x)|$ est bornée par M sur $[a,b]$ pour tout $x \in [a,b]$, alors $\forall x \in [a,b]$,

$$|f(x) - P_n(x)| \leq \frac{M}{n!}|\omega_n(x)| \leq \frac{M}{n!}(b-a)^n$$

Preuve: (à faire)

2.3 Implémentation avec Python

```
from scipy.interpolate import lagrange

x = np.array([1, 2, 3]) #  remplacer par les valeurs de x_1, x_2, x_3
y = np.array([2, 3, 1]) #  remplacer par les valeurs de y_1, y_2, y_3
p = lagrange(x,y)
print(p) # affiche le polyn me
print(p(2.5)) #  value  le polyn me en x=2.5
```

3 Construction des polynômes d'interpolation de Lagrange

3.1 Interpolation dans la base canonique (Vandermonde)

3.1.1 Construction

Soit $P(x) = \sum_{i=0}^{n-1} a_i x^i \in \mathbb{P}_{n-1}[x]$. On cherche les coefficients $a_0, \ldots, a_{n-1}$ tels que $P(x_k) = y_k$ pour $k = 1, \ldots, n$.

$$\sum_{i=0}^{n-1} a_i x_k^i = y_k, \quad k = 1, \ldots, n$$

Ce qui conduit au système linéaire matriciel suivant :

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$V(x_1, \ldots, x_n)\mathbf{a} = \mathbf{y}$$

où $V(x_1, \ldots, x_n)$ est la matrice de Vandermonde.

C'est une matrice pleine, souvent mal conditionnée, mais facile à construire.

```
def VDM_Mat(x):
    n = len(x)
    V = np.zeros((n, n))
    for i in range(n):
        for j in range(n):
            V[i, j] = x[i]**j
    return V

def VDM_Poly(x,y):
    M = VDM_Mat(x)
    a = np.linalg.solve(M, y)
    return a
```

3.2 Evaluation efficace : Algorithme de Horner

3.2.1 Proposition: Algorithme de Horner

Proposition 3.1. Soit $P(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n$ un polynôme. On définit la suite $(q_k)_{k=0}^n$ par :

$$\begin{cases} q_0 = a_0 \\ q_k = q_{k-1}x + a_k, \quad k = 1, \ldots, n \end{cases}$$

Alors $q_n = P(x)$.

Exemple: $P(x) = x^2 + 2x + 1 = (x+1)^2$

Pour évaluer $P(2)$: $q_0 = a_0 = 1$ $q_1 = q_0 \times 2 + a_1 = 1 \times 2 + 2 = 4$ $q_2 = q_1 \times 2 + a_2 = 4 \times 2 + 1 = 9 = P(2) = 2^2 + 2 \times 2 + 1 = 9$

```
def Horner(P, xx):
    y = 0
    for a in P:
        y = y*xx + a
    return y

def IntVal_VDM (x, y, xx):
    a = VDM_Poly(x,y)
    YY = Horner(a[::-1], xx) # reverse a pour correspondre   l'ordre des coefficients dan
    return YY
```

3.3 Interpolation dans la base duale: Formule de Lagrange et points barycentriques

3.3.1 Construction

L'idée est de prendre pour base de $\mathbb{P}_{n-1}[x]$ l'image réciproque de la base canonique de $\mathbb{R}^n$ par l'application Φ définie dans le théorème d'existence et unicité. On cherche donc une base $\{\mathcal{L}_j\}_{j=1}^n$ de $\mathbb{P}_{n-1}[x]$ telle que

$$\mathcal{L}_j(x_i) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$$

On construit les polynômes de Lagrange $\mathcal{L}_j(x)$ comme suit :

$$\mathcal{L}_j(x) = \prod_{\substack{i=1 \\ i\neq j}}^{n} \frac{x - x_i}{x_j - x_i} = \frac{\prod_{i\neq j}(x - x_i)}{\prod_{i\neq j}(x_j - x_i)}$$

Le polynôme interpolateur de Lagrange s'écrit alors :

$$P(x) = \sum_{j=1}^{n} y_j \mathcal{L}_j(x)$$