République Tunisienne
Ministère de l'Enseignement
Supérieur et de la Recherche Scientifique

Université de Gabès
Ecole Nationale d'Ingénieurs de Gabès
Département Génie Electrique-Automatique

Cours Analyse Numérique

Réalisé par : DEHRI Khadija

Maitre Assistante en Génie Electrique-Automatique

Membre de Laboratoire de recherche
Commande Numérique des Procédés Industriels (LAB CONPRI)

République Tunisienne
Ministère de l'Enseignement
Supérieur et de la Recherche Scientifique

ENIG

Université de Gabès
Ecole Nationale d'Ingénieurs de Gabès
Département Génie Electrique-Automatique

Chapitre 5
Interpolation polynomiale

Réalisé par : DEHRI Khadija

Maitre Assistante en Génie Electrique-Automatique

Membre de Laboratoire de recherche
Commande Numérique des Procédés Industriels (LAB CONPRI)

Interpolation polynomiale

Introduction

Objectif : Étant donné un ensemble de couples $(x_i, f(x_i) = y_i, i = 1 \ldots n)$ (résultats expérimentaux, par exemple), le problème consiste à trouver un modèle mathématique (polynomial, trigonométrique, exponentiel, etc.) qui permet de reconstruire la fonction $f(x)$ "au mieux".

Autrement dit on cherche une fonction simple et facile à manipuler qui interpole les points $(x_i, f(x_i) = y_i)$.

Ensuite, on peut trouver la valeur de y à n'importe quelle autre valeur de x.

Ceci s'appelle l'interpolation.

Parmi les méthodes de l'interpolation d'une fonction inconnue explicitement, on cite les interpolations polynomiales.

Idée : Trouver un polynôme $P(x)$ de degré n passant par les $(n+1)$ points.

- Interpolation polynomiale de Lagrange
- Interpolation polynomiale de Newton
- Interpolation polynomiale de Hermite

Interpolation polynomiale

Principe

Considérons *n+1* couples $(x_0, y_0), (x_1, y_1), \ldots\ldots, (x_{n-1}, y_{n-1}), (x_n, y_n)$

Le problème est de trouvé un polynôme de degré n $P_n(x) = \sum_{k=0}^{n} a_k x^k$ tel que $P_n(x_i) = y_i \;\; \forall i \in [0 \; n]$

Théorème d'existence et d'unicité

Étant donné n+1 points distincts $(x_i, \; i = 1 \ldots n)$ et n+1 valeurs correspondantes $(y_i, \; i = 0 \ldots n)$ alors il existe un unique poynome $P_n(x)$ de degré n tel que $P_n(x_i) = y_i, i = 0 \ldots n.$

Démonstration

Calcul: résolution du système linéaire suivant:

$$\begin{matrix} P_n(x_0) = y_0 \\ P_n(x_1) = y_1 \\ \ldots \\ P_n(x_n) = y_n \end{matrix} \Leftrightarrow \begin{bmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \ldots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \ldots \\ y_n \end{bmatrix}$$

n+1 équations à n+1 inconnus

$$\Leftrightarrow \quad Va = y \qquad V : \text{matrice de Vandermonde}$$

$$\Delta = \det(V) = \prod_{0 \le j < i \le n} x_j - x_i \neq 0$$

Cela implique que le système linéaire admet une solution unique

La méthode consistant à inverser la matrice de Vandermonde ou de résoudre le système linéaire est coûteuse **Complexité du calcul : n^3**

Pour éviter ces problèmes, plusieurs méthodes explicites (sans résolution de systèmes linéaires) existent qui évitent de calculer les coefficients du polynôme : **Lagrange, Newton, …** **Complexité du calcul : n^2**

Interpolation polynomiale de Lagrange

Principe

Étant donné n+1 points distincts $(x_i,\ i=1\ldots n)$ et n+1 valeurs correspondantes $(y_i,\ i=0\ldots n)$ alors il existe un unique poynome $P_n(x)$ de degré n tel que $P_n(x_i)=y_i, i=0\ldots n$.

$$P_n\left(x\right)=L_0\left(x\right)y_0+L_1\left(x\right)y_1+L_2\left(x\right)y_2+\ldots\ldots\ldots+L_n\left(x\right)y_n$$

$$P_n(x)=\sum_{i=0}^{n} y_i L_i(x),\quad \text{avec}\quad L_i(x)=\prod_{\substack{k=0\\k\neq i}}^{n}\frac{x-x_k}{x_i-x_k}$$

$L_i(x)$ sont de degré n,

$L_i(x_i)=1, i=0..n$

$L_i(x_j)=0, i\neq j$

Exemple 1

On connaît 2 points x_0, y_0 , x_1, y_1

$$P_1\ x\ =a_0+a_1x \qquad a_0=\frac{y_0x_1-y_1x_0}{x_1-x_0} \qquad a_1=\frac{y_1-y_0}{x_1-x_0}$$

$$P_1\ x\ =y_0L_0(x)+y_1L_1(x)$$

$$L_0(x)=\frac{x_1-x}{x_1-x_0} \qquad L_1(x)=\frac{x-x_0}{x_1-x_0}$$

Exemple 2

x	0	1	2
y	2	4	6

$$P_2(x) = y_0L_0(x) + y_1L_1(x) + y_2L_2(x)$$

$$L_0(x) = \frac{x - x_1}{x_0 - x_1}\,\frac{x - x_2}{x_0 - x_2} = \frac{x-1 \quad x-2}{2}$$

$$L_1(x) = \frac{x - x_0}{x_1 - x_0}\,\frac{x - x_2}{x_1 - x_2} = -x \quad x-2$$

$$L_2(x) = \frac{x - x_0}{x_2 - x_0}\,\frac{x - x_1}{x_2 - x_1} = \frac{x \quad x-1}{2}$$

$$P_2(x) = 2x + 2$$

Exemple 3

On connaît 3 points $(0,1)$, $(2,5)$ et $(4,17)$

$$P_2 \;\; x \; = L_0 \;\; x \;\; y_0 + L_1 \;\; x \;\; y_1 + L_2 \;\; x \;\; y_2$$

$$L_0(x) = \frac{(x-2)(x-4)}{8}$$

$$L_1(x) = \frac{x(x-4)}{-4}$$

$$L_2(x) = \frac{x(x-2)}{8}$$

En simplifiant, on trouve $P_2(x)=x^2+1$

Interpolation polynomiale de Lagrange

Polynôme nodal

Le polynome d'interpolation de Lagrange $P_n(x)$ peut s'écrire sous la forme suivante :

$$P_n(x) = \sum_{i=0}^{n} \frac{W_{n+1}(x)}{(x - x_i)\dot{W}_{n+1}(x_i)} y_i,$$

avec $W_{n+1}(x)$ est le polynome de nodal de degré n+1

$$W_{n+1}(x) = \prod_{i=0}^{n} (x - x_i) \quad \text{et} \quad \dot{W}_{n+1}(x_i) = \prod_{\substack{j=0 \\ i \neq j}}^{n} (x_i - x_j)$$

Erreur d'interpolation

Théorème Soit le support $\{x_i, y_i \equiv f(x_i)\}, i = 0,\dots,n$ d'interpolation d'une fonction f quelconque. On définit par $P_n(x)$ le polynôme d'interpolation correspondant.

Soit x une valeur appartenant à l'intervalle I_x (le plus petit intervalle contenant les points x_i),

On suppose que f(x) est de classe $\mathbb{C}^{n+1}(I_x)$

L'erreur d'interpolation au point x s'écrit :

$$E_n(x) = f(x) - P_n(x) = \underbrace{\left(\prod_{j=0}^{n} (x - x_j)\right)}_{W_{n+1}(x)} \frac{f^{(n+1)}(\xi)}{(n+1)!}, \xi \in I_x$$

E_n s'annule aux $n+1$ points x_i.

Erreur d'interpolation

Majoration de l'erreur d'interpolation

- si f est $n+1$ dérivable sur $[a,b]$, $\forall\, x \in [a,b]$, $\left\|f^{(n+1)}\right\|_\infty = \max_{x\in[a,b]}\left|f^{(n+1)}(x)\right|$
- I_x le plus petit intervalle fermé contenant x et les x_i
- alors, il existe $\xi \in I_x$ tel que
- NB : ξ dépend de x

• Utilité = on contrôle l'erreur d'interpolation donc la qualité de l'approximation

$$\left|E_n(x)\right| = \left|f(x) - P_n(x)\right| \le \frac{\left\|f^{(n+1)}\right\|_\infty}{(n+1)!}\left|W_{n+1}(x)\right| \le \frac{\left\|f^{(n+1)}\right\|_\infty}{(n+1)!}\left\|W_{n+1}\right\|_\infty \le \frac{\left\|f^{(n+1)}\right\|_\infty}{(n+1)!}(b-a)^{n+1}$$

Exemple Avec qu'elle précision peut-on calculer sin(1) à l'aide de l'interpolation polynomiale de Lagrange aux points $x_0 = 0,\ x_1 = \frac{\pi}{6},\ x_2 = \frac{\pi}{4},\ x_3 = \frac{\pi}{3},\ x_4 = \frac{\pi}{2}$

$$f(x) = \sin(x),\ [a,b] = \left[0, \frac{\pi}{2}\right], n = 4; \left\|f^{(n+1)}\right\|_\infty = \max_{x\in[a,b]}\left|f^{(n+1)}(x)\right| = 1$$

$$\left|E_4(1)\right| = \left|f(1) - P_4(1)\right| \le \frac{\left\|f^{(n+1)}\right\|_\infty}{(n+1)!}\left|b-a\right|^{n+1} = 0.0797$$

$$\left|E_4(1)\right| = \left|f(1) - P_4(1)\right| \le \frac{\left\|f^{(n+1)}\right\|_\infty}{(n+1)!}\prod_{i=0}^{4}\left|1 - x_i\right| = 5.8 * 10^{-5}$$

Interpolation polynomiale de Lagrange

Algorithme de Lagrange

Initialisation (x_i,y_i), i=0..n et x

$$
\begin{array}{l}
\text{pour } i = 1 \text{ jusqu'à } n+1 \text{ faire} \\
L_i \leftarrow 1 \\
\quad \text{pour } j = 1 \text{ jusqu'à } n+1 \\
\quad\quad \text{si } j \neq i \text{ alors} \\
\quad\quad\quad L_i \leftarrow L_i * \dfrac{x - x_j}{x_i - x_j} \\
\quad\quad \text{fin si} \\
\quad \text{fin pour} \\
\quad y \leftarrow y + y_i * L_i \\
\text{fin pour}
\end{array}
$$

Complexité du calcul : n^2

Remarques

- Le calcul de l'erreur d'interpolation dépend de la connaissance de f.
- L'erreur est faible près d'un nœud.
- L'erreur est autant plus petite que la fonction est lisse.
- Les polynômes de Lagrange ne sont pas pratiques puisque d'un point de vue numérique il est difficile de déduire L_{i+1} à partir de L_i

Solution Polynômes d'interpolation de Newton

Interpolation polynomiale de Newton

Principe

Étant donné n+1 points distincts $(x_i, y_i,\ i = 0 \ldots n)$ et le polynome d'interpolation de Newton $P_n(x)$ de degré n tel que

$$P_n(x) = \sum_{k=0}^{n} a_k x^k \text{ tel que } P_n(x_i) = y_i \ \forall i \in [0\ n]$$

$$P_n(x) = P_{n-1}(x) + Q_n(x)$$

$Q_n(x)$ est un polynome de degré n tel que

$$Q_n(x) = a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1}) = a_n W_n(x)$$

$$Q_n(x_i) = P_n(x_i) - P_{n-1}(x_i)\ \forall i \in [0\ n-1]$$

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots + a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

d'ou, on déduit : $P_n(x_n) = P_{n-1}(x_n) + Q_n(x_n) \Leftrightarrow f(x_n) = P_{n-1}(x_n) + a_n W_n(x_n)$

$$a_n = \frac{f(x_n) - P_{n-1}(x_n)}{W_n(x_n)}$$

il faut déterminer $a_i, \forall i \in [0\ n-1]$

Différences divisées de Newton

Différences divisées de Newton

Le coefficient a_n est appelé n^ème^ différence divisée de Newton

$$a_n = f\left[x_0, x_1, \ldots, x_n\right] = \frac{f(x_n) - P_{n-1}(x_n)}{W_n(x_n)}$$

Par conséquent, on obtient la formulation récursive suivante :

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

On a $W_{n+1}(x) = (x - x_0)(x - x_1)\ldots(x - x_n)$

On peut écrire alors :

$$P_0(x) = W_0(x) f\left[x_0\right] = f\left[x_0\right]$$

$$P_1(x) = P_0(x) + W_1(x) f\left[x_0, x_1\right] = f\left[x_0\right] + (x - x_0) f\left[x_0, x_1\right]$$

$$P_2(x) = P_1(x) + W_2(x) f\left[x_0, x_1, x_2\right] = f\left[x_0\right] + (x - x_0) f\left[x_0, x_1\right] + (x - x_0)(x - x_1) f\left[x_0, x_1, x_2\right]$$

$$\vdots$$

d'ou, on déduit : $P_n(x) = \sum_{k=0}^{n} W_k(x) f\left[x_0, x_1, \ldots x_k\right]$

$$P_n(x) = f\left[x_0\right] + f\left[x_0, x_1\right](x - x_0) + f\left[x_0, x_1, x_2\right](x - x_0)(x - x_1) + \ldots + f\left[x_0, x_1, \ldots x_n\right](x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

A partir de l'expression du polynome de Lagrange,

on peut obtenir l'expression $f\left[x_0, x_1, \ldots x_n\right] = \sum_{i=0}^{n} \frac{y_i}{\prod_{\substack{j=0 \\ j \neq i}}^{n} (x_i - x_j)}$

Interpolation polynomiale de Newton

Différences divisées de Newton

$$a_2 = f[x_0, x_1, x_2]$$

$$f[x_0,x_1,x_2] = \frac{\dfrac{f(x_2)-f(x_1)}{x_2-x_1} - \dfrac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2-x_0} = \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}$$

On peut montrer que la $k^{\text{ème}}$ différence divisée de Newton a la forme suivante :

$k=0,\qquad f[x_i] = f(x_i)$

$k=1,\qquad f[x_i, x_{i+1}] = \dfrac{f[x_{i+1}] - f[x_i]}{x_{i+1}-x_i} = f[x_{i+1}, x_i]$

$k \geq 2,\qquad f[x_i, x_{i+1}, \ldots, x_{i+k}] = \dfrac{f[x_{i+1}, \ldots, x_{i+k}] - f[x_i, x_{i+1}, \ldots, x_{i+k-1}]}{x_{i+k}-x_i}$

$$a_0 = f[x_0]$$

$$a_1 = f[x_0, x_1]$$

$$a_2 = f[x_0, x_1, x_2]$$

$$a_n = f\ x_0, x_1, \ldots, x_n$$

Différences divisées de Newton

x_i	$f\ x_i$	$Df\ x_i$	$D^2f\ x_i$	$\ldots$	$D^nf\ x_i$
x_0	$f(x_0)$				
x_1	$f(x_1)$	$f[x_0,x_1]=\dfrac{f[x_1]-f[x_0]}{x_1-x_0}$			
x_2	$f(x_2)$	$f[x_1,x_2]=\dfrac{f[x_2]-f[x_1]}{x_2-x_1}$	$f[x_0,x_1,x_2]=\dfrac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}$		
x_3	$f(x_3)$	$f[x_2,x_3]=\dfrac{f[x_3]-f[x_2]}{x_3-x_2}$	$f[x_1,x_2,x_3]=\dfrac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1}$	$\ddots$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
x_n	$f(x_n)$	$f[x_{n-1},x_n]=\dfrac{f[x_n]-f[x_{n-1}]}{x_n-x_{n-1}}$	$f[x_{n-2},x_{n-1},x_n]=\dfrac{f[x_{n-1},x_n]-f[x_{n-2},x_{n-1}]}{x_n-x_{n-2}}$	$\ldots$	$f\ x_0, x_1, \ldots, x_n$

Interpolation polynomiale de Newton

Différences divisées de Newton

Exemple 1

x_i	$f[x_i]$
1	3
2	6
3	19
5	99

x_i	$f[x_i]$	$Df[x_i]$	$D^2f[x_i]$	$D^nf[x_i]$
1	3			
2	6	3		
3	19	13	5	
5	99	40	9	1

$$P_3(x) = \sum_{k=0}^{3} W_k(x) f[x_0, x_1, \dots x_k]$$

$$P_3(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2)$$

$$= 3 + 3(x-1) + 5(x-1)(x-2) + (x-1)(x-2)(x-3)$$

Différences divisées de Newton

Exemple 2

x_i	$f(x_i)$
2.0	0.85467
2.3	0.75682
2.6	0.43126
2.9	0.22364
3.2	0.08567

	x_i	$f(x_i)$	$Df(x_i)$	$D^2f(x_i)$	$D^3f(x_i)$	$D^4f(x_i)$
0	**2.0**	**0.85467**				
1	**2.3**	**0.75682**	**-0.32617**			
2	**2.6**	**0.43126**	**-1.08520**	**-1.26505**		
3	**2.9**	**0.22364**	**-0.69207**	**0.65522**	**2.13363**	
4	**3.2**	**0.08567**	**-0.45990**	**0.38695**	**-0.29808**	**-2.02642**

$$P_4(x)=a_0+a_1(x-x_0)+a_2(x-x_0)(x-x_1)+a_3(x-x_0)(x-x_1)(x-x_2)$$
$$+a_4(x-x_0)(x-x_1)(x-x_2)(x-x_3)$$

$$P_4(x)=0.85467-0.32617(x-2.0)-1.26505(x-2.0)(x-2.3)$$
$$+2.13363(x-2.0)(x-2.3)(x-2.6)-2.02642(x-2.0)(x-2.3)(x-2.6)(x-2.9)$$

$$f(2.8)\approx P(2.8)=0.275$$

Algorithme de Newton

Initialisation (x_i, y_i), i=0..n et x

pour $i = 1$ jusqu'à n faire

$\quad F(i,1) \leftarrow y(i)$

fin pour

pour $i = 1$ jusqu'à $n+1$ faire

$\quad$ pour $j = 1$ jusqu'à $i-1$

$$F(i, j+1) \leftarrow \frac{F(i, j) - F(i-1, j)}{x(i) - x(i-j)}$$

$\quad$ fin pour

fin pour

pour $i = 1$ jusqu'à n

$\quad a(i) \leftarrow F(i,i)$

fin pour

$n = \text{length}(x) - 1;$

$y = F(n+1, n+1)$

pour $i = n:-1:1$ faire

$y = y.*(z - x(i)) + F(i,i);$

fin pour

Complexité du calcul : n^2

Interpolation polynomiale de Newton

Remarques

- Pour n+1 points, il est nécessaire de calculer une matrice triangulaire inférieure de taille n dans laquelle n(n+1)/2 éléments différents de zéro.
- L'erreur est faible près d'un nœud.
- n(n+1) additions et n(n+1)/2 divisions sont nécessaires pour construire la matrice de Newton.
- Si on dispose d'un nouveau nœud x_{n+1}, an aurait à calculer simplement une ligne supplémentaire de n+1 éléments.
- Pour construire P_{n+1} à partir de P_n, 2(n+1) additions et (n+1) divisions sont nécessaires donc un cout O(n), ceci n'est pas le cas pour Lagrange ou il est nécessaire de répéter toute la procédure avec un cout O(n^2)

$$|E_n(x)| = |f(x) - P_n(x)| \le \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}|W_{n+1}(x)| \le \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}\|W_{n+1}\|_\infty \le \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}(b-a)^{n+1}$$

Convergence uniforme

L'étude de la convergence uniforme analyse le comportement de la norme de maximum de l'erreur quand $n \to \infty$

Soient $x_0,\ x_0, \ldots, x_n,$ n+1 points distincts dans $[a\ b]$ et $P_n(x)$ le polynome d'interpolation, il est toujours possible de trouver la fonction f pour laquelle $\lim_{n\to\infty} \| f - P_n \|_\infty \neq 0$

L'interpolation polynomiale ne permet pas d'approcher toute fonction continue. Ceci est le cas pour des fonctions dont les points singuliers sont trop proche à l'intervalle $[a\, b]$.

Notons, ceci n'est pas en contradiction avec le théorème de Weirstrass selon lequel pour toute fonction continue f de classe C^1 sur $[a\, b]$ il existe toujours une série des polynomes P_n tels que $\lim_{n\to\infty} \| f - P_n \|_\infty = 0$

Malheureusement, ces polynomes ne peuvent pas être obtenus par interpolation polynomiale

Interpolation polynomiale de Newton

Contre exemple

Supposons qu'on approche la fonction f(x)= $\frac{1}{1+x^2}$ ou f(x)=2(1+tanh(x))- $\frac{x}{10}$

en utilisant l'interpolation de Lagrange avec noeuds équirépartis.

Puisque les dérivées de la fonction ne sont pas bornées, on peut vérifier qu'il existe des points x à l'intérieur de l'intervalle de l'interpolation tels que : $\lim_{n\to\infty} \left| f(x) - P_n(x) \right| \neq 0$

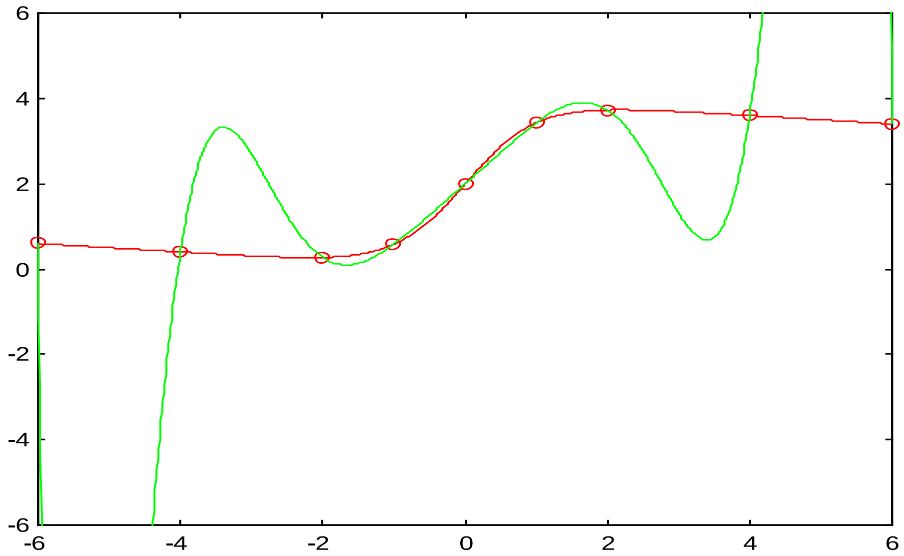
c.à.d l'approximation se dégrade en ajoutant des données.

Remarques :

- les polynomes d'interpolation présentent des oscillations qui augmentent avec le degré du polynome
-une petite variation des données, noeuds équirépartis, cause une large variation de l'approximation

$\Rightarrow$ utilisation de l'interpolation de Hermite (informations dur les dérivées de la fonction à approcher)

$\Rightarrow$ interpolation polynomiale par morceaux (splines)



f(x)=2(1+tanh(x)) - x/10

Conditionnement de problème

Le conditionnement du problème d'interpolation peut être décrit en fonction des polynomes caractéristiques par la constante de Lebergue associé aux noeuds d'interpolation.

$$\Lambda_n = \left\| \sum_{i=0}^{n} |L_i(x)| \right\|_\infty$$

Supposons de perturber les données de la manière suivante :

$\hat{y}_i = y_i + \delta_i$ et soit $\hat{P}_n$ le polynome perturbé résultant,

il est possible de montrer que : $\dfrac{\left\| P_n - \hat{P}_n \right\|_\infty}{\left\| P_n \right\|_\infty} \leq \Lambda_n \leq \dfrac{\left\| \delta_n \right\|_\infty}{\left\| f \right\|_\infty}$

En particulier pour des noeuds équirépartis, on peut montrer que $\Lambda_n \simeq \dfrac{2^{n+1}}{e^n \log n}$

Interpolation polynomiale de Hermite

Principe

On suppose que f est de classe C^1 sur $[a, b]$. On peut généraliser l'interpolation de Lagrange d'une fonction f pour chercher un polynome qui passe pas seulement par les points $(xi , f(xi))$, mais dont les dérivées coincident à certains points nodaux avec les dérivées de la fonction f.

Le polynôme d'interpolation utilisant les valeurs de f et de sa dérivée est appelé polynome de Hermite.

Théorème d'existence et d'unicité

Étant donné n+1 points distincts $(x_i,\ i=0\ldots n)$, il existe un unique polynôme $P_{2n+1}(x)$ nommé polynôme d'interpolation de Hermite tel que :

$$P_{2n+1}(x)=\sum_{k=0}^{2n+1}a_k x^k=\sum_{k=0}^{n}\left(f(x_k)H_k(x)+\dot{f}(x_k)\hat{H}_k(x)\right) \text{ avec } P_{2n+1}(x_i)=f(x_i) \text{ et } \dot{P}_{2n+1}(x_i)=\dot{f}(x_i) \quad \forall i \in$$

$$\begin{cases} H_k(x)=\left(L_k(x)\right)^2\left(1-2\dot{L}_k(x_k)(x-x_k)\right) \\ \hat{H}_k(x)=\left(L_k(x)\right)^2(x-x_k) \end{cases}, k=0\ldots n$$

$$\left|E_{2n+1}(x)\right|=\left|f(x)-P_{2n+1}(x)\right|\le\frac{\left\|f^{(2n+2)}\right\|_\infty}{(2n+2)!}\left(\left\|W_{n+1}\right\|_\infty\right)^2\le\frac{\left\|f^{(2n+2)}\right\|_\infty}{(2n+2)!}(b-a)^{2n+2}$$

Remarque Comme pour l'interpolation de Lagrange, cette forme n'est pas adaptée à un algorithme. Quand on rajoute un nœud il faut tout recalculer. Il existe aussi pour les polynômes d'interpolation d'Hermite une écriture à l'aide des différences divisées

Interpolation polynomiale de Hermite

Principe Étant donné n+1 points distincts ($x_i,\ i=0\ldots n$), on pose $z_{2i}=z_{2i+1}=x_i, i=0\ldots n$

$$f\left[z_{2i}\ z_{2i+1}\right]=\dot{f}\left(z_{2i}\right)=\dot{f}\left(x_i\right)$$

z	$f(z)$	$\mathrm{Df}\ \mathrm{z_i}$	$\mathrm{D^2f}\ \mathrm{z_i}$	$\cdots$
$z_0=x_0$	$f[z_0]=f(x_0)$			
$z_1=x_0$	$f[z_1]=f(x_0)$	$f[z_0,z_1]=f'(x_0)$		
$z_2=x_1$	$f[z_2]=f(x_1)$	$f[z_1,z_2]=\dfrac{f[z_2]-f[z_1]}{z_2-z_1}$	$f[z_0,z_1,z_2]=\dfrac{f[z_1,z_2]-f[z_0,z_1]}{z_2-z_0}$	
$z_3=x_1$	$f[z_3]=f(x_1)$	$f[z_2,z_3]=f'(x_1)$	$f[z_1,z_2,z_3]=\dfrac{f[z_2,z_3]-f[z_1,z_2]}{z_3-z_1}$	
$z_4=x_2$	$f[z_4]=f(x_2)$	$f[z_3,z_4]=\dfrac{f[z_4]-f[z_3]}{z_4-z_3}$	$f[z_2,z_3,z_4]=\dfrac{f[z_3,z_4]-f[z_2,z_3]}{z_4-z_2}$	
$z_5=x_2$	$f[z_5]=f(x_2)$	$f[z_4,z_5]=f'(x_2)$	$f[z_3,z_4,z_5]=\dfrac{f[z_4,z_5]-f[z_3,z_4]}{z_5-z_3}$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	

$$P_{2n+1}(x)=\sum_{k=0}^{2n+1}W_k(z)\,f\left[z_0,z_1,\ldots,z_k\right]$$

$$P_{2n+1}(x)=f\left[z_0\right]+f\left[z_0,z_1\right](x-z_0)+\ldots+f\left[z_0,z_1,\ldots,z_n\right](x-z_0)(x-z_1)\ldots(x-z_{n-1})$$

Interpolation polynomiale de Hermite

Exemple

k	x_k	$f(x_k)$	$f'(x_k)$
0	1.3	0.6200860	−0.5220232
1	1.6	0.4554022	−0.5698959
2	1.9	0.2818186	−0.5811571

z_i	$f\ z_i$	$Df\ z_i$	$D^2f\ z_i$	$D^3f\ z_i$	$D^4f\ z_i$	$D^5f\ z_i$
1.3	0.6200860					
1.3	0.6200860	−0.5220232				
1.6	0.4554022	−0.5489460	−0.0897427			
1.6	0.4554022	−0.5698959	−0.0698330	0.0663657		
1.9	0.2818186	−0.5786120	−0.0290537	0.0679655	0.0026663	
1.9	0.2818186	−0.5811571	−0.0084837	0.0685667	0.0010020	−0.0027738