Interpolation polynomiale

Polynome de Lagrange

Définition : *Polynômes de Lagrange*

Pour les points $x_0,\ x_1,\ \ldots,\ x_n$ donnés et distincts, les n polynômes de Lagrange de degré n sont définis par

$$L_i(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)} = \prod_{\substack{0\leq j\leq n\\ j\neq i}} \frac{(x-x_j)}{(x_i-x_j)}, \qquad (3.3)$$

pour $i = 0, 1, \ldots, n$ et $x \in \mathbb{R}$.

La fonction L_i est un polynôme de degré n tel que

$$\boxed{L_i(x_j) = \begin{cases} 1 & \text{si } i = j, \\ 0 & \text{si } i \neq j. \end{cases}}$$

Le polynôme de Lagrange L_i est donc le polynôme d'interpolation associé aux points $\{x_0, \ldots, x_n\}$, qui vaut 1 en xi et 0 ailleurs. Le polynôme P d'interpolation de Lagrange aux points (x_i, y_i) pour i = 0, . . . ,n est alors donné par :

$$p(x) = \sum_{i=0}^{n} y_i L_i(x).$$

Base de Newton - Différences divisées

Définition : *Base de Newton*

On appelle base de Newton relative à $\{x_0, \ldots, x_n\}$, la base formée des $n+1$ polynômes

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

Tout polynôme $p \in \mathbb{P}_n$ peut s'écrire sous la forme

$$p(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \cdots + a_n \prod_{i=0}^{n-1}(x-x_i),$$

Le polynôme d'interpolation p de degré $\leq n$ de f en $x_0,\ \ldots,\ x_n$ s'écrit

$$p(x) = f(x_0) + f[x_0, x_1](x-x_0) + \cdots + f[x_0, \ldots, x_n] \prod_{i=0}^{n-1}(x-x_i),$$

Notation et Définition. *Différences divisées*

Soient $\{x_k\}_{k\geq 0}$ des points deux à deux distincts. Les différences divisées de la fonction f sont définies par récurrence de la façon suivante.

Différence divisée d'ordre 0 : $\quad f[x_i] = f(x_i) = y_i$

Différence divisée d'ordre 1 : $\quad f[x_i, x_j] = \dfrac{f(x_i) - f(x_j)}{x_i - x_j} = \dfrac{y_i - y_j}{x_i - x_j}, \quad i \neq j$

Différence divisée d'ordre $k \neq 0$:

$$f[x_l, \dots, x_{l+k}] = \frac{f[x_l, \dots, x_{l+k-1}] - f[x_{l+1}, \dots, x_{l+k}]}{x_l - x_{l+k}} \tag{3.6}$$

Proposition 1 :

i) On a, pour $k \geq 1$, $\quad f[x_l, \dots, x_{l+k}] = \displaystyle\sum_{i=l}^{l+k} \frac{f(x_i)}{\displaystyle\prod_{\substack{l \leq j \leq l+k \\ j \neq i}} (x_i - x_j)}$.

ii) Les valeurs des différences divisées ne dépendent pas de l'ordre des points.

Calcul des différences divisées

Les différences divisées d'ordre k sont calculées à partir des différences divisées d'ordre k − 1 par la relation de récurrence (3.6) (et non pas en utilisant la Proposition 1). On construit ainsi le tableau suivant :

$$\begin{array}{c|c|ccccccc}
x_0 & y_0 = \boxed{f(x_0)} & & & & & & & \\
& & \searrow & & & & & & \\
x_1 & y_1 = f(x_1) & \rightarrow & \boxed{f[x_0, x_1]} & & & & & \\
& & \searrow & & \searrow & & & & \\
x_2 & y_2 = f(x_2) & \rightarrow & f[x_1, x_2] & \rightarrow & \boxed{f[x_0, x_1, x_2]} & & & \\
& & \searrow & & \searrow & & \searrow & & \\
x_3 & y_3 = f(x_3) & \rightarrow & f[x_2, x_3] & \rightarrow & f[x_1, x_2, x_3] & \rightarrow & \boxed{f[x_0, x_1, x_2, x_3]} & \\
\vdots & \vdots & & \vdots & & & & \ddots & \\
\vdots & \vdots & \searrow & \vdots & \searrow & & & & \searrow \\
x_n & y_n = f(x_n) & \rightarrow & f[x_{n-1}, x_n] & \rightarrow & f[x_{n-2}, x_{n-1}, x_n] \ \dots & \dots & \dots \rightarrow & \boxed{f[x_0, \dots, x_n]}
\end{array}$$

Les coefficients de p dans la base de Newton sont lus en haut sur la diagonale. Si on ajoute un point supplémentaire (x_{n+1}, y_{n+1}) il faut effectuer n+1 calculs en plus pour obtenir $f[x_0, \dots, x_n, x_{n+1}]$.

Exemple : déterminer les coefficients dans la base de Newton du polynôme d'interpolation de degré 3 tel que p(−2) = −31, p(0) = −1, p(1) = −1 et p(4) = 83.

$$\begin{array}{c|ccccc}
 & & a_0 & & & \\
x_i & y_i \ \swarrow & & a_1 & & \\
\hline
-2 & \boxed{-31} & \swarrow & & a_2 & \\
0 & -1 & \boxed{15} & \swarrow & & a_3 \\
1 & -1 & 0 & \boxed{-5} & \swarrow & \\
4 & 83 & 28 & 7 & \boxed{2} &
\end{array}$$

On obtient ainsi

$$p(x) = -31 + 15(x+2) - 5(x+2)x + 2(x+2)x(x-1)$$

Une fois les coefficients du polynôme obtenus, il faut évaluer la valeur de ce polynôme en une valeur x donnée. On utilise pour cela le schéma de Horner.

Evaluation du polynôme - Schéma d'Horner

Dans la base de Newton, on a $\mathbf{P(x) = c_0+c_1(x-x_0)+\cdots+c_n(x-x_0)\dots(x-x_{n-1})}$.

où les $c_i=f[x_0,x_1,\dots x_i]$, i=0,1,…,n

- **Calculer P(x) pour une valeur x donnée**

Si on calcule par différences divisées les coefficients c_i de P dans la base de Newton et qu'on évalue ensuite le polynôme en x, le nombre d'opérations effectuées est alors le suivant :

- **Evaluation des coefficients :**

1ère colonne : 2n soustractions + n divisions = 3n opérations.
2ème colonne : 3(n – 1) opérations.
⋮
nème colonne : 3 opérations.

Le coût total pour calculer les coefficients dans la base de Newton par différence divisées est donc de :

$$3\left(n+(n-1)+\cdots 1\right)=\frac{3n(n+1)}{2}\simeq\frac{3n^2}{2}\text{ opérations pour } n \text{ grand.}$$

- **Evaluation du polynôme par un schéma de Horner :**

Pour évaluer le polynôme P en un point x donné, on écrit l'expression de P dans la base de Newton sous la forme suivante (schéma de Horner) :

$$P_n(x)=c_0+(x-x_1)(c_1+(x-x_2)c_1+(x-x_3)(\dots\dots.)))$$

On effectue ainsi 2n additions et n multiplications, soit un coût de 3n opérations pour le schéma de Horner.

L'algorithme d'évaluation d'un polynôme d'interpolation par la forme de Newton

Input $\{x_1,\dots,x_{n+1},d_1,\dots,d_{n+1}\}$
Input x
$p\leftarrow d_{n+1}$
for $i=n,n-1,\dots,1$
  $p\leftarrow p*(x-x_i)+d_i$
Output $\{x,p\}$

Les différences divisées peuvent aussi être utilisées dans la formule du schéma de Horner (ici pour $n=4$) :

$$P(x)=d_{1,1}+(x-x_1)\{d_{1,2}+(x-x_2)[d_{1,3}+(x-x_3)(d_{1,4})]\}.$$

Erreur d'interpolation

Théorème 3 : *Erreur d'interpolation*

Soit f une fonction $(n+1)$ fois dérivable et soit p son polynôme d'interpolation de degré n associés aux points $x_0, \dots, x_n$ distincts. Alors pour tout $x\in\mathbb{R}$, il existe un réel $\theta_x\in]\min(x,x_i),\max(x,x_i)[$ tel que

$$f(x)-p(x)=\frac{1}{(n+1)!}\Pi_{n+1}(x)\,f^{(n+1)}(\theta_x),\qquad\text{avec}\qquad \Pi_{n+1}(x)=\prod_{i=0}^{n}(x-x_i).$$

Problème de convergence de l'interpolation

L'erreur d'interpolation dépend essentiellement de deux termes ; d'une part de la fonction n+1 qui ne dépend que de la répartition des points x_i et non de f et d'autre part de la dérivée $(n+1)^{ème}$ de f qui au contraire ne dépend pas des x_i. On se place désormais sur un intervalle sur un intervalle [a, b] avec $a = x_0 < x_1 < \cdots < xn = b$. On peut alors estimer la fonction Π_{n+1} pour des abscisses xi quelconques.

Lemme 2 : Pour des points $x_0, \cdots, x_n$ quelconques, on a l'estimation

$$\max_{x\in[a,b]} |\Pi_{n+1}(x)| \leq h^{n+1}n! \quad \text{où} \quad h = \max_i h_i \text{ avec } h_i = |x_{i+1} - x_i|.$$

Pour une fonction f qui est n + 1 fois dérivable, on considère son polynôme d'interpolation P aux points {x0, . . . , xn}

$$\max_{x\in[a,b]} |f(x) - p(x)| \leq \frac{1}{(n+1)!} \max_{x\in[a,b]} |\Pi_{n+1}(x)| \max_{x\in[a,b]} \left|f^{(n+1)}(x)\right|.$$

Compte tenu du Lemme 2, on obtient alors l'estimation d'erreur suivante :

$$\max_{x\in[a,b]} |f(x) - p(x)| \leq \frac{h^{n+1}}{(n+1)} \max_{x\in[a,b]} \left|f^{(n+1)}(x)\right|.$$

Points d'interpolation équidistants

On considère une subdivision régulière de l'intervalle [a, b] en (n + 1) points (n ≥ 1) :

$$x_0 = a, \quad x_1 = a+h, \quad x_2 = a+2h, \quad \ldots \quad , \quad x_n = a+nh, \quad \text{avec } h = \frac{b-a}{n}.$$

$$\text{Dans ce cas, on a } \frac{h^{n+1}}{(n+1)} = \frac{1}{(n+1)}\left(\frac{b-a}{n}\right)^{n+1} \longrightarrow 0 \text{ quand } n \to +\infty.$$

Compte tenu de l'estimation d'erreur, on serait donc tenté de penser que l'erreur entre f et son polynôme d'interpolation P tend vers 0 quand n → +∞. Mais attention car la $(n+1)^{ème}$ dérivée de f dépend de n et en fait peut croître très rapidement avec n. En général, P ne converge pas vers f lorsque n tend vers +∞. L'exemple suivant illustre une telle situation de non-convergence.

Phénomène de Runge

On considère la fonction $f(x) = \frac{1}{1+x^2}$ sur l'intervalle [−5, 5].

On note P_n le polynôme d'interpolation de f aux n+1 points équidistants dans l'intervalle [−5, 5]. On observe alors (Fig.1) quand n augmente, des problèmes d'oscillations aux extrémités de l'intervalle. En fait $|f^{(n)}(5)|$ devient rapidement grand avec n. On montre que pour

$$|x| \geq 3.83\cdots, \text{ on a } |f(x) - p(x)| \to +\infty \text{ quand } n \to +\infty.$$

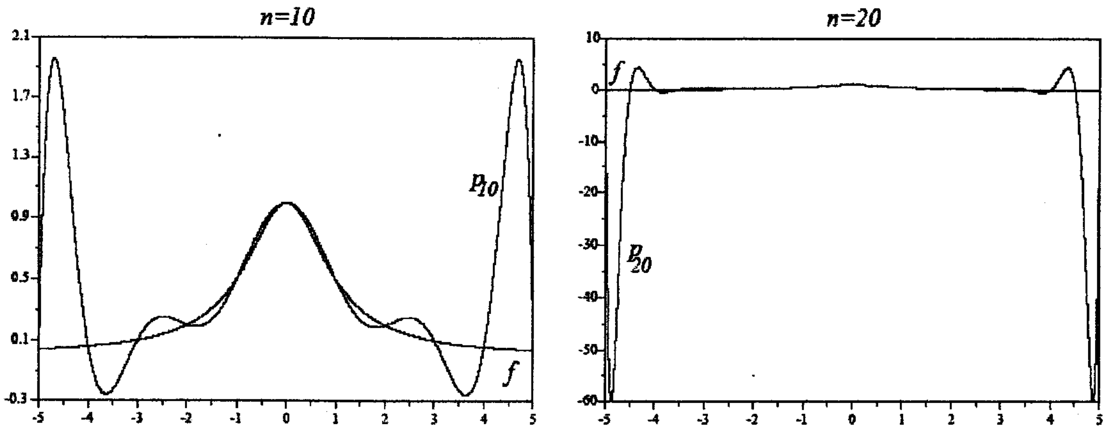

Fig. 1 – Phénomène de Runge : Interpolation polynomiale avec points équidistants.

Abscisses de Tchebichev

En général, il n'y a pas de convergence du polynôme d'interpolation lorsqu'on choisit les points d'interpolation répartis de façon uniforme dans un intervalle fermé borné. Mais existe-t-il une répartition (évidemment non uniforme) des points d'interpolation pour lesquels il y convergence ? Une réponse est fournie par les abscisses de Tchebychev.

Abscisses de Tchebichev

Le problème est le suivant : étant donnés l'intervalle $[-1,1]$ et un entier $n \in \mathbb{N}$, trouver $n+1$ points $\{x_0, x_1, \ldots, x_n\}$ distincts qui minimisent $\|\Pi_{n+1}\|_\infty = \max_{x\in[-1,1]} |\Pi_{n+1}(x)|$.

L'existence de ces points est donnée par le théorème suivant.

Théorème 4 : *Abscisses de Tchebichev*

1) La fonction $T_{n+1}(x) = \cos((n+1)\arccos(x))$ définie pour $x \in [-1,1]$ est un polynôme de degré $n+1$ en x, appelé polynôme de Tchebichev. Ce polynôme admet exactement $n+1$ racines distinctes dans $[-1,1]$, appelées abscisses de Tchebichev.

2) Les $(n+1)$ racines du polynôme de Tchebichev T_{n+1} minimisent la quantité $\|\Pi_{n+1}\|_\infty$.

On montre facilement par récurrence que les T_n vérifient

$$\begin{cases} T_0(x) = 1, \qquad T_1(x) = x, \\ T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x). \end{cases}$$

Ces relations permettent d'établir que T_{n+1} est bien un polynôme de degré n+1. Cherchons à présent les racines de ce polynôme.

Les racines de $T_{n+1}(x)$ sont données par :

$$x_i = \cos\left(\frac{(2i+1)\pi}{2(n+1)}\right) \qquad \textbf{avec } i = 0,\ldots,n$$

Abscisses de Tchebychev sur un intervalle [a, b] :

$$t_i = \frac{(a+b)}{2} + \frac{(b-a)}{2}x_i,$$

où les x_i sont les abscisses de Tchebychev données sur l'intervalle [−1, 1].

La fig.2 suivante montre ce que donnent les abscisses de Tchebychev avec l'exemple du phénomène de Runge.

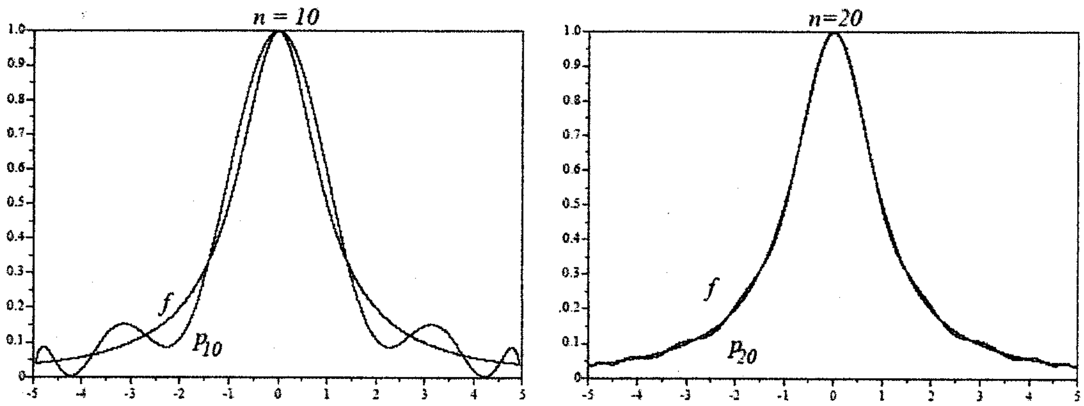

Fig. 2 – Interpolation avec les abscisses de Tchebychev.

Algorithme de Horner d'évaluation d'un polynôme $P=a_0+a_1x+...+a_nx^n$

function Pn=Horner(a,x,t)

<u>entrée :</u> a : le vecteur ligne des coefficients du polynome p

t : point où on veut évaluer le polynôme P

<u>sortie :</u> P_n : valeur du polynôme au point t : $P_n=P(t)$

> **P←a_n**
> **Pour k=n-1 à 0**
> **P←t*P+a_k**
> **end**

Algorithme de Horner d'évaluation du polynôme d'interpolation de Newton

P=c(1) + c(2)*(t-x(1)) + c(3)*(t-x(1))*(t-x(2)) + ... + c(n)*(t-x(1))*...*(t-x(n-1))

function Pn=Horner2(c,x,t)

<u>entrée</u> c :[c_0,..c_n] le vecteur ligne des différences divisées

x : vecteur [x

t : point où on veut évaluer le polynôme P

<u>sortie</u> P_n : valeur du polynôme au point t : $P_n=P(t)$

> **P←d_n**
> **Pour k=n-1 à 0**
> **P←P*(x-t_i)+d_i**
> **end**

```
function [p]=horner(c,x,t)

 % Evaluation du polynome en un point t :
 % P(t)=c(1) + c(2)*(t-x(1)) + c(3)*(t-x(1))*(t-x(2)) + ... + c(n)*(t-x(1))*...*(t-x(n-1))
 % par l'algorithme de horner

 n=length(c)
 p=c(n)
 for k=n-1:-1:1
   p=c(k)+(t-x(k))*p
 end
```

```
function [p]=horner2(c,x,t)

 % Evaluation du polynome aux points t=[t0,t1,...tk]
 % P(t)=c(1) + c(2)*(t-x(1)) + c(3)*(t-x(1))*(t-x(2)) + ...par l'algorithme de Horner
 % t est un vecteur d'instants (ou une matrice)
 n=length(x);
 p=c(n)*ones(size(t));
 for k=n-1:-1:1
    p= p.*(t-x(k)*ones(size(k)))+c(k)*ones(size(k));
 end
```

~~Exercices~~